

# RH850/F1L Group

User's Manual: Hardware

Renesas microcontroller  
RH850 Family

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
  2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
  3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
  4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
  5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
- Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
  7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
  8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
  9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
  10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
  11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## Notes for CMOS devices

- (1) Voltage application waveform at input pin: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).
- (2) Handling of unused input pins: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
- (3) Precaution against ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.
- (4) Status before initialization: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.
- (5) Power ON/OFF sequence: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.
- (6) Input of signal during power off state: Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

# How to Use This Manual

**Readers** This manual is intended for users who wish to understand the functions of the RH850/F1L and design application systems using the RH850/F1L microcontrollers.

**Purpose** This manual is intended to give users an understanding of the hardware functions of the RH850/F1L shown in the *Organization* below.

**Organization** This manual is divided into two parts: Hardware (this manual) and software (RH850G3K User's Manual: Software).

Hardware	Software
Pin functions	Overview
CPU function	Processor Model
On-chip peripheral functions	Register Reference
Flash memory programming	Exceptions and Interrupts
	Memory Management
	Instruction Reference
	Reset
	Appendix

**How to read this manual** It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the RH850/F1L.

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See RH850G3K User's Manual: Software available separately.

The “yyy bit of the xxx register” is described as the “xxx.yyy bit” in this manual. Note with caution that if “xxx.yyy” is described as is in a program, however, the compiler/assembler cannot recognize it correctly.

**Conventions**

- Data significance: Higher digits on the left and lower digits on the right
- Active low representation: xxx (overscore over pin or signal name)
- Memory map address: Higher addresses on the top and lower addresses on the bottom
- Note: Footnote for item marked with Note in the text
- Caution: Information requiring particular attention
- Remark: Supplementary information
- Numeric representation:
  - Binary ... xxxx or xxxx<sub>B</sub>
  - Decimal ... xxxx
  - Hexadecimal ... xxxx<sub>H</sub>
- Prefix indicating power of 2 (address space, memory capacity):
  - K (kilo):  $2^{10} = 1,024$
  - M (mega):  $2^{20} = 1,024^2$
  - G (giga):  $2^{30} = 1,024^3$

## Description of Registers

Each register description includes the register access, register address, and register value after a reset, a bit chart illustrating the arrangement of bits and a table of bits, describing the meaning of the bit settings.

The standard format for bit charts and tables is shown below.

**Table 14.19 CSIGnCFG0 Register Contents (1/2)**

Bit Position	Bit Name	Function																				
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																				
29, 28	CSIGnPS[1:0]	Specifies parity.																				
		<table border="1"> <thead> <tr> <th>CSIGn PS1</th> <th>CSIGn PS0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No parity transmitted</td> <td>No parity is waited for.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Add parity bit fixed at 0</td> <td>Parity bit is waited for but not judged.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Add odd parity</td> <td>Odd parity bit is waited for.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Add even parity</td> <td>Even parity bit is waited for.</td> </tr> </tbody> </table>	CSIGn PS1	CSIGn PS0	Transmission	Reception	0	0	No parity transmitted	No parity is waited for.	0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.	1	0	Add odd parity	Odd parity bit is waited for.	1	1	Add even parity	Even parity bit is waited for.
CSIGn PS1	CSIGn PS0	Transmission	Reception																			
0	0	No parity transmitted	No parity is waited for.																			
0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.																			
1	0	Add odd parity	Odd parity bit is waited for.																			
1	1	Add even parity	Even parity bit is waited for.																			
27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits																				
<b>CAUTION</b>																						
Do not set bits CSIGnCFG0.CSIGnDLS[3:0] to values 1 to 6 when the extended data length function is disabled with bit CSIGnCTRL1.CSIGnEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits.																						
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																				

### (1) Access

The register can be accessed in the bit unit indicated here.

### (2) Address

This is the register address.

For base address, see description of base address in each section.

(3) Value after a reset (in hexadecimal notation)

This is the value of all bits of the register after a reset. Values for bytes are given as numbers in the range from 0 to 9 and letters from A to F or as X where they are undefined.

(4) Bit position

This is the bit number.

The bits are numbered from 31 to 0 for 32-bit registers, 15 to 0 for 16-bit registers, and 7 to 0 for 8-bit registers.

(5) Bit name

Bit name or field name is indicated.

If it is required to clearly identify the digits of a bit field, this is done by using a form such as CSIGnDLS[3:0] in the example above.

Reserved bits are indicated by a dash (—).

(6) Value after a reset (in binary notation)

This is the bit value after a reset.

0 : The value after a reset is 0.

1 : The value after a reset is 1.

— : The value after a reset is undefined.

(7) R/W

This is the bit attribute of all bits of the register.

R/W : The bit or field is readable and writable.

R : The bit or field is readable.

Note that all reserved bits are indicated as R. When written, the value specified in the bit chart or the value after a reset should be written.

W : This bit or field is writable. When read, the value is undefined. If a value is indicated in the bit chart, the value is returned.

(8) Function

This is function of the bit.

# Table of Contents

Section 1	Overview .....	63
1.1	RH850/F1L Product Features.....	63
1.1.1	RH850/F1L Functions.....	64
1.1.2	Internal Block Diagram .....	68
Section 2	Pin Function .....	74
2.1	Pin Connection Diagram.....	74
2.2	Pin Description.....	96
2.3	Pin Functions During and After Reset .....	101
2.4	Port State in Standby Mode.....	102
2.5	Recommended Connection of Unused Pins.....	103
2.6	RH850/F1L Port Features.....	104
2.6.1	Port Group .....	104
2.6.2	Port Group Index n .....	104
2.6.3	Register Base Address.....	104
2.6.4	Clock Supply.....	104
2.7	Port Functions.....	105
2.7.1	Functional Overview .....	105
2.7.2	Terms .....	106
2.7.2.1	JTAG Ports.....	106
2.7.3	Overview of Pin Functions .....	107
2.7.4	Pin Data Input/Output.....	109
2.7.4.1	Output Data .....	109
2.7.4.2	Input Data.....	109
2.7.4.3	Writing to the Pn Register .....	110
2.8	Schematic View of Port Control .....	111
2.9	Port Group Configuration Registers.....	112
2.9.1	Overview.....	112
2.9.2	Pin Function Configuration .....	115
2.9.2.1	PM <sub>Cn</sub> / JPMC0 — Port Mode Control Register .....	115
2.9.2.2	PMCSR <sub>n</sub> / JPMCSR0 — Port Mode Control Set/Reset Register.....	116
2.9.2.3	PIPC <sub>n</sub> — Port IP Control Register .....	117
2.9.2.4	PM <sub>n</sub> / APM <sub>n</sub> / JPM0 — Port Mode Register .....	118
2.9.2.5	PMSR <sub>n</sub> / APMSR <sub>n</sub> / JPMSR0 — Port Mode Set/Reset Register.....	119
2.9.2.6	PIBC <sub>n</sub> / APIBC <sub>n</sub> / JPIBC0 / IPIBC0 — Port Input Buffer Control Register .....	120
2.9.2.7	PFC <sub>n</sub> / JPFC0 — Port Function Control Register .....	121
2.9.2.8	PFCE <sub>n</sub> — Port Function Control Expansion Register .....	122
2.9.2.9	PFCAE <sub>n</sub> — Port Function Control Additional Expansion Register.....	123
2.9.3	Pin Data Input/Output.....	125
2.9.3.1	PBDC <sub>n</sub> / APBDC <sub>n</sub> / JPBDC0 — Port Bidirection Control Register .....	125
2.9.3.2	PPR <sub>n</sub> / APPR <sub>n</sub> / JPPR0 / IPPR0 — Port Pin Read Register .....	126
2.9.3.3	P <sub>n</sub> / AP <sub>n</sub> / JP0 — Port Register .....	127
2.9.3.4	PNOT <sub>n</sub> / APNOT <sub>n</sub> / JPNOT0 — Port NOT Register .....	128
2.9.3.5	PSR <sub>n</sub> / APSR <sub>n</sub> / JPSR0 — Port Set/Reset Register.....	129
2.9.4	Configuration of Electrical Characteristics .....	130

2.9.4.1	PUn / JPU0 — Pull-Up Option Register .....	130
2.9.4.2	PDn / JPD0 — Pull-Down Option Register.....	131
2.9.4.3	PDSCn — Port Drive Strength Control Register .....	132
2.9.4.4	PODCn / JPODC0 — Port Open Drain Control Register .....	133
2.9.4.5	PISn — Port Input Buffer Selection Register.....	134
2.9.4.6	JPISA0 — Port Input Buffer Selection Advanced Register .....	135
2.9.5	Port Register Protection .....	135
2.9.6	Flowchart Example for Port Settings .....	136
2.9.6.1	Batch Setting.....	136
2.9.6.2	Individual Settings .....	137
2.10	Port (General I/O) Function Overview .....	140
2.10.1	JTAG Port 0 (JP0) .....	141
2.10.1.1	Alternative Function.....	141
2.10.1.2	Control Registers.....	142
2.10.2	Port 0 (P0) .....	143
2.10.2.1	Alternative Functions.....	143
2.10.2.2	Control Registers.....	145
2.10.3	Port 1 (P1) .....	148
2.10.3.1	Alternative Functions.....	148
2.10.3.2	Control Registers.....	149
2.10.4	Port 2 (P2) .....	150
2.10.4.1	Alternative Functions.....	150
2.10.4.2	Control Registers.....	151
2.10.5	Port 8 (P8) .....	152
2.10.5.1	Alternative Functions.....	152
2.10.5.2	Control Registers.....	153
2.10.6	Port 9 (P9) .....	155
2.10.6.1	Alternative Functions.....	155
2.10.6.2	Control Registers.....	156
2.10.7	Port 10 (P10) .....	158
2.10.7.1	Alternative Functions.....	158
2.10.7.2	Control Registers.....	160
2.10.8	Port 11 (P11) .....	163
2.10.8.1	Alternative Functions.....	163
2.10.8.2	Control Registers.....	166
2.10.9	Port 12 (P12) .....	169
2.10.9.1	Alternative Functions.....	169
2.10.9.2	Control Registers.....	170
2.10.10	Port 18 (P18) .....	172
2.10.10.1	Alternative Functions.....	172
2.10.10.2	Control Registers.....	173
2.10.11	Port 20 (P20) .....	174
2.10.11.1	Alternative Functions.....	174
2.10.11.2	Control Registers.....	175
2.10.12	Analog Port 0 (AP0).....	177
2.10.12.1	Alternative Functions.....	177
2.10.12.2	Control Registers.....	178
2.10.13	Analog Port 1 (AP1).....	180

2.10.13.1	Alternative Functions.....	180
2.10.13.2	Control Registers.....	181
2.10.14	Input Port 0 (IP0) .....	182
2.10.14.1	Alternative Functions.....	182
2.10.14.2	Control Registers.....	183
2.11	Port (Special I/O) Function Overview .....	184
2.11.1	Special I/O after Reset .....	184
2.11.1.1	P0_0: RESETOUT .....	184
2.11.1.2	JP0_0 to JP0_6: Debug Interface .....	185
2.11.1.3	FPDR(JP0_0), FPDT(JP0_1), FPCK(JP0_2): Flash Programmer .....	185
2.11.1.4	Mode Pins .....	185
2.11.1.5	IP0_0: XT2 .....	185
2.11.2	A/D Input Alternative I/O.....	186
2.11.3	Special I/O Control .....	188
2.11.3.1	Direct I/O Control (PIPC).....	188
2.11.3.2	Input Buffer Control (PISn, JPISA0).....	190
2.11.3.3	Output Buffer Control (PDSC) .....	192
2.12	Noise Filter & Edge/Level Detector.....	194
2.12.1	Port Filter Assignment .....	194
2.12.1.1	Input Pins that Incorporate Analog Filter Type A.....	194
2.12.1.2	Input Pins that Incorporate Analog Filter Type B.....	195
2.12.1.3	Input Pins that Incorporate Analog Filter Type C .....	195
2.12.1.4	Input Pins that Incorporate Digital Filter Type D.....	196
2.12.1.5	Input Pins that Incorporate Digital Filter Type E.....	198
2.12.2	Clock Supply for Port Filters .....	202
2.13	Description of Port Noise Filter & Edge/Level Detection .....	203
2.13.1	Overview.....	203
2.13.1.1	Analog Filter Types .....	203
2.13.1.2	Digital Filter Types.....	203
2.13.2	Analog Filters.....	204
2.13.2.1	Analog Filter Characteristic .....	204
2.13.2.2	Analog Filter Control Registers .....	204
2.13.2.3	Analog Filter in Standby Mode .....	204
2.13.3	Digital Filters.....	207
2.13.3.1	Digital Filter Characteristic .....	207
2.13.3.2	Digital Filter Groups.....	208
2.13.3.3	Digital Filters in Standby Mode.....	208
2.13.3.4	Digital Filter Control Registers.....	209
2.13.4	Filter Control Registers.....	210
2.13.4.1	FCLA0CTLm_<name> — Filter Control Register.....	211
2.13.4.2	DNFA<name>CTL — Digital Noise Elimination Control Register .....	212
2.13.4.3	DNFA<name>EN — Digital Noise Elimination Enable Register .....	213
Section 3	CPU System .....	214
3.1	Overview.....	214
3.1.1	Architectural Features .....	214
3.1.1.1	CPU .....	214

3.1.1.2	Coprocessor .....	214
3.1.1.3	Exceptions and Interrupts.....	214
3.1.1.4	Memory Management .....	215
3.1.1.5	System Register Protection.....	215
3.1.1.6	Functions Not Supported.....	215
3.1.2	Power Supply and Clock .....	215
3.1.3	RH850/F1L CPU Subsystem.....	216
3.2	Processor Model.....	217
3.2.1	CPU Operating Mode .....	217
3.2.2	Hardware Thread.....	217
3.2.3	CPU Data Address and Physical Program Address Space.....	218
3.2.3.1	Program Space and Data Space.....	219
3.2.3.2	Wrap-Around of Data Space .....	220
3.2.3.3	Wrap-Around of Program Space .....	220
3.2.4	RH850/F1L Memory Map .....	221
3.2.4.1	Code Flash Area .....	222
3.2.4.2	Data Flash Area .....	223
3.2.4.3	Primary Local RAM Area.....	224
3.2.4.4	Secondary Local RAM Area.....	225
3.2.4.5	Retention RAM Area .....	226
3.3	Register Reference.....	227
3.3.1	PC — Program Counter .....	227
3.3.2	Basic System Registers.....	228
3.3.2.1	EIPC — Status Save Register when Acknowledging EI Level Exception .....	229
3.3.2.2	EIPSW — Status Save Register when Acknowledging EI Level Exception.....	230
3.3.2.3	FEPC — Status Save Register when Acknowledging FE Level Exception.....	231
3.3.2.4	FEPSW — Status Save Register when Acknowledging FE Level Exception .....	232
3.3.2.5	PSW — Program Status Word.....	233
3.3.2.6	CTPC — Status Save Register when Executing CALLT.....	235
3.3.2.7	CTBP — CALLT Base Pointer .....	236
3.3.2.8	HTCFG0 — Thread Configuration.....	236
3.3.2.9	MEI — Memory Error Information .....	236
3.3.2.10	RBASE — Reset Vector Base Address .....	239
3.3.2.11	EBASE — Exception Handler Vector Address.....	239
3.3.2.12	INTBP — Base Address of the Interrupt Handler Table.....	240
3.3.2.13	PID — Processor ID .....	240
3.3.2.14	SCBP — SYSCALL Base Pointer .....	241
3.3.2.15	MCFG0 — Machine Configuration 0 .....	241
3.3.2.16	MCTL — Machine Control.....	242
3.3.3	Interrupt Function Registers .....	242
3.3.3.1	ISPR — Priority of Interrupt being Serviced .....	243
3.3.3.2	PMR — Interrupt Priority Masking.....	244
3.3.3.3	ICSR — Interrupt Control Status .....	244
3.3.3.4	INTCFG — Interrupt Function Setting.....	245
3.3.4	MPU Function Registers.....	246
3.3.4.1	MPM — Memory Protection Operation Mode .....	248
3.3.4.2	MPRC — MPU Region Control .....	249
3.3.4.3	MPBRGN — MPU Base Region .....	249

3.3.4.4	MPTRGN — MPU End Region .....	249
3.3.4.5	MPLAn — Protection Area Minimum Address .....	250
3.3.4.6	MPUAn — Protection Area Maximum Address.....	250
3.3.4.7	MPATn — Protection Area Attribute.....	251
3.3.5	Cache Operation Function Registers .....	252
3.4	Instruction Reference.....	253
3.5	System Error Notification Setting Registers.....	253
3.5.1	SEG_CONT — SYSERR Exception Notification Setting Registers .....	254
3.5.2	SEG_FLAG — System Error Source Storing Registers .....	255
3.6	Usage Notes .....	256
3.6.1	Synchronization of Store Instruction Completion and Subsequent Instruction Execution	256
3.6.2	System Register Hazards.....	257
3.6.3	Accessing Registers by Using Bit-Manipulation Instructions.....	257
3.6.4	Overwriting Context upon Acknowledgment of Multiple Exceptions.....	258
3.6.5	Detection of ECC Errors during Prefetching.....	258
<b>Section 4</b>	<b>Write-Protected Registers .....</b>	<b>259</b>
4.1	Overview.....	259
4.1.1	Functional Overview .....	259
4.1.2	Writing Procedure to Write-Protected Registers.....	259
4.1.3	Interrupt during Write Protection Unlock.....	260
4.1.4	Emulation Break during Write Protection Unlock Sequence .....	261
4.1.5	Write-Protection Target Registers .....	261
4.2	Registers.....	264
4.2.1	List of Registers.....	264
4.2.2	Details of Control Protection Cluster Registers .....	266
4.2.2.1	PROTCMDn — Protection Command Register .....	266
4.2.2.2	PROTSn — Protection Status Register.....	267
4.2.3	Details of Clock Monitor Protection Cluster Registers .....	268
4.2.3.1	CLMANPCMD — CLMAN Protection Command Register .....	268
4.2.3.2	CLMANPS — CLMAN Protection Status Register .....	269
4.2.3.3	PROTCMDCLMA — Clock Monitor Test Protection Command Register .....	270
4.2.3.4	PROTSCLMA — Clock Monitor Test Protection Status Register.....	271
4.2.4	Details of Core Voltage Monitor Register .....	272
4.2.4.1	PROTCMDCVM — Core Voltage Monitor Protection Command Register .....	272
4.2.4.2	PROTSCVM — Core Voltage Monitor Protection Status Register.....	273
4.2.5	Details of Port Protection Cluster Registers .....	274
4.2.5.1	PPCMDn — Port Protection Command Register.....	274
4.2.5.2	PPROTSn — Port Protection Status Register.....	275
4.2.6	Details of Self-Programming Protection Cluster Registers .....	276
4.2.6.1	FLMDPCMD — FLMD Protection Command Register .....	276
4.2.6.2	FLMDPS — FLMD Protection Error Status Register.....	277
<b>Section 5</b>	<b>Operating Mode .....</b>	<b>278</b>
<b>Section 6</b>	<b>Exceptions/Interrupts .....</b>	<b>279</b>

6.1	Features.....	279
6.2	RH850/F1L Interrupt Sources.....	281
6.2.1	Interrupt Sources.....	281
6.2.1.1	FE Level Non-Maskable Interrupts.....	281
6.2.1.2	FE Level Maskable Interrupts.....	282
6.2.1.3	EI Level Maskable Interrupts.....	283
6.2.2	FE Level Non-Maskable Interrupt Sources.....	293
6.2.2.1	WDTNMIF — FENMI Factor Register.....	293
6.2.2.2	WDTNMIFC — WDTNMI Factor Clear Register.....	294
6.2.3	FE Level Maskable Interrupt Sources .....	295
6.2.3.1	FEINTF — FEINT Factor Register .....	295
6.2.3.2	FEINTFMSK — FEINT Factor Mask Register.....	297
6.2.3.3	FEINTFC — FEINT Factor Clear Register .....	299
6.3	Edge/Level Detection.....	301
6.4	Interrupt Controller Control Registers .....	302
6.4.1	ICxxx — EI Level Interrupt Control Registers.....	302
6.4.2	IMR <sub>m</sub> — EI Level Interrupt Mask Registers ( $m = 0$ to 8).....	304
6.4.3	FNC — FE Level NMI Status Register .....	305
6.4.4	FIC — FE Level Maskable Interrupt Status Register.....	306
6.5	EI Level Maskable Interrupt Select Registers .....	307
6.5.1	SELB_INTC1 — INTC1 Interrupt Select Register .....	307
6.5.2	SELB_INTC2 — INTC2 Interrupt Select Register .....	309
6.6	Interrupt Function System Registers .....	311
6.6.1	ISPR — Priority of Interrupt being Serviced .....	311
6.6.2	PMR — Interrupt Priority Masking .....	311
6.6.3	ICSR — Interrupt Control Status .....	311
6.6.4	INTCFG — Interrupt Function Setting .....	311
6.7	Operation when Acknowledging an Interrupt.....	312
6.7.1	Exception Source Codes When an SYSERR Exception Occurs.....	314
6.8	Return from Interrupts.....	315
6.9	Interrupt Operation.....	316
6.9.1	Interrupt Mask Function of EI Level Maskable Interrupt (EIINT) .....	316
6.9.2	Interrupt Priority Level Judgment.....	316
6.9.2.1	Comparison with the Priority Level of the Interrupt Currently being Handled.....	317
6.9.2.2	Masking through Priority Mask Register (PMR) .....	317
6.9.2.3	The Requested Interrupt Source with the Highest Priority Level is Selected .....	317
6.9.2.4	Interrupt Hold by CPU .....	317
6.9.3	Interrupt Request Acknowledgement Conditions and the Priority .....	320
6.9.4	Exception Priority of Interrupts and the Priority Mask.....	320
6.9.5	Interrupt Priority Mask .....	320
6.9.6	Priority Mask Function .....	321
6.9.7	Exception Management.....	321
6.10	Exception Handler Address .....	322
6.10.1	Direct Vector Method.....	323
6.10.2	Table Reference Method.....	325

Section 7 DMA .....	327
7.1 Features of RH850/F1L DMA .....	327
7.1.1 Number of Channels.....	327
7.1.2 Register Base Address.....	327
7.1.3 Interrupt Requests .....	328
7.1.4 DMA Trigger Factors .....	329
7.2 Overview.....	333
7.2.1 Functional Overview .....	333
7.2.1.1 DMA Transfer Function .....	333
7.2.1.2 DTFR Function .....	335
7.2.1.3 DMA Access Memory Map.....	335
7.2.1.4 Prioritization of Channels .....	335
7.2.2 Terms .....	336
7.2.3 Block Diagram .....	337
7.3 Registers.....	338
7.3.1 List of Registers.....	338
7.3.2 Enabling or Disabling Writing Control Registers.....	339
7.3.3 DMAC Control Registers .....	340
7.3.3.1 DTRE0 — DMA Transfer Request Control Register 0 .....	340
7.3.3.2 DMCM0 — DMA Channel Master Setting Register 0.....	340
7.3.3.3 DSAm — DMA Source Address Register .....	341
7.3.3.4 DDAm — DMA Destination Address Register.....	343
7.3.3.5 DTClm — DMA Transfer Count Register ( $m = 0$ to 15) .....	345
7.3.3.6 DTCTm — DMA Transfer Control Register ( $m = 0$ to 15) .....	346
7.3.3.7 DTSm — DMA Transfer Status Register ( $m = 0$ to 15) .....	347
7.3.4 DTFR Control Registers .....	349
7.3.4.1 DTFRm — DMA Trigger Factor Register $m$ ( $m = 0$ to 15).....	349
7.3.4.2 DRQCLR — DMA Request Clear Register .....	350
7.3.4.3 DRQSTR — DMA Request Check Register.....	350
7.4 Functions .....	351
7.4.1 DMAC Transfer Modes.....	351
7.4.1.1 Single Transfer Mode (when hardware DMA transfer request is generated) .....	351
7.4.1.2 Single-Step Transfer Mode (when software DMA transfer request is generated). ....	352
7.4.2 DMAC Channel Priority Control.....	353
7.4.3 Valid DMAC Transfer Request Conditions .....	354
7.4.4 Suspending/Resuming/Forcibly Terminating DMA Transfer .....	355
7.4.4.1 Suspending or Resuming DMA Transfer for All Channels through Software.....	355
7.4.4.2 Suspending or Resuming DMA Transfer for Each Channel using DMA Transfer Enable Bit (DTSmDTE) .....	355
7.4.4.3 Suspending or Resuming DMA Transfer by using Software DMA Transfer Request Bit (DTSmSR) .....	355
7.4.4.4 Forcibly Terminating Suspended DMA Transfer .....	355
7.4.5 Error Response .....	356
7.4.5.1 Suspending DMA Transfer by Error Response.....	356
7.4.5.2 Canceling Transfer Suspend by Error Response.....	356
7.4.6 Stand-By Support .....	356
7.4.7 Loop function .....	356

7.5	Protection Function .....	357
7.6	DMAC Transfer Setting Flow .....	358
<b>Section 8</b>	<b>Reset Controller .....</b>	<b>359</b>
8.1	Overview .....	359
8.1.1	Reset Sources .....	359
8.1.2	Reset Controller Redundancy .....	360
8.1.3	Reset Output ( <u>RESETOUT</u> ) .....	361
8.1.4	Reset Flag .....	361
8.1.5	Clock Supply.....	361
8.2	Configuration .....	362
8.2.1	Block Diagram .....	362
8.3	Registers.....	365
8.3.1	Reset Controller Registers Overview .....	365
8.3.2	Details of Reset Flag Registers .....	366
8.3.2.1	RESF — Reset Factor Register .....	366
8.3.2.2	RESFC — Reset Factor Clear Register .....	368
8.3.2.3	RESFR — Redundant Reset Factor Register .....	369
8.3.2.4	RESFCR — Redundant Reset Factor Clear Register.....	371
8.3.3	Details of Software Reset Control Registers .....	372
8.3.3.1	SWRESA — Software Reset Register .....	372
8.4	Functional Description .....	373
8.4.1	Reset Flags .....	373
8.4.2	Power-On-Clear (POC) Reset .....	373
8.4.3	Low-Voltage Indicator (LVI) Reset.....	376
8.4.4	Core Voltage Monitor (CVM) Reset.....	377
8.4.5	External Reset ( <u>RESET</u> ).....	378
8.4.6	Watchdog Timer (WDTA) Reset.....	379
8.4.7	Software Reset.....	379
8.4.8	Clock Monitor (CLMA) Reset.....	380
8.4.9	Debugger Reset .....	380
<b>Section 9</b>	<b>Supply Voltage Monitor .....</b>	<b>381</b>
9.1	Overview .....	381
9.1.1	Functional Overview .....	381
9.1.2	Power-On Clear (POC).....	381
9.1.3	Low Voltage Indicator Circuit (LVI) .....	381
9.1.3.1	LVI Reference Voltage .....	381
9.1.3.2	LVI Reset (LVIRES) .....	382
9.1.3.3	LVI Interrupt (INTLVL / INTLVIH) .....	382
9.1.3.4	LVI Setting Procedure .....	383
9.1.3.5	Clock Supply to the LVI .....	383
9.1.4	Core Voltage Monitor (CVM) .....	384
9.1.4.1	CVM Reset ( <u>CVMRES</u> ) .....	384
9.1.4.2	CVM Setting .....	384
9.1.4.3	Diagnostic (DIAG) Mode .....	384

9.1.4.4	Clock Supply to the CVM .....	385
9.1.5	RAM Retention Voltage Indicator (Very-Low-Voltage Detection Circuit:VLVI) .....	386
9.1.5.1	Clock Supply to the VLVI.....	386
9.1.5.2	Retention RAM Content Retention.....	386
9.1.6	Block Diagram .....	387
9.2	Registers.....	388
9.2.1	List of Registers.....	388
9.2.2	Low-Voltage Indicator Reset Control Registers.....	389
9.2.2.1	LVICNT — LVI Control Register.....	389
9.2.3	Core Voltage Monitor Control Registers.....	390
9.2.3.1	CVMF — CVM Factor Register .....	390
9.2.3.2	CVMDE — CVM Detection Enable Register .....	391
9.2.3.3	CVMDIAG — CVM Diagnostic Mode Setting Register.....	392
9.2.4	Very-Low-Voltage Detection Control Registers .....	393
9.2.4.1	VLVF — Very-Low-Voltage Detection Register.....	393
9.2.4.2	VLVFC — Very-Low-Voltage Detection Clear Register .....	394
<b>Section 10</b>	<b>Clock Controller .....</b>	<b>395</b>
10.1	Features of Clock Controller of RH850/F1L.....	395
10.2	Configuration of Clock Controller.....	397
10.2.1	Clock Generation Circuits.....	399
10.2.2	Clock Selection.....	400
10.2.3	Clock Domains .....	401
10.2.4	Resetting Clock Oscillators.....	401
10.3	Clock Oscillators .....	402
10.3.1	Main Oscillator (MainOSC).....	402
10.3.2	Sub Oscillator (SubOSC).....	404
10.3.3	High Speed Internal Oscillator (HS IntOSC).....	405
10.3.4	Low Speed Internal Oscillator (LS IntOSC) .....	406
10.3.5	PLL .....	407
10.3.5.1	PLL Parameters .....	408
10.4	Registers.....	410
10.4.1	List of Registers.....	410
10.4.2	Clock Oscillator Registers .....	412
10.4.2.1	MOSCE — MainOSC Enable Register .....	412
10.4.2.2	MOSCS — MainOSC Status Register .....	413
10.4.2.3	MOSCC — MainOSC Control Register.....	414
10.4.2.4	MOSCST — MainOSC Stabilization Time Register .....	415
10.4.2.5	MOSCSTPM — MainOSC Stop Mask Register .....	416
10.4.2.6	SOSCE — SubOSC Enable Register .....	417
10.4.2.7	SOSCS — SubOSC Status Register .....	418
10.4.2.8	SOSCST — SubOSC Stabilization Time Register .....	419
10.4.2.9	ROSCE — HS IntOSC Enable Register.....	420
10.4.2.10	ROSCS — HS IntOSC Status Register.....	421
10.4.2.11	ROSCSTPM — HS IntOSC Stop Mask Register .....	422
10.4.2.12	PLLE — PLL Enable Register .....	423
10.4.2.13	PLLS — PLL Status Register .....	424

10.4.2.14	PLL C — PLL Control Register .....	425
10.4.3	Clock Selector Control Register .....	434
10.4.3.1	WDTA0 Clock Domain C_AWO_WDTA.....	434
10.4.3.2	TAUJ Clock Domain C_AWO_TAUJ.....	437
10.4.3.3	RTCA Clock Domain C_AWO_RTCA.....	442
10.4.3.4	ADCA0 Clock Domain C_AWO_ADCA.....	447
10.4.3.5	FOUT Clock Domain C_AWO_FOUT .....	452
10.4.3.6	CPU Clock Domain C_ISO_CPUCLK.....	455
10.4.3.7	Peripheral Clock Domains C_ISO_PERI1 and C_ISO_PERI2 .....	459
10.4.3.8	RLIN Clock Domains C_ISO_LIN.....	463
10.4.3.9	ADCA1 Clock Domain C_ISO_ADCA .....	468
10.4.3.10	RS-CAN Clock Domains C_ISO_CAN and C_ISO_CANOSC .....	472
10.4.3.11	CSI Clock Domain C_ISO_CSI .....	478
10.5	Clock Domain Setting Method .....	480
10.5.1	Clock Domain Setting .....	480
10.5.1.1	Overview of Clock Selector Register.....	480
10.5.1.2	Setting Procedure for Clock Domain .....	481
10.5.2	Stopping the Clock in Stand-By Mode.....	481
10.5.3	Clock Domain Settings .....	482
10.6	Frequency Output Function (FOUT) .....	484
10.6.1	Functional Overview .....	484
10.6.2	Registers .....	485
10.6.2.1	List of Registers.....	485
10.6.2.2	Clock Supply .....	485
10.6.2.3	FOUTDIV — Clock Division Ratio Register.....	486
10.6.2.4	FOUTSTAT — Clock Divider Status Register.....	487
10.7	Clock Monitor A (CLMA) .....	488
10.7.1	Features of RH850/F1L CLMA .....	488
10.7.1.1	Number of Channels .....	488
10.7.1.2	Register Base Addresses.....	488
10.7.1.3	Clock Supply .....	488
10.7.1.4	Reset Sources.....	489
10.7.1.5	Internal Input/Output Signals .....	489
10.7.2	Overview.....	489
10.7.2.1	Functional Overview .....	489
10.7.3	Enabling CLMA.....	491
10.7.4	Functions .....	491
10.7.4.1	Detection of Abnormal Clock Frequencies .....	491
10.7.4.2	Notification of Abnormal Clock Frequency .....	493
10.7.5	Registers .....	494
10.7.5.1	List of Registers.....	494
10.7.5.2	CLMACTL0 — CLMA Control Register 0 .....	495
10.7.5.3	CLMACMPH — CLMA Compare Register H .....	496
10.7.5.4	CLMACMPL — CLMA Compare Register L .....	497
10.7.5.5	CLMATEST — CLMA Test Register .....	498
10.7.5.6	CLMATESTS — CLMA Test Status Register.....	499
10.7.5.7	CLMAEMU0 — CLMA Emulation Register 0 .....	500
10.7.6	Usage Notes for CLMA.....	501

Section 11 Stand-By Controller (STBC).....	502
11.1 Functions .....	502
11.1.1 Types of Stand-By Mode .....	502
11.1.2 Wake-Up Control.....	503
11.1.2.1 Wake-Up Factors for Stand-By Modes.....	503
11.1.2.2 Setting of Wake-Up Factors .....	505
11.1.3 On-Chip Debug Wake-Up.....	508
11.1.4 I/O Buffer Control.....	509
11.1.4.1 I/O Buffer Hold State .....	509
11.1.4.2 I/O Buffers during STOP Mode .....	509
11.1.4.3 I/O Buffers during DeepSTOP Mode.....	509
11.1.5 Transition to Stand-By Mode .....	510
11.1.6 Clock Supply.....	510
11.2 Registers.....	511
11.2.1 List of Registers.....	511
11.2.2 Details of Stand-By Controller Control Registers .....	512
11.2.2.1 STBC0PSC — Power Save Control Register .....	512
11.2.2.2 STBC0STPT — Power Stop Trigger Register.....	513
11.2.2.3 WUF0/WUF20/WUF_ISO0 — Wake-Up Factor Registers.....	514
11.2.2.4 WUFMSK0/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers...	515
11.2.2.5 WUFC0/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers .....	516
11.2.2.6 IOHOLD — I/O Buffer Hold Control Register .....	517
11.3 Mode Transition .....	518
11.3.1 STOP Mode.....	518
11.3.2 DeepSTOP Mode .....	520
11.3.3 Cyclic RUN Mode .....	523
11.3.4 Cyclic STOP Mode .....	525
11.4 Writing to the Stand-By Controller Related Registers.....	526
11.5 Cautions when Using Stand-By Modes .....	526
11.5.1 Cautions Concerning Transitioning to DeepSTOP Mode When Using a Debugger.....	526
Section 12 Low-Power Sampler (LPS) .....	527
12.1 Features of RH850/F1L LPS .....	527
12.1.1 Number of Units .....	527
12.1.2 Register Base Address.....	528
12.1.3 Clock Supply.....	528
12.1.4 Interrupt Request.....	528
12.1.5 Reset Sources .....	529
12.1.6 External Input/Output Signals.....	529
12.1.7 Internal Input/Output Signals .....	529
12.2 Overview.....	530
12.2.1 Functional Overview .....	530
12.3 Registers.....	531
12.3.1 List of Registers.....	531
12.3.2 SCTRLR — LPS Control Register .....	532

12.3.3	EVFR — Event Flag Register .....	534
12.3.4	DPSELRL0 — DPIN Select Register 0.....	535
12.3.5	DPSELRM — DPIN Select Register M.....	536
12.3.6	DPSELRH — DPIN Select Register H .....	537
12.3.7	DPDSR0 — DPIN Data Set Register 0 .....	538
12.3.8	DPDSRM — DPIN Data Set Register M .....	539
12.3.9	DPDSRH — DPIN Data Set Register H .....	540
12.3.10	DPDIMR0 — DPIN Data Input Monitor Register 0 .....	541
12.3.11	DPDIMR1 — DPIN Data Input Monitor Register 1 .....	541
12.3.12	DPDIMR2 — DPIN Data Input Monitor Register 2 .....	542
12.3.13	DPDIMR3 — DPIN Data Input Monitor Register 3 .....	542
12.3.14	DPDIMR4 — DPIN Data Input Monitor Register 4 .....	543
12.3.15	DPDIMR5 — DPIN Data Input Monitor Register 5 .....	543
12.3.16	DPDIMR6 — DPIN Data Input Monitor Register 6 .....	544
12.3.17	DPDIMR7 — DPIN Data Input Monitor Register 7 .....	544
12.3.18	CNTVAL — Count Value Register.....	545
12.3.19	SOSTR — LPS Operation Status Register .....	546
12.4	Digital Input Mode.....	547
12.5	Analog Input Mode.....	552

<b>Section 13</b>	<b>External Memory Access Controller (MEMC) .....</b>	<b>556</b>
13.1	Features of MEMC.....	556
13.1.1	Products that Incorporate MEMC .....	556
13.1.2	Register Base Address.....	556
13.1.3	Clock Supply.....	557
13.1.4	Reset Sources .....	557
13.1.5	External Input/Output Signals.....	557
13.2	Overview.....	558
13.2.1	Functional Overview .....	558
13.2.1.1	Multiplexed Bus .....	558
13.2.1.2	Chip Select Output Function .....	558
13.2.1.3	Bus Sizing Function.....	558
13.2.1.4	Data Endian Setting Function.....	558
13.2.1.5	Programmable Wait Setting Functions .....	558
13.2.1.6	External Wait Function .....	559
13.3	Registers.....	560
13.3.1	List of Registers.....	560
13.3.2	BSC — Data Bus Width Configuration Register.....	561
13.3.3	DEC — Data Endian Configuration Register.....	562
13.3.4	DWC — Data Wait Configuration Register.....	563
13.3.5	DHC — Data Hold Wait Configuration Register .....	564
13.3.6	AWC — Address Wait Configuration Register .....	565
13.3.7	ICC — Idle Cycle Configuration Register .....	566
13.4	Functions .....	567
13.4.1	Bus Control Functions .....	567

13.4.1.1	Chip Select Output Function .....	567
13.4.1.2	Bus Sizing Function.....	568
13.4.1.3	Data Endian Setting Function.....	568
13.4.2	Wait Functions.....	569
13.4.2.1	Programmable Data Wait Function .....	570
13.4.2.2	External Wait Function .....	571
13.4.2.3	Data Hold Wait Function .....	572
13.4.2.4	Address Setup Wait and Address Hold Wait Functions .....	573
13.4.2.5	Idle Insertion Function .....	574
13.4.3	Memory Connection Example .....	575
13.4.3.1	SRAM Connection Example.....	575
13.4.4	Data Access Flow.....	576
13.4.4.1	Data Flow for Byte Access (for Reading and Writing) .....	577
13.4.4.2	Data Flow for Half-Word Read Access.....	579
13.4.4.3	Data Flow for Half-Word Write Access.....	582
13.4.4.4	Data Flow for Word Read Access .....	585
13.4.4.5	Data Flow for Word Write Access .....	590
<b>Section 14</b>	<b>Clocked Serial Interface G (CSIG).....</b>	<b>595</b>
14.1	Features of RH850/F1L CSIG .....	595
14.1.1	Number of Units.....	595
14.1.2	Register Base Address.....	595
14.1.3	Clock Supply.....	595
14.1.4	Interrupt Requests .....	596
14.1.5	Reset Sources .....	596
14.1.6	External Input/Output Signals.....	597
14.1.7	Data Consistency Check .....	597
14.2	Overview.....	598
14.2.1	Functional Overview .....	598
14.2.2	Functional Overview Description .....	598
14.2.3	Block Diagram .....	599
14.3	Registers.....	600
14.3.1	List of Registers.....	600
14.3.2	CSIGnCTL0 — CSIGn Control Register 0.....	601
14.3.3	CSIGnCTL1 — CSIGn Control Register 1.....	602
14.3.4	CSIGnCTL2 — CSIGn Control Register 2.....	604
14.3.5	CSIGnSTR0 — CSIGn Status Register 0.....	605
14.3.6	CSIGnSTCR0 — CSIGn Status Clear Register 0 .....	607
14.3.7	CSIGnBCTL0 — CSIGn Rx-Only Mode Control Register 0 .....	608
14.3.8	CSIGnCFG0 — CSIGn Configuration Register 0 .....	609
14.3.9	CSIGnTX0W — CSIGn Transmission Register 0 for Word Access .....	611
14.3.10	CSIGnTX0H — CSIGn Transmission Register 0 for Half Word Access.....	612
14.3.11	CSIGnRX0 — CSIGn Reception Register 0.....	612
14.3.12	CSIGnEMU — CSIGn Emulation Register .....	613
14.3.13	List of Cautions.....	614
14.4	Interrupt Sources .....	615

14.4.1	Interrupt Delay .....	615
14.4.2	INTCSIGTIC (Communication Status Interrupt) .....	616
14.4.3	INTCSIGTIR (Reception Status Interrupt) .....	617
14.4.4	INTCSIGTIRE (Communication Error Interrupt) .....	617
14.5	Operation .....	618
14.5.1	Master/Slave Mode .....	618
14.5.1.1	Master Mode .....	618
14.5.1.2	Slave Mode .....	619
14.5.2	Master/Slave Connections .....	620
14.5.2.1	One Master and One Slave .....	620
14.5.2.2	One Master and Multiple Slaves .....	621
14.5.3	Transmission Clock Selection .....	622
14.5.4	Data Transfer Modes .....	623
14.5.4.1	Transmit-Only Mode .....	623
14.5.4.2	Receive-Only Mode .....	623
14.5.4.3	Transmit/Receive Mode .....	623
14.5.5	Data Length Selection .....	624
14.5.5.1	Data Length Selection without Extended Length .....	624
14.5.5.2	Data Length Selection with Extended Data Length .....	624
14.5.6	Serial Data Direction Selection Function .....	626
14.5.7	Communication in Slave Mode .....	627
14.5.8	Handshake Function .....	628
14.5.8.1	Slave Mode .....	628
14.5.8.2	Master Mode .....	629
14.5.9	Loop-Back Mode .....	630
14.5.10	Error Detection .....	631
14.5.10.1	Data Consistency Check .....	631
14.5.10.2	Parity Check .....	632
14.5.10.3	Overrun Error .....	632
14.6	Operating Procedures .....	634
14.6.1	Master Mode Transmission/Reception by DMA .....	634
<b>Section 15</b>	<b>Clocked Serial Interface H (CSIH)</b> .....	<b>636</b>
15.1	Features of RH850/F1L CSIH .....	636
15.1.1	Number of Units .....	636
15.1.2	Register Base Address .....	637
15.1.3	Clock Supply .....	637
15.1.4	Interrupt Requests .....	638
15.1.5	Reset Sources .....	638
15.1.6	External Input/Output Signals .....	639
15.1.7	Data Consistency Check .....	640
15.2	Overview .....	641
15.2.1	Functional Overview .....	641
15.2.2	Functional Overview Description .....	642
15.2.3	Block Diagram .....	643
15.3	Registers .....	644

15.3.1	List of Registers.....	644
15.3.2	CSIHnCTL0 — CSIHn Control Register 0 .....	645
15.3.3	CSIHnCTL1 — CSIHn Control Register 1 .....	646
15.3.4	CSIHnCTL2 — CSIHn Control Register 2 .....	648
15.3.5	CSIHnSTR0 — CSIHn Status Register 0 .....	650
15.3.6	CSIHnSTCR0 — CSIHn Status Clear Register 0.....	654
15.3.7	CSIHnMCTL0 — CSIHn Memory Control Register 0 .....	655
15.3.8	CSIHnMCTL1 — CSIHn Memory Control Register 1 .....	656
15.3.9	CSIHnMCTL2 — CSIHn Memory Control Register 2 .....	657
15.3.10	CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0.....	659
15.3.11	CSIHnCFGx — CSIHn Configuration Register x.....	661
15.3.12	CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access .....	666
15.3.13	CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access.....	668
15.3.14	CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access .....	669
15.3.15	CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access .....	670
15.3.16	CSIHnEMU — CSIHn Emulation Register .....	670
15.3.17	CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3).....	671
15.3.18	List of Cautions.....	672
15.4	Interrupt Sources .....	674
15.4.1	Overview.....	674
15.4.2	Interrupt Delay .....	675
15.4.3	INTCSIHTIC (Communication Status Interrupt) .....	675
15.4.3.1	INTCSIHTIC in Direct Access Mode .....	676
15.4.3.2	INTCSIHTIC in FIFO Mode .....	677
15.4.3.3	INTCSIHTIC in Job Mode.....	678
15.4.4	INTCSIHTIR (Reception Status Interrupt) .....	679
15.4.4.1	INTCSIHTIR in Direct Access Mode .....	679
15.4.4.2	INTCSIHTIR in Dual Buffer Mode .....	680
15.4.5	INTCSIHTIRE (Communication Error Interrupt) .....	680
15.4.6	INTCSIHTIJC (Job Completion Interrupt).....	681
15.5	Operation .....	682
15.5.1	Operating Modes (Master/Slave).....	682
15.5.1.1	Master Mode .....	682
15.5.1.2	Slave Mode .....	683
15.5.2	Master/Slave Connections.....	684
15.5.2.1	One Master and One Slave.....	684
15.5.2.2	One Master and Multiple Slaves .....	684
15.5.3	Chip Selection (CS) Features.....	686
15.5.3.1	Configuration Registers.....	686
15.5.3.2	CS Example .....	688
15.5.3.3	Job Concept .....	688
15.5.4	Details of Chip Select Timing .....	689
15.5.4.1	Changing the Clock Phase.....	689
15.5.4.2	Changing the Data Phase .....	690
15.5.5	Transmission Clock Selection .....	691
15.5.6	CSIH Buffer Memory .....	693
15.5.6.1	FIFO Mode .....	693

15.5.6.2	Dual Buffer Mode .....	694
15.5.6.3	Transmit-Only Buffer Mode .....	694
15.5.6.4	Direct Access Mode .....	694
15.5.7	Data Transfer Modes .....	695
15.5.7.1	Transmit-Only Mode .....	695
15.5.7.2	Receive-Only Mode .....	695
15.5.7.3	Transmit/Receive Mode .....	695
15.5.7.4	Summary .....	695
15.5.8	Data Length Selection .....	696
15.5.8.1	Data Length from 2 to 16 Bits .....	696
15.5.8.2	Data Length Greater than 16 Bits .....	697
15.5.9	Serial Data Direction Selection .....	699
15.5.10	Slave Select (SS) Function .....	700
15.5.10.1	Communication Timing Using SS Function .....	700
15.5.10.2	CSIHTSSO Operation .....	701
15.5.11	Handshake Function .....	702
15.5.11.1	Slave Mode .....	702
15.5.11.2	Master Mode .....	705
15.5.12	Error Detection .....	706
15.5.12.1	Data Consistency Check .....	706
15.5.12.2	Parity Check .....	708
15.5.12.3	Time-Out Error .....	709
15.5.12.4	Overflow Error .....	710
15.5.12.5	Overrun Error .....	712
15.5.13	Loop-Back Mode .....	715
15.5.14	CPU-Controlled High Priority Communication Function .....	716
15.5.15	Enforced Chip Select Idle Setting .....	719
15.6	Operating Procedures .....	720
15.6.1	Procedures in Direct Access Mode .....	720
15.6.1.1	Transmission/Reception in Master Mode when Job Mode is Disabled .....	720
15.6.1.2	Transmission/Reception in Master Mode when Job Mode is Enabled .....	722
15.6.2	Procedures in Transmit-Only Buffer Mode .....	724
15.6.2.1	Transmission/Reception in Master Mode when Job Mode is Disabled .....	724
15.6.2.2	Transmission/Reception in Master Mode when Job Mode is Enabled .....	726
15.6.3	Procedures in Dual Buffer Mode .....	728
15.6.3.1	Transmission/Reception in Master Mode when Job Mode is Disabled .....	728
15.6.3.2	Transmission/Reception in Master Mode when Job Mode is Enabled .....	730
15.6.3.3	Transmit/Receive in Slave Mode when Job Mode is Disabled .....	732
15.6.4	Procedures in FIFO Mode .....	734
15.6.4.1	Transmission/Reception in Master Mode when Job Mode is Disabled .....	734
15.6.4.2	Transmit/Receive Mode when Job Mode is Enabled in Master Mode .....	736
15.7	Detection and Correction of Errors in CSIHn RAM .....	738
15.7.1	ECC for the CSIHn RAM .....	738
15.7.2	Interrupt Requests .....	738
15.7.3	ECCCSIHnCTL — CSIHn ECC Control Register .....	739
15.7.4	ECCCSIHnTMC — CSIHn ECC Test Mode Control Register .....	741
15.7.5	ECCCSIHnTED — CSIHn ECC Encode/Decode Input/Output Replacement Test Register .....	743

15.7.6	ECCCSIHnTRC — CSI <sub>n</sub> ECC Redundant Bit Data Control Test Register.....	744
15.7.7	ECCCSIHnSYND — CSI <sub>n</sub> ECC Decode Syndrome Data Register.....	744
15.7.8	ECCCSIHnHORD — CSI <sub>n</sub> ECC 7-Bit Redundant Bit Data Hold Test Register .....	745
15.7.9	ECCCSIHnECRD — CSI <sub>n</sub> ECC Encode Test Register.....	745
15.7.10	ECCCSIHnERDB — CSI <sub>n</sub> ECC Redundant Bit Input/Output Replacement Buffer Register .....	746
15.7.11	SELB_READTEST — ECCREAD Test Select Register.....	747
<b>Section 16</b>	<b>LIN Master Interface (RLIN2).....</b>	<b>748</b>
16.1	Features of RH850/F1L RLIN2.....	748
16.1.1	Number of Units and Channels .....	748
16.1.2	Register Base Address.....	749
16.1.3	Clock Supply.....	749
16.1.4	Interrupt Request.....	749
16.1.5	Reset Sources .....	750
16.1.6	External Input/Output Signals.....	750
16.2	Overview.....	751
16.2.1	Functional Overview .....	751
16.2.2	Block Diagram .....	752
16.3	Registers.....	753
16.3.1	List of Registers.....	753
16.3.2	Global Registers .....	755
16.3.2.1	RLN24nGLWBR / RLN21nGLWBR — LIN Wake-Up Baud Rate Select Register	755
16.3.2.2	RLN24nGLRP0 / RLN21nGLRP0 — LIN Baud Rate Prescaler 0 Register.....	756
16.3.2.3	RLN24nGLRP1 / RLN21nGLRP1 — LIN Baud Rate Prescaler 1 Register.....	757
16.3.2.4	RLN24nGLSTC / RLN21nGLSTC — LIN Self-Test Control Register .....	758
16.3.3	Channel Registers .....	759
16.3.3.1	RLN24nmLiMD / RLN21nmLiMD — LIN Mode Register.....	759
16.3.3.2	RLN24nmLiBFC / RLN21nmLiBFC — LIN Break Field Configuration Register....	760
16.3.3.3	RLN24nmLiSC / RLN21nmLiSC — LIN Space Configuration Register .....	761
16.3.3.4	RLN24nmLiWUP / RLN21nmLiWUP — LIN Wake-Up Configuration Register....	762
16.3.3.5	RLN24nmLiIE / RLN21nmLiIE — LIN Interrupt Enable Register .....	763
16.3.3.6	RLN24nmLiEDE / RLN21nmLiEDE — LIN Error Detection Enable Register.....	764
16.3.3.7	RLN24nmLiCUC / RLN21nmLiCUC — LIN Control Register.....	766
16.3.3.8	RLN24nmLiTRC / RLN21nmLiTRC — LIN Transmission Control Register .....	767
16.3.3.9	RLN24nmLiMST / RLN21nmLiMST — LIN Mode Status Register .....	768
16.3.3.10	RLN24nmLiST / RLN21nmLiST — LIN Status Register .....	769
16.3.3.11	RLN24nmLiEST / RLN21nmLiEST — LIN Error Status Register.....	771
16.3.3.12	RLN24nmLiDFC / RLN21nmLiDFC — LIN Data Field Configuration Register .....	773
16.3.3.13	RLN24nmLiIDB / RLN21nmLiIDB — LIN ID Buffer Register.....	775
16.3.3.14	RLN24nmLiCBR / RLN21nmLiCBR — LIN Checksum Buffer Register .....	776
16.3.3.15	RLN24nmLiDBRb / RLN21nmLiDBRb — LIN Data Buffer b Register .....	777
16.4	Interrupt Sources .....	779
16.5	Modes .....	780
16.6	LIN Reset Mode .....	782
16.7	LIN Operation Mode .....	783

16.8	LIN Wake-Up Mode .....	783
16.9	Header Transmission/Response Transmission/Response Reception.....	784
16.9.1	Header Transmission .....	784
16.9.2	Response Transmission .....	785
16.9.3	Response Reception .....	786
16.10	Data Transmission/Reception.....	787
16.10.1	Data Transmission.....	787
16.10.2	Data Reception.....	788
16.11	Transmission/Reception Data Buffering .....	789
16.11.1	Transmission of LIN Frames .....	789
16.11.2	Reception of LIN Frames.....	790
16.12	Wake-Up Transmission/Reception .....	791
16.12.1	Wake-Up Transmission .....	791
16.12.2	Wake-Up Reception .....	792
16.12.3	Wake-Up Collision.....	792
16.13	Status.....	793
16.14	Error Status.....	794
16.14.1	Types of Error Statuses.....	794
16.14.2	Target Time Domain for Error Detection .....	795
16.15	LIN Self-Test Mode.....	796
16.15.1	Transition to LIN Self-Test Mode.....	797
16.15.2	Transmission in LIN Self-Test Mode .....	798
16.15.3	Reception in LIN Self-Test Mode.....	799
16.15.4	Exiting LIN Self-Test Mode.....	800
16.16	Baud Rate Generator.....	801
<b>Section 17</b>	<b>LIN/UART Interface (RLIN3) .....</b>	<b>802</b>
17.1	Features of RH850/F1L RLIN3.....	802
17.1.1	Number of Units and Channels .....	802
17.1.2	Register Base Address.....	803
17.1.3	Clock Supply.....	803
17.1.4	Interrupt Request.....	804
17.1.5	Reset Sources .....	804
17.1.6	External Input/output Signals.....	805
17.2	Overview.....	806
17.2.1	Functional Overview .....	806
17.2.2	Block Diagram .....	810
17.2.3	Terms used in block diagram .....	810
17.3	Registers.....	811
17.3.1	List of Registers.....	811
17.3.2	LIN Master Related Registers .....	813
17.3.2.1	RLN3nLWBR — LIN Wake-Up Baud Rate Select Register .....	813
17.3.2.2	RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register .....	814
17.3.2.3	RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register .....	815

17.3.2.4	RLN3nLSTC — LIN Self-Test Control Register .....	816
17.3.2.5	RLN3nLMD — LIN Mode Register .....	817
17.3.2.6	RLN3nLBFC — LIN Break Field Configuration Register.....	819
17.3.2.7	RLN3nLSC — LIN Space Configuration Register .....	820
17.3.2.8	RLN3nLWUP — LIN Wake-Up Configuration Register.....	821
17.3.2.9	RLN3nLIE — LIN Interrupt Enable Register.....	822
17.3.2.10	RLN3nLEDE —LIN Error Detection Enable Register.....	824
17.3.2.11	RLN3nLCUC — LIN Control Register .....	826
17.3.2.12	RLN3nLTRC — LIN Transmission Control Register .....	827
17.3.2.13	RLN3nLMST — LIN Mode Status Register.....	828
17.3.2.14	RLN3nLST — LIN Status Register.....	829
17.3.2.15	RLN3nLEST — LIN Error Status Register .....	831
17.3.2.16	RLN3nLDFC — LIN Data Field Configuration Register .....	833
17.3.2.17	RLN3nLIDB — LIN ID Buffer Register .....	835
17.3.2.18	RLN3nLCBR — LIN Checksum Buffer Register .....	836
17.3.2.19	RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8) .....	837
17.3.3	LIN Slave Related Registers .....	839
17.3.3.1	RLN3nLWBR — LIN Wake-Up Baud Rate Select Register .....	839
17.3.3.2	RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register.....	840
17.3.3.3	RLN3nLSTC — LIN Self-Test Control Register .....	841
17.3.3.4	RLN3nLMD — LIN Mode Register .....	842
17.3.3.5	RLN3nLBFC — LIN Break Field Configuration Register.....	843
17.3.3.6	RLN3nLSC — LIN Space Configuration Register .....	844
17.3.3.7	RLN3nLWUP — LIN Wake-Up Configuration Register.....	845
17.3.3.8	RLN3nLIE — LIN Interrupt Enable Register.....	846
17.3.3.9	RLN3nLEDE — LIN Error Detection Enable Register.....	848
17.3.3.10	RLN3nLCUC — LIN Control Register .....	850
17.3.3.11	RLN3nLTRC — LIN Transmission Control Register .....	851
17.3.3.12	RLN3nLMST — LIN Mode Status Register.....	852
17.3.3.13	RLN3nLST — LIN Status Register.....	853
17.3.3.14	RLN3nLEST — LIN Error Status Register .....	855
17.3.3.15	RLN3nLDFC — LIN Data Field Configuration Register .....	857
17.3.3.16	RLN3nLIDB — LIN ID Buffer Register .....	859
17.3.3.17	RLN3nLCBR — LIN Checksum Buffer Register .....	860
17.3.3.18	RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8) .....	861
17.3.4	UART Related Registers .....	862
17.3.4.1	RLN3nLWBR — LIN Wake-Up Baud Rate Select Register .....	862
17.3.4.2	RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register.....	863
17.3.4.3	RLN3nLMD — UART Mode Register.....	864
17.3.4.4	RLN3nLBFC — UART Configuration Register.....	865
17.3.4.5	RLN3nLSC — UART Space Configuration Register.....	867
17.3.4.6	RLN3nLEDE —UART Error Detection Enable Register.....	868
17.3.4.7	RLN3nLCUC — UART Control Register .....	869
17.3.4.8	RLN3nLTRC — UART Transmission Control Register .....	870
17.3.4.9	RLN3nLMST — UART Mode Status Register.....	871
17.3.4.10	RLN3nLST — UART Status Register.....	872
17.3.4.11	RLN3nLEST — UART Error Status Register .....	874
17.3.4.12	RLN3nLDFC — UART Data Field Configuration Register .....	876
17.3.4.13	RLN3nLIDB — UART ID Buffer Register .....	877

17.3.4.14	RLN3nLUDB0 — UART Data Buffer 0 Register.....	877
17.3.4.15	RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8) .....	878
17.3.4.16	RLN3nLUOER — UART Operation Enable Register .....	879
17.3.4.17	RLN3nLUOR1 — UART Option Register 1.....	880
17.3.4.18	RLN3nLUTDR — UART Transmission Data Register .....	882
17.3.4.19	RLN3nLURDR — UART Reception Data Register .....	883
17.3.4.20	RLN3nLUWTDR — UART Wait Transmission Data Register.....	884
17.4	Interrupt Sources .....	885
17.5	Modes .....	886
17.6	LIN Reset Mode.....	888
17.7	LIN Mode .....	889
17.7.1	LIN Master Mode .....	892
17.7.1.1	Header Transmission .....	892
17.7.1.2	Response Transmission.....	893
17.7.1.3	Response Reception.....	894
17.7.2	LIN Slave Mode .....	895
17.7.2.1	Header Reception .....	895
17.7.2.2	Response Transmission.....	898
17.7.2.3	Response Reception.....	899
17.7.2.4	No-response Request .....	900
17.7.3	Data Transmission/Reception .....	901
17.7.3.1	Data Transmission .....	901
17.7.3.2	Data Reception.....	902
17.7.4	Transmission/Reception Data Buffering .....	903
17.7.4.1	Transmission of LIN Frames .....	903
17.7.4.2	Reception of LIN Frames .....	904
17.7.4.3	Multi-Byte Response Transmission/Reception Function.....	905
17.7.5	Wake-up Transmission/Reception.....	906
17.7.5.1	Wake-up Transmission.....	906
17.7.5.2	Wake-up Reception.....	907
17.7.5.3	Wakeup Collision.....	907
17.7.6	Statuses.....	908
17.7.7	Error Statuses.....	910
17.7.7.1	LIN Master Mode .....	910
17.7.7.2	LIN Slave Mode .....	912
17.8	UART Mode .....	914
17.8.1	Transmission .....	914
17.8.1.1	Continuous Transmission.....	916
17.8.1.2	UART Buffer Transmission.....	917
17.8.1.3	Data Transmission .....	919
17.8.1.4	Transmission Start Wait Function .....	920
17.8.2	Reception .....	921
17.8.2.1	Data Reception.....	922
17.8.3	Expansion Bits.....	923
17.8.3.1	Expansion Bit Transmission .....	923
17.8.3.2	Expansion Bit Reception .....	923
17.8.3.3	Expansion Bit Reception (with Expansion Bit Comparison) .....	924
17.8.3.4	Expansion Bit Reception (with Data Comparison) .....	925

17.8.4	Statuses.....	926
17.8.5	Error Statuses.....	927
17.9	LIN Self-Test Mode.....	928
17.9.1	Transition to LIN Self-Test Mode.....	930
17.9.2	Transmission in LIN Master Self-Test Mode .....	931
17.9.3	Reception in LIN Master Self-Test Mode.....	932
17.9.4	Transmission in LIN Slave Self-Test Mode .....	933
17.9.5	Reception in LIN Slave Self-Test Mode.....	935
17.9.6	Exiting LIN Self-Test Mode.....	936
17.10	Baud Rate Generator.....	937
17.10.1	LIN Master Mode .....	937
17.10.2	LIN Slave Mode .....	938
17.10.3	UART Mode.....	939
17.11	Noise Filter.....	940
17.12	Usage Notes .....	942
17.12.1	Note on LIN Master mode .....	942
17.12.2	Note on LIN Slave mode (Fixed baud rate) .....	942
17.12.3	Note on LIN Slave mode (Auto baud rate) .....	942
<b>Section 18</b>	<b>I<sup>2</sup>C Bus Interface (RIIC)</b> .....	<b>943</b>
18.1	Features of RH850/F1L RIIC .....	943
18.1.1	Number of Units and Channels .....	943
18.1.2	Register Base Address.....	943
18.1.3	Clock Supply.....	943
18.1.4	Interrupt Requests .....	944
18.1.5	Reset Sources .....	944
18.1.6	External Input/Output Signals.....	944
18.2	Overview.....	945
18.2.1	Functional Overview .....	945
18.2.2	Block Diagram .....	947
18.3	Registers.....	949
18.3.1	List of Registers.....	949
18.3.2	RIICnCR1 — I <sup>2</sup> C Bus Control Register 1 .....	950
18.3.3	RIICnCR2 — I <sup>2</sup> C Bus Control Register 2 .....	953
18.3.4	RIICnMR1 — I <sup>2</sup> C Bus Mode Register 1 .....	957
18.3.5	RIICnMR2 — I <sup>2</sup> C Bus Mode Register 2 .....	959
18.3.6	RIICnMR3 — I <sup>2</sup> C Bus Mode Register 3 .....	961
18.3.7	RIICnFER — I <sup>2</sup> C Bus Function Enable Register .....	964
18.3.8	RIICnSER — I <sup>2</sup> C Bus Status Enable Register .....	966
18.3.9	RIICnIER — I <sup>2</sup> C Bus Interrupt Enable Register .....	968
18.3.10	RIICnSR1 — I <sup>2</sup> C Bus Status Register 1 .....	970
18.3.11	RIICnSR2 — I <sup>2</sup> C Bus Status Register 2 .....	973
18.3.12	RIICnSARy — I <sup>2</sup> C Slave Address Register y (y = 0 to 2).....	978
18.3.13	RIICnBRL — I <sup>2</sup> C Bus Bit Rate Low-Level Register.....	980
18.3.14	RIICnBRH — I <sup>2</sup> C Bus Bit Rate High-Level Register .....	981

18.3.15	RIICnDRT — I <sup>2</sup> C Bus Transmit Data Register .....	984
18.3.16	RIICnDRR — I <sup>2</sup> C Bus Receive Data Register.....	985
18.3.17	RIICnDRS — I <sup>2</sup> C Bus Shift Register .....	986
18.4	Interrupt Sources .....	987
18.5	Operation .....	988
18.5.1	Communication Data Format.....	988
18.5.2	Initial Settings .....	989
18.5.3	Master Transmit Operation.....	990
18.5.4	Master Receive Operation.....	994
18.5.5	Slave Transmit Operation.....	1000
18.5.6	Slave Receive Operation.....	1003
18.6	SCL Synchronization Circuit.....	1006
18.7	Facility for Delaying SDA Output .....	1007
18.8	Digital Noise-Filter Circuits .....	1008
18.9	Address Match Detection.....	1009
18.9.1	Slave-Address Match Detection .....	1009
18.9.2	Detection of the General Call Address .....	1011
18.9.3	Device-ID Address Detection .....	1012
18.10	Automatic Low-Hold Function for SCL.....	1014
18.10.1	Function to Prevent Wrong Transmission of Transmit Data.....	1014
18.10.2	NACK Reception Transfer Suspension Function .....	1015
18.10.3	Function to Prevent Failure to Receive Data.....	1016
18.11	Arbitration-Lost Detection Functions.....	1018
18.11.1	Master Arbitration-Lost Detection (MALE Bit).....	1018
18.11.2	Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) .....	1020
18.11.3	Slave Arbitration-Lost Detection (SALE Bit) .....	1021
18.12	Start Condition/Restart Condition/Stop Condition Issuing Function .....	1022
18.12.1	Issuing a Start Condition .....	1022
18.12.2	Issuing a Restart Condition .....	1022
18.12.3	Issuing a Stop Condition.....	1024
18.13	Bus Hanging .....	1025
18.13.1	Timeout Function.....	1025
18.13.2	Extra SCL Clock Cycle Output Function.....	1027
18.13.3	RIIC Reset and Internal Reset.....	1028
18.14	Reset Function of RIIC .....	1029
<b>Section 19</b>	<b>CAN Interface (RS-CAN) .....</b>	<b>1031</b>
19.1	Features of RH850/F1L RS-CAN .....	1031
19.1.1	Number of Units and Channels .....	1031
19.1.2	Register Base Address.....	1032
19.1.3	Clock Supply.....	1033
19.1.4	Interrupt Requests .....	1034
19.1.5	Reset Sources .....	1035
19.1.6	External Input/Output Signals.....	1035

19.2	Overview.....	1036
19.2.1	Functional Overview .....	1036
19.2.2	Block Diagram .....	1038
19.3	Registers.....	1039
19.3.1	List of Registers.....	1039
19.3.2	RSCAN0CmCFG — Channel Configuration Register (m = 0 to 5).....	1073
19.3.3	RSCAN0CmCTR — Channel Control Register (m = 0 to 5).....	1075
19.3.4	RSCAN0CmSTS — Channel Status Register (m = 0 to 5) .....	1079
19.3.5	RSCAN0CmERFL — Channel Error Flag Register (m = 0 to 5) .....	1081
19.3.6	RSCAN0GCFG — Global Configuration Register.....	1085
19.3.7	RSCAN0GCTR — Global Control Register.....	1088
19.3.8	RSCAN0GSTS — Global Status Register.....	1090
19.3.9	RSCAN0GERFL — Global Error Flag Register.....	1091
19.3.10	RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0.....	1093
19.3.11	RSCAN0GTINTSTS1 — Global TX Interrupt Status Register 1.....	1096
19.3.12	RSCAN0GTSC — Global Timestamp Counter Register.....	1098
19.3.13	RSCAN0GAFLECTR — Receive Rule Entry Control Register .....	1099
19.3.14	RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0 .....	1100
19.3.15	RSCAN0GAFLCFG1 — Receive Rule Configuration Register 1 .....	1102
19.3.16	RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15).....	1103
19.3.17	RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15).....	1105
19.3.18	RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15) .....	1106
19.3.19	RSCAN0GAFLP1j — Receive Rule Pointer 1 Register (j = 0 to 15) .....	1108
19.3.20	RSCAN0RMNB — Receive Buffer Number Register .....	1109
19.3.21	RSCAN0RMNDy — Receive Buffer New Data Register (y = 0 to 2).....	1110
19.3.22	RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 95) .....	1111
19.3.23	RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 95).....	1112
19.3.24	RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 95) .....	1113
19.3.25	RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 95) .....	1114
19.3.26	RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7) .....	1115
19.3.27	RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7) .....	1117
19.3.28	RSCAN0RFPCTR <sub>x</sub> — Receive FIFO Buffer Pointer Control Register (x = 0 to 7) .....	1119
19.3.29	RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7).....	1120
19.3.30	RSCAN0RFPTR <sub>x</sub> — Receive FIFO Buffer Access Pointer Register (x = 0 to 7) .....	1121
19.3.31	RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7) .....	1122
19.3.32	RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7) .....	1123
19.3.33	RSCAN0CFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 17) .....	1124
19.3.34	RSCAN0CFSTSK — Transmit/receive FIFO Buffer Status Register (k = 0 to 17).....	1128
19.3.35	RSCAN0CFPCTR <sub>k</sub> — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 17) .....	1131
19.3.36	RSCAN0CFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 17) .....	1133

19.3.37	RSCAN0CFPTRk — Transmit/receive FIFO Buffer Access Pointer Register ( $k = 0$ to 17) .....	1135
19.3.38	RSCAN0CFDF0k — Transmit/receive FIFO Buffer Access Data Field 0 Register ( $k = 0$ to 17) .....	1137
19.3.39	RSCAN0CFDF1k — Transmit/receive FIFO Buffer Access Data Field 1 Register ( $k = 0$ to 17) .....	1138
19.3.40	RSCAN0FESTS — FIFO Empty Status Register .....	1139
19.3.41	RSCAN0FFSTS — FIFO Full Status Register .....	1141
19.3.42	RSCAN0FMSTS — FIFO Message Lost Status Register .....	1143
19.3.43	RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register .....	1145
19.3.44	RSCAN0CFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register .....	1146
19.3.45	RSCAN0CFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register .....	1148
19.3.46	RSCAN0TMCp — Transmit Buffer Control Register ( $p = 0$ to 95) .....	1149
19.3.47	RSCAN0TMSTSp — Transmit Buffer Status Register ( $p = 0$ to 95) .....	1151
19.3.48	RSCAN0TMTRSTS $y$ — Transmit Buffer Transmit Request Status Register ( $y = 0$ to 2) .....	1153
19.3.49	RSCAN0TMTARSTS $y$ — Transmit Buffer Transmit Abort Request Status Register ( $y = 0$ to 2) .....	1155
19.3.50	RSCAN0TMTCS $y$ — Transmit Buffer Transmit Complete Status Register ( $y = 0$ to 2) .....	1157
19.3.51	RSCAN0TMTASTS $y$ — Transmit Buffer Transmit Abort Status Register ( $y = 0$ to 2) .....	1159
19.3.52	RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register ( $y = 0$ to 2) .....	1161
19.3.53	RSCAN0TMIDp — Transmit Buffer ID Register ( $p = 0$ to 95) .....	1163
19.3.54	RSCAN0TMPTRP — Transmit Buffer Pointer Register ( $p = 0$ to 95) .....	1165
19.3.55	RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register ( $p = 0$ to 95) .....	1166
19.3.56	RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register ( $p = 0$ to 95) .....	1167
19.3.57	RSCAN0TXQCCm — Transmit Queue Configuration and Control Register ( $m = 0$ to 5) .....	1168
19.3.58	RSCAN0TXQSTS $m$ — Transmit Queue Status Register ( $m = 0$ to 5) .....	1170
19.3.59	RSCAN0TXQPCTR $m$ — Transmit Queue Pointer Control Register ( $m = 0$ to 5) .....	1172
19.3.60	RSCAN0THLCCm — Transmit History Configuration and Control Register ( $m = 0$ to 5) .....	1173
19.3.61	RSCAN0THLSTS $m$ — Transmit History Status Register ( $m = 0$ to 5) .....	1175
19.3.62	RSCAN0THLACCM — Transmit History Access Register ( $m = 0$ to 5) .....	1177
19.3.63	RSCAN0THLPCTR $m$ — Transmit History Pointer Control Register ( $m = 0$ to 5) .....	1178
19.3.64	RSCAN0GTSTCFG — Global Test Configuration Register .....	1179
19.3.65	RSCAN0GTSTCTR — Global Test Control Register .....	1181
19.3.66	RSCAN0GLOCKK — Global Lock Key Register .....	1182
19.3.67	RSCAN0RPGACC $r$ — RAM Test Page Access Register ( $r = 0$ to 63) .....	1183
19.4	Interrupt Sources .....	1184
19.5	CAN Modes .....	1188
19.5.1	Global Modes .....	1188
19.5.1.1	Global Stop Mode .....	1190
19.5.1.2	Global Reset Mode .....	1190
19.5.1.3	Global Test Mode .....	1190
19.5.1.4	Global Operating Mode .....	1190

19.5.2	Channel Modes .....	1191
19.5.2.1	Channel Stop Mode.....	1192
19.5.2.2	Channel Reset Mode.....	1192
19.5.2.3	Channel Halt Mode.....	1192
19.5.2.4	Channel Communication Mode .....	1193
19.5.2.5	Bus-off State.....	1193
19.6	Reception Function.....	1196
19.6.1	Data Processing Using the Receive Rule Table.....	1196
19.6.1.1	Acceptance Filter Processing.....	1197
19.6.1.2	DLC Filter Processing .....	1198
19.6.1.3	Routing Processing .....	1198
19.6.1.4	Label Addition Processing.....	1198
19.6.1.5	Mirror Function Processing .....	1198
19.6.1.6	Timestamp.....	1199
19.7	Transmission Functions.....	1200
19.7.1	Transmit Priority Determination .....	1201
19.7.2	Transmission Using Transmit Buffers.....	1201
19.7.2.1	Transmit Abort Function .....	1201
19.7.2.2	One-Shot Transmission Function (Retransmission Disabling Function) .....	1202
19.7.3	Transmission Using FIFO Buffers .....	1202
19.7.3.1	Interval Transmission Function .....	1203
19.7.4	Transmission Using Transmit Queues .....	1205
19.7.5	Transmit History Function.....	1205
19.8	Gateway Function.....	1207
19.9	Test Function.....	1208
19.9.1	Standard Test Mode .....	1208
19.9.2	Listen-Only Mode .....	1208
19.9.3	Self-Test Mode (Loopback Mode) .....	1209
19.9.3.1	Self-Test Mode 0 (External Loopback Mode).....	1209
19.9.3.2	Self-Test Mode 1 (Internal Loopback Mode) .....	1210
19.9.4	RAM Test.....	1211
19.9.5	Inter-Channel Communication Test.....	1211
19.10	RS-CAN Setting Procedure .....	1212
19.10.1	Initial Settings .....	1212
19.10.1.1	Clock Setting .....	1213
19.10.1.2	Bit Timing Setting .....	1213
19.10.1.3	Communication Speed Setting.....	1214
19.10.1.4	Receive Rule Setting.....	1215
19.10.1.5	Buffer Setting.....	1216
19.10.2	Reception Procedure.....	1218
19.10.2.1	Receive Buffer Reading Procedure.....	1218
19.10.2.2	FIFO Buffer Reading Procedure.....	1220
19.10.3	Transmission Procedure.....	1223
19.10.3.1	Procedure for Transmission from Transmit Buffers.....	1223
19.10.3.2	Procedure for Transmission from Transmit/Receive FIFO Buffers .....	1227
19.10.3.3	Procedure for Transmission from the Transmit Queue .....	1231
19.10.3.4	Transmit History Buffer Reading Procedure.....	1232

19.10.4	Test Settings.....	1233
19.10.4.1	Self-Test Mode Setting Procedure .....	1233
19.10.4.2	Procedure for Releasing the Protection .....	1234
19.10.4.3	RAM Test Setting Procedure.....	1235
19.10.4.4	Inter-Channel Communication Test Setting Procedure.....	1236
19.11	Detection and Correction of Errors in RS-CAN RAM.....	1237
19.11.1	ECC for the RSCAN0 RAM .....	1237
19.11.2	Interrupt Request.....	1237
19.11.3	ECCR CAN0CTL — RSCAN0 ECC Control Register .....	1238
19.11.4	ECCR CAN0TMC — RSCAN0 ECC Test Mode Control Register .....	1240
19.11.5	ECCR CAN0TED — RSCAN0 ECC Encode/Decode Input/Output Replacement Test Register .....	1242
19.11.6	ECCR CAN0TRC — RSCAN0 ECC Redundant Bit Data Control Test Register .....	1243
19.11.7	ECCR CAN0SYND — RSCAN0 ECC Decode Syndrome Data Register .....	1243
19.11.8	ECCR CAN0HORD — RSCAN0 ECC 7-Bit Redundant Bit Data Hold Test Register ....	1244
19.11.9	ECCR CAN0ECRD — RSCAN0 ECC Encode Test Register .....	1244
19.11.10	ECCR CAN0ERDB — RSCAN0 ECC Redundant Bit Input/Output Replacement Buffer Register .....	1245
19.11.11	SELB_READTEST — ECCREAD Test Select Register.....	1245
19.12	Notes on the RS-CAN Module.....	1246
<b>Section 20</b>	<b>Window Watchdog Timer (WDTA).....</b>	<b>1247</b>
20.1	Features of RH850/F1L WDTA.....	1247
20.1.1	Number of Units and Channels .....	1247
20.1.2	Register Base Address.....	1247
20.1.3	Clock Supply.....	1247
20.1.4	Interrupt Requests.....	1248
20.1.5	Reset Sources .....	1248
20.2	Overview.....	1249
20.2.1	Functional Overview .....	1249
20.2.2	Block Diagram .....	1251
20.3	Registers.....	1252
20.3.1	List of Registers.....	1252
20.3.2	WDTAnWDTE — WDTA Enable Register.....	1253
20.3.3	WDTAnEVAC — WDTA Enable VAC Register .....	1255
20.3.4	WDTAnREF — WDTA Reference Value Register .....	1256
20.3.5	WDTAnMD — WDTA Mode Register.....	1257
20.4	Interrupt Sources .....	1258
20.5	Functions .....	1259
20.5.1	WDTA after Reset Release .....	1259
20.5.1.1	Start Modes .....	1259
20.5.1.2	WDTA Settings after Reset Release .....	1259
20.5.1.3	Default Start Mode Timing.....	1260
20.5.1.4	Software Trigger Start Mode Timing .....	1261
20.5.2	WDTA Trigger.....	1262
20.5.2.1	Calculating an Activation Code when the VAC Function is Used.....	1262

20.5.3	WDTA Error Detection.....	1263
20.5.3.1	WDTA Error Mode.....	1263
20.5.4	75% Interrupt Request Signals.....	1265
20.5.5	Window Function.....	1266
<b>Section 21</b>	<b>OS Timer (OSTM).....</b>	<b>1267</b>
21.1	Features of RH850/F1L OSTM.....	1267
21.1.1	Number of Units.....	1267
21.1.2	Register Base Address.....	1267
21.1.3	Clock Supply.....	1268
21.1.4	Interrupt Request.....	1268
21.1.5	Reset Sources.....	1268
21.2	Overview.....	1269
21.2.1	Functional Overview .....	1269
21.2.2	Block Diagram .....	1269
21.2.3	Count Clock.....	1270
21.2.4	Interrupt Sources (OSTMTINT) .....	1270
21.3	Registers.....	1271
21.3.1	List of Registers.....	1271
21.3.2	OSTMnCMP — OSTMn Compare Register.....	1272
21.3.3	OSTMnCNT — OSTMn Counter Register.....	1273
21.3.4	OSTMnTE — OSTMn Count Enable Status Register .....	1274
21.3.5	OSTMnTS — OSTMn Count Start Trigger Register.....	1275
21.3.6	OSTMnTT — OSTMn Count Stop Trigger Register .....	1275
21.3.7	OSTMnCTL — OSTMn Control Register .....	1276
21.3.8	OSTMnEMU — OSTMn Emulation Register.....	1277
21.4	Operation .....	1278
21.4.1	Starting and Stopping OSTM.....	1278
21.4.2	Interval Timer Mode.....	1278
21.4.2.1	Basic Operation in Interval Timer Mode .....	1278
21.4.2.2	Operation when OSTMnCMP = 0000 0000 <sub>H</sub> .....	1281
21.4.2.3	Setting Procedure for Interval Timer Mode .....	1281
21.4.3	Free-Run Compare Mode.....	1282
21.4.3.1	Basic Operation in Free-Run Compare Mode .....	1282
21.4.3.2	Operation when OSTMnCMP = 0000 0000 <sub>H</sub> .....	1283
21.4.3.3	Setting Procedure for Free-Run Compare Mode .....	1284
<b>Section 22</b>	<b>Timer Array Unit B (TAUB) .....</b>	<b>1285</b>
22.1	Features of RH850/F1L TAUB.....	1285
22.1.1	Number of Units and Channels .....	1285
22.1.2	Register Base Address.....	1285
22.1.3	Clock Supply.....	1286
22.1.4	Interrupt Requests.....	1286
22.1.5	Reset Sources .....	1287
22.1.6	External Input/Output Signals.....	1287

22.2	Overview.....	1289
22.2.1	Functional Overview .....	1289
22.2.2	Terms .....	1290
22.2.3	Functional List of Timer Operations.....	1292
22.2.4	Input/Output Interrupt Request Signals .....	1293
22.2.5	Block Diagram .....	1294
22.2.6	Description of Blocks.....	1295
22.3	Registers.....	1296
22.3.1	List of Registers.....	1296
22.3.2	Details of TAUBn Prescaler Registers.....	1297
22.3.2.1	TAUBnTPS — TAUBn Prescaler Clock Select Register .....	1297
22.3.3	Details of TAUBn Control Registers .....	1300
22.3.3.1	TAUBnCDRm — TAUBn Channel Data Register .....	1300
22.3.3.2	TAUBnCNTm — TAUBn Channel Counter Register.....	1301
22.3.3.3	TAUBnCMORm — TAUBn Channel Mode OS Register.....	1302
22.3.3.4	TAUBnCMURm — TAUBn Channel Mode User Register .....	1305
22.3.3.5	TAUBnCSRm — TAUBn Channel Status Register .....	1306
22.3.3.6	TAUBnCSCm — TAUBn Channel Status Clear Register .....	1306
22.3.3.7	TAUBnTS — TAUBn Channel Start Trigger Register .....	1307
22.3.3.8	TAUBnTE — TAUBn Channel Enable Status Register.....	1307
22.3.3.9	TAUBnTT — TAUBn Channel Stop Trigger Register.....	1308
22.3.4	Details of TAUBn Simultaneous Rewrite Registers.....	1309
22.3.4.1	TAUBnRDE — TAUBn Channel Reload Data Enable Register .....	1309
22.3.4.2	TAUBnRDS — TAUBn Channel Reload Data Control Channel Select Register	1309
22.3.4.3	TAUBnRDM — TAUBn Channel Reload Data Mode Register.....	1310
22.3.4.4	TAUBnRDC — TAUBn Channel Reload Data Control Register .....	1310
22.3.4.5	TAUBnRDT — TAUBn Channel Reload Data Trigger Register .....	1311
22.3.4.6	TAUBnRSF — TAUBn Channel Reload Status Register .....	1311
22.3.5	Details of TAUBn Output Registers .....	1312
22.3.5.1	TAUBnTOE — TAUBn Channel Output Enable Register .....	1312
22.3.5.2	TAUBnTO — TAUBn Channel Output Register .....	1312
22.3.5.3	TAUBnTOM — TAUBn Channel Output Mode Register .....	1313
22.3.5.4	TAUBnTOC — TAUBn Channel Output Configuration Register .....	1313
22.3.5.5	TAUBnTOL — TAUBn Channel Output Level Register .....	1314
22.3.6	Details of TAUBn Dead Time Output Registers .....	1315
22.3.6.1	TAUBnTDE — TAUBn Channel Dead Time Output Enable Register .....	1315
22.3.6.2	TAUBnTDL — TAUBn Channel Dead Time Output Level Register .....	1315
22.3.7	TAUBn Emulation Register.....	1316
22.3.7.1	TAUBnEMU — TAUBn Emulation Register .....	1316
22.4	General Operating Procedure.....	1317
22.5	Concepts of Synchronous Channel Operation .....	1318
22.5.1	Rules of Synchronous Channel Operation Function .....	1318
22.5.2	Simultaneous Start and Stop of Synchronous Channel Counters.....	1320
22.5.2.1	Simultaneous Start and Stop within the Same Unit.....	1320
22.6	Simultaneous Rewrite.....	1321
22.6.1	Introduction .....	1321
22.6.2	How to Control Simultaneous Rewrite.....	1322

22.6.2.1	Initial Settings .....	1323
22.6.2.2	Start Counter and Count Operation.....	1323
22.6.2.3	Simultaneous Rewrite .....	1323
22.6.3	Other General Rules of Simultaneous Rewrite.....	1324
22.6.4	Types of Simultaneous Rewrite.....	1325
22.6.4.1	Simultaneous Rewrite when the Master Channel (Re)Starts Counting (Method A) .....	1325
22.6.4.2	Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel (Method B).....	1326
22.6.4.3	Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm (Method C1) .....	1328
22.7	Channel Output Modes.....	1330
22.7.1	General Procedures for Specifying a Channel Output Mode .....	1332
22.7.2	Channel Output Modes Controlled Independently by TAUBn Signals .....	1333
22.7.2.1	Independent Channel Output Mode 1 .....	1333
22.7.2.2	Independent Channel Output Mode 2 .....	1333
22.7.3	Channel Output Modes Controlled Synchronously by TAUBn Signals .....	1333
22.7.3.1	Synchronous Channel Output Mode 1 .....	1333
22.7.3.2	Synchronous Channel Output Mode 2 .....	1333
22.7.3.3	Synchronous Channel Output Mode 2 with Dead Time Output .....	1334
22.8	Start Timing in Each Operating Modes.....	1335
22.8.1	Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode.....	1335
22.8.2	Event Count Mode.....	1336
22.8.3	Other Operating Modes .....	1336
22.9	TAUBTOUTm Output and INTTAUBnIm Generation when Counter Starts or Restarts.....	1337
22.10	Interrupt Generation upon Overflow .....	1338
22.10.1	Example of Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function .....	1339
22.10.2	Example of Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement) .....	1340
22.10.3	Example of Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function.....	1341
22.10.4	Example of Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection) .....	1342
22.11	TAUBTTINm Edge Detection.....	1343
22.12	Independent Channel Operation Functions .....	1344
22.12.1	Interval Timer Function.....	1344
22.12.1.1	Overview .....	1344
22.12.1.2	Equations .....	1344
22.12.1.3	Block Diagram and General Timing Diagram.....	1345
22.12.1.4	Register Settings .....	1346
22.12.1.5	Operating Procedure for Interval Timer Function .....	1348
22.12.1.6	Specific Timing Diagrams.....	1349
22.12.2	TAUBTTINm Input Interval Timer Function .....	1353
22.12.2.1	Overview .....	1353
22.12.2.2	Block Diagram and General Timing Diagram.....	1354

22.12.2.3	Register Settings .....	1355
22.12.2.4	Operating Procedure for TAUBTTINm Input Interval Timer Function .....	1357
22.12.2.5	Specific Timing Diagrams.....	1358
22.12.3	Clock Divide Function.....	1359
22.12.3.1	Overview .....	1359
22.12.3.2	Block Diagram and General Timing Diagram.....	1360
22.12.3.3	Register Settings .....	1361
22.12.3.4	Operating Procedure for Clock Divide Function.....	1362
22.12.3.5	Specific Timing Diagrams.....	1363
22.12.4	External Event Count Function.....	1365
22.12.4.1	Overview .....	1365
22.12.4.2	Equations .....	1365
22.12.4.3	Block Diagram and General Timing Diagram.....	1366
22.12.4.4	Register Settings .....	1367
22.12.4.5	Operating Procedure for External Event Count Function .....	1368
22.12.4.6	Specific Timing Diagrams.....	1369
22.12.5	One-Pulse Output Function .....	1371
22.12.5.1	Overview .....	1371
22.12.5.2	Equations .....	1371
22.12.5.3	Block Diagram and General Timing Diagram.....	1372
22.12.5.4	Register Settings .....	1373
22.12.5.5	Operating Procedure for One-Pulse Output Function .....	1374
22.12.6	TAUBTTINm Input Pulse Interval Measurement Function .....	1375
22.12.6.1	Overview .....	1375
22.12.6.2	Equations .....	1376
22.12.6.3	Block Diagram and General Timing Diagram.....	1377
22.12.6.4	Register Settings .....	1378
22.12.6.5	Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function.....	1379
22.12.6.6	Specific Timing Diagrams: Overflow Behavior .....	1380
22.12.7	TAUBTTINm Input Signal Width Measurement Function .....	1384
22.12.7.1	Overview .....	1384
22.12.7.2	Equations .....	1385
22.12.7.3	Block Diagram and General Timing Diagram.....	1385
22.12.7.4	Register Settings .....	1386
22.12.7.5	Operating Procedure for TAUBTTINm Input Signal Width Measurement Function .....	1387
22.12.7.6	Specific Timing Diagrams: Overflow Behavior .....	1388
22.12.8	TAUBTTINm Input Position Detection Function .....	1392
22.12.8.1	Overview .....	1392
22.12.8.2	Equations .....	1392
22.12.8.3	Block Diagram and General Timing Diagram.....	1393
22.12.8.4	Register Settings .....	1394
22.12.8.5	Operating Procedure for TAUBTTINm Input Position Detection Function .....	1395
22.12.8.6	Specific Timing Diagrams.....	1396
22.12.9	TAUBTTINm Input Period Count Detection Function .....	1397
22.12.9.1	Overview .....	1397
22.12.9.2	Equations .....	1397
22.12.9.3	Block Diagram and General Timing Diagram.....	1398

22.12.9.4 Register Settings .....	1399
22.12.9.5 Operating Procedure for TAUBTTINm Input Period Count Detection Function ..	1400
22.12.9.6 Specific Timing Diagrams.....	1401
22.12.10 TAUBTTINm Input Pulse Interval Judgment Function.....	1402
22.12.10.1 Overview .....	1402
22.12.10.2 Block Diagram and General Timing Diagram.....	1403
22.12.10.3 Register Settings .....	1403
22.12.10.4 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function.	1405
22.12.11 TAUBTTINm Input Signal Width Judgment Function .....	1406
22.12.11.1 Overview .....	1406
22.12.11.2 Block Diagram and General Timing Diagram.....	1407
22.12.11.3 Register Settings .....	1408
22.12.11.4 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function ..	1409
22.12.12 Overflow Interrupt Output Function (during TAUBTTINm Width Measurement) .....	1410
22.12.12.1 Overview .....	1410
22.12.12.2 Block Diagram and General Timing Diagram.....	1411
22.12.12.3 Register Settings .....	1412
22.12.12.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement) .....	1414
22.12.13 Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)	1415
22.12.13.1 Overview .....	1415
22.12.13.2 Block Diagram and General Timing Diagram.....	1416
22.12.13.3 Register Settings .....	1417
22.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection) .....	1419
22.13 Independent Channel Simultaneous Rewrite Functions.....	1420
22.13.1 Simultaneous Rewrite Trigger Generation Function Type 1.....	1420
22.13.1.1 Overview .....	1420
22.13.1.2 Equations .....	1421
22.13.1.3 Block Diagram and General Timing Diagram.....	1422
22.13.1.4 Register Settings for The Upper Channel .....	1424
22.13.1.5 Register Settings for the Lower Channel(s) .....	1425
22.13.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1 .....	1426
22.14 Synchronous Channel Operation Functions .....	1427
22.14.1 PWM Output Function .....	1427
22.14.1.1 Overview .....	1427
22.14.1.2 Equations .....	1428
22.14.1.3 Block Diagram and General Timing Diagram.....	1428
22.14.1.4 Register Settings for the Master Channel .....	1430
22.14.1.5 Register Settings for the Slave Channel(s) .....	1432
22.14.1.6 Operating Procedure for PWM Output Function .....	1434
22.14.1.7 Specific Timing Diagrams.....	1435
22.14.2 One-Shot Pulse Output Function.....	1439
22.14.2.1 Overview .....	1439
22.14.2.2 Equations .....	1440
22.14.2.3 Block Diagram and General Timing Diagram.....	1441
22.14.2.4 Register Settings for the Master Channel .....	1443

22.14.2.5 Register Settings for the Slave Channel .....	1445
22.14.2.6 Operating Procedure for One-Shot Pulse Output Function.....	1447
22.14.2.7 Specific Timing Diagrams.....	1448
22.14.3 Delay Pulse Output Function.....	1454
22.14.3.1 Overview .....	1454
22.14.3.2 Equations .....	1455
22.14.3.3 Block Diagram and General Timing Diagram.....	1457
22.14.3.4 Register Settings for the Master Channel .....	1459
22.14.3.5 Register Settings for Slave Channel 1 .....	1461
22.14.3.6 Register Settings For Slave Channel 2 .....	1463
22.14.3.7 Register Settings for Slave Channel 3 .....	1465
22.14.3.8 Operating Procedure for Delay Pulse Output Function .....	1466
22.14.3.9 Specific Timing Diagrams.....	1468
22.14.4 AD Conversion Trigger Output Function Type 1 .....	1470
22.14.4.1 Overview .....	1470
22.14.4.2 Block Diagram and General Timing Diagram.....	1470
22.14.5 Triangle PWM Output Function .....	1472
22.14.5.1 Overview .....	1472
22.14.5.2 Equations .....	1473
22.14.5.3 Block Diagram and General Timing Diagram.....	1474
22.14.5.4 Register Settings for the Master Channel .....	1476
22.14.5.5 Register Settings for the Slave Channel(s) .....	1478
22.14.5.6 Operating Procedure for Triangle PWM Output Function .....	1480
22.14.5.7 Specific Timing Diagrams.....	1481
22.14.6 Triangle PWM Output Function with Dead Time .....	1483
22.14.6.1 Overview .....	1483
22.14.6.2 Equations .....	1485
22.14.6.3 Block Diagram and General Timing Diagram.....	1486
22.14.6.4 Register Settings for the Master Channel .....	1488
22.14.6.5 Register Settings for Slave Channel 2 .....	1490
22.14.6.6 Register Settings for Slave Channel 3 .....	1492
22.14.6.7 Operating Procedure for Triangle PWM Output Function with Dead Time .....	1494
22.14.6.8 Specific Timing Diagrams.....	1495
22.14.7 AD Conversion Trigger Output Function Type 2 .....	1506
22.14.7.1 Overview .....	1506
22.14.7.2 Block Diagram and General Timing Diagram.....	1506
<b>Section 23 Timer Array Unit D (TAUD) .....</b>	<b>1508</b>
23.1 Features of RH850/F1L TAUD .....	1508
23.1.1 Number of Units and Channels .....	1508
23.1.2 Register Base Address.....	1508
23.1.3 Clock Supply.....	1509
23.1.4 Interrupt Requests.....	1509
23.1.5 Reset Sources .....	1509
23.1.6 External Input/Output Signals.....	1510
23.1.7 Internal Input/Output Signals .....	1511
23.1.8 TAUD0 Input Selection .....	1511
23.1.8.1 SELB_TAUD0I — TAUDTTINm Input Signal Selection Register.....	1512

23.2	Overview.....	1513
23.2.1	Functional Overview .....	1513
23.2.2	Terms .....	1514
23.2.3	Functional List of Timer Operations.....	1515
23.2.4	TAUD I/O and Interrupt Request Signals .....	1516
23.2.5	Block Diagram .....	1517
23.2.6	Description of Blocks.....	1518
23.3	Registers.....	1519
23.3.1	List of Registers.....	1519
23.3.2	Details of TAUDn Prescaler Registers .....	1520
23.3.2.1	TAUDnTPS — TAUDn Prescaler Clock Select Register.....	1520
23.3.2.2	TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register .....	1523
23.3.3	Details of TAUDn Control Registers .....	1524
23.3.3.1	TAUDnCDRm — TAUDn Channel Data Register .....	1524
23.3.3.2	TAUDnCNTm — TAUDn Channel Counter Register .....	1525
23.3.3.3	TAUDnCMORm — TAUDn Channel Mode OS Register .....	1526
23.3.3.4	TAUDnCMURm — TAUDn Channel Mode User Register .....	1529
23.3.3.5	TAUDnCSRm — TAUDn channel status register .....	1530
23.3.3.6	TAUDnCSCm — TAUDn Channel Status Clear Register .....	1531
23.3.3.7	TAUDnTS — TAUDn Channel Start Trigger Register.....	1531
23.3.3.8	TAUDnTE — TAUDn Channel Enable Status Register .....	1532
23.3.3.9	TAUDnTT — TAUDn Channel Stop Trigger Register .....	1532
23.3.4	Details of TAUDn Simultaneous Rewrite Registers.....	1533
23.3.4.1	TAUDnRDE — TAUDn Channel Reload Data Enable Register.....	1533
23.3.4.2	TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register	1533
23.3.4.3	TAUDnRDM — TAUDn Channel Reload Data Mode Register .....	1534
23.3.4.4	TAUDnRDC — TAUDn Channel Reload Data Control Register .....	1534
23.3.4.5	TAUDnRDT — TAUDn Channel Reload Data Trigger Register.....	1535
23.3.4.6	TAUDnRSF — TAUDn Channel Reload Status Register.....	1535
23.3.5	Details of TAUDn Output Registers.....	1536
23.3.5.1	TAUDnTOE — TAUDn Channel Output Enable Register .....	1536
23.3.5.2	TAUDnTO — TAUDn Channel Output Register.....	1536
23.3.5.3	TAUDnTOM — TAUDn Channel Output Mode Register.....	1537
23.3.5.4	TAUDnTOC — TAUDn Channel Output Configuration Register.....	1537
23.3.5.5	TAUDnTOL — TAUDn Channel Output Level Register .....	1538
23.3.6	Details of TAUDn Dead Time Output Registers .....	1539
23.3.6.1	TAUDnTDE — TAUDn Channel Dead Time Output Enable Register.....	1539
23.3.6.2	TAUDnTDM — TAUDn Channel Dead Time Output Mode Register .....	1539
23.3.6.3	TAUDnTDL — TAUDn Channel Dead Time Output Level Register.....	1540
23.3.7	Details of TAUDn Real-time/Modulation Output Registers.....	1541
23.3.7.1	TAUDnTRE — TAUDn Channel Real-time Output Enable Register .....	1541
23.3.7.2	TAUDnTRC — TAUDn Channel Real-time Output Control Register .....	1541
23.3.7.3	TAUDnTRO — TAUDn Channel Real-time Output Register.....	1542
23.3.7.4	TAUDnTME — TAUDn Channel Modulation Output Enable Register .....	1542
23.3.8	TAUDn Emulation Register .....	1543
23.3.8.1	TAUDnEMU — TAUDn Emulation Register.....	1543
23.4	Operating Procedure .....	1544
23.5	Concepts of Synchronous Channel Operation .....	1545

23.5.1	Rules of Synchronous Channel Operation .....	1545
23.5.2	Simultaneous Start and Stop of Synchronous Channel Counters.....	1547
23.5.2.1	Simultaneous Start and Stop within the Same Unit.....	1547
23.5.2.2	Simultaneous Start between the Units .....	1547
23.6	Simultaneous Rewrite.....	1548
23.6.1	Overview of Operations .....	1548
23.6.2	How to Control Simultaneous Rewrite.....	1550
23.6.2.1	Initial Settings .....	1551
23.6.2.2	Start Counter and Count Operation.....	1551
23.6.2.3	Simultaneous Rewrite .....	1551
23.6.3	Other General Rules of Simultaneous Rewrite.....	1552
23.6.4	Types of Simultaneous Rewrite .....	1553
23.6.4.1	Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)	1553
23.6.4.2	Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B).....	1555
23.6.4.3	Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1).....	1556
23.6.4.4	Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2) .....	1558
23.7	Channel Output Modes .....	1560
23.7.1	General Procedures for Specifying a Channel Output Mode .....	1562
23.7.2	Channel Output Modes Controlled Independently by TAUDn Signals .....	1563
23.7.2.1	Independent Channel Output Mode 1 .....	1563
23.7.2.2	Independent Channel Output Mode 1 with Real-Time Output .....	1563
23.7.2.3	Independent Channel Output Mode 2 .....	1564
23.7.3	Channel Output Modes Controlled Synchronously by TAUDn Signals .....	1565
23.7.3.1	Synchronous Channel Output Mode 1 .....	1565
23.7.3.2	Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output.....	1565
23.7.3.3	Synchronous Channel Output Mode 2 .....	1565
23.7.3.4	Synchronous Channel Output Mode 2 with Dead Time Output .....	1566
23.7.3.5	Synchronous Channel Output Mode 2 with One-Phase PWM Output.....	1567
23.7.3.6	Synchronous Channel Output Mode 2 with Complementary Modulation Output	1568
23.7.3.7	Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output.....	1568
23.8	Start Timing in Each Operating Modes .....	1569
23.8.1	Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode.....	1569
23.8.2	Event Count Mode.....	1570
23.8.3	Other Operating Modes .....	1570
23.9	TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts .....	1571
23.10	Interrupt Generation upon Overflow .....	1572
23.10.1	Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function .....	1573
23.10.2	Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width) .....	1574

23.10.3	Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function.....	1575
23.10.4	Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count).....	1576
23.11	TAUDTTINm Edge Detection .....	1577
23.12	Independent Channel Operation Functions .....	1578
23.12.1	Interval Timer Function.....	1578
23.12.1.1	Overview .....	1578
23.12.1.2	Equations .....	1578
23.12.1.3	Block Diagram and General Timing Diagram.....	1579
23.12.1.4	Register Settings .....	1580
23.12.1.5	Operating Procedure for Interval Timer Function .....	1582
23.12.1.6	Specific Timing Diagrams.....	1583
23.12.2	TAUDTTINm Input Interval Timer Function .....	1587
23.12.2.1	Overview .....	1587
23.12.2.2	Equations .....	1587
23.12.2.3	Block Diagram and General Timing Diagram.....	1588
23.12.2.4	Register Settings .....	1589
23.12.2.5	Operating Procedure for TAUDTTINm Input Interval Timer Function .....	1591
23.12.2.6	Specific Timing Diagrams.....	1592
23.12.3	Clock Divide Function.....	1593
23.12.3.1	Overview .....	1593
23.12.3.2	Equations .....	1594
23.12.3.3	Block Diagram and General Timing Diagram.....	1594
23.12.3.4	Register Settings .....	1595
23.12.3.5	Operating Procedure for Clock Divide Function .....	1597
23.12.3.6	Specific Timing Diagrams.....	1598
23.12.4	External Event Count Function.....	1600
23.12.4.1	Overview .....	1600
23.12.4.2	Equations .....	1600
23.12.4.3	Block Diagram and General Timing Diagram.....	1601
23.12.4.4	Register Settings .....	1602
23.12.4.5	Operating Procedure for External Event Count Function .....	1603
23.12.4.6	Specific Timing Diagrams.....	1604
23.12.5	Delay Count Function .....	1606
23.12.5.1	Overview .....	1606
23.12.5.2	Equations .....	1606
23.12.5.3	Block Diagram and General Timing Diagram.....	1607
23.12.5.4	Register Settings .....	1608
23.12.5.5	Operating Procedure for Delay Count Function .....	1609
23.12.6	One-Pulse Output Function .....	1610
23.12.6.1	Overview .....	1610
23.12.6.2	Equations .....	1610
23.12.6.3	Block Diagram and General Timing Diagram.....	1611
23.12.6.4	Register Settings .....	1612
23.12.6.5	Operating Procedure for One-Pulse Output Function .....	1614
23.12.7	TAUDTTINm Input Pulse Interval Measurement Function .....	1615
23.12.7.1	Overview .....	1615
23.12.7.2	Equations .....	1616

23.12.7.3	Block Diagram and General Timing Diagram.....	1617
23.12.7.4	Register Settings .....	1618
23.12.7.5	Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function.....	1619
23.12.7.6	Specific Timing Diagrams: Overflow Operation.....	1620
23.12.8	TAUDTTINm Input Signal Width Measurement Function .....	1623
23.12.8.1	Overview .....	1623
23.12.8.2	Equations .....	1624
23.12.8.3	Block Diagram and General Timing Diagram.....	1624
23.12.8.4	Register Settings .....	1626
23.12.8.5	Operating Procedure for TAUDTTINm Input Signal Width Measurement Function .....	1627
23.12.8.6	Specific Timing Diagrams: Overflow Operation.....	1628
23.12.9	TAUDTTINm Input Position Detection Function .....	1632
23.12.9.1	Overview .....	1632
23.12.9.2	Equations .....	1632
23.12.9.3	Block Diagram and General Timing Diagram.....	1633
23.12.9.4	Register Settings .....	1634
23.12.9.5	Operating Procedure for TAUDTTINm Input Position Detection Function .....	1635
23.12.9.6	Specific Timing Diagrams.....	1636
23.12.10	TAUDTTINm Input Period Count Detection Function.....	1637
23.12.10.1	Overview .....	1637
23.12.10.2	Equations .....	1637
23.12.10.3	Block Diagram and General Timing Diagram.....	1638
23.12.10.4	Register Settings .....	1639
23.12.10.5	Operating Procedure for TAUDTTINm Input Period Count Detection Function ..	1640
23.12.10.6	Specific Timing Diagrams.....	1641
23.12.11	TAUDTTINm Input Pulse Interval Judgment Function .....	1642
23.12.11.1	Overview .....	1642
23.12.11.2	Block Diagram and General Timing Diagram.....	1643
23.12.11.3	Register Settings .....	1644
23.12.11.4	Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function.	1645
23.12.12	TAUDTTINm Input Signal Width Judgment Function .....	1646
23.12.12.1	Overview .....	1646
23.12.12.2	Block Diagram and General Timing Diagram.....	1647
23.12.12.3	Register Settings .....	1648
23.12.12.4	Operating Procedure for TAUDTTINm Input Signal Width Judgment Function ..	1649
23.12.13	Overflow Interrupt Output Function (during TAUDTTINm Width Measurement) .....	1650
23.12.13.1	Overview .....	1650
23.12.13.2	Block Diagram and General Timing Diagram.....	1651
23.12.13.3	Register Settings .....	1652
23.12.13.4	Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement) .....	1653
23.12.14	Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection) .....	1654
23.12.14.1	Overview .....	1654
23.12.14.2	Block Diagram and General Timing Diagram.....	1655
23.12.14.3	Register Settings .....	1656
23.12.14.4	Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection) .....	1657

23.12.15 One-Phase PWM Output Function .....	1658
23.12.15.1 Overview .....	1658
23.12.15.2 Block Diagram and General Timing Diagram.....	1659
23.12.15.3 Register Settings for Lower Channels.....	1661
23.12.15.4 Register Settings for Upper Channels.....	1663
23.12.15.5 Operating Procedure for One-phase PWM Output Function.....	1664
23.13 Independent Channel Real-Time Functions .....	1665
23.13.1 Real-Time Output Function Type 1 .....	1665
23.13.1.1 Overview .....	1665
23.13.1.2 Equations .....	1666
23.13.1.3 Block Diagram and General Timing Diagram.....	1666
23.13.1.4 Register Settings for Upper Channels.....	1668
23.13.1.5 Register Settings for Lower Channels.....	1670
23.13.1.6 Operating Procedure for Real-Time Output Function Type 1.....	1671
23.13.1.7 Specific Timing Diagrams.....	1672
23.13.2 Real-Time Output Function Type 2 .....	1673
23.13.2.1 Overview .....	1673
23.13.2.2 Block Diagram and General Timing Diagram.....	1674
23.13.2.3 Register Settings for Upper Channels.....	1676
23.13.2.4 Register Settings for Lower Channels.....	1678
23.13.2.5 Operating Procedure for Real-Time Output Function Type 2.....	1679
23.13.2.6 Specific Timing Diagrams.....	1680
23.14 Independent Channel Simultaneous Rewrite Functions.....	1681
23.14.1 Simultaneous Rewrite Trigger Generation Function Type 1.....	1681
23.14.1.1 Overview .....	1681
23.14.1.2 Equations .....	1682
23.14.1.3 Block Diagram and General Timing Diagram.....	1683
23.14.1.4 Register Settings for Upper Channels.....	1685
23.14.1.5 Register Settings for Lower Channels.....	1686
23.14.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1 .....	1687
23.14.2 Simultaneous Rewrite Trigger Generation Function Type 2.....	1688
23.14.2.1 Overview .....	1688
23.14.2.2 Block Diagram and General Timing Diagram.....	1689
23.14.2.3 Register Settings for Upper Channels.....	1691
23.14.2.4 Register Settings for Lower Channels.....	1693
23.14.2.5 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2 .....	1694
23.15 Synchronous Channel Operation Functions .....	1696
23.15.1 PWM Output Function .....	1696
23.15.1.1 Overview .....	1696
23.15.1.2 Equations .....	1697
23.15.1.3 Block Diagram and General Timing Diagram.....	1697
23.15.1.4 Register Settings for the Master Channel .....	1699
23.15.1.5 Register Settings for Slave Channels.....	1701
23.15.1.6 Operating Procedure for PWM Output Function .....	1703
23.15.1.7 Specific Timing Diagrams.....	1704
23.15.2 One-Shot Pulse Output Function.....	1707

23.15.2.1	Overview .....	1707
23.15.2.2	Equations .....	1708
23.15.2.3	Block Diagram and General Timing Diagram.....	1708
23.15.2.4	Register Settings for the Master Channel .....	1710
23.15.2.5	Register Settings for Slave Channels.....	1712
23.15.2.6	Operating Procedure for One-Shot Pulse Output Function.....	1714
23.15.2.7	Specific Timing Diagrams.....	1715
23.15.3	Trigger Start PWM Output Function.....	1719
23.15.3.1	Overview .....	1719
23.15.3.2	Equations .....	1720
23.15.3.3	Block Diagram and General Timing Diagram.....	1720
23.15.3.4	Register Settings for the Master Channel .....	1722
23.15.3.5	Register Settings for Slave Channels.....	1724
23.15.3.6	Operating Procedure for Trigger Start PWM Output Function .....	1726
23.15.3.7	Specific Timing Diagrams.....	1727
23.15.4	Delay Pulse Output Function.....	1730
23.15.4.1	Overview .....	1730
23.15.4.2	Equations .....	1731
23.15.4.3	Block Diagram and General Timing Diagram.....	1732
23.15.4.4	Register Settings for the Master Channel .....	1734
23.15.4.5	Register Settings for Slave Channel 1 .....	1736
23.15.4.6	Register Settings for Slave Channel 2 .....	1738
23.15.4.7	Register Settings for Slave Channel 3 .....	1740
23.15.4.8	Operating Procedure for Delay Pulse Output Function .....	1742
23.15.4.9	Specific Timing Diagrams.....	1744
23.15.5	Offset Trigger Output Function.....	1746
23.15.5.1	Overview .....	1746
23.15.5.2	Equations .....	1746
23.15.5.3	Block Diagram and General Timing Diagram.....	1747
23.15.5.4	Register Settings for the Master Channel .....	1749
23.15.5.5	Register Settings for Slave Channels.....	1751
23.15.5.6	Operating Procedure for Offset Trigger Output Function .....	1753
23.15.5.7	Specific Timing Diagrams.....	1754
23.15.6	A/D Conversion Trigger Output Function Type 1 .....	1756
23.15.6.1	Overview .....	1756
23.15.6.2	Block Diagram and General Timing Diagram.....	1756
23.15.7	Triangle PWM Output Function .....	1758
23.15.7.1	Overview .....	1758
23.15.7.2	Equations .....	1759
23.15.7.3	Block Diagram and General Timing Diagram.....	1760
23.15.7.4	Register Settings for the Master Channel .....	1762
23.15.7.5	Register Settings for Slave Channels.....	1764
23.15.7.6	Operating Procedure for Triangle PWM Output Function .....	1766
23.15.7.7	Specific Timing Diagrams.....	1767
23.15.8	Triangle PWM Output Function with Dead Time .....	1769
23.15.8.1	Overview .....	1769
23.15.8.2	Equations .....	1771
23.15.8.3	Block Diagram and General Timing Diagram.....	1772
23.15.8.4	Register Settings for the Master Channel .....	1774

23.15.8.5	Register Settings for Slave Channel 2 .....	1776
23.15.8.6	Register Settings for Slave Channel 3 .....	1778
23.15.8.7	Operating Procedure for Triangle PWM Output Function with Dead Time .....	1780
23.15.8.8	Specific Timing Diagrams.....	1781
23.15.9	A/D Conversion Trigger Output Function Type 2 .....	1783
23.15.9.1	Overview .....	1783
23.15.9.2	Block Diagram and General Timing Diagram.....	1783
23.15.10	Interrupt Request Signals Culling Function .....	1785
23.15.10.1	Overview .....	1785
23.15.10.2	Equations .....	1786
23.15.10.3	Block Diagram and General Timing Diagram.....	1786
23.15.10.4	Register Settings for the Master Channel .....	1788
23.15.10.5	Register Settings for the Slave Channel .....	1790
23.15.10.6	Operating Procedure for Interrupt Request Signals Culling Function .....	1791
23.15.10.7	Specific Timing Diagrams.....	1792
23.16	Synchronous Non-Complementary and Complementary Modulation Output Functions .....	1793
23.16.1	Non-Complementary Modulation Output Function Type 1 .....	1793
23.16.1.1	Overview .....	1793
23.16.1.2	Equations .....	1795
23.16.1.3	Block Diagram and General Timing Diagram.....	1796
23.16.1.4	Register Settings for the Master Channel .....	1798
23.16.1.5	Register Settings for Slave Channel 1 .....	1800
23.16.1.6	Register Settings for Slave Channels 2 to 7.....	1802
23.16.1.7	Operating Procedure for Non-Complementary Modulation Output Function Type 1 .....	1804
23.16.1.8	Specific Timing Diagrams.....	1806
23.16.2	Non-Complementary Modulation Output Function Type 2 .....	1807
23.16.2.1	Overview .....	1807
23.16.2.2	Equations .....	1809
23.16.2.3	Block Diagram and General Timing Diagram.....	1810
23.16.2.4	Register Settings the Master Channel.....	1812
23.16.2.5	Register Settings for Slave Channel 1 .....	1814
23.16.2.6	Register settings for slave channels 2 to 7.....	1816
23.16.2.7	Operating Procedure for Non-Complementary Modulation Output Function Type 2 .....	1818
23.16.2.8	Specific Timing Diagrams.....	1820
23.16.3	Complementary Modulation Output Function .....	1821
23.16.3.1	Overview .....	1821
23.16.3.2	Equations .....	1824
23.16.3.3	Block Diagram and General Timing Diagram.....	1825
23.16.3.4	Register Settings for the Master Channel .....	1827
23.16.3.5	Register Settings for Slave Channel 1 .....	1829
23.16.3.6	Register settings for slave channels 2, 4, and 6.....	1831
23.16.3.7	Register settings for slave channels 3, 5, and 7.....	1833
23.16.3.8	Operating Procedure for Complementary Modulation Output Function .....	1835
23.16.3.9	Specific Timing Diagrams.....	1837
Section 24	Timer Array Unit J (TAUJ).....	1839
24.1	Features of RH850/F1L TAUJ .....	1839

24.1.1	Number of Units.....	1839
24.1.2	Register Base Address.....	1839
24.1.3	Clock Supply.....	1840
24.1.4	Interrupt Requests .....	1840
24.1.5	Reset Sources .....	1840
24.1.6	External Input/Output Signals.....	1841
24.1.7	Internal Input/Output Signals .....	1841
24.1.8	TAUJ0 Input Selection.....	1842
24.1.8.1	SELB_TAUJ0I — TAUJTTINm Input Signal Selection Register .....	1843
24.2	Overview.....	1844
24.2.1	Functional Overview .....	1844
24.2.2	Terms .....	1845
24.2.3	Functional List of Timer Operations.....	1846
24.2.4	TAUJ I/O and Interrupt Request Signals .....	1846
24.2.5	Block Diagram .....	1847
24.2.6	Description of Block Diagram .....	1848
24.3	Registers.....	1849
24.3.1	List of Registers.....	1849
24.3.2	Details of TAUJn Prescaler Registers .....	1850
24.3.2.1	TAUJnTPS — TAUJn Prescaler Clock Select Register .....	1850
24.3.2.2	TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register.....	1853
24.3.3	Details of TAUJn Control Registers.....	1854
24.3.3.1	TAUJnCDRm — TAUJn Channel Data Register.....	1854
24.3.3.2	TAUJnCNTm — TAUJn Channel Counter Register.....	1855
24.3.3.3	TAUJnCMORm — TAUJn Channel Mode OS Register.....	1857
24.3.3.4	TAUJnCMURm — TAUJn Channel Mode User Register.....	1860
24.3.3.5	TAUJnCSRm — TAUJn Channel Status Register .....	1861
24.3.3.6	TAUJnCSCm — TAUJn Channel Status Clear Trigger Register .....	1861
24.3.3.7	TAUJnTS — TAUJn Channel Start Trigger Register .....	1862
24.3.3.8	TAUJnTE — TAUJn Channel Enable Status Register .....	1862
24.3.3.9	TAUJnTT — TAUJn Channel Stop Trigger Register .....	1863
24.3.4	Details of TAUJn Simultaneous Rewrite Register .....	1864
24.3.4.1	TAUJnRDE — TAUJn Channel Reload Data Enable Register .....	1864
24.3.4.2	TAUJnRDM — TAUJn Channel Reload Data Mode Register .....	1864
24.3.4.3	TAUJnRDT — TAUJn Channel Reload Data Trigger Register .....	1865
24.3.4.4	TAUJnRSF — TAUJn Channel Reload Status Register .....	1865
24.3.5	Details of TAUJn Output Registers.....	1866
24.3.5.1	TAUJnTOE — TAUJn Channel Output Enable Register.....	1866
24.3.5.2	TAUJnTO — TAUJn Channel Output Register .....	1866
24.3.5.3	TAUJnTOM — TAUJn Channel Output Mode Register .....	1867
24.3.5.4	TAUJnTOC — TAUJn Channel Output Configuration Register .....	1867
24.3.5.5	TAUJnTOL — TAUJn Channel Output Level Register.....	1869
24.3.5.6	TAUJnEMU — TAUJn Emulation Register .....	1869
24.4	Operating Procedure .....	1870
24.5	Concepts of Synchronous Channel Operation Function.....	1871
24.5.1	Rules of Synchronous Channel Operation Function .....	1871
24.5.2	Simultaneous Start and Stop of Synchronous Channel Counters.....	1873

24.5.2.1	Simultaneous Start and Stop within a TAUJ Unit .....	1873
24.5.2.2	Simultaneous Start between TAUJ Units .....	1873
24.6	Simultaneous Rewrite .....	1874
24.6.1	How to Control Simultaneous Rewrite .....	1874
24.6.1.1	Initial Settings .....	1875
24.6.1.2	Start Counter and Count Operation .....	1875
24.6.1.3	Simultaneous Rewrite .....	1875
24.6.2	Other General Rules for Simultaneous Rewrite .....	1875
24.6.3	Simultaneous Rewrite Procedure .....	1876
24.7	Channel Output Modes .....	1878
24.7.1	General Procedures for Specifying a Channel Output Mode .....	1880
24.7.2	Channel Output Modes Controlled Independently by TAUJn Signals .....	1881
24.7.2.1	Independent Channel Output Mode 1 .....	1881
24.7.3	Channel Output Modes Controlled Synchronously by TAUJn Signals .....	1881
24.7.3.1	Synchronous Channel Output Mode 1 .....	1881
24.8	Start Timing in Each Operating Modes .....	1882
24.8.1	Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode .....	1882
24.8.2	Other Operating Modes .....	1883
24.9	TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts .....	1884
24.10	Interrupt Generation upon Overflow .....	1885
24.10.1	Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function .....	1886
24.11	TAUJTTINm Edge Detection .....	1887
24.12	Independent Channel Operation Functions .....	1888
24.12.1	Interval Timer Function .....	1888
24.12.1.1	Overview .....	1888
24.12.1.2	Equations .....	1888
24.12.1.3	Block Diagram and General Timing Diagram .....	1889
24.12.1.4	Register Settings .....	1890
24.12.1.5	Operating Procedure for Interval Timer Function .....	1892
24.12.1.6	Specific Timing Diagrams .....	1893
24.12.2	TAUJTTINm Input Interval Timer Function .....	1895
24.12.2.1	Overview .....	1895
24.12.2.2	Equations .....	1895
24.12.2.3	Block Diagram and General Timing Diagram .....	1896
24.12.2.4	Register Settings .....	1897
24.12.2.5	Operating Procedure for TAUJTTINm Input Interval Timer Function .....	1899
24.12.2.6	Specific Timing Diagrams .....	1900
24.12.3	TAUJTTINm Input Pulse Interval Measurement Function .....	1901
24.12.3.1	Overview .....	1901
24.12.3.2	Equations .....	1902
24.12.3.3	Block Diagram and General Timing Diagram .....	1902
24.12.3.4	Register Settings .....	1904
24.12.3.5	Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function .....	1905
24.12.3.6	Specific Timing Diagrams: Overflow Behavior .....	1906

24.12.4 TAUJTTINm Input Signal Width Measurement Function.....	1910
24.12.4.1 Overview .....	1910
24.12.4.2 Equations .....	1911
24.12.4.3 Block Diagram and General Timing Diagram.....	1911
24.12.4.4 Register Settings .....	1912
24.12.4.5 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function .....	1913
24.12.4.6 Specific Timing Diagrams: Overflow Behavior .....	1914
24.12.5 TAUJTTINm Input Position Detection Function.....	1918
24.12.5.1 Overview .....	1918
24.12.5.2 Equations .....	1918
24.12.5.3 Block Diagram and General Timing Diagram.....	1919
24.12.5.4 Register Settings .....	1920
24.12.5.5 Operating Procedure for TAUJTTINm Input Position Detection Function .....	1921
24.12.5.6 Specific Timing Diagrams.....	1922
24.12.6 TAUJTTINm Input Period Count Detection Function.....	1923
24.12.6.1 Overview .....	1923
24.12.6.2 Equations .....	1923
24.12.6.3 Block Diagram and General Timing Diagram.....	1924
24.12.6.4 Register Settings .....	1925
24.12.6.5 Operating Procedure for TAUJTTINm Input Period Count Detection Function...	1926
24.12.6.6 Specific Timing Diagrams.....	1927
24.12.7 Overflow Interrupt Output Function (during TAUJTTINm Width Measurement).....	1928
24.12.7.1 Overview .....	1928
24.12.7.2 Block Diagram and General Timing Diagram.....	1929
24.12.7.3 Register Settings .....	1930
24.12.7.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement) .....	1931
24.12.8 Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection).....	1932
24.12.8.1 Overview .....	1932
24.12.8.2 Block Diagram and General Timing Diagram.....	1933
24.12.8.3 Register Settings .....	1934
24.12.8.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection) .....	1935
24.13 Synchronous Channel Operation Functions .....	1936
24.13.1 PWM Output Function .....	1936
24.13.1.1 Overview .....	1936
24.13.1.2 Equations .....	1938
24.13.1.3 Block Diagram and General Timing Diagram.....	1938
24.13.1.4 Register Settings for the Master Channel .....	1940
24.13.1.5 Register Settings for the Slave Channel(s) .....	1942
24.13.1.6 Operating Procedure for PWM Output Function .....	1944
24.13.1.7 Specific Timing Diagrams.....	1945
<b>Section 25 Real-Time Clock (RTCA) .....</b>	<b>1948</b>
25.1 Features of RH850/F1L RTCA .....	1948
25.1.1 Number of Units and Channels .....	1948
25.1.2 Register Base Address.....	1948
25.1.3 Clock Supply.....	1949

25.1.4	Interrupt Requests .....	1949
25.1.5	Reset Sources .....	1949
25.1.6	External Input/Output Signals .....	1949
25.2	Overview .....	1950
25.2.1	Functional Overview .....	1950
25.2.2	Block Diagram .....	1950
25.2.3	Description of Blocks .....	1951
25.3	Registers .....	1952
25.3.1	List of Registers .....	1952
25.3.2	Details of RTCA Control Registers .....	1953
25.3.2.1	RTCAnCTL0 — RTCA Control Register 0 .....	1953
25.3.2.2	RTCAnCTL1 — RTCA Control Register 1 .....	1954
25.3.2.3	RTCAnCTL2 — RTCA Control Register 2 .....	1955
25.3.3	Details of RTCA Sub-Counter Registers .....	1957
25.3.3.1	RTCAnSUBC — RTCA Sub-Count Register .....	1957
25.3.3.2	RTCAnSRBU — RTCA Sub-Count Register Read Buffer .....	1958
25.3.3.3	RTCAnSUBU — RTCA Clock Error Correction Register .....	1959
25.3.3.4	RTCAnSCMP — RTCA Sub-Counter Compare Register .....	1960
25.3.4	Details of RTCA Clock Counter and Buffer Registers .....	1961
25.3.4.1	RTCAnSECC — RTCA Seconds Count Register .....	1961
25.3.4.2	RTCAnSEC — RTCA Seconds Count Buffer Register .....	1962
25.3.4.3	RTCAnMINC — RTCA Minutes Count Register .....	1963
25.3.4.4	RTCAnMIN — RTCA Minutes Count Buffer Register .....	1964
25.3.4.5	RTCAnHOURC — RTCA Hours Count Register .....	1965
25.3.4.6	RTCAnHOUR — RTCA Hours Count Buffer Register .....	1967
25.3.4.7	RTCAnWEEKC — RTCA Day of the Week Count Register .....	1968
25.3.4.8	RTCAnWEEK — RTCA Day of the Week Count Buffer Register .....	1969
25.3.4.9	RTCAnDAYC — RTCA Day of the Month Count Register .....	1970
25.3.4.10	RTCAnDAY — RTCA Day of the Month Count Buffer Register .....	1971
25.3.4.11	RTCAnMONC — RTCA Month Count Register .....	1972
25.3.4.12	RTCAnMONTH — RTCA Month Count Buffer Register .....	1973
25.3.4.13	RTCAnYEARC — RTCA Year Count Register .....	1974
25.3.4.14	RTCAnYEAR — RTCA Year Count Buffer Register .....	1975
25.3.5	Details of RTCA Special Counter and Buffer Registers .....	1976
25.3.5.1	RTCAnTIMEC — RTCA Time Count Register .....	1976
25.3.5.2	RTCAnTIME — RTCA Time Count Buffer Register .....	1977
25.3.5.3	RTCAnCALC — RTCA Calendar Count Register .....	1978
25.3.5.4	RTCAnCAL — RTCA Calendar Count Buffer Register .....	1979
25.3.6	Details of RTCA Alarm Setting Registers .....	1980
25.3.6.1	RTCAnALM — RTCA Alarm Minute Setting Register .....	1980
25.3.6.2	RTCAnALH — RTCA Alarm Hour Setting Register .....	1981
25.3.6.3	RTCAnALW — RTCA Alarm Day of the Week Setting Register .....	1982
25.3.7	RTCA Emulation Register .....	1983
25.3.7.1	RTCAnEMU — RTCA Emulation Register .....	1983
25.4	Operation .....	1984
25.4.1	Clock Counter Format .....	1985
25.4.2	Fixed Interval Interrupt Function .....	1985

25.4.3	Alarm Interrupt Function .....	1985
25.4.4	Clock Error Correction .....	1986
25.4.4.1	Setting the Correction Value and the Operator .....	1988
25.4.4.2	Impact of the Repetition Interval.....	1988
25.4.4.3	Sample Settings .....	1989
25.5	Procedures for Setup, Writing and Reading .....	1991
25.5.1	Initial Setting of the RTCA .....	1991
25.5.1.1	RTCA Stop Procedure.....	1991
25.5.1.2	RTCA Initialization Procedure .....	1992
25.5.2	Updating Clock Counters.....	1993
25.5.3	Reading Clock Counters.....	1994
25.5.3.1	Procedure for Reading Count Buffer Registers.....	1994
25.5.3.2	Procedure for Reading Counter Registers Directly .....	1996
25.5.4	Reading RTCAnSRBU .....	1997
25.5.5	Writing to RTCAnSUBU.....	1998
25.5.6	Writing to RTCAnSCMP .....	1999
25.6	Timing Diagrams.....	2000
25.6.1	Timing of Counter Start .....	2000
25.6.2	Timing of Clock Counter Update while Counter Is Enabled .....	2001
25.6.3	Timing of Sub-Counter Read Buffer Reading while Counter is Enabled .....	2002
<b>Section 26</b>	<b>Encoder Timer (ENCA).....</b>	<b>2003</b>
26.1	Features of RH850/F1L ENCA .....	2003
26.1.1	Number of Units and Channels .....	2003
26.1.2	Register Base Address.....	2003
26.1.3	Clock Supply.....	2004
26.1.4	Interrupt Requests .....	2004
26.1.5	Reset Sources .....	2004
26.1.6	External Input/Output Signals.....	2004
26.1.7	Internal Input/Output Signals .....	2005
26.2	Overview.....	2006
26.2.1	Functional Overview .....	2006
26.2.2	Block Diagram .....	2007
26.3	Registers.....	2008
26.3.1	List of Registers.....	2008
26.3.2	ENCAnCTL — ENCAn Control Register .....	2009
26.3.3	ENCAnIOC0 — ENCAn I/O Control Register 0.....	2011
26.3.4	ENCAnIOC1 — ENCAn I/O Control Register 1 .....	2012
26.3.5	ENCAnFLG — ENCAn Status Flag Register .....	2014
26.3.6	ENCAnFGC — ENCAn Status Flag Clear Register .....	2015
26.3.7	ENCAnCCR0 — ENCAn Capture/Compare Register 0 .....	2016
26.3.8	ENCAnCCR1 — ENCAn Capture/Compare Register 1 .....	2017
26.3.9	ENCAnCNT — ENCAn Counter Register .....	2018
26.3.10	ENCAnTE — ENCAn Timer Enable Status Register .....	2019
26.3.11	ENCAnTS — ENCAn Timer Start Trigger Register.....	2020

26.3.12 ENCAAnTT — ENCAAn Timer Stop Trigger Register .....	2021
26.3.13 ENCAAnEMU — ENCAAn Emulation Register.....	2022
<b>26.4 Operation .....</b>	<b>2023</b>
26.4.1 Timer Counter Operation.....	2023
26.4.2 Up/Down Control of Timer Counter .....	2025
26.4.2.1 When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 00 <sub>B</sub> ...	2025
26.4.2.2 When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 01 <sub>B</sub> ...	2026
26.4.2.3 When the ENCAAnUDS1 and ENCAAnUDS0 Bits in the ENCAAnCTL Register = 10 <sub>B</sub> .....	2027
26.4.2.4 When ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 11 <sub>B</sub> .....	2028
26.4.3 Timer Counter Clear Control by Encoder Input .....	2029
26.4.3.1 Clearing Method when ENCAAnSCE = 0 .....	2029
26.4.3.2 Clearing Method when ENCAAnSCE = 1 .....	2029
26.4.4 Functions of ENCAAnCCR0 .....	2030
26.4.4.1 Compare Function.....	2030
26.4.4.2 Capture Function.....	2030
26.4.5 Functions of ENCAAnCCR1 .....	2031
26.4.5.1 Compare Function.....	2031
26.4.5.2 Capture Function.....	2032
26.4.5.3 Timer Counter Clearing upon Compare Register Match .....	2033
26.4.6 Startup/Stop of Timer Counter.....	2034
26.4.6.1 Startup of Timer.....	2034
26.4.6.2 Stop of Timer.....	2034
<b>26.5 ENCAAn Setting Sequences.....</b>	<b>2035</b>
26.5.1 ENCAAn Setting Procedure.....	2035
26.5.1.1 Initial Setting Procedure for the Counter .....	2035
26.5.1.2 Initial Setting Procedure for Counter Clear.....	2036
26.5.1.3 Setting Procedure for ENCAAnCCR0 Register .....	2036
26.5.1.4 Setting Procedure for ENCAAnCCR1 Register .....	2036
<b>26.6 Timing Chart .....</b>	<b>2037</b>
26.6.1 Overflow Occurrence and Overflow Flag Clear Operation .....	2037
26.6.2 Underflow Occurrence and Underflow Flag Clear Operation .....	2038
26.6.3 Count Clearing and Capture Operation by Encoder Clear Input (ENCAAnEC Pin).....	2039
26.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin).....	2040
26.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin).....	2041
26.6.6 Overflow Operation Immediately after Startup .....	2042
26.6.7 Underflow Operation Immediately after Startup .....	2043
26.6.8 Using the ENCAAnLDE Function Immediately after Startup .....	2044
26.6.9 ENCAAnLDE Function (Loading Count Value).....	2045
26.6.10 Conflict between ENCAAnLDE Function (Loading Counter Value) and Rewrite of ENCAAnCCR0 Register .....	2047
26.6.11 Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAAnEC Pin) .....	2048
26.6.12 Up-count after Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input .....	2050
26.6.13 Capture Operation between Count Clocks (ENCAAnCCR1).....	2051

26.6.14	Capture Operation between Count Clocks (ENCA <sub>n</sub> CCR0) .....	2052
26.6.15	Encoder Operation when Compare Match Clear Control is Enabled and ENCA <sub>n</sub> CTS = 0 .....	2053
26.6.16	Encoder Operation when Compare Match Clear Control is Enabled and ENCA <sub>n</sub> CTS = 1 .....	2054
26.6.17	Encoder Operation when Compare Match Clear Control is Disabled .....	2055
26.6.18	Capture Operation Performed upon Clearing by ENCA <sub>n</sub> EC, ENCA <sub>n</sub> E0, or ENCA <sub>n</sub> E1 when ENCA <sub>n</sub> SCE = 1 .....	2056
26.6.18.1	Accompanying capture operation.....	2056
26.6.18.2	When the Timing of the ENCA <sub>n</sub> EC Input is Later than that of the ENCA <sub>n</sub> E1 Input during Up-count (When ENCA <sub>n</sub> ACL = 1, ENCA <sub>n</sub> BCL = 0, ENCA <sub>n</sub> ZCL = 1, and ENCA <sub>n</sub> UDS[1:0] = 11 <sub>B</sub> ).....	2057
26.6.18.3	When the Timing of the ENCA <sub>n</sub> EC Input is the Same as that of the ENCA <sub>n</sub> E1 Input during Up-count (When ENCA <sub>n</sub> ACL = 1, ENCA <sub>n</sub> BCL = 0, ENCA <sub>n</sub> ZCL = 1, and ENCA <sub>n</sub> UDS[1:0] = 11 <sub>B</sub> ).....	2058
26.6.18.4	When the Timing of the ENCA <sub>n</sub> EC Input is Earlier than that of the ENCA <sub>n</sub> E1 Input during Up-count (When ENCA <sub>n</sub> ACL = 1, ENCA <sub>n</sub> BCL = 0, ENCA <sub>n</sub> ZCL = 1, and ENCA <sub>n</sub> UDS[1:0] = 11 <sub>B</sub> ).....	2058
26.6.18.5	When the Timing of the ENCA <sub>n</sub> EC Input is Later than that of the ENCA <sub>n</sub> E1 Input during Down-count (When ENCA <sub>n</sub> ACL = 1, ENCA <sub>n</sub> BCL = 0, ENCA <sub>n</sub> ZCL = 1, and ENCA <sub>n</sub> UDS[1:0] = 11 <sub>B</sub> ).....	2059
26.6.19	Capture Operation Performed upon Clearing by ENCA <sub>n</sub> EC when ENCA <sub>n</sub> SCE = 0.....	2060
<b>Section 27</b>	<b>Motor Control .....</b>	<b>2061</b>
27.1	Features of RH850/F1L Motor Control .....	2061
27.1.1	Number of Units and Channels .....	2061
27.1.2	Register Base Address.....	2062
27.1.3	Clock Supply .....	2062
27.1.4	Interrupt Request.....	2062
27.1.5	Reset Sources .....	2063
27.1.6	External Input/Output Signal.....	2063
27.1.7	Internal Output Signal.....	2064
27.2	Overview.....	2065
27.2.1	Functional Overview .....	2065
27.2.2	Basic Structure of Motor Control .....	2066
27.2.3	Block Diagram .....	2067
27.2.4	Definition of Terms .....	2068
27.3	Registers.....	2069
27.3.1	List of Registers.....	2069
27.3.2	TAPAnCTL0 — TAPA Control Register 0 .....	2070
27.3.3	TAPAnCTL1 — TAPA Control Register 1 .....	2071
27.3.4	TAPAnFLG — TAPA Flag Register.....	2072
27.3.5	TAPAnACWE — TAPA Asynchronous Hi-Z Control Write Enable Register .....	2073
27.3.6	TAPAnACTS — TAPA Asynchronous Hi-Z Control Start Trigger Register .....	2073
27.3.7	TAPAnACTT — TAPA Asynchronous Hi-Z Control Stop Trigger Register .....	2074
27.3.8	TAPAnOPHS — TAPA Hi-Z Start Trigger Register.....	2074
27.3.9	TAPAnOPHT — TAPA Hi-Z Stop Trigger Register .....	2075
27.3.10	TAPAnEMU — TAPA Emulation Register.....	2075

27.4	Asynchronous Hi-Z Control Function.....	2076
27.4.1	Overview.....	2076
27.4.2	System Configuration Example .....	2076
27.4.3	Basic Operation.....	2078
27.4.4	Asynchronous Hi-Z Control Using Software Trigger.....	2080
27.4.5	Operating Procedure .....	2081
27.4.6	TAPA0 Hi-Z Control Input Selection .....	2082
27.4.7	Registers .....	2083
27.4.7.1	PIC0HIZCEN $n$ — Hi-Z Output Control Register $n$ ( $n = 0$ ) .....	2083
27.5	INT Signal Output Selection Function.....	2084
27.5.1	Configuration of the INT Signal Output Selection Function .....	2084
27.5.2	Block Diagram .....	2085
27.5.3	Registers .....	2086
27.5.3.1	PIC0REG2n0 — Timer I/O Control Register 2n0 ( $n = 0$ ) .....	2086
27.6	A/D Converter Conversion Trigger Selection Function.....	2087
27.6.1	Configuration of A/D Converter Conversion Trigger Selection Function .....	2087
27.6.2	Block Diagram .....	2088
27.6.3	Waveforms of A/D Converter Conversion Trigger Output Control Operation in Triangle PWM Mode.....	2089
27.6.4	Operating Procedure for A/D Converter Conversion Trigger Selection Function .....	2091
27.7	ADCA Trigger Selection Function.....	2092
27.7.1	Functional Overview .....	2092
27.7.2	Configuration .....	2092
27.7.3	Registers .....	2093
27.7.3.1	PIC0ADTEN4nj — A/D Conversion Trigger Output Control Register 4nj ( $n = 0, j = 0$ to 2) .....	2093
27.7.4	Example of Operation.....	2094
27.7.5	Setup Flow.....	2094
27.8	Simultaneous Start Trigger Function .....	2095
27.8.1	Functional Overview .....	2095
27.8.2	Configuration .....	2095
27.8.3	Registers .....	2096
27.8.3.1	PIC0SSER0 — Simultaneous Start Control Register 0.....	2096
27.8.3.2	PIC0SSER2 — Simultaneous Start Control Register 2.....	2096
27.8.3.3	PIC0SST — Simultaneous Start Trigger Control Register .....	2097
27.8.4	Example of Operation.....	2098
27.8.5	Setup Flow.....	2099
27.9	Trigger & Pulse Width Measuring Function .....	2100
27.9.1	Functional Overview .....	2100
27.9.2	Configuration .....	2100
27.9.3	Registers .....	2102
27.9.3.1	PIC0REG31 — Timer I/O Control Register 31 .....	2102
27.9.4	Example of Operation.....	2104
27.9.5	Setup Flow.....	2106
27.9.6	Setting Examples for Operation Functions .....	2109
27.10	A/D Trigger Encoder Capture Function .....	2112

27.10.1	Functional Overview .....	2112
27.10.2	Configuration .....	2112
27.10.3	Registers .....	2113
27.10.3.1	PIC0REG30 — Timer I/O Control Register 30 .....	2113
27.10.4	Example of Operation.....	2114
27.10.5	Setup Flow.....	2115
27.10.6	Setting Examples for Operation Functions .....	2116
27.11	Three-Phase PWM Output with Dead Time .....	2117
27.11.1	Functional Overview .....	2117
27.11.2	Configuration .....	2117
27.11.3	Registers .....	2120
27.11.3.1	PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0) .....	2120
27.11.3.2	PIC0HIZCENn — Hi-Z Output Control Register n (n = 0) .....	2121
27.11.4	Operation Example.....	2122
27.11.4.1	Pwm Output.....	2122
27.11.4.2	One-Phase PWM Output.....	2122
27.11.4.3	SR Flip-Flop Circuit .....	2123
27.11.5	Setup Flow.....	2131
27.11.6	Setting Examples for Operation Functions .....	2133
27.11.6.1	TAUDn Settings (Active High Example) .....	2133
27.11.6.2	PIC Settings .....	2138
27.12	High-accuracy Triangle PWM Output with Dead Time .....	2139
27.12.1	Functional Overview .....	2139
27.12.2	Configuration .....	2140
27.12.3	Registers .....	2142
27.12.3.1	PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0) .....	2142
27.12.3.2	PIC0REG2n1 — Timer I/O Control Register 2n1 (n = 0) .....	2143
27.12.3.3	PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0) .....	2145
27.12.3.4	PIC0REG2n3 — Timer I/O Control Register 2n3 (n = 0) .....	2146
27.12.3.5	PIC0HIZCENn — Hi-Z Output Control Register n (n = 0) .....	2148
27.12.4	Operation Example.....	2149
27.12.4.1	Triangle PWM Output with Dead Time.....	2149
27.12.4.2	One-shot Pulse Output.....	2149
27.12.4.3	U phase Combination Circuit (PFN001) .....	2150
27.12.4.4	Logical Operation Circuit (FN0i) (i = 0 or 1) .....	2152
27.12.5	Setup Flow.....	2162
27.12.6	Setting Examples for Operation Functions .....	2164
27.12.6.1	TAUDn settings (active high example) .....	2164
27.12.6.2	PIC Settings (Active High Example) .....	2170
27.13	Delay Pulse Output with Dead Time.....	2171
27.13.1	Functional Overview .....	2171
27.13.2	Configuration .....	2171
27.13.2.1	TAUDn configuration .....	2172
27.13.3	Registers .....	2174
27.13.3.1	PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0) .....	2174
27.13.3.2	PIC0HIZCENn — Hi-Z Output Control Register n (n = 0) .....	2175
27.13.4	Operation Example.....	2176

27.13.4.1	Delay pulse output function .....	2176
27.13.4.2	One-phase PWM output.....	2176
27.13.5	Setup Flow.....	2182
27.13.6	Setting Examples for Operation Functions .....	2185
27.13.6.1	TAUDn Settings.....	2185
27.13.6.2	Peripheral Interconnections Settings.....	2190
<b>Section 28</b>	<b>PWM Output/Diagnostic (PWM-Diag).....</b>	<b>2191</b>
28.1	Features of RH850/F1L PWM-Diag .....	2191
28.1.1	Number of Units and Channels .....	2191
28.1.2	Register Base Address.....	2192
28.1.3	Clock Supply.....	2192
28.1.4	Interrupt Requests .....	2193
28.1.5	Reset Sources .....	2195
28.1.6	External Input/Output Signals.....	2195
28.1.7	Internal Signals.....	2195
28.1.8	Functional Overview .....	2196
28.1.9	Block Diagram .....	2198
28.2	Registers.....	2199
28.2.1	List of Registers.....	2199
28.2.1.1	PWBAnBRSm Register .....	2200
28.2.1.2	PWBAnTE Register.....	2201
28.2.1.3	PWBAnTS Register.....	2202
28.2.1.4	PWBAnTT Register.....	2203
28.2.1.5	PWBAnEMU Register .....	2204
28.2.1.6	PWGAnCTL — PWGA Control Register .....	2205
28.2.1.7	PWGAnCNT — PWM Cycle Count Register.....	2205
28.2.1.8	PWGAnCSDR — PWM Output Set Condition Register .....	2206
28.2.1.9	PWGAnCRDR — PWM Output Reset Condition Register .....	2206
28.2.1.10	PWGAnCTDR — PWGA_TRGOUTn Generation Condition Register .....	2207
28.2.1.11	PWGAnCSBR — PWGAnCSDR Buffer Register.....	2207
28.2.1.12	PWGAnCRBR — PWGAnCRDR Buffer Register .....	2208
28.2.1.13	PWGAnCTBR — PWGAnCTDR Buffer Register .....	2208
28.2.1.14	PWGAnRSF — Buffer Register Reload Status Register.....	2209
28.2.1.15	PWGAnRDT — Buffer Register Reload Trigger Register .....	2209
28.2.1.16	SLPWGAK — PWGA Synchronous Trigger Register (k = 0 to 2) .....	2210
28.2.1.17	PWSAnCTL Register.....	2211
28.2.1.18	PWSAnSTR Register .....	2211
28.2.1.19	PWSAnSTC Register .....	2212
28.2.1.20	PWSAnQUEj (j = 0 to 7) Register .....	2212
28.2.1.21	PWSAnPVCRx_y (x = 00, 02, 04 ... 70, y = 01, 03, 05 ... 71) Register.....	2213
28.2.1.22	PWSAnEMU — Emulation Control Register .....	2214
28.3	Operating Procedure .....	2215
28.4	Operation Waveform of PWM-Diag .....	2217
28.4.1	PWM Waveform Output by PWGA and Operation Waveform for A/D Conversion Trigger Output .....	2217
28.4.1.1	Basic Operation Waveform of PWGA .....	2217

28.4.1.2	Operation Waveform when Simultaneous Rewrite for PWGA is Executed.....	2218
28.4.1.3	Operation Waveform when Stopping and Resuming PWGA Operation .....	2219
28.4.1.4	Waveforms of PWGA Operation with Specific Settings .....	2221
28.4.2	Operation Waveform when A/D Conversion Trigger Occurs in PWSA .....	2222
28.5	PWM-Diag Related Functions in A/D Converter (ADCA) .....	2223
28.5.1	ADCA registers when the PWM-Diag function is used.....	2223
<b>Section 29</b>	<b>A/D Converter (ADCA).....</b>	<b>2224</b>
29.1	Features of RH850/F1L ADCA .....	2224
29.1.1	Number of Units and Channels .....	2224
29.1.2	Register Base Address.....	2225
29.1.3	Clock Supply.....	2225
29.1.4	Interrupt Requests.....	2227
29.1.5	Reset Sources .....	2227
29.1.6	External Input/Output Signals.....	2227
29.2	Overview.....	2230
29.2.1	Functional Overview .....	2230
29.2.2	Block Diagram .....	2232
29.3	Registers.....	2237
29.3.1	List of Registers.....	2237
29.3.2	ADCA Specific Registers.....	2239
29.3.2.1	ADCAnVCRj — Virtual Channel Register j.....	2239
29.3.2.2	ADCAnPWDVCR — PWM-Diag Virtual Channel Register .....	2241
29.3.2.3	ADCAnDRj — Data Register j .....	2242
29.3.2.4	ADCAnDIRj — Data Supplementary Information Register j .....	2244
29.3.2.5	ADCAnPWDTNSDR — PWM-Diag Data Register.....	2245
29.3.2.6	ADCAnPWDDIR — PWM-Diag Data Supplementary Information Register.....	2246
29.3.2.7	ADCAnADHALTR — A/D Force Halt Register .....	2247
29.3.2.8	ADCAnADCR — A/D Control Register.....	2248
29.3.2.9	ADCAnMPXCURR — MPX Current Register .....	2250
29.3.2.10	ADCAnTHSMPSTCR — T&H Sampling Start Control Register.....	2251
29.3.2.11	ADCAnTHCR — T&H Control Register.....	2252
29.3.2.12	ADCAnTHAHLDSCTR — T&H Group A Hold Start Control Register .....	2253
29.3.2.13	ADCAnTBHLDSTCR — T&H Group B Hold Start Control Register .....	2254
29.3.2.14	ADCAnTHACR — T&H Group A Control Register.....	2255
29.3.2.15	ADCAnTHBCR — T&H Group B Control Register .....	2256
29.3.2.16	ADCAnTHER — T&H Enable Register .....	2258
29.3.2.17	ADCAnTHGSR — T&H Group Select Register.....	2259
29.3.2.18	ADCAnSMPCR — Sampling Control Register.....	2261
29.3.2.19	ADCAnSFTCR — Safety Control Register.....	2262
29.3.2.20	ADCAnULLMTBR0 to 2 — Upper Limit/Lower Limit Table Registers 0 to 2 .....	2263
29.3.2.21	ADCAnECR — Error Clear Register .....	2264
29.3.2.22	ADCAnULER — Upper Limit/Lower Limit Error Register .....	2265
29.3.2.23	ADCAnOWER — Overwrite Error Register .....	2267
29.3.3	Scan Group (SG) Specific Registers .....	2268
29.3.3.1	ADCAnSGSTCRx — Scan Group x Start Control Register .....	2268
29.3.3.2	ADCAnSGCRx — Scan Group x Control Register.....	2269

29.3.3.3	ADCAnPWDGCR — PWM-Diag Scan Group Control Register .....	2270
29.3.3.4	ADCAnSGVCSPx — Scan Group x Start Virtual Channel Pointer .....	2271
29.3.3.5	ADCAnSGVCEPx — Scan Group x End Virtual Channel Pointer.....	2272
29.3.3.6	ADCAnSGMCYCRx — Scan Group x Multicycle Register .....	2273
29.3.3.7	ADCAnPWDGSEFCR — PWM-Diag Scan End Flag Clear Register .....	2274
29.3.3.8	ADCAnSGSEFCRx — Scan Group x Scan End Flag Clear Register .....	2275
29.3.3.9	ADCAnSGSTR — Scan Group Status Register.....	2276
29.3.4	Hardware Trigger Specific Register.....	2278
29.3.4.1	ADCAnSGTSELx — Scan Group x Start Trigger Control Register x .....	2278
29.3.5	Self-Diagnosis Specific Registers.....	2280
29.3.5.1	ADCAnDGCTL0 — Self-Diagnosis Control Register 0 .....	2280
29.3.5.2	ADCAnDGCTL1 — Self-Diagnosis Control Register 1 .....	2281
29.3.5.3	ADCAnPDCTL1 — Pull Down Control Register 1 .....	2282
29.3.5.4	ADCAnPDCTL2 — Pull Down Control Register 2.....	2283
29.3.6	Emulation Specific Register.....	2284
29.3.6.1	ADCAnEMU — Emulation Control Register.....	2284
29.4	Operation .....	2285
29.4.1	Initial Setting .....	2285
29.4.2	Trigger Input .....	2286
29.4.3	Ending A/D Conversion .....	2287
29.4.4	Example of Scan Group Operation.....	2288
29.4.5	Channel Repeat Mode.....	2290
29.4.6	Example of Simultaneous Track and Hold Operation.....	2292
29.4.7	A/D Conversion with External Analog Multiplexer .....	2294
29.4.7.1	A/D Conversion with PWM-Diag Enabled .....	2296
29.4.8	Example of Synchronous Suspend and Resume Operation .....	2298
29.4.9	Example of Asynchronous Suspend and Resume Operation .....	2299
29.4.10	Error Detecting Functions.....	2300
29.4.10.1	Upper-Limit/Lower-Limit Error Detecting Function .....	2300
29.4.10.2	Overwrite Error Detecting Function .....	2300
29.4.10.3	SVSTOP Operation.....	2300
29.4.11	Activating Scan Group by a Hardware Trigger .....	2304
29.4.11.1	Stopping Scan Group by ADHALT .....	2304
29.4.12	Scan End Interrupt Request .....	2305
29.4.13	A/D Error Interrupt Request .....	2306
29.5	Self-Diagnostic Function.....	2307
29.5.1	Diagnosis of A/D Conversion Circuit .....	2308
29.5.1.1	Diagnostic procedure .....	2308
29.5.2	Diagnosis of Channel Multiplexer .....	2309
29.5.2.1	Diagnostic procedure .....	2310
29.5.3	Diagnosis of Open Pins .....	2311
29.5.3.1	Diagnostic procedure .....	2312
29.5.4	Diagnosis of T&H Circuit .....	2313
29.5.4.1	Diagnostic Procedure (in case of T&H circuit ch0 diagnosis).....	2313
29.5.4.2	Diagnosis Mechanism .....	2315
29.6	Definition of A/D Conversion Accuracy .....	2316
29.7	Usage Notes .....	2317

29.7.1	Range of Channel Input Voltage .....	2317
29.7.2	Notes on Application Design .....	2317
<b>Section 30 Key Return (KR).....</b>		<b>2322</b>
30.1	Features of RH850/F1L KR .....	2322
30.1.1	Number of Units and Channels .....	2322
30.1.2	Register Base Address.....	2322
30.1.3	Clock Supply.....	2323
30.1.4	Interrupt Requests.....	2323
30.1.5	Reset Sources .....	2323
30.1.6	External Input/Output Signals.....	2323
30.2	Overview.....	2324
30.2.1	Functional Overview .....	2324
30.2.2	Block Diagram .....	2324
30.3	Registers.....	2325
30.3.1	List of Registers.....	2325
30.3.2	KRnKRM — Key Return Mode Register.....	2325
30.4	Operation .....	2326
30.4.1	Interrupt Request INTKRn .....	2326
<b>Section 31 Functional Safety .....</b>		<b>2327</b>
31.1	Overview .....	2327
<b>Section 32 Data CRC (DCRA) .....</b>		<b>2329</b>
32.1	Features of RH850/F1L DCRA.....	2329
32.1.1	Number of Units.....	2329
32.1.2	Register Base Address.....	2329
32.1.3	Clock Supply.....	2330
32.1.4	Reset Sources .....	2330
32.2	Overview.....	2331
32.2.1	Functional Overview .....	2331
32.2.2	Block Diagram .....	2331
32.2.3	Operational Circuit.....	2332
32.3	Registers.....	2333
32.3.1	List of Registers.....	2333
32.3.2	DCRAnCIN — CRC Input Register .....	2334
32.3.3	DCRAnCOUT — CRC Data Register.....	2335
32.3.4	DCRAnCTL — CRC Control Register .....	2336
32.4	Operation .....	2337
<b>Section 33 Intelligent Cryptographic Unit (ICUSB)</b> .....		<b>2338</b>
<b>Section 34 On-Chip Debug Unit (OCD)</b> .....		<b>2339</b>
34.1	Overview of RH850/F1L OCD .....	2339

34.1.1	Functional Overview .....	2339
34.2	Peripheral Break Control .....	2341
34.3	Registers.....	2343
34.3.1	EPC — Emulation Peripheral Control Register .....	2343
34.4	Cautions on Using On-Chip Debugging.....	2344
34.4.1	Treatment of Devices Used for Debugging .....	2344
34.4.2	Reset Assertion When a Debugger is connected.....	2344
34.4.3	Transition to DeepSTOP Mode When a Debugger is connected.....	2344
<b>Section 35</b>	<b>Flash Memory .....</b>	<b>2345</b>
35.1	Features.....	2345
35.2	Structure of Memory .....	2346
35.2.1	Mapping of Code Flash Memory .....	2346
35.2.2	Mapping of Data Flash Memory .....	2347
35.3	Operating Modes Associated with Flash Memory .....	2348
35.4	Functions .....	2349
35.4.1	Functional Overview .....	2349
35.5	Serial Programming .....	2354
35.5.1	Environments for Programming.....	2354
35.6	Communication Modes .....	2355
35.6.1	Asynchronous Flash Programming Interface - 1-Wire UART .....	2355
35.6.2	Asynchronous Flash Programming Interface - 2-Wire UART .....	2355
35.6.3	Synchronous Flash Programming Interface CSI .....	2355
35.6.4	Selection of Communication Method.....	2356
35.7	Self-Programming.....	2357
35.7.1	Outline .....	2357
35.7.2	Background Operation.....	2358
35.7.3	Enabling Self-Programming.....	2358
35.7.3.1	FLMDCNT Register.....	2358
35.8	Reading Flash Memory.....	2359
35.8.1	Reading Code Flash Memory.....	2359
35.8.2	Reading Data Flash Memory .....	2359
35.8.2.1	EEPRDCYCL — Data Flash Wait Cycle Control Register .....	2360
35.8.2.2	PRDNAME <sub>n</sub> — Product Name Storage Register (n = 1 to 3) .....	2361
35.9	ECC Error Detection and Correction for Code Flash Memory.....	2364
35.9.1	Outline of ECC Functions for Code Flash Memory .....	2364
35.9.2	Interrupt Request.....	2364
35.9.3	Registers .....	2365
35.9.3.1	List of Registers.....	2365
35.9.3.2	CFECCCTL — Code Flash ECC Control Register.....	2366
35.9.3.3	CFFSTERSTR — Code Flash First Error Status Register .....	2367
35.9.3.4	CFFSTSTC — Code Flash Error Status Clear Register .....	2368
35.9.3.5	CFOVFSTR — Code Flash Error Overflow Status Register .....	2368
35.9.3.6	CFOVFSTC — Code Flash Error Overflow Status Clear Register.....	2369

35.9.3.7	CFERRINT — Code Flash Error Notification Control Register .....	2369
35.9.3.8	CFFSTEADR — Code Flash First Error Address Register .....	2370
35.9.3.9	CFTSTCTL — Code Flash Test Control Register .....	2371
35.10	ECC Error Detection and Correction for Data Flash Memory .....	2372
35.10.1	Outline of ECC Functions for Data Flash Memory .....	2372
35.10.2	Interrupt Request.....	2372
35.10.3	Registers .....	2373
35.10.3.1	List of Registers.....	2373
35.10.3.2	DFECCCTL — Data Flash ECC Control Register.....	2373
35.10.3.3	DFERSTR — Data Flash Error Status Register.....	2374
35.10.3.4	DFERSTC — Data Flash Error Status Clear Register .....	2375
35.10.3.5	DFERRINT — Data Flash Error Notification Control Register .....	2375
35.10.3.6	DFTSTCTL — Data Flash Test Control Register .....	2376
35.11	Option Bytes .....	2377
35.11.1	Option Byte Setting.....	2377
35.11.2	OPBT0 — Option Byte 0 .....	2378
35.12	Usage Notes .....	2380
<b>Section 36</b>	<b>RAM .....</b>	<b>2381</b>
36.1	Features.....	2381
36.2	Memory Configuration .....	2382
36.3	ECC Error Detection and Correction for Local RAM.....	2383
36.3.1	Outline of ECC Functions for Local RAM .....	2383
36.3.2	Interrupt Request.....	2384
36.4	Registers.....	2384
36.4.1	List of Registers.....	2384
36.4.2	LRECCCTL — Local RAM ECC Control Register.....	2385
36.4.3	LRFSTERSTR — Local RAM First Error Status Register .....	2386
36.4.4	LRSTCLR — Local RAM Error Status Clear Register.....	2387
36.4.5	LROVFSTR — Local RAM Error Overflow Status Register .....	2388
36.4.6	LROVFSTC — Local RAM Error Overflow Status Clear Register.....	2389
36.4.7	LRFSTEADR0 — Local RAM First Error Address Register 0 .....	2390
36.4.8	LRERRINT — Local RAM Error Notification Control Register.....	2391
36.4.9	LRTSTCTL — Local RAM Test Control Register .....	2392
36.4.10	LRTDATBF0 — Local RAM Test Data Read Buffer 0 .....	2393
<b>Section 37</b>	<b>Boundary Scan .....</b>	<b>2394</b>
37.1	Overview.....	2394
37.2	Features.....	2394
37.3	External Input/Output Pins.....	2396
37.4	Register Descriptions.....	2397
37.4.1	Instruction Register (SDIR).....	2398
37.4.2	ID Register (SDID).....	2398
37.4.3	Bypass Register (SDBPR).....	2398
37.4.4	Boundary Scan Register (SDBSR).....	2398

37.5	Operation .....	2399
37.5.1	TAP Controller .....	2399
37.5.2	Supported Instructions .....	2400
37.5.2.1	BYPASS .....	2400
37.5.2.2	SAMPLE/PRELOAD .....	2400
37.5.2.3	EXTEST .....	2400
37.5.2.4	IDCODE .....	2400
37.5.3	Pins Subject to Boundary Scan .....	2401
37.6	Usage Notes .....	2402
Section 38 Power Supply and Power Domains .....		2403
38.1	Function .....	2403
38.1.1	Power Supply Pins .....	2404
38.1.2	Block Diagram of Power Domains .....	2405
38.1.3	Power Domains Arrangement .....	2407
Appendix A. Pin List .....		2408

## Section 1 Overview

### 1.1 RH850/F1L Product Features

The features of the RH850/F1L are described below.

The RH850/F1L is a 32-bit single-chip microcontroller with a **G3K core**. The key features of the F1L are low power consumption, high computational processing power, and a wide variety of internal peripheral functions. To reduce supply current in a variety of applications, a wide range of power reducing measures are available. For example, there is a Low Power Sampler (LPS), that can poll signals input to the analog and digital input pins without using the CPU, and DeepSTOP mode, in which the power supply to the most circuits of the microcontroller can be turned off.

#### Applications

The RH850/F1L is ideal for automotive electronics such as BCM (body control module), gateway, HVAC, lighting modules, and many others.

### 1.1.1 RH850/F1L Functions

**Table 1.1 Overview of Product (1/2)**

Product Name		RH850/F1L										
		48 pin	64 pin	80 pin	100 pin	144 pin	176 pin					
Memory		See Table 1.2, Product Lineup.										
External memory interface (MEMC)		Not provided										
Provided (20 bit Address Bus)												
CPU	CPU System	G3K										
	CPU frequency	80 MHz max. (ECO)*2	80 MHz max. (ECO/Gateway-512 KB) <sup>2</sup> 96 MHz max. (ADVANCED/PREMIUM/Gateway-1 MB)									
	System Register Protection Unit (SRP)	Provided										
	Memory Protection Unit (MPU)	Provided										
DMA		16 channels										
Operating clock	Main Oscillator (MainOSC)	8 to 24 MHz										
	Low Speed Internal Oscillator (LS IntOSC)	240 kHz (typ.)										
	High Speed Internal Oscillator (HS IntOSC)	8 MHz (typ.)										
	PLL	ECO/Gateway-512 KB: 80 MHz max., ADVANCED/PREMIUM/Gateway-1 MB: 96 MHz max.										
	Sub Oscillator (SubOSC)	Not provided				32.768 kHz						
I/O port		33	49	65	81	120	150					
A/D converter	ADC0	Physical input channels	Total 12 ch (12 bit resolution: 8 ch + 10 bit resolution: 4 ch)	Total 21 ch (12 bit resolution: 10 ch + 10 bit resolution: 11 ch)	Total 25 ch (12 bit resolution: 11 ch + 10 bit resolution: 14 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)					
		External multiplexer support for channel number extension	Provided									
		Channels with T&H	3		6							
	ADC1	Physical input channels	Not provided				Total 12 ch (12 bit resolution: 8 ch + 10 bit resolution: 4 ch)					
		External multiplexer support for channel number extension	Not provided									
		Channels with T&H	Not provided									
Timer		Timer Array Unit D (TAUD)	1 unit (16 bit resolution timers × 16 channels /unit)									
		Timer Array Unit B (TAUB)	Not provided			1 unit (16 bit resolution timers × 16 channels /unit)	2 units (16 bit resolution timers × 16 channels /unit)					
		Timer Array Unit J (TAUJ)	1 unit (32 bit resolution timers × 4 channels /unit)			2 units (32 bit resolution timers × 4 channels /unit)						
		Operating System Timer (OSTM)	1 unit									
		Real-Time Counter (RTCA)	Not provided				1 unit					
		Encoder Timer (ENCA)	1 unit									
		Window Watchdog Timer A (WDTA)	2 units									

Table 1.1 Overview of Product (2/2)

Product Name		RH850/F1L											
		48 pin	64 pin	80 pin	100 pin	144 pin	176 pin						
Serial interfaces	Clocked Serial Interface G (CSIG)	1 channel				2 channels							
	Clocked Serial Interface H (CSIH)	1 channel		3 channels	4 channels								
	CAN Interface (RS-CAN)	1 channel (Total 80 message buffers)	3 channels (Total 240 message buffers)	3 channels (Total 240 message buffers)	3 channels (Total 240 message buffers) for except F1L Gateway  6 channels (Total 480 message buffers) for F1L Gateway	4 channels (Total 320 message buffers) for Code Flash size 1 MB and less 6 channels (Total 480 message buffers) for Code Flash size 1.5 MB and more	6 channels (Total 480 message buffers)						
	LIN/UART Interface (RLIN3)	1 channel	2 channels	3 channels	4 channels	6 channels							
External Interrupts	LIN Master Interface (RLIN2)	2 channels			3 channels	6 channels	10 channels						
	I <sup>2</sup> C Interface (RIIC)	1 channel											
Other functions	Maskable	8		12	13	16							
	Non-maskable (NMI)	1											
	Clock Monitors (CLMA)	For PLL, HS IntOSC, MainOSC											
	Data CRC (DCRA)	1 channel		4 channels									
	Low-Voltage Indicator (LVI)	Provided											
	Power-On-Clear (POC)	Provided											
	Core Voltage Monitors (CVM)	Provided											
	Error Correction Coding (ECC)	For Code Flash, Data Flash, Primary Local RAM , Secondary Local RAM, Retention RAM, CSIH, RS-CAN											
	Low Power Sampling (LPS)	Provided											
	PWM diagnosis (PWM-Diag)	13 channels	24 channels		48 channels	64 channels	72 channels						
	Motor Control	1 unit											
	Key Return (KR)	6 channels	8 channels										
	CLOCK OUTPUT (FOUT)	Provided											
	RESET OUTPUT (RESETOUT)	Provided											
Voltage supply	ICUSB	Not provided	Not provided (ECO/Gateway-512 KB)* <sup>2</sup> Provided (ADVANCED/PREMIUM/Gateway-1 MB)										
	On-Chip debug (OCD)	Provided											
	Boundary Scan	Provided											
	Internal supply	V <sub>POC</sub> * <sup>1</sup> to 5.5 V											
A/D Converter supplies	Input/output buffer supplies	V <sub>POC</sub> * <sup>1</sup> to 5.5 V											
	A/D Converter supplies	3.0 to 5.5 V											
Package		48 pin LQFP	64 pin LQFP	80 pin LQFP	100 pin LQFP	144 pin LQFP	176pin LQFP						

Note 1. V<sub>POC</sub> is the voltage level of power-on-clear circuit.

Refer to the data sheet for detailed specifications of electrical values.

Note 2. ECO (products in the eco range) is a standard product operating at 80 MHz.

ADVANCED (products in the advanced range) is a product having an ICUSB<sup>(\*)</sup> and whose CPU performance is enhanced to operate at 96 MHz.

PREMIUM (Products in the premium range) is a product having an ICUSB<sup>(\*)</sup> and 64-Kbyte data flash memory and whose CPU performance is enhanced to operate at 96 MHz.

Gateway-512KB (512-Kbyte Gateway products) is a 100 pin product having six CAN channels.

Gateway-1MB (1-Mbyte Gateway products) is a 100 pin product having six CAN channels and an ICUSB<sup>(\*)</sup>, and whose CPU performance is enhanced to operate at 96 MHz.

"F1L for Gateway" is the general term for both Gateway-512KB and Gateway-1MB.

\*ICUSB: Intelligent Cryptographic Unit Slave B

Table 1.2 Product Lineup (1/2)

Pin Count	CPU frequency	Memory				CAN Interface (RS-CAN)	Operating Temperature (Ta)	Part Name
		Code Flash	Primary Local RAM	Secondary Local RAM	Data Flash			
48 pins	80 MHz max.	256 kB	—	Not provided	32 kB	32 kB	1 channel (Total 80 message buffers)	R7F7010083Afp R7F7010084Afp ECO
		384 kB	16 kB				R7F7010093Afp R7F7010094Afp ECO	
64 pins	80 MHz max.	512 kB	32 kB				R7F7010103Afp R7F7010104Afp ECO	
		256 kB	—	Not provided	32 kB	32 kB	3 channels (Total 240 message buffers)	R7F7010113Afp R7F7010114Afp ECO
80 pins	80 MHz max.	384 kB	16 kB				R7F7010123Afp R7F7010124Afp ECO	
		512 kB	32 kB				R7F7010133Afp R7F7010134Afp ECO	
100 pins	80 MHz max.	768 kB	64 kB				R7F7010143Afp R7F7010144Afp ECO	
		1024 kB	96 kB				R7F7010153Afp R7F7010154Afp ECO	
144 pins	80 MHz max.	256 kB	—	Not provided	32 kB	32 kB	3 channels (Total 240 message buffers)	R7F7010163Afp R7F7010164Afp ECO
		384 kB	16 kB				R7F7010173Afp R7F7010174Afp ECO	
176 pins	80 MHz max.	512 kB	32 kB				R7F7010183Afp R7F7010184Afp ECO	
		768 kB	64 kB				R7F7010193Afp R7F7010194Afp ECO	
96 MHz max.	80 MHz max.	1024 kB	96 kB				R7F7010203Afp R7F7010204Afp ECO	
		256 kB	—	Not provided	32 kB	32 kB	3 channels (Total 240 message buffers)	R7F7010213Afp R7F7010214Afp ECO
128 pins	96 MHz max.	384 kB	16 kB				R7F7010223Afp R7F7010224Afp ECO	
		512 kB	32 kB				R7F7010233Afp R7F7010234Afp ECO	
144 pins	96 MHz max.	768 kB	64 kB				R7F7010243Afp R7F7010244Afp ECO	
		1024 kB	96 kB				R7F7010253Afp R7F7010254Afp ECO	
176 pins	96 MHz max.	512 kB	32 kB				R7F701023Afp R7F701024Afp Gateway-512KB	
		1024 kB	96 kB				R7F7010033Afp R7F7010034Afp Gateway-1MB	
2048 pins	80 MHz max.	768 kB	64 kB	Not provided	32 kB	32 kB	4 channels (Total 320 message buffers)	R7F7010283Afp R7F7010284Afp ECO
		1024 kB	96 kB				R7F7010293Afp R7F7010294Afp ECO	
2208 pins	80 MHz max.	128 kB					R7F7010303Afp R7F7010304Afp ECO	
		2048 kB	128 kB	32 kB			R7F70101063Afp R7F7010064Afp ECO	
2368 pins	80 MHz max.	768 kB	64 kB	Not provided	32 kB	32 kB	6 channels (Total 480 message buffers)	R7F7010323Afp R7F7010324Afp ECO
		1024 kB	96 kB				R7F7010333Afp R7F7010334Afp ECO	
2528 pins	80 MHz max.	128 kB					R7F7010343Afp R7F7010344Afp ECO	
		2048 kB	128 kB	32 kB			R7F7010073Afp R7F7010074Afp ECO	

Table 1.2 Product Lineup (2/2)

Pin Count	CPU frequency	Memory				CAN Interface (RS-CAN)	Operating Temperature (Ta) -40°C to +105°C -40°C to +125°C	Part Name
		Code Flash	Primary Local RAM	Secondary Local RAM	Data Flash			
64 pins	96 MHz max.	768 KB 1024 KB	64 KB 96 KB	Not Provided	32 KB	32 KB (Total 240 message buffers)	R7F7010403Afp R7F7010413Afp	R7F7010404Afp R7F7010414Afp
80 pins	96 MHz max.	768 KB 1024 KB	64 KB 96 KB	Not Provided	32 KB	32 KB (Total 240 message buffers)	R7F7010423Afp R7F7010433Afp	R7F7010424Afp R7F7010434Afp
100 pins	96 MHz max.	768 KB 1024 KB	64 KB 96 KB	Not Provided	32 KB	32 KB (Total 240 message buffers)	R7F7010443Afp R7F7010453Afp	R7F7010444Afp R7F7010454Afp
144 pins	96 MHz max.	768 KB 1024 KB 1536 KB 2048 KB	64 KB 96 KB 128 KB 128 KB	Not Provided	32 KB	32 KB (Total 320 message buffers)	R7F7010463Afp R7F7010473Afp	R7F7010464Afp R7F7010474Afp
176 pins	96 MHz max.	768 KB 1024 KB 1536 KB 2048 KB	64 KB 96 KB 128 KB 128 KB	Not Provided	32 KB	32 KB (Total 480 message buffers)	R7F7010483Afp R7F7010493Afp	R7F7010484Afp R7F7010494Afp
144pins	96 MHz max.	1536 KB 2048 KB	128 KB 128 KB	Not Provided	64 KB	32 KB (Total 480 message buffers)	R7F7010503Afp R7F7010513Afp	R7F7010504Afp R7F7010514Afp
176pins	96 MHz max.	1536 KB 2048 KB	128 KB 128 KB	Not Provided	64 KB	32 KB (Total 480 message buffers)	R7F7010523Afp R7F7010533Afp	R7F7010524Afp R7F7010534Afp
							R7F7010543Afp R7F7010553Afp R7F7010563Afp R7F7010573Afp	R7F7010544Afp R7F7010554Afp R7F7010564Afp R7F7010574Afp

### 1.1.2 Internal Block Diagram

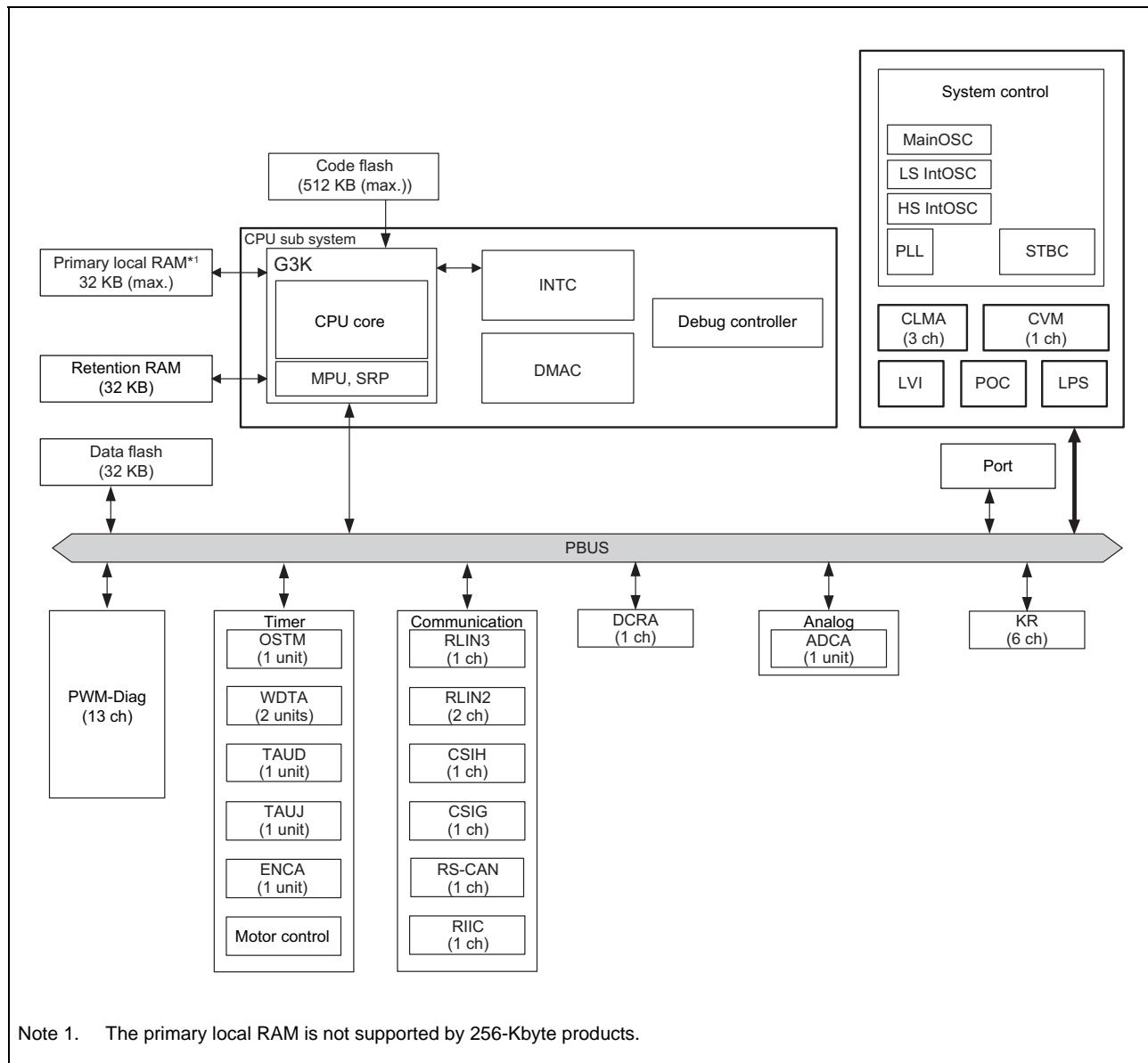


Figure 1.1 Internal Block Diagram (RH850/F1L 48 pin)

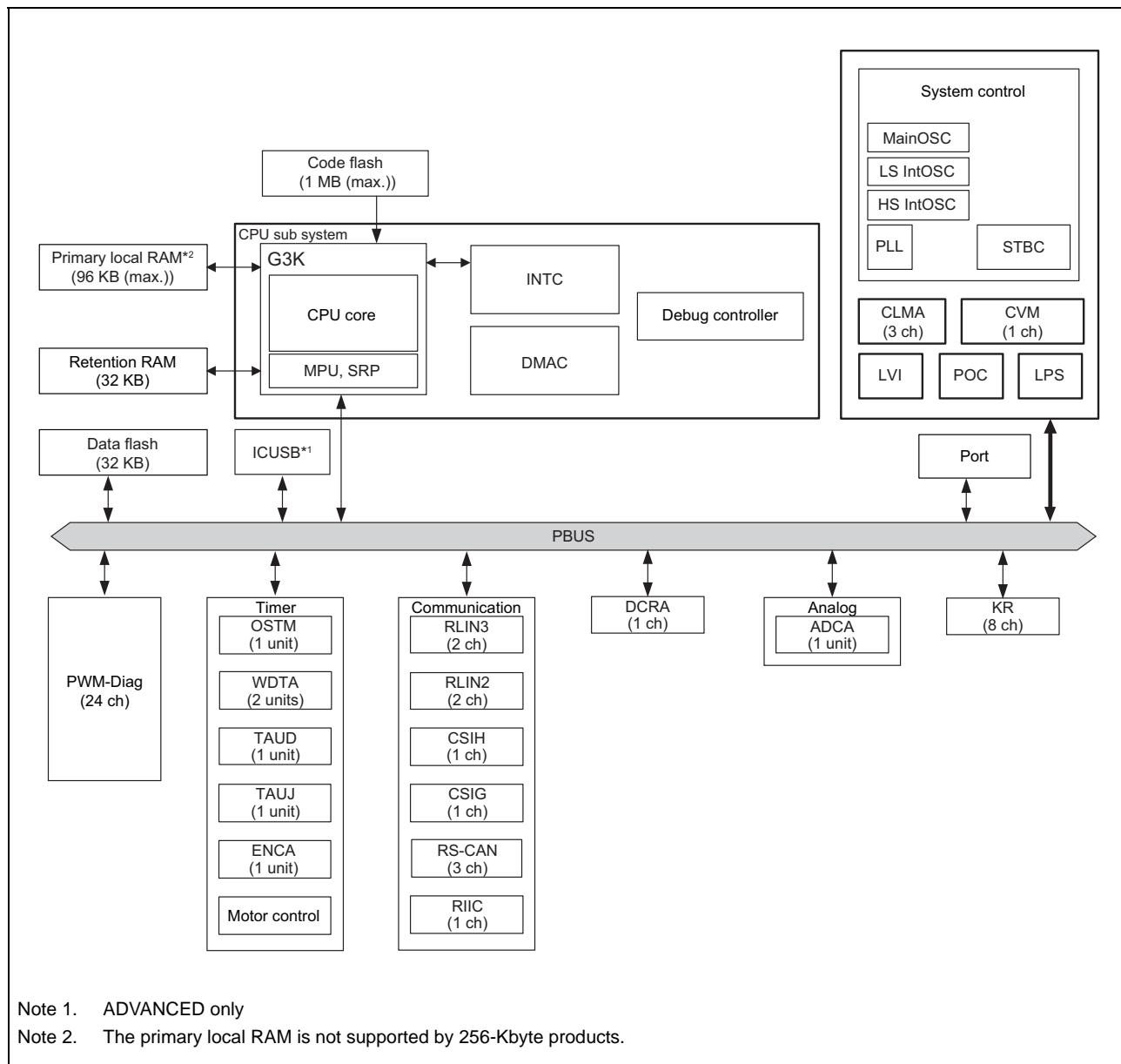


Figure 1.2 Internal Block Diagram (RH850/F1L 64 pin)

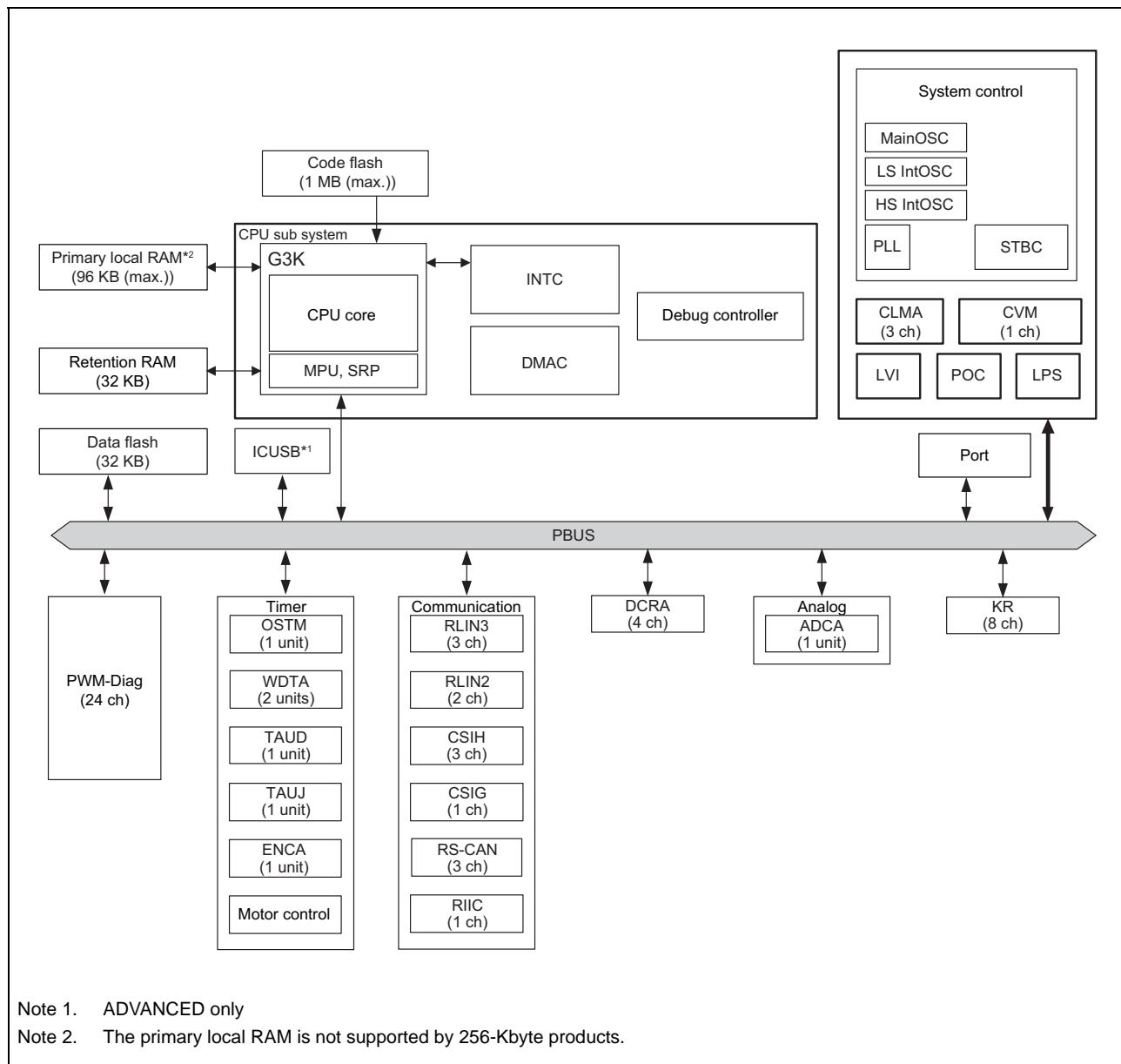


Figure 1.3 Internal Block Diagram (RH850/F1L 80 pin)

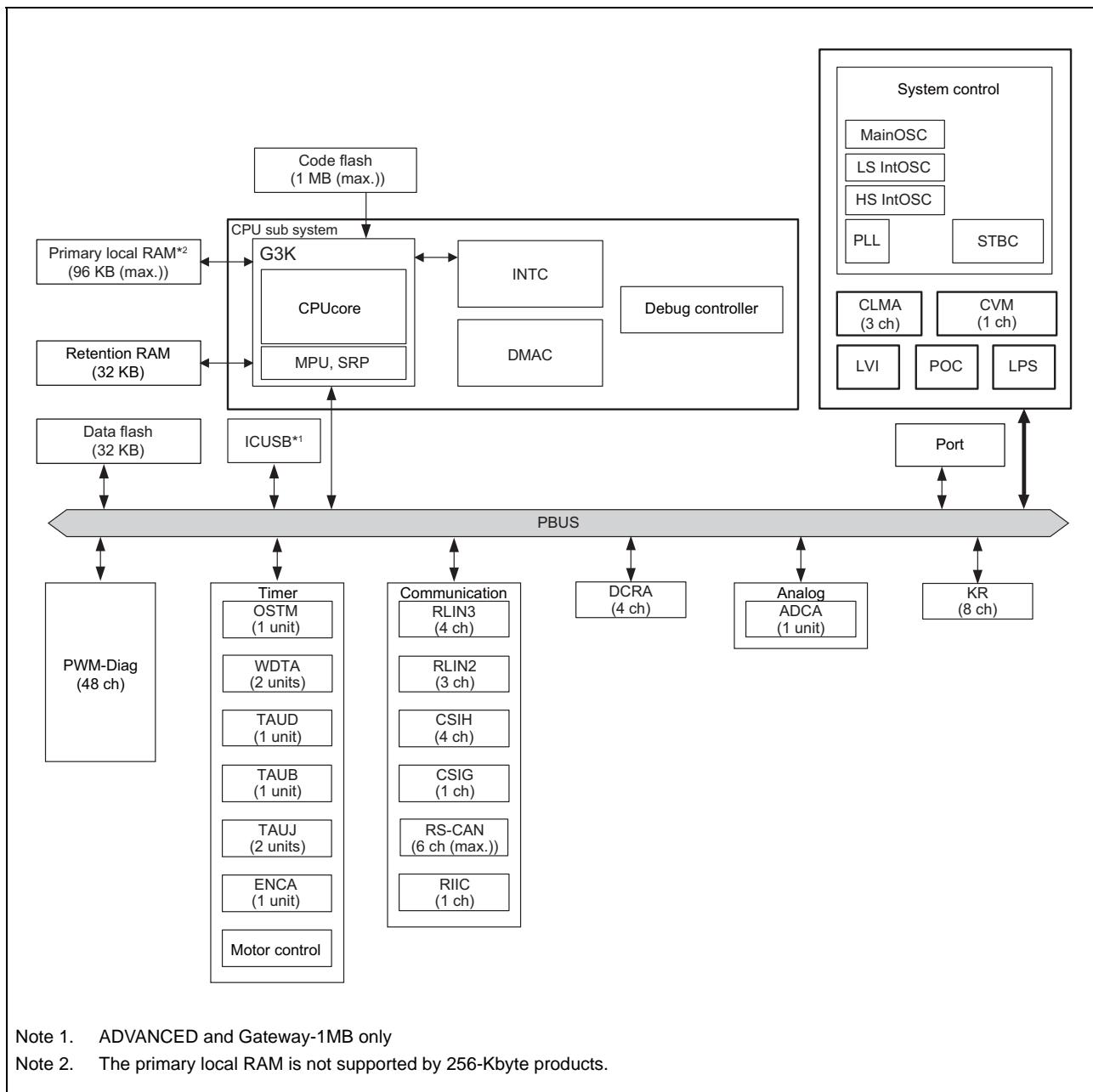


Figure 1.4 Internal Block Diagram (RH850/F1L 100 pin)

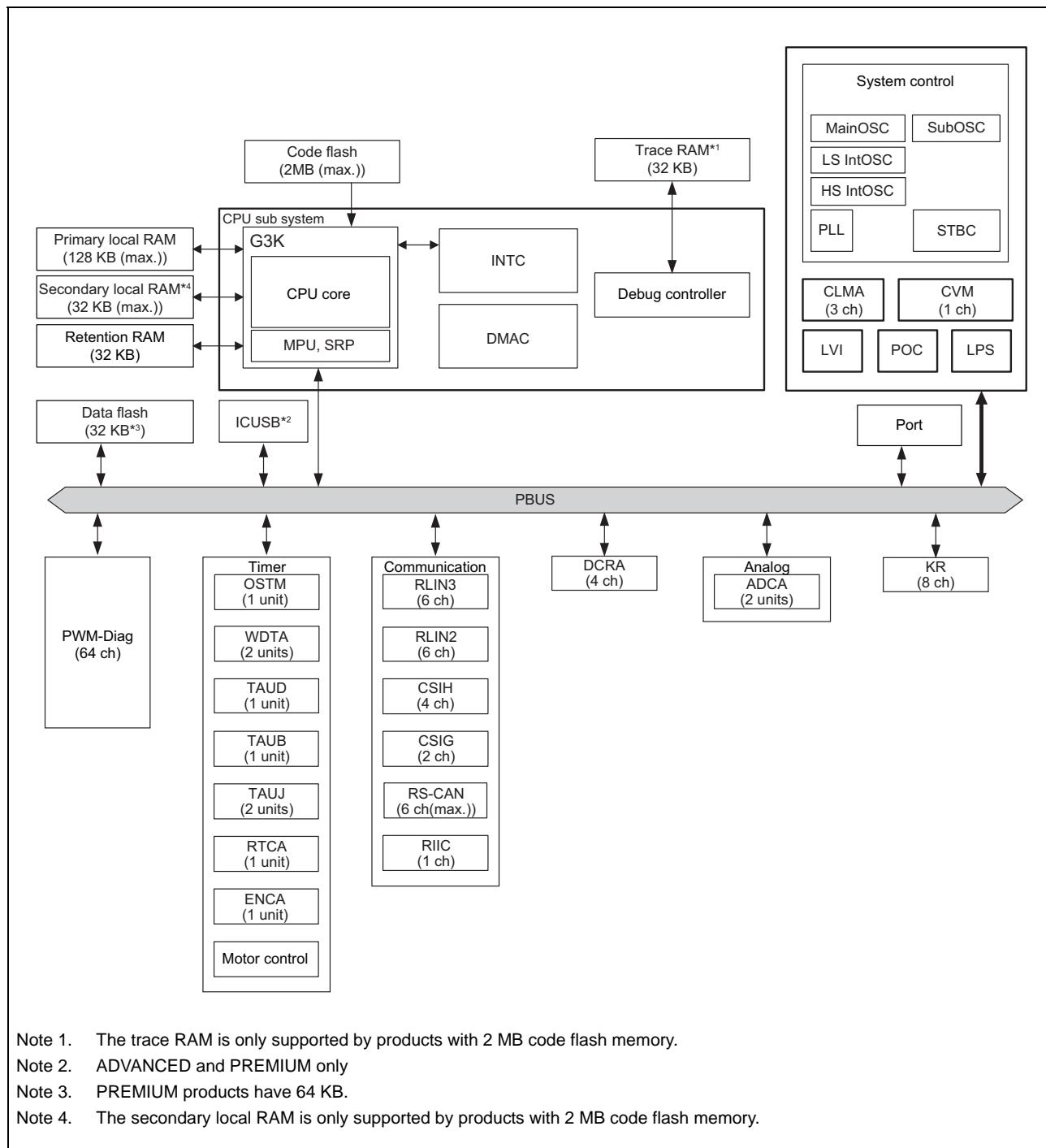


Figure 1.5 Internal Block Diagram (RH850/F1L 144 pin)

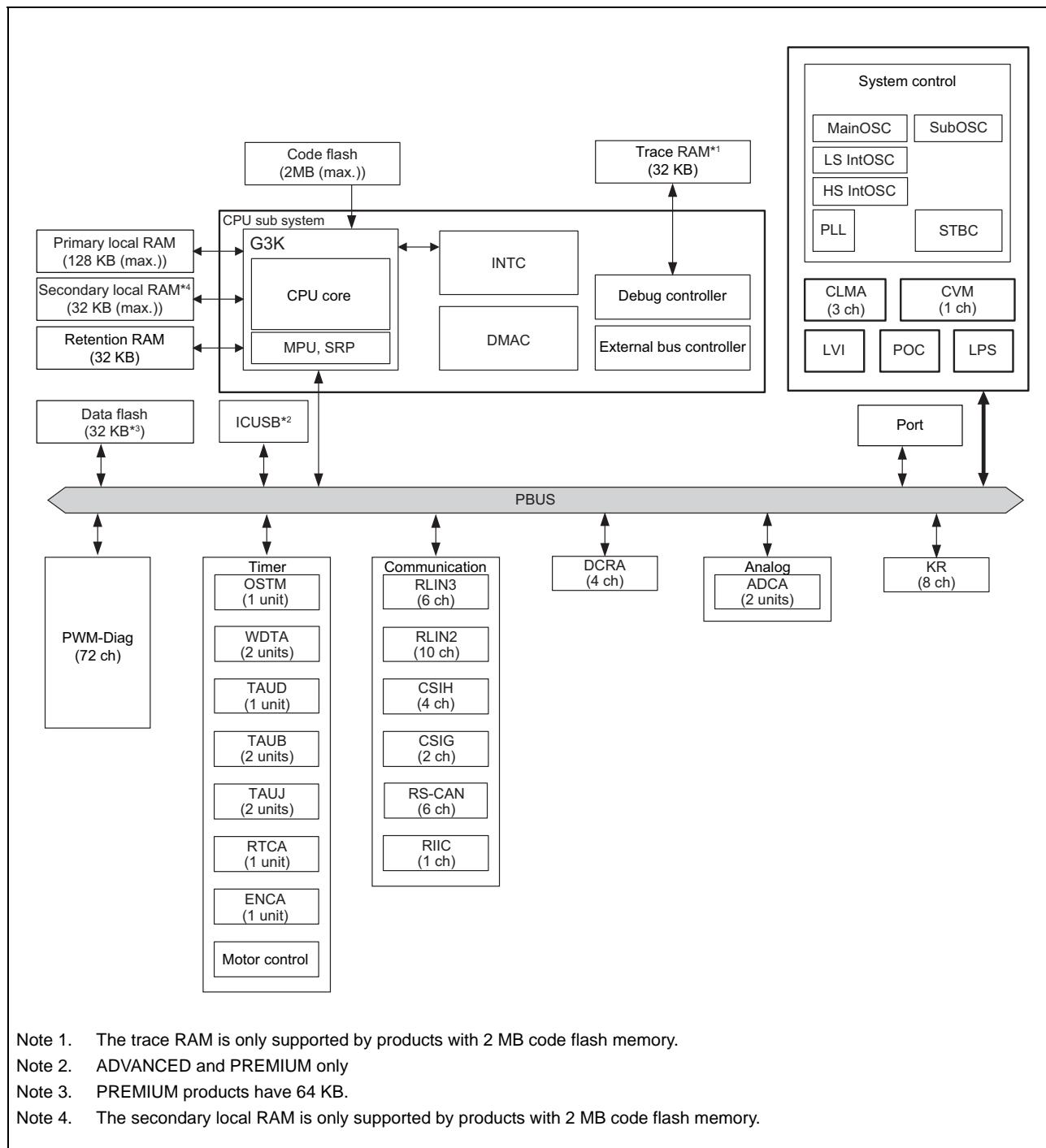


Figure 1.6 Internal Block Diagram (RH850/F1L 176 pin)

## Section 2 Pin Function

This section describes the pin and port functions.

**Section 2.1** to **Section 2.5** describe the pin connections and respective pins.

**Section 2.6** to **Section 2.13** describe the general port functions.

### 2.1 Pin Connection Diagram

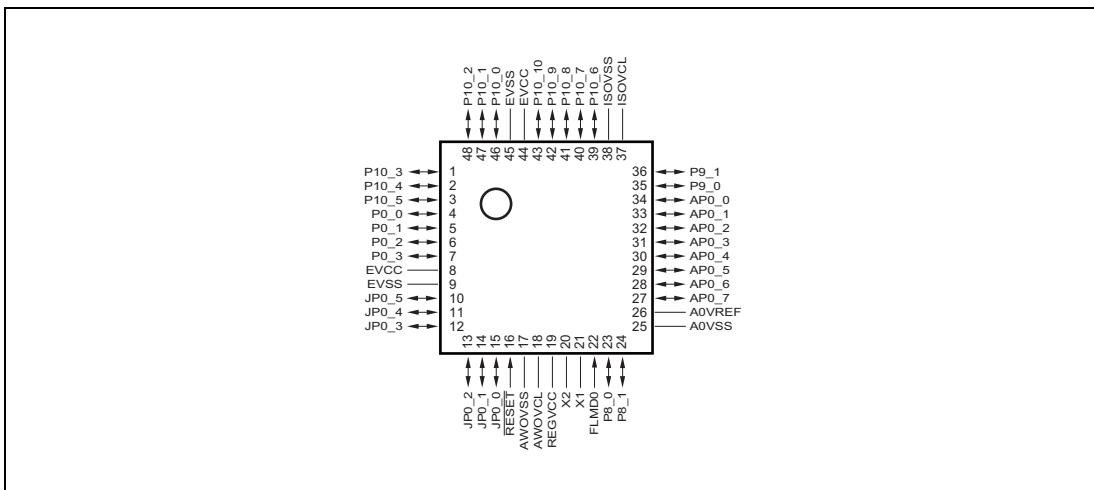


Figure 2.1 Pin Connection Diagram (48 pin LQFP)

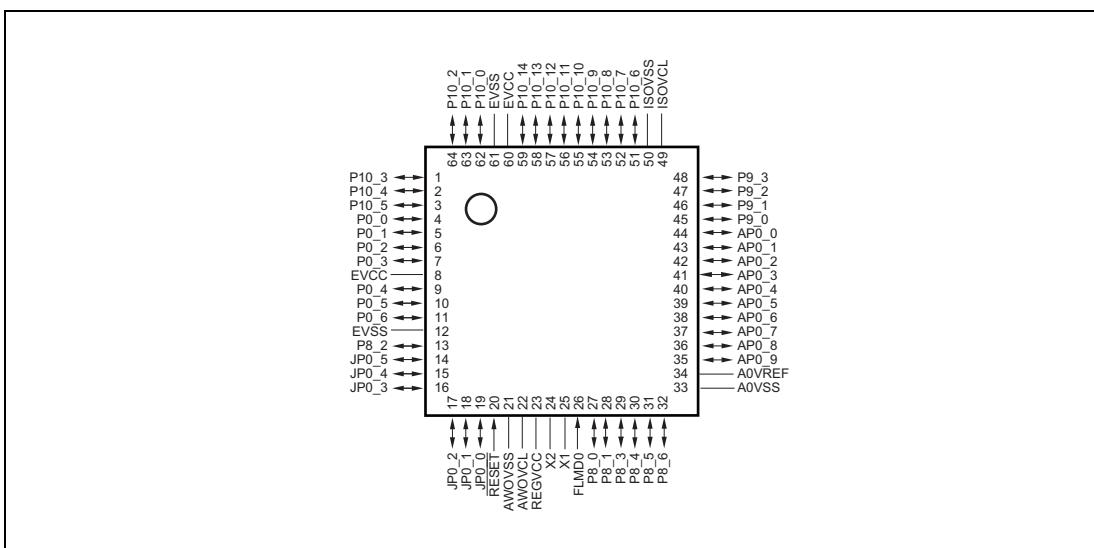


Figure 2.2 Pin Connection Diagram (64 pin LQFP)

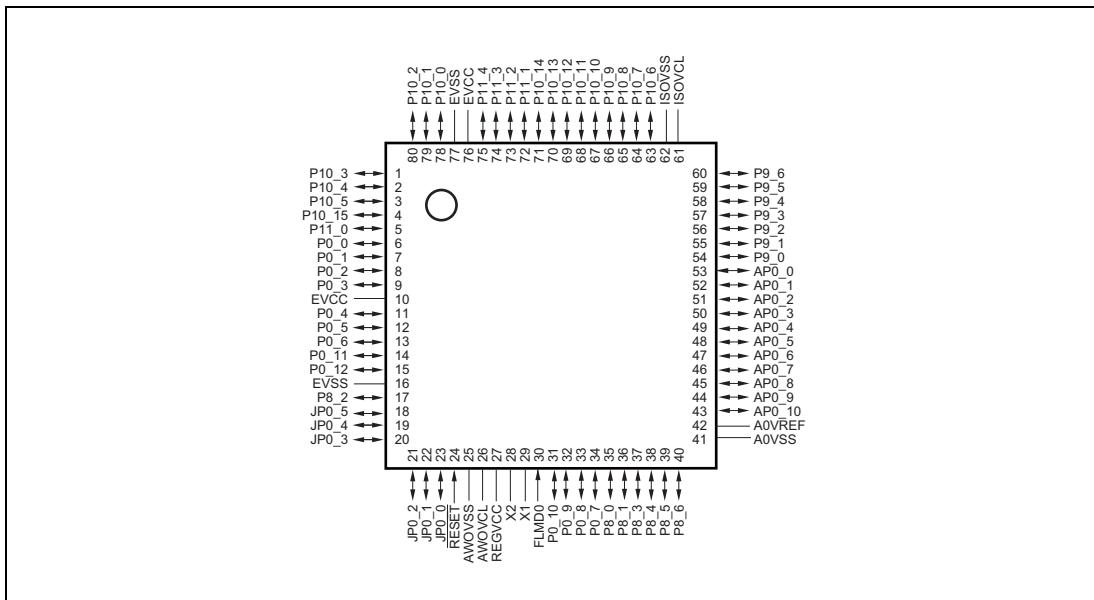


Figure 2.3 Pin Connection Diagram (80 pin LQFP)

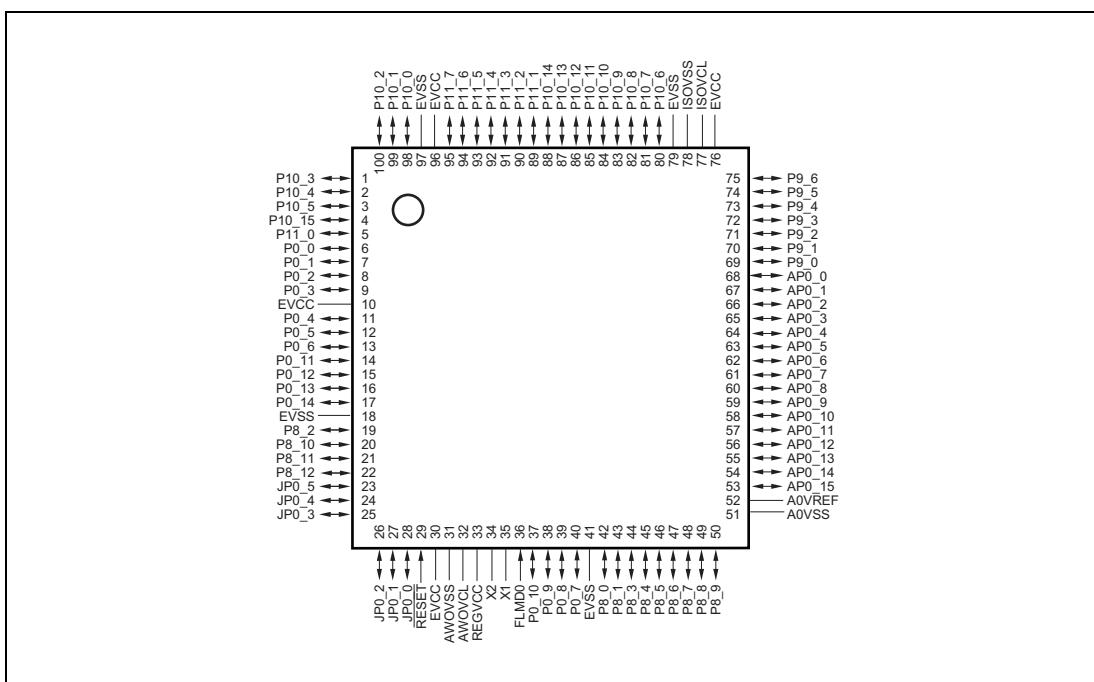
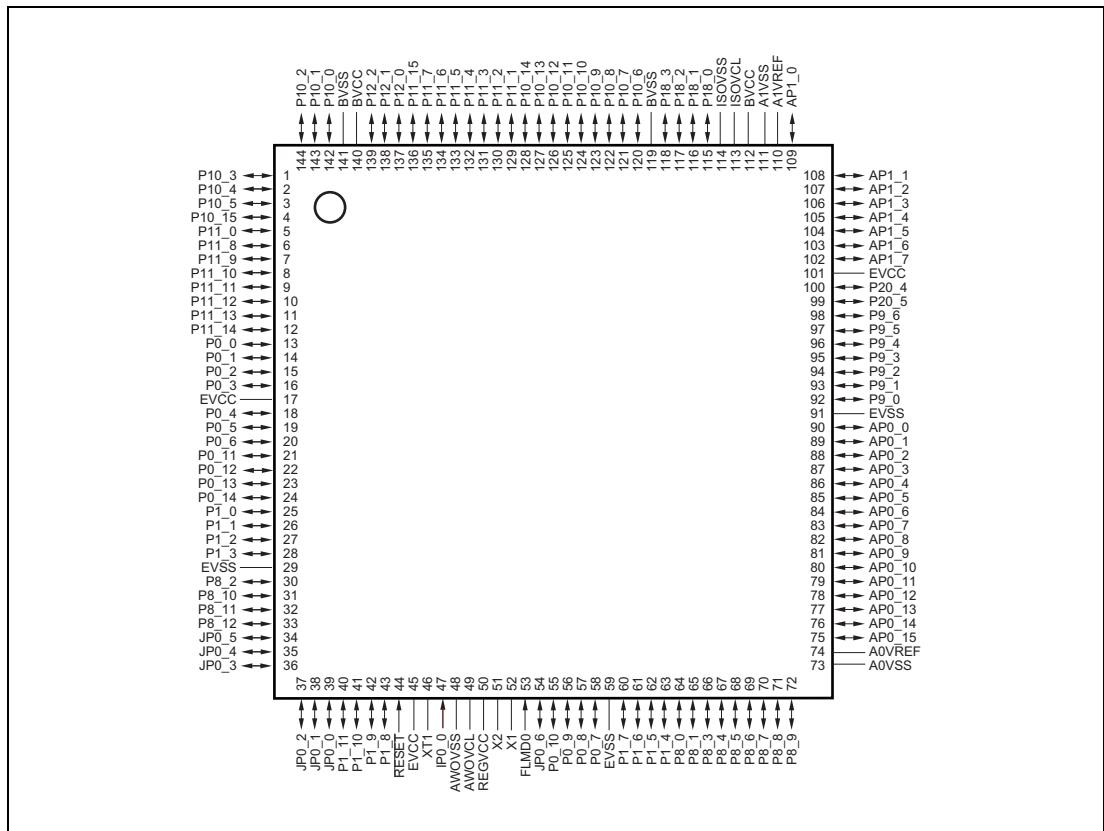


Figure 2.4 Pin Connection Diagram (100 pin LQFP)



## Figure 2.5 Pin Connection Diagram (144 pin LQFP)

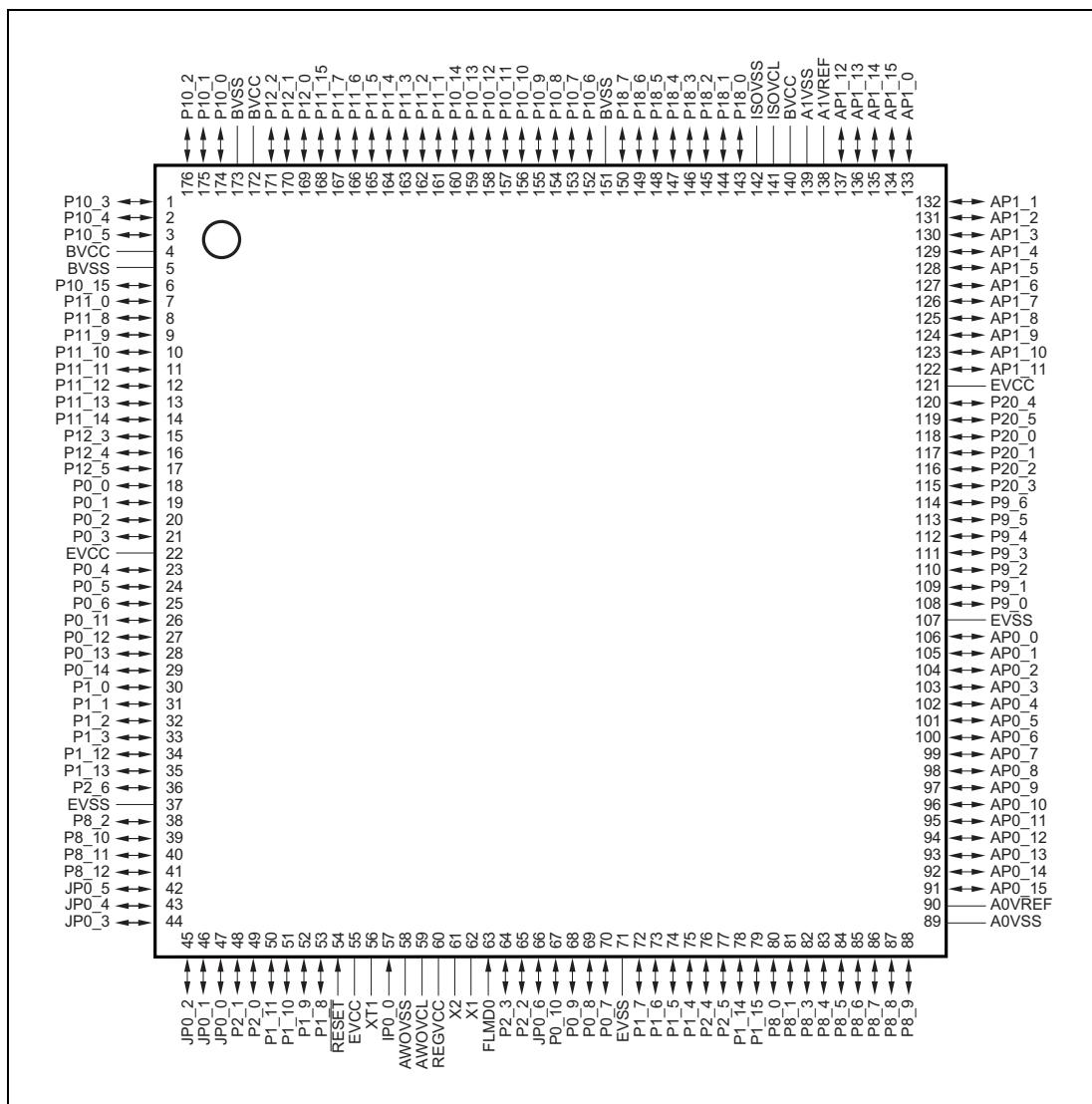


Figure 2.6 Pin Connection Diagram (176 pin LQFP)

**Table 2.1 Pin Assignment 48 pin LQFP (1/2)**

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / CSIG0SSI
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / CSIH0SSI / DPO / RESETOUT
5	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
6	P0_2 / TAUD0I6 / TAUD0O6 / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
7	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / DPIN1 / CSIH0SO / INTP10
8	EVCC
9	EVSS
10	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / DCURDY / LPDCLKOUT
11	JP0_4 / DCUTRST
12	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
13	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
14	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
15	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
16	RESET
17	AWOVSS
18	AWOVCL
19	REGVCC
20	X2
21	X1
22	FLMD0
23	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / INTP4 / CSIH0CSS0 / ADCA0I0S
24	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / INTP5 / ADCA0I1S
25	A0VSS
26	A0VREF
27	AP0_7 / ADCA0I7
28	AP0_6 / ADCA0I6
29	AP0_5 / ADCA0I5
30	AP0_4 / ADCA0I4
31	AP0_3 / ADCA0I3
32	AP0_2 / ADCA0I2
33	AP0_1 / ADCA0I1
34	AP0_0 / ADCA0I0
35	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / KR0I4 / ADCA0I2S
36	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / ADCA0I3S
37	ISOVCL
38	ISOVSS
39	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2
40	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O
41	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1

**Table 2.1 Pin Assignment 48 pin LQFP (2/2)**

Pin No.	Pin Name
42	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX /INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
43	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
44	EVCC
45	EVSS
46	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP
47	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / MODE0
48	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / MODE1

**Table 2.2 Pin Assignment 64 pin LQFP (1/2)**

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / CSIG0SSI
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / CSIH0SSI / DPO / RESETOUT
5	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
6	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
7	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
8	EVCC
9	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / SELDP0 / DPIN8
10	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1
11	P0_6 / INTP2 / DPIN10 / SELDP2
12	EVSS
13	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / PWGA22O / ADCA0I4S
14	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / DCURDY / LPDCLKOUT
15	JP0_4 / DCUTRST
16	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
17	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
18	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
19	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
20	RESET
21	A0VSS
22	A0VCL
23	REGVCC
24	X2
25	X1
26	FLMD0
27	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
28	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / ADCA0I1S
29	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / PWGA23O / ADCA0I5S
30	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / ADCA0I6S
31	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / ADCA0I7S
32	P8_6 / NMI / ADCA0I8S
33	A0VSS
34	A0VREF
35	AP0_9 / ADCA0I9
36	AP0_8 / ADCA0I8
37	AP0_7 / ADCA0I7
38	AP0_6 / ADCA0I6
39	AP0_5 / ADCA0I5
40	AP0_4 / ADCA0I4
41	AP0_3 / ADCA0I3

**Table 2.2 Pin Assignment 64 pin LQFP (2/2)**

<b>Pin No.</b>	<b>Pin Name</b>
42	AP0_2 / ADCA0I2
43	AP0_1 / ADCA0I1
44	AP0_0 / ADCA0I0
45	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / KR0I4 / ADCA0I2S
46	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / ADCA0I3S
47	P9_2 / KR0I6 / PWGA20O / TAPA0ES0 / ADCA0I9S
48	P9_3 / KR0I7 / PWGA21O / ADCA0I10S
49	ISOVCL
50	ISOVSS
51	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1
52	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
53	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1
54	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RY1 / CSIH0RY0
55	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
56	P10_11 / PWGA16O / RLIN31RX / INTP11
57	P10_12 / PWGA17O / RLIN31TX
58	P10_13 / CSIH0SSI / PWGA18O
59	P10_14 / PWGA19O
60	EVCC
61	EVSS
62	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP
63	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / MODE0
64	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / MODE1

**Table 2.3 Pin Assignment 80 pin LQFP (1/2)**

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / CSIH1SSI
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / CSIG0SSI
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P10_15
5	P11_0 / CSIH2RYI / CSIH2RYO
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / CSIH0SSI / DPO / RESETOUT
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
14	P0_11 / RIIC0SDA / CSIH1CSS2
15	P0_12 / RIIC0SCL
16	EVSS
17	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
18	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / DCURDY / LPDCLKOUT
19	JP0_4 / DCUTRST
20	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
21	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
22	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
23	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
24	RESET
25	AWVSS
26	AWVCL
27	REGVCC
28	X2
29	X1
30	FLMD0
31	P0_10 / INTP3 / CSIH1CSS1 / DPIN11
32	P0_9 / INTP12 / CSIH1CSS0 / DPIN7
33	P0_8 / RLIN21TX / DPIN6 / CSIH1SSI
34	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO
35	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
36	P8_1 / TAPA0ES0 / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
37	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
38	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / ADCA0I6S
39	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / ADCA0I7S
40	P8_6 / NMI / CSIH0CSS4 / ADCA0I8S
41	A0VSS

**Table 2.3 Pin Assignment 80 pin LQFP (2/2)**

Pin No.	Pin Name
42	A0VREF
43	AP0_10 / ADCA0I10
44	AP0_9 / ADCA0I9
45	AP0_8 / ADCA0I8
46	AP0_7 / ADCA0I7
47	AP0_6 / ADCA0I6
48	AP0_5 / ADCA0I5
49	AP0_4 / ADCA0I4
50	AP0_3 / ADCA0I3
51	AP0_2 / ADCA0I2
52	AP0_1 / ADCA0I1
53	AP0_0 / ADCA0I0
54	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
55	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
56	P9_2 / KR0I6 / PWGA20O / TAPA0ES0 / CSIH2CSS2 / ADCA0I9S
57	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / ADCA0I10S
58	P9_4 / CSIH0CSS5 / ADCA0I11S
59	P9_5 / CSIH0CSS6 / ADCA0I12S
60	P9_6 / CSIH0CSS7 / ADCA0I13S
61	ISOVCL
62	ISOVSS
63	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1
64	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
65	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1
66	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RY1 / CSIH0RY0
67	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
68	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0
69	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1
70	P10_13 / CSIH0SSI / PWGA18O / RLIN32RX / INTP12
71	P10_14 / PWGA19O / RLIN32TX
72	P11_1 / CSIH2SSI / RLIN20RX
73	P11_2 / CSIH2SO / RLIN20TX
74	P11_3 / CSIH2SC
75	P11_4 / CSIH2SI
76	EVCC
77	EVSS
78	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI
79	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
80	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

**Table 2.4 Pin Assignment 100 pin LQFP (1/3)**

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / CSIH1SSI
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / CSIG0SSI
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RY1 / CSIG0RYO
4	P10_15 / CSIH3RY1 / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9
5	P11_0 / CSIH2RY1 / CSIH2RYO / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / CSIH0SSI / DPO / RESETOUT
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
14	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8
15	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
16	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX <sup>*1</sup> / INTP5
17	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX <sup>*1</sup>
18	EVSS
19	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
20	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
21	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
22	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
23	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / DCURDY / LPDCLKOUT
24	JP0_4 / DCUTRST
25	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
26	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
27	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
28	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
29	RESET
30	EVCC
31	AWOVSS
32	AWOVCL
33	REGVCC
34	X2
35	X1
36	FLMD0
37	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX <sup>*1</sup>
38	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX <sup>*1</sup> / INTP4
39	P0_8 / RLIN21TX / DPIN6 / CSIH1SSI / TAUB0I2 / TAUB0O2 / CAN3TX <sup>*1</sup>
40	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RY1 / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX <sup>*1</sup> / INTP3
41	EVSS

**Table 2.4 Pin Assignment 100 pin LQFP (2/3)**

Pin No.	Pin Name
42	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
43	P8_1 / TAPA0ES0 / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
44	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
45	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
46	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / PWGA37O / ADCA0I7S
47	P8_6 / NMI / CSIH0CSS4 / PWGA38O / ADCA0I8S
48	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
49	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
50	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
51	A0VSS
52	A0VREF
53	AP0_15 / ADCA0I15
54	AP0_14 / ADCA0I14
55	AP0_13 / ADCA0I13
56	AP0_12 / ADCA0I12
57	AP0_11 / ADCA0I11
58	AP0_10 / ADCA0I10
59	AP0_9 / ADCA0I9
60	AP0_8 / ADCA0I8
61	AP0_7 / ADCA0I7
62	AP0_6 / ADCA0I6
63	AP0_5 / ADCA0I5
64	AP0_4 / ADCA0I4
65	AP0_3 / ADCA0I3
66	AP0_2 / ADCA0I2
67	AP0_1 / ADCA0I1
68	AP0_0 / ADCA0I0
69	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
70	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
71	P9_2 / KR0I6 / PWGA20O / TAPA0ES0 / CSIH2CSS2 / ADCA0I9S
72	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / ADCA0I10S
73	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
74	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
75	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
76	EVCC
77	ISOVCL
78	ISOVSS
79	EVSS
80	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1
81	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
82	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1

**Table 2.4 Pin Assignment 100 pin LQFP (3/3)**

Pin No.	Pin Name
83	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RY1 / CSIH0RY0
84	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
85	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1
86	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
87	P10_13 / <u>CSIH0SSI</u> / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5
88	P10_14 / PWGA19O / RLIN32TX / <u>CSIH3SSI</u> / TAUB0I7 / TAUB0O7
89	P11_1 / <u>CSIH2SSI</u> / RLIN20RX / PWGA26O / TAUB0I13 / TAUB0O13
90	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15
91	P11_3 / CSIH2SC / CAN3RX* <sup>1</sup> / INTP3 / PWGA28O
92	P11_4 / CSIH2SI / CAN3TX* <sup>1</sup> / PWGA29O
93	P11_5 / CAN5RX* <sup>1</sup> / INTP5 / RLIN33TX / PWGA30O / CSIH3SI
94	P11_6 / RLIN33RX / INTP13 / CAN5TX* <sup>1</sup> / PWGA31O / CSIH3SO
95	P11_7 / INTP5 / PWGA32O / CSIH3SC
96	EVCC
97	EVSS
98	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI
99	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
100	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Note 1. CANmTX and CANmRX (m = 3 to 5) are provided only in the F1L for Gateway.

**Table 2.5 Pin Assignment 144 pin LQFP (1/4)**

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / CSIH1SSI
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / CSIG0SSI
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9
5	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11
6	P11_8 / CSIG1SSI / RLIN35TX / PWGA48O
7	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O
8	P11_10 / CSIG1SC / PWGA50O
9	P11_11 / CSIG1SI / RLIN25TX / PWGA51O
10	P11_12 / RLIN25RX / PWGA52O
11	P11_13 / RLIN24RX / PWGA53O
12	P11_14 / RLIN24TX / PWGA54O
13	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / CSIH0SSI / DPO / RESETOUT
14	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
15	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
16	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
17	EVCC
18	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8
19	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
20	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
21	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8
22	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
23	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX*2 / INTP5
24	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX*2
25	P1_0 / RLIN33RX / INTP13
26	P1_1 / RLIN33TX
27	P1_2 / CAN3RX / INTP3
28	P1_3 / CAN3TX / DPIN23
29	EVSS
30	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
31	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
32	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
33	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
34	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / DCURDY / LPDCLKOUT
35	JP0_4 / DCUTRST
36	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
37	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
38	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
39	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
40	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22
41	P1_10 / RLIN24RX / DPIN21

**Table 2.5 Pin Assignment 144 pin LQFP (2/4)**

Pin No.	Pin Name
42	P1_9 / RLIN34TX / DPIN20
43	P1_8 / RLIN34RX / INTP14
44	RESET
45	EVCC
46	XT1
47	IP0_0 / XT2
48	AWOVSS
49	AWOVCL
50	REGVCC
51	X2
52	X1
53	FLMD0
54	JP0_6 / EVTO* <sup>1</sup>
55	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX* <sup>2</sup>
56	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX* <sup>2</sup> / INTP4
57	P0_8 / RLIN21TX / DPIN6 / CSIH1SSI / TAUB0I2 / TAUB0O2 / CAN3TX
58	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RY1 / CSIH1RY0 / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
59	EVSS
60	P1_7 / ADCA1TRG1 / RLIN25TX / DPIN19
61	P1_6 / RLIN25RX / DPIN18
62	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17
63	P1_4 / RLIN35RX / INTP15
64	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
65	P8_1 / TAPA0ES0 / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
66	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
67	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
68	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
69	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S
70	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
71	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
72	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
73	A0VSS
74	A0VREF
75	AP0_15 / ADCA0I15
76	AP0_14 / ADCA0I14
77	AP0_13 / ADCA0I13
78	AP0_12 / ADCA0I12
79	AP0_11 / ADCA0I11
80	AP0_10 / ADCA0I10
81	AP0_9 / ADCA0I9
82	AP0_8 / ADCA0I8

**Table 2.5 Pin Assignment 144 pin LQFP (3/4)**

<b>Pin No.</b>	<b>Pin Name</b>
83	AP0_7 / ADCA0I7
84	AP0_6 / ADCA0I6
85	AP0_5 / ADCA0I5
86	AP0_4 / ADCA0I4
87	AP0_3 / ADCA0I3
88	AP0_2 / ADCA0I2
89	AP0_1 / ADCA0I1
90	AP0_0 / ADCA0I0
91	EVSS
92	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
93	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
94	P9_2 / KR0I6 / PWGA20O / TAPA0ES0 / CSIH2CSS2 / ADCA0I9S
95	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / ADCA0I10S
96	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
97	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
98	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
99	P20_5 / RLIN23TX / PWGA60O
100	P20_4 / RLIN23RX / PWGA59O
101	EVCC
102	AP1_7 / ADCA1I7
103	AP1_6 / ADCA1I6
104	AP1_5 / ADCA1I5
105	AP1_4 / ADCA1I4
106	AP1_3 / ADCA1I3
107	AP1_2 / ADCA1I2
108	AP1_1 / ADCA1I1
109	AP1_0 / ADCA1I0
110	A1VREF
111	A1VSS
112	BVCC
113	ISOVCL
114	ISOVSS
115	P18_0 / CSIG1RY1 / CSIG1RY0 / PWGA61O / ADCA1I0S
116	P18_1 / PWGA62O / ADCA1I1S
117	P18_2 / PWGA63O / ADCA1I2S
118	P18_3 / ADCA1I3S
119	BVSS
120	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1
121	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
122	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1
123	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RY1 / CSIH0RY0

**Table 2.5 Pin Assignment 144 pin LQFP (4/4)**

Pin No.	Pin Name
124	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
125	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1
126	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
127	P10_13 / CSIH0SSI / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5
128	P10_14 / ADCA1TRG0 / PWGA19O / RLIN32TX / CSIH3SSI / TAUB0I7 / TAUB0O7
129	P11_1 / CSIH2SSI / RLIN20RX / PWGA26O / TAUB0I13 / TAUB0O13
130	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15
131	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O
132	P11_4 / CSIH2SI / CAN3TX / PWGA29O
133	P11_5 / CAN5RX*2 / INTP5 / RLIN33TX / PWGA30O / CSIH3SI
134	P11_6 / RLIN33RX / INTP13 / CAN5TX*2 / ADCA1TRG1 / PWGA31O / CSIH3SO
135	P11_7 / INTP5 / PWGA32O / CSIH3SC
136	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O
137	P12_0 / CAN2TX / PWGA56O
138	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O
139	P12_2 / RLIN34TX / PWGA58O
140	BVCC
141	BVSS
142	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI
143	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
144	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Note 1. The EVTO pin is only available in devices with 2-MB code flash memory.

Note 2. The CANmTX and CANmRX pins (m = 4, 5) are only available in devices with 1.5- and 2-MB code flash memories.

**Table 2.6 Pin Assignment 176 pin LQFP (1/5)**

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / CSIH1SSI / MEMC0CLK
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / CSIG0SSI
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	BVCC
5	BVSS
6	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / MEMC0RD
7	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / MEMC0WR
8	P11_8 / CSIG1SSI / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / MEMC0CS0
9	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / MEMC0CS1
10	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / MEMC0CS2
11	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / MEMC0CS3
12	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / MEMC0WAIT
13	P11_13 / RLIN24RX / PWGA53O / TAUB1I4 / TAUB1O4 / MEMC0BEN0
14	P11_14 / RLIN24TX / PWGA54O / TAUB1I6 / TAUB1O6 / MEMC0BEN1
15	P12_3 / RLIN27RX / PWGA68O
16	P12_4 / RLIN27TX / PWGA69O
17	P12_5 / PWGA70O
18	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / CSIH0SSI / DPO / RESETOUT
19	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
20	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
21	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
22	EVCC
23	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8
24	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
25	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
26	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX
27	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
28	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
29	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
30	P1_0 / RLIN33RX / INTP13
31	P1_1 / RLIN33TX
32	P1_2 / CAN3RX / INTP3
33	P1_3 / CAN3TX / DPIN23
34	P1_12 / CAN4RX / INTP4
35	P1_13 / CAN4TX
36	P2_6
37	EVSS
38	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
39	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
40	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
41	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S

Table 2.6 Pin Assignment 176 pin LQFP (2/5)

Pin No.	Pin Name
42	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / DCURDY / LPDCLKOUT
43	JP0_4 / DCUTRST
44	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
45	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
46	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
47	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
48	P2_1 / RLIN27TX
49	P2_0 / RLIN27RX
50	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22
51	P1_10 / RLIN24RX / DPIN21
52	P1_9 / RLIN34TX / DPIN20
53	P1_8 / RLIN34RX / INTP14
54	RESET
55	EVCC
56	XT1
57	IP0_0 / XT2
58	AWOVSS
59	AWOVCL
60	REGVCC
61	X2
62	X1
63	FLMD0
64	P2_3 / RLIN28TX
65	P2_2 / RLIN28RX
66	JP0_6 / EVTO*1
67	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
68	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
69	P0_8 / RLIN21TX / DPIN6 / CSIH1SSI / TAUB0I2 / TAUB0O2 / CAN3TX
70	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RY1 / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
71	EVSS
72	P1_7 / ADCA1TRG1 / RLIN25TX / DPIN19
73	P1_6 / RLIN25RX / DPIN18
74	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17
75	P1_4 / RLIN35RX / INTP15
76	P2_4 / RLIN29RX
77	P2_5 / RLIN29TX
78	P1_14 / RLIN23RX
79	P1_15 / RLIN23TX
80	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
81	P8_1 / TAPA0ES0 / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
82	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S

**Table 2.6 Pin Assignment 176 pin LQFP (3/5)**

<b>Pin No.</b>	<b>Pin Name</b>
83	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
84	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
85	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S
86	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
87	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
88	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
89	A0VSS
90	A0VREF
91	AP0_15 / ADCA0I15
92	AP0_14 / ADCA0I14
93	AP0_13 / ADCA0I13
94	AP0_12 / ADCA0I12
95	AP0_11 / ADCA0I11
96	AP0_10 / ADCA0I10
97	AP0_9 / ADCA0I9
98	AP0_8 / ADCA0I8
99	AP0_7 / ADCA0I7
100	AP0_6 / ADCA0I6
101	AP0_5 / ADCA0I5
102	AP0_4 / ADCA0I4
103	AP0_3 / ADCA0I3
104	AP0_2 / ADCA0I2
105	AP0_1 / ADCA0I1
106	AP0_0 / ADCA0I0
107	EVSS
108	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
109	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
110	P9_2 / KR0I6 / PWGA20O / TAPA0ES0 / CSIH2CSS2 / ADCA0I9S
111	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / ADCA0I10S
112	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
113	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
114	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
115	P20_3 / CAN4TX / PWGA67O
116	P20_2 / CAN4RX / INTP4 / PWGA66O
117	P20_1 / RLIN26TX / PWGA65O
118	P20_0 / RLIN26RX / PWGA64O
119	P20_5 / RLIN23TX / PWGA60O
120	P20_4 / RLIN23RX / PWGA59O
121	EVCC
122	AP1_11 / ADCA1I11
123	AP1_10 / ADCA1I10

**Table 2.6 Pin Assignment 176 pin LQFP (4/5)**

<b>Pin No.</b>	<b>Pin Name</b>
124	AP1_9 / ADCA1I9
125	AP1_8 / ADCA1I8
126	AP1_7 / ADCA1I7
127	AP1_6 / ADCA1I6
128	AP1_5 / ADCA1I5
129	AP1_4 / ADCA1I4
130	AP1_3 / ADCA1I3
131	AP1_2 / ADCA1I2
132	AP1_1 / ADCA1I1
133	AP1_0 / ADCA1I0
134	AP1_15 / ADCA1I15
135	AP1_14 / ADCA1I14
136	AP1_13 / ADCA1I13
137	AP1_12 / ADCA1I12
138	A1VREF
139	A1VSS
140	BVCC
141	ISOVCL
142	ISOVSS
143	P18_0 / CSIG1RY1 / CSIG1RY0 / PWGA61O / ADCA1I0S
144	P18_1 / PWGA62O / ADCA1I1S
145	P18_2 / PWGA63O / ADCA1I2S
146	P18_3 / PWGA71O / ADCA1I3S
147	P18_4 / CSIH1CSS4 / ADCA1I4S
148	P18_5 / CSIH1CSS5 / ADCA1I5S
149	P18_6 / ADCA1I6S
150	P18_7 / ADCA1I7S
151	BVSS
152	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0
153	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1
154	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1
155	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RY1 / CSIH0RY0 / MEMC0AD3
156	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4
157	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
158	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
159	P10_13 / CSIH0SSI / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5 / MEMC0AD7
160	P10_14 / ADCA1TRG0 / PWGA19O / RLIN32TX / CSIH3SSI / TAUB0I7 / TAUB0O7 / MEMC0AD8
161	P11_1 / CSIH2SSI / RLIN20RX / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
162	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10
163	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11
164	P11_4 / CSIH2SI / CAN3TX / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12

**Table 2.6 Pin Assignment 176 pin LQFP (5/5)**

Pin No.	Pin Name
165	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13
166	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14
167	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15
168	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / MEMC0ASTB
169	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / MEMC0A16
170	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
171	P12_2 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18
172	BVCC
173	BVSS
174	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / MEMC0A19
175	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
176	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Note 1. The  $\overline{EVTO}$  pin is only available in devices with 2-MB code flash memory.

内部时钟的输出

## 2.2 Pin Description

Table 2.7 Pin Functions (1/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
AnVREF	✓	✓	✓	✓	✓	✓	—	ADCAn voltage supply and reference voltage	ADCAn
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
AnVSS	✓	✓	✓	✓	✓	✓	—	ADCAn ground	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
ADCAnIm	✓	✓	✓	✓	✓	✓	I	ADCAn input channel m with 12-bit resolution	
	n = 0, m = 0 to 7	n = 0, m = 0 to 9	n = 0, m = 0 to 10	n = 0, m = 0 to 15	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15			
ADCA0ImS	✓	✓	✓	✓	✓	✓	I	ADCA0 input channel m with 10-bit resolution	
	m = 0 to 3	m = 0 to 10	m = 0 to 13	m = 0 to 19	m = 0 to 19	m = 0 to 19			
ADCA1ImS	—	—	—	—	✓	✓	I	ADCA1 input channel m with 10-bit resolution	
					m = 0 to 3	m = 0 to 7			
ADCA0SELy	✓	✓	✓	✓	✓	✓	O	External MPX selection pin y for ADCA0 input	
	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2			
ADCAnTRGy	✓	✓	✓	✓	✓	✓	I	ADCAn external trigger pin y	
	n = 0, y = 0 to 2	n = 0, y = 0 to 2	n = 0, y = 0 to 2	n = 0, y = 0 to 2	n = 0, 1, y = 0 to 2	n = 0, 1, y = 0 to 2			
AP0_m	✓	✓	✓	✓	✓	✓	IO	Analog port 0_m	Port
	m = 0 to 7	m = 0 to 9	m = 0 to 10	m = 0 to 15	m = 0 to 15	m = 0 to 15			
AP1_m	—	—	—	—	✓	✓	IO	Analog port 1_m	
					m = 0 to 7	m = 0 to 15			
APO	✓	✓	✓	✓	✓	✓	O	Port output signal for analog input	LPS
AWOVCL	✓	✓	✓	✓	✓	✓	—	Voltage regulator for AWO area capacitor connection	Power
AWOVSS	✓	✓	✓	✓	✓	✓	—	Internal logic for AWO area ground	
BVCC	—	—	—	—	✓	✓	—	Port buffer voltage supply	
BVSS	—	—	—	—	✓	✓	—	Port buffer ground	
CANmRX	✓	✓	✓	✓	✓	✓	I	CANm receive data input	RS-CANn
	m = 0	m = 0 to 2	m = 0 to 2	m = 0 to 2, m = 0 to 5*2	m = 0 to 3, m = 0 to 5*3	m = 0 to 5			
CANmTX	✓	✓	✓	✓	✓	✓	O	CANm transmit data output	
	m = 0	m = 0 to 2	m = 0 to 2	m = 0 to 2, m = 0 to 5*2	m = 0 to 3, m = 0 to 5*3	m = 0 to 5			
CSCXFOUT	✓	✓	✓	✓	✓	✓	O	<u>Clock output</u>	Clock
CSIGnRYI	✓	✓	✓	✓	✓	✓	I	CSIGn ready (1) / busy (0) input signal	CSIGn
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnRYO	✓	✓	✓	✓	✓	✓	O	CSIGn ready (1) / busy (0) output signal	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSC	✓	✓	✓	✓	✓	✓	IO	CSIGn serial clock signal	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSI	✓	✓	✓	✓	✓	✓	I	CSIGn serial data input	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSO	✓	✓	✓	✓	✓	✓	O	CSIGn serial data output	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSSI	✓	✓	✓	✓	✓	✓	I	CSIGn SS function control input signal	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			

Table 2.7 Pin Functions (2/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
CSIHnCSS0	✓	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 0	CSIHn
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS1	✓	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 1	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS2	—	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 2	
	n = 0	n = 0 to 2	n = 0 to 3						
CSIHnCSS3	—	✓	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 3	
	n = 0	n = 0 to 2	n = 0 to 3						
CSIHnCSS4	—	—	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 4	
			n = 0	n = 0, 1	n = 0 to 2	n = 0 to 2			
CSIHnCSS5	—	—	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 5	
			n = 0	n = 0, 1	n = 0 to 2	n = 0 to 2			
CSIHnCSS6	—	—	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 6	
			n = 0	n = 0	n = 0	n = 0			
CSIHnCSS7	—	—	✓	✓	✓	✓	O	CSIHn serial peripheral chip select signal 7	
			n = 0	n = 0	n = 0	n = 0			
CSIHnRYI	✓	✓	✓	✓	✓	✓	I	CSIHn ready (1) / busy (0) input signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnRYO	✓	✓	✓	✓	✓	✓	O	CSIHn ready (1) / busy (0) output signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSC	✓	✓	✓	✓	✓	✓	IO	CSIHn serial clock signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSI	✓	✓	✓	✓	✓	✓	I	CSIHn serial data input	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSO	✓	✓	✓	✓	✓	✓	O	CSIHn serial data output	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSSI	✓	✓	✓	✓	✓	✓	I	CSIHn slave select input signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
DCURDY	✓	✓	✓	✓	✓	✓	O	Debug ready	OCD
DCUTCK	✓	✓	✓	✓	✓	✓	I	Debug clock	
DCUTDI	✓	✓	✓	✓	✓	✓	I	Debug data input	
DCUTDO	✓	✓	✓	✓	✓	✓	O	Debug data output	
DCUTMS	✓	✓	✓	✓	✓	✓	I	Debug mode select	
DCUTRST	✓	✓	✓	✓	✓	✓	I	Debug reset	
DPInm	✓	✓	✓	✓	✓	✓	I	Digital port input m	LPS
	m = 0 to 2	m = 0 to 4, 8 to 10	m = 0 to 11	m = 0 to 16	m = 0 to 23	m = 0 to 23			
DPO	✓	✓	✓	✓	✓	✓	O	Output for digital input port	
ENCA0E0	✓	✓	✓	✓	✓	✓	I	ENCA0 encoder input 0	ENCA0
ENCA0E1	✓	✓	✓	✓	✓	✓	I	ENCA0 encoder input 1	
ENCA0TINm	✓	✓	✓	✓	✓	✓	I	ENCA0 capture trigger input m	
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0, 1	m = 0, 1	m = 0, 1			
ENCA0EC	✓	✓	✓	✓	✓	✓	I	ENCA0 encoder clear input	
EVCC	✓	✓	✓	✓	✓	✓	—	Port buffer voltage supply	Power
EVSS	✓	✓	✓	✓	✓	✓	—	Port buffer ground	
EVTO	—	—	—	—	✓ *1	✓ *1	O	Event output	TEU_OUT
FLMD0	✓	✓	✓	✓	✓	✓	I	Operating mode select pin 0	Mode
FLMD1	✓	✓	✓	✓	✓	✓	I	Operating mode select pin 1	

Table 2.7 Pin Functions (3/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
INTPm	✓	✓	✓	✓	✓	✓	I	External interrupt input m	INTC
	$m = 0 \text{ to } 5, \quad m = 0 \text{ to } 5, \quad m = 0 \text{ to } 8, \quad m = 0 \text{ to } 8, \quad m = 0 \text{ to } 15 \quad m = 0 \text{ to } 15$ 10, 11      10, 11      10 to 12      10 to 13								
IP0_0	—	—	—	—	✓	✓	I	Input port 0_0	Port
ISOVCL	✓	✓	✓	✓	✓	✓	—	Voltage regulator for ISO area capacitor connection	Power
ISOVSS	✓	✓	✓	✓	✓	✓	—	Internal logic for ISO area ground	
JP0_m	✓	✓	✓	✓	✓	✓	IO	JTAG port 0_m	Port
	$m = 0 \text{ to } 5 \quad m = 0 \text{ to } 6 \quad m = 0 \text{ to } 6$								
KR0Im	✓	✓	✓	✓	✓	✓	I	KR0 key input signal	KR0
	$m = 0 \text{ to } 5 \quad m = 0 \text{ to } 7 \quad m = 0 \text{ to } 7$								
LPDCLK	✓	✓	✓	✓	✓	✓	I	LPD clock input (4-pin)	LPD
LPDCLKOUT	✓	✓	✓	✓	✓	✓	O	LPD clock output (4-pin)	
LPDI	✓	✓	✓	✓	✓	✓	I	LPD data input (4-pin)	
LPDIO	✓	✓	✓	✓	✓	✓	IO	LPD data input / output (1-pin)	
LPDO	✓	✓	✓	✓	✓	✓	O	LPD data output (4-pin)	
MEMC0Am	—	—	—	—	—	✓	O	MEMC0 address m	MEMC0
	$m = 16 \text{ to } 19$								
MEMC0ADm	—	—	—	—	—	✓	IO	MEMC0 address / data m	
	$m = 0 \text{ to } 15$								
MEMC0ASTB	—	—	—	—	—	✓	O	MEMC0 address strobe	
MEMC0BENm	—	—	—	—	—	✓	O	MEMC0 byte enable m	
	$m = 0, 1$								
MEMC0CLK	—	—	—	—	—	✓	O	MEMC0 clock output	
MEMC0CSm	—	—	—	—	—	✓	O	MEMC0 chip select m	
	$m = 0 \text{ to } 3$								
MEMC0RD	—	—	—	—	—	✓	O	MEMC0 read strobe	
MEMC0WAIT	—	—	—	—	—	✓	I	MEMC0 wait input	
MEMC0WR	—	—	—	—	—	✓	O	MEMC0 write strobe	
MODEm	✓	✓	✓	✓	✓	✓	I	Sub operating mode select (Boundary scan)	Mode
	$m = 0, 1 \quad m = 0, 1$								
NMI	✓	✓	✓	✓	✓	✓	I	External non-maskable interrupt input	INTC
P0_m	✓	✓	✓	✓	✓	✓	IO	Port 0_m	Port
	$m = 0 \text{ to } 3 \quad m = 0 \text{ to } 6 \quad m = 0 \text{ to } 12 \quad m = 0 \text{ to } 14 \quad m = 0 \text{ to } 14 \quad m = 0 \text{ to } 14$								
P1_m	—	—	—	—	—	✓	IO	Port 1_m	
	$m = 0 \text{ to } 11 \quad m = 0 \text{ to } 15$								
P2_m	—	—	—	—	—	✓	IO	Port 2_m	
	$m = 0 \text{ to } 6$								
P8_m	✓	✓	✓	✓	✓	✓	IO	Port 8_m	
	$m = 0, 1 \quad m = 0 \text{ to } 6 \quad m = 0 \text{ to } 6 \quad m = 0 \text{ to } 12 \quad m = 0 \text{ to } 12 \quad m = 0 \text{ to } 12$								

Table 2.7 Pin Functions (4/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
P9_m	✓	✓	✓	✓	✓	✓	IO	Port 9_m	Port
	m = 0, 1	m = 0 to 3	m = 0 to 6	m = 0 to 6	m = 0 to 6	m = 0 to 6			
P10_m	✓	✓	✓	✓	✓	✓	IO	Port 10_m	
	m = 0 to 10	m = 0 to 14	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
P11_m	—	—	✓	✓	✓	✓	IO	Port 11_m	
			m = 0 to 4	m = 0 to 7	m = 0 to 15	m = 0 to 15			
P12_m	—	—	—	—	✓	✓	IO	Port 12_m	
					m = 0 to 2	m = 0 to 5			
P18_m	—	—	—	—	✓	✓	IO	Port 18_m	
					m = 0 to 3	m = 0 to 7			
P20_m	—	—	—	—	✓	✓	IO	Port 20_m	
					m = 4, 5	m = 0 to 5			
PWGAnO	✓	✓	✓	✓	✓	✓	O	PWGAn output signal	PWM-Diag
	n = 0 to 12	n = 0 to 23	n = 0 to 23	n = 0 to 47	n = 0 to 63	n = 0 to 71			
REGVCC	✓	✓	✓	✓	✓	✓	—	Voltage regulators voltage supply	Power
RESET	✓	✓	✓	✓	✓	✓	I	External reset input	Reset
RESETOUT	✓	✓	✓	✓	✓	✓	O	Reset output	
RIIC0SCL	✓	✓	✓	✓	✓	✓	IO	RIIC0 serial clock	RIIC0
RIIC0SDA	✓	✓	✓	✓	✓	✓	IO	RIIC0 serial data	
RLIN2mRX	✓	✓	✓	✓	✓	✓	I	RLIN2m receive data input	RLIN2m
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0 to 2	m = 0 to 5	m = 0 to 9			
RLIN2mTX	✓	✓	✓	✓	✓	✓	O	RLIN2m transmit data output	
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0 to 2	m = 0 to 5	m = 0 to 9			
RLIN3nRX	✓	✓	✓	✓	✓	✓	I	RLIN3n receive data input	RLIN3n
	n = 0	n = 0, 1	n = 0 to 2	n = 0 to 3	n = 0 to 5	n = 0 to 5			
RLIN3nTX	✓	✓	✓	✓	✓	✓	O	RLIN3n transmit data output	
	n = 0	n = 0, 1	n = 0 to 2	n = 0 to 3	n = 0 to 5	n = 0 to 5			
RTCA0OUT	—	—	—	—	✓	✓	O	RTCA0 1Hz output	RTCA0
SELDPk	—	✓	✓	✓	✓	✓	O	External multiplexer selection output signal k for digital port	LPS
		k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2			
TAPA0ESO	✓	✓	✓	✓	✓	✓	I	Hi-Z control	Motor control
TAPA0UN	✓	✓	✓	✓	✓	✓	O	Motor control output U phase (negative)	
TAPA0UP	✓	✓	✓	✓	✓	✓	O	Motor control output U phase (positive)	
TAPA0VN	✓	✓	✓	✓	✓	✓	O	Motor control output V phase (negative)	
TAPA0VP	✓	✓	✓	✓	✓	✓	O	Motor control output V phase (positive)	
TAPA0WN	✓	✓	✓	✓	✓	✓	O	Motor control output W phase (negative)	
TAPA0WP	✓	✓	✓	✓	✓	✓	O	Motor control output W phase (positive)	
TAUD0Im	✓	✓	✓	✓	✓	✓	I	TAUD0 channel input m	TAUD0
	m = 0 to 15	m = 0 to 15	m = 0 to 15						
TAUD0Om	✓	✓	✓	✓	✓	✓	O	TAUD0 channel output m	
	m = 0 to 15	m = 0 to 15	m = 0 to 15						
TAUBnIm	—	—	—	✓	✓	✓	I	TAUBn channel input m	TAUBn
				n = 0, m = 0 to 15	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15			
TAUBnOm	—	—	—	✓	✓	✓	O	TAUBn channel output m	
				n = 0, m = 0 to 15	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15			

**Table 2.7 Pin Functions (5/5)**

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
TAUJnIm	✓ n = 0, m = 0 to 3	✓ n = 0, m = 0 to 3	✓ n = 0, m = 0 to 3	✓ n = 0, 1, m = 0 to 3	✓ n = 0, 1, m = 0 to 3	✓ n = 0, 1, m = 0 to 3	I	TAUJn channel input m	TAUJn
TAUJnOm	✓ n = 0, m = 0 to 3	✓ n = 0, m = 0 to 3	✓ n = 0, m = 0 to 3	✓ n = 0, 1, m = 0 to 3	✓ n = 0, 1, m = 0 to 3	✓ n = 0, 1, m = 0 to 3	O	TAUJn channel output m	
X1, X2	✓	✓	✓	✓	✓	✓	—	MainOSC connections	MOSC
XT1, XT2	—	—	—	—	✓	✓	—	SubOSC connections	SOSC

Note 1. The EVTO pin is only available in devices with 2-MB code flash memory.

Note 2. "m = 0 to 5" is only available for F1L for Gateway.

Note 3. "m = 0 to 5" is only available in devices with 1.5- and 2- MB code flash memories.

### CAUTION

- When pin functions for a peripheral module are allocated to multiple pins, use the pins from the same port group or nearby pins as the pins for a given channel.  
(e.g.) When RS-CAN channel 0 is used:  
 CAN0TX P0\_0 P10\_1  
 CAN0RX P0\_1 P10\_0  
 Use one of the following pin combinations:  
 – P0\_0 and P0\_1.  
 – P10\_0 and P10\_1.  
 • The combinations of P0\_0 and P10\_0, and P0\_1 and P10\_1 are not allowed.

## 2.3 Pin Functions During and After Reset

Table 2.8 Pin Functions During and After Reset

Pins	During Reset <sup>*4</sup>	After Reset <sup>*4</sup>
JP0_0	High impedance	JP0_0 : High impedance Serial programming mode: FPDR, FPDT (1 wire UART) FPDR (2 wire UART) Nexus I/F : DCUTDI input LPD (4-pin) : LPDI input LPD (1-pin) : LPDIO input/output
JP0_1	High impedance	JP0_1 : High impedance Serial programming mode: FPDT Nexus I/F : DCUTDO output LPD (4-pin) : LPDO output LPD (1-pin) : High impedance
JP0_2	High impedance	JP0_2 : High impedance Serial programming mode: FPCK Nexus I/F : DCUTCK input LPD (4-pin) : LPDCLK input LPD (1-pin) : High impedance
JP0_3	High impedance	JP0_3 : High impedance Serial programming mode: High impedance Nexus I/F : DCUTMS input LPD (4-pin) : High impedance LPD (1-pin) : High impedance
JP0_4	Input <sup>*3</sup>	JP0_4 : High impedance Serial programming mode: High impedance Nexus I/F : DCUTRST input <sup>*1</sup> LPD (4-pin) : High impedance LPD (1-pin) : High impedance
JP0_5	High impedance	JP0_5 : High impedance Serial programming mode: High impedance Nexus I/F : DCURDY output LPD (4-pin) : LPDCLKOUT output LPD (1-pin) : High impedance
JP0_6	High impedance	JP0_6 : High impedance Serial programming mode: High impedance Nexus I/F : EVTO output <sup>*5</sup> LPD (4-pin) : High impedance LPD (1-pin) : High impedance
P0_0	Output <sup>*2</sup>	Output <sup>*2</sup>
P0_1 to P0_14	High impedance	High impedance
P1, P2, P8 to P12, P18, P20 (except P10_1, P10_2, and P10_8)	High impedance	High impedance
P10_1	High impedance	High impedance (FLMD = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE0 input (FLMD0 = 1, FLMD1 = 1)
P10_2	High impedance	High impedance (FLMD = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE1 input (FLMD0 = 1, FLMD1 = 1)
P10_8	High impedance	High impedance (FLMD = 0) FLMD1 input (FLMD0 = 1)
FLMD0	Input	Input
RESET	Input	Input
AP0, AP1	High impedance	High impedance

Note 1. When Nexus is enabled and no external device is connected, the level of the pin must always be fixed to low.

Note 2. RESETOUT is output. For details, see **Section 2.11, Port (Special I/O) Function Overview**.

Note 3. When the power is turned on or when a reset is deasserted, the JP0\_4 pin should be driven Low.

Note 4. It is shown that some pin status is high impedance during and after reset. However the output pin has a possibility to become unstable at the transition moment (less than 12ns) of external and internal reset. After that, pin status will be Hi-Z. The phenomenon of unstable output period and influence of the pin level after the unstable situation, it depends on customer board and measurement conditions.

Note 5. The EVTO pin is only available in devices with 2-MB code flash memory.

## 2.4 Port State in Standby Mode

For the port states in standby mode, see **Section 11.1.4, I/O Buffer Control**.

## 2.5 Recommended Connection of Unused Pins

If the pins are not used, it is recommended to connect them as shown below.

**Table 2.9 Recommended Connection of Unused Pins**

Pin	Recommended Connection of Unused Pins
A0VREF, A1VREF <sup>3</sup>	Connected to EVCC or BVCC <sup>3</sup>
A0VSS, A1VSS <sup>3</sup>	Connected to EVSS or BVSS <sup>3</sup>
<u>RESET</u>	Connected to EVCC
XT1	Connected to AWOVSS
IP0_0	Connected to REGVCC or AWOVSS via a resistor (bit 0 of IPBIC0 = 1) Open (bit 0 of IPBIC0 = 0)
JP0 (excluding JP0_4) P0 (excluding P0_0) P1 P2 P8 P9 P20	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P0_0	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
JP0_4	Connected to EVSS via a resistor <sup>2</sup>
P10_1, P10_2, P10_8	Input: Open (when the PIBCn_m and PMCn_m bits are 0) For 48 pins, 64 pins, 80 pins, 100 pins: Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) For 144 pins and 176 pins: Connected to BVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P10 (excluding P10_1, P10_2, P10_8) P11 P12 P18	Input: Open (when the PIBCn_m and PMCn_m bits are 0) For 48 pins, 64 pins, 80 pins, 100 pins: Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) For 144 pins and 176 pins: Connected to BVCC or BVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
AP0	Input: Open (when the PIBCn_m bit is 0) Connected to A0VREF or A0VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
AP1	Input: Open (when the PIBCn_m bit is 0) Connected to A1VREF or A1VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
Nexus/LPD I/F (JP0)	DCUTDI/LPDI/LPDIO (JP0_0): Connected to EVCC via a resistor DCUTDO/LPDO (JP0_1): Open DCUTCK/LPDCLK (JP0_2): Open DCUTMS (JP0_3): Connected to EVCC via a resistor <u>DCUTRST</u> (JP0_4): Connected to EVSS via a resistor <sup>2</sup> <u>DCURDY</u> /LPDCLKOUT (JP0_5): Open <u>EVTO</u> (JP0_6): Open <sup>1</sup>

Note 1. The EVTO pin is only available in devices with 2-MB code flash memory.

Note 2. For details, see the specifications of the development tool.

Note 3. Only in 144 and 176 pin devices.

## 2.6 RH850/F1L Port Features

### 2.6.1 Port Group

The RH850/F1L provides the following port groups, indicated by the numbers in the table below.

**Table 2.10 Port Groups in RH850/F1L**

No. of Pins	Port Group	RH850/F1L
48 pins	Number	6
	Name	P0, P8 to P10, JP0, AP0
64 pins	Number	6
	Name	P0, P8 to P10, JP0, AP0
80 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0
100 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0
144 pins	Number	13
	Name	P0, P1, P8 to P12, P18, P20, JP0, AP0, AP1, IP0
176 pins	Number	14
	Name	P0 to P2, P8 to P12, P18, P20, JP0, AP0, AP1, IP0

### 2.6.2 Port Group Index n

Throughout this section, the port groups are identified by using the index “n” (n = 0 to 2, 8 to 12, 18, and 20). For example, the port mode control register of the Pn pin is PMCn.

### 2.6.3 Register Base Address

Port and JTAG port base addresses are listed in the following table.

Port and JTAG port register addresses are given as offsets from the base addresses.

**Table 2.11 Register Base Addresses**

Base Address Name	Base Address
<PORTn_base>	FFC1 0000 <sub>H</sub>
<JPORT0_base>	FFC2 0000 <sub>H</sub>

### 2.6.4 Clock Supply

The clock supply to ports is shown in the following table.

**Table 2.12 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
Port	Register access clock	CPUCLK2

## 2.7 Port Functions

The microcontroller has various pins for input/output functions, known as ports. The ports are organized in port groups.

The RH850/F1L also has several control registers to enable pins to be used as other than general purpose input/output pins.

For a description of the terms pin, port, or port group, see **Section 2.7.2, Terms**.

### 2.7.1 Functional Overview

- All the port settings can be specified individually.
- The maximum number of bits (pins) in a port is 16.
- The output level of any pin can be set independently without affecting the other pins in the same port.
- Input buffers are enabled through registers settings.
- Pin level is read by dedicated port-pin-read register (PPR)
- All possible port functions are shown in the tables listed below.

**Section 2.9.2, Pin Function Configuration, Table 2.40, Table 2.42, Table 2.44, Table 2.46, Table 2.48, Table 2.50, Table 2.52, Table 2.54, Table 2.56, Table 2.58, Table 2.60, Table 2.62, Table 2.64 and Table 2.66.**

#### CAUTION

Some input or output functions may be assigned to more than one port. Only activate a given function on a single pin. Do not activate a function on multiple pins at the same time. This also applies in cases where multiple peripheral functions are assigned to a single multiplexed function and only one of these functions is used.

#### [Example]

INTP0 is assigned to the following pins on this device. However, the INTP0 function should not be activated on more than one pin. After activating the function on one pin, do not activate it on another.

- JP0\_0 (1st input alternative function)
- P0\_1 (2nd, 3rd input alternative function)
- P10\_0 (2nd input alternative function)

In the above case, when the 1st input alternative function (INTP0) of JP0\_0 is selected, using the 2nd input alternative function (CAN0RX/INTP0) of P0\_1 only for the CAN signal is also prohibited.

## 2.7.2 Terms

The following terms are used in this section:

### Pin

Denotes the physical pin. Every pin is denoted by a unique pin number.

A pin can be used in several modes. Each pin is assigned a name that reflects its function, which is determined by the selected mode.

### Port group

Denotes a group of pins. All the pins of a specific port group are controlled by the same port control register.

### Port mode and ports

A pin in port mode works as a general purpose input/output pin. It is then called “port”.

The corresponding name is Pn\_m. For example, P0\_7 denotes port 7 of port group 0. It is referenced as “port P0\_7”.

### Alternative mode

In alternative mode, a pin can be used for various non-general-purpose input/output functions, such as the input/output pin of on-chip peripherals.

The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

Note that two different names can refer to the same physical pin, for example P0\_0 and INTP0. The different names indicate the function of the pin at that time.

### 2.7.2.1 JTAG Ports

The JTAG port groups are used for connecting a debugger for on-chip debugging. These are special port groups provided because the microcontroller cannot be used for the user's application while on-chip debugging is being executed. When a debugger is not connected and the microcontroller is operating normally, these port groups can be used in the same way as the other port groups.

JTAG port group registers and bit names are prefixed by a “J”. For example, JP0 denotes JTAG port group 0, and JPM0.JPM0\_m denotes the JPM0\_m port mode bit of the JPM0 port mode register.

#### NOTE

In this section, the description about all ports and their registers other than PFCEn, PFCAEn, PIPCn, PDSCn, and PISn applies to the JTAG port unless otherwise specified.

### 2.7.3 Overview of Pin Functions

Pins can operate in three modes.

- Port mode (PMCn.PMCn\_m bit = 0)

GPIO

A pin in port mode operates as a general purpose input/output pin. The I/O mode is selected by setting the PMn.PMn\_m bit.

- Software I/O control alternative mode (PMCn.PMCn\_m bit = 1, PIPCn.PIPCn\_m bit = 0)

In this mode, the pins operate as alternative functions. The I/O mode is selected by setting the PMn.PMn\_m bit. **设置PMn.PMn\_m bit 来选择I/O mode**

- Direct I/O control alternative mode (PMCn.PMCn\_m bit = 1, PIPCn.PIPCn\_m bit = 1)

In this mode, the pins operate as alternative functions. Unlike the software I/O control alternative mode, however, the I/O mode is directly controlled by the alternative function.

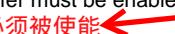
**不同于 software I/O control alternative mode , I/O 的 mode 直接由 alternative function 控制**

An overview of the register settings is given in the tables below.

Table 2.13 Pin Function Configuration (Overview)

Mode	Bit			I/O
	PMCn_m	PMn_m	PIPCn_m	
Port mode	0	0	X	O output
		1*1		I input
Software I/O control alternative mode	1	0	0	O
		1	0	I
Direct I/O control alternative mode		X	1	Controlled by the alternative function

Note 1. The input buffer must be enabled (PIBCn\_m = 1).

**输入缓存必须被使能** 

- Software I/O control alternative mode (PIPCn.PIPCn\_m = 0)
  - Output (PMn\_m = 0): Alternative output mode 1 to Alternative output mode 5
  - Input (PMn\_m = 1): Alternative input mode 1 to Alternative input mode 5
- Direct I/O control alternative mode (PIPCn.PIPCn\_m = 1)
  - The I/O mode for Alternative output mode 1 to Alternative output mode 5 and Alternative input mode 1 to Alternative input mode 5 is directly selected by the alternative function.

Table 2.14 Alternative Mode Selection Overview (PMCn.PMCn\_m Bit = 1)

Mode	Register					I/O
	PIPC*1	PM*1	PFCAE	PFCE	PFC	
Alternative output mode 1 (ALT-OUT1)	0	0	0	0	0	O
Alternative input mode 1 (ALT-IN1)		1				I
Alternative output mode 2 (ALT-OUT2)		0	0	0	1	O
Alternative input mode 2 (ALT-IN2)		1				I
Alternative output mode 3 (ALT-OUT3)		0	0	1	0	O
Alternative input mode 3 (ALT-IN3)		1				I
Alternative output mode 4 (ALT-OUT4)		0	0	1	1	O
Alternative input mode 4 (ALT-IN4)		1				I
Alternative output mode 5 (ALT-OUT5)		0	1	0	0	O
Alternative input mode 5 (ALT-IN5)		1				I
Other than above	Setting prohibited					

Note 1. If PIPCn.PIPCn\_m = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

If a pin is in alternative mode (PMCn.PMCn\_m = 1), one of up to five alternative functions can be selected for that pin by using the PFCn, PFCEn, and PFCAEn registers.

这三个寄存器来选择1-5之间的alternative function

## 2.7.4 Pin Data Input/Output

以下的寄存器用于数据的输出/输入  
The registers used for data input/output are described below.

The location that is read via the PPRn register differs depending on the pin mode.  
**Pin mode不同，通过PPRn寄存器读取的也不一样**

### 2.7.4.1 Output Data

In the port mode (PMCn.PMCn\_m = 0), the value of the Pn.Pn\_m bit is output from the Pn\_m pin.

### 2.7.4.2 Input Data

When the PPRn register is read, either the value of the Pn\_m pin, the value of the Pn.Pn\_m bit, or the value output by the alternative function is returned. **通过读取PPRn寄存器，Pn\_m的值，Pn.Pn\_m bit值或者是复用功能的输出值将被返回**

Which value is returned depends on the pin mode and setting of several control bits.  
**哪个值被返回取决于Pin mode和几个控制位的设置**

The different PPRn read modes are shown in the table below.

Table 2.15 PPRn\_m Read Values

PMCn_m	PMn_m	PIBCn_m	PIPCn_m	PODCn_m	Mode	PPRn_m Read Value	
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m bit	
		1		X	Port input, input buffer enabled	Pn_m pin	
	0	X	0		Port push-pull output	Pn.Pn_m bit*1	
				1	Port open-drain output		
1	1	X	0	X	Software I/O control alternative input	Pn_m pin	
	0	X		0	Software I/O control alternative push-pull output	Output signal from the alternative function*1	
				1	Software I/O control alternative open-drain output		
	X	X	1	0	Direct I/O control alternative input or push-pull output	I/O port in alternative mode: <ul style="list-style-type: none"><li>Input: Pn_m pin</li><li>Output: <u>Output signal from the alternative function*1</u></li></ul>	
				1	Direct I/O control alternative input or open-drain output		

Note 1. When PBDCn\_m = 1, the level of the Pn\_m pin is returned by the PPRn\_m bit.

The control registers in the above table have the following effects:

- **PMCn.PMCn\_m bit** 区分是port mode还是alternative mode  
This bit selects port mode (PMCn\_m = 0) or alternative mode (PMCn\_m = 1).
- **PMn.PMn\_m bit** 当处于Port mode或者software I/Ocontrol alternative mode的时候，此位决定是输入模式还是输入模式  
This bit selects input (PMn\_m = 1) or output (PMn\_m = 0) when the port mode (PMCn\_m = 0) and software I/O control alternative mode (PMCn\_m = 1, PIPCn\_m = 0) have been selected.
- **PIBCn.PIBCn\_m bit** 在Port mode的输入模式下，此位决定是使能还是禁止input buffer  
This bit disables (PIBCn\_m = 0) or enables (PIBCn\_m = 1) the input buffer in input port mode (PMCn\_m = 0 and PMn\_m = 1). If the input buffer is disabled, PPRn\_m reads the Pn.Pn\_m bit; otherwise the Pn\_m pin level is returned. **如果input buffer被禁止，PPRn\_m读取Pn.Pn\_m bit，否则Pn\_m pin电平被返回**
- **PIPCn.PIPCn\_m bit**  
This bit selects software I/O control alternative mode or direct I/O control alternative mode. **此位区分是software I/O alternative mode还是 direct I/O control alternative mode**

- PODCn.PODCn\_m bit

This bit selects push-pull output (PODCn\_m = 0) or open-drain output (PODCn\_m = 1).  
此位选择是推挽输出还是开漏输出

- PBDCn.PBDCn\_m bit

In output mode, when this bit is set to 1, the pin enters the bidirectional mode. In bidirectional mode, the level of the signal on a Pn\_m pin can be read from PPRn.PPRn\_m.

**CAUTION** 在输出模式下，当此位被置位，此pin进入bidirectional mode，在此mode下，Pn\_m的信号电平可以从 PPRn.PPRn\_m读取

在Pn\_m作为alternative input功能时，使能 bidirectional mode，这样Pn\_m上的电平可以通过 PPRn.PPRn\_m来读取

When using Pn\_m as an alternative output function (PMCn.PMCn\_m = 1, PMn.PMn\_m = 0), the level of the Pn\_m pin can be read at the PPRn.PPRn\_m bit by enabling bidirectional mode (PBDCn.PBDCn\_m = 1).

Note, however, that in this case, the level of the Pn\_m pin will be input to the alternative input function that the Pn\_m pin is being used as.

#### 2.7.4.3 Writing to the Pn Register

The data to be output via port Pn\_m in port mode (PMCn.PMCn\_m = 0) is held in port register Pn.

Pn data can be overwritten in two ways: 重写Pn数据的方式有两种

- By writing data directly to the Pn register. 1. 将数据直接写入Pn寄存器

In this case, new data can be written directly to the Pn register.

- By performing an indirect bitwise operation (a “set”, “reset”, or “not” operation) on the Pn register. 2. 使用间接的位反转操作 (set reset 或者not)

An indirect bitwise operation (“set”, “reset”, or “not”) can be performed on the Pn register by using the following two registers:

- Port set reset register PSRn

If the PSRn.PSRn\_m bit = 1, the value of the Pn.Pn\_m bit is determined by the value of the PSRn.PSRn\_m bit. 对应高16位 Pn.Pn\_m bit取决于 PSRn.PSRn\_m 的bit  
也就是说无需直接操作Pn寄存器既可以set或者reset Pn\_m bit

In other words, the Pn\_m bit can be set or reset without writing directly to the Pn register.

- Port NOT register PNOTn

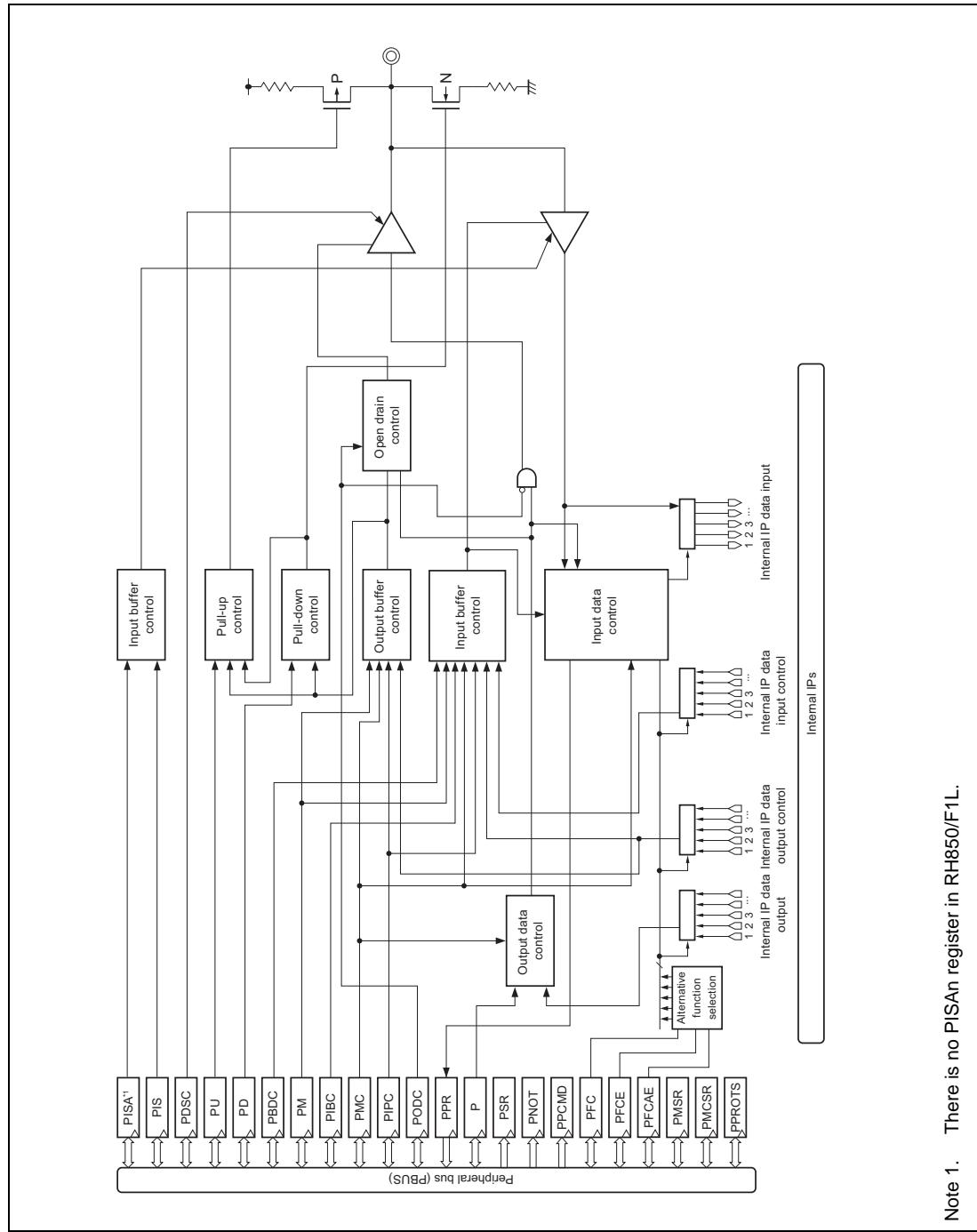
By setting PNOTn.PNOTn\_m bit to 1, the Pn.Pn\_m bit can be inverted without writing directly to the Pn register.

An indirect bitwise operation on the Pn register (“set”, “reset”, or “not”) has no effect on the bits that do not need to be updated, allowing you to overwrite only the bit or bits that need to be overwritten.

在Pn register上进行的间接的位反转操作不影响 不需要更新的bit，只是允许你重写需要重新写入的bit或者bits

## 2.8 Schematic View of Port Control

The following figure is a schematic view of the port control functions.



**Figure 2.7 Schematic View of Port Control**

### CAUTION

Use documented alternative functions only.

## 2.9 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then describes all registers in detail. The configuration registers are grouped as follows:

- **Section 2.9.2, Pin Function Configuration**
- **Section 2.9.3, Pin Data Input/Output**
- **Section 2.9.4, Configuration of Electrical Characteristics**

### 2.9.1 Overview

The following registers are used for setting the individual pins of the port groups.

For details on <PORTn\_base> and <JPORT0\_base>, see **Section 2.6.3, Register Base Address**.

**Table 2.16 Port Group Configuration Registers (1/2)**

Register Name	Symbol	Address
<b>Pin function configuration</b>		
Port mode control register	PMCn	<PORTn_base> + 0400 <sub>H</sub> + n × 4
	JPMCO	<JPORT0_base> + 0040 <sub>H</sub>
Port mode control set/reset register	PMCSRn	<PORTn_base> + 0900 <sub>H</sub> + n × 4
	JPMCSR0	<JPORT0_base> + 0090 <sub>H</sub>
Port IP control register	PIPCn	<PORTn_base> + 4200 <sub>H</sub> + n × 4
Port mode register	PMn	<PORTn_base> + 0300 <sub>H</sub> + n × 4
APMn: analog port mode		
	APMn	<PORTn_base> + 03C8 <sub>H</sub> + n × 4
	JPM0	<JPORT0_base> + 0030 <sub>H</sub>
Port mode set/reset register	PMSRn	<PORTn_base> + 0800 <sub>H</sub> + n × 4
	APMSRn	<PORTn_base> + 08C8 <sub>H</sub> + n × 4
	JPMR0	<JPORT0_base> + 0080 <sub>H</sub>
Port input buffer control register	PIBCn	<PORTn_base> + 4000 <sub>H</sub> + n × 4
	APIBCn	<PORTn_base> + 40C8 <sub>H</sub> + n × 4
	JPIBC0	<JPORT0_base> + 0400 <sub>H</sub>
	IPIBC0	<PORTn_base> + 40F0 <sub>H</sub>
Port function control register	PFCn	<PORTn_base> + 0500 <sub>H</sub> + n × 4
	JPFC0	<JPORT0_base> + 0050 <sub>H</sub>
Port function control expansion register	PFCEn	<PORTn_base> + 0600 <sub>H</sub> + n × 4
Port function control additional expansion register	PFCAEn	<PORTn_base> + 0A00 <sub>H</sub> + n × 4
<b>Pin data input/output</b>		
Port bidirection control register	PBDCn	<PORTn_base> + 4100 <sub>H</sub> + n × 4
	APBDCn	<PORTn_base> + 41C8 <sub>H</sub> + n × 4
	JPBDC0	<JPORT0_base> + 0410 <sub>H</sub>
Port pin read register	PPRn	<PORTn_base> + 0200 <sub>H</sub> + n × 4
	APPn	<PORTn_base> + 02C8 <sub>H</sub> + n × 4
	JPPR0	<JPORT0_base> + 0020 <sub>H</sub>
	IPPR0	<PORTn_base> + 02F0 <sub>H</sub>
Port register	Pn	<PORTn_base> + 0000 <sub>H</sub> + n × 4
	APn	<PORTn_base> + 00C8 <sub>H</sub> + n × 4
	JPO	<JPORT0_base> + 0000 <sub>H</sub>

**Table 2.16 Port Group Configuration Registers (2/2)**

Register Name	Symbol	Address
Port NOT register	PNOTn	<PORTn_base> + 0700 <sub>H</sub> + n × 4
	APNOTn	<PORTn_base> + 07C8 <sub>H</sub> + n × 4
	JPNOT0	<JPORT0_base> + 0070 <sub>H</sub>
Port set/reset register	PSRn	<PORTn_base> + 0100 <sub>H</sub> + n × 4
	APSrn	<PORTn_base> + 01C8 <sub>H</sub> + n × 4
	JPSR0	<JPORT0_base> + 0010 <sub>H</sub>
<b>Configuration of electrical characteristics</b>		
Pull-up option register	PUn	<PORTn_base> + 4300 <sub>H</sub> + n × 4
	JPU0	<JPORT0_base> + 0430 <sub>H</sub>
Pull-down option register	PDn	<PORTn_base> + 4400 <sub>H</sub> + n × 4
	JPD0	<JPORT0_base> + 0440 <sub>H</sub>
Port drive strength control register	PDSCn	<PORTn_base> + 4600 <sub>H</sub> + n × 4
Port open drain control register	PODCn	<PORTn_base> + 4500 <sub>H</sub> + n × 4
	JPODC0	<JPORT0_base> + 0450 <sub>H</sub>
Port input buffer selection register	PISn	<PORTn_base> + 4700 <sub>H</sub> + n × 4
Port input buffer selection advanced register	JPISA0	<JPORT0_base> + 04A0 <sub>H</sub>
<b>Port protection</b>		
Port protection command register	PPCMDn	<PORTn_base> + 4C00 <sub>H</sub> + n × 4
	JPPCMD0	<JPORT0_base> + 04C0 <sub>H</sub>
Port protection status register	PPROTSn	<PORTn_base> + 4B00 <sub>H</sub> + n × 4
	JPPROTS0	<JPORT0_base> + 04B0 <sub>H</sub>

**Index n**

In **Table 2.16, Port Group Configuration Registers**, the index “n” in register symbols denotes the actual indices of the individual port groups. For example, PMCn generically indicates a port mode control register for port group n (Pn). The values for n differ according to the number of pins on the device in the way shown in **Table 2.17**.

**Table 2.17 Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Symbols**

Number of Pins on the Device	Port Groups	Values for “n”
48 pins	P0, P8, P9, P10	0, 8, 9, 10
	AP0	0
64 pins	P0, P8, P9, P10	0, 8, 9, 10
	AP0	0
80 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0
100 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0
144 pins	P0, P1, P8, P9, P10, P11, P12, P18, P20	0, 1, 8, 9, 10, 11, 12, 18, 20
	AP0, AP1	0, 1
176 pins	P0, P1, P2, P8, P9, P10, P11, P12, P18, P20	0, 1, 2, 8, 9, 10, 11, 12, 18, 20
	AP0, AP1	0, 1

### JTAG port registers

JTAG port registers are not explicitly described in the following register descriptions.

All descriptions (except for those of the PFCEn register, PFCAEn register, PIPCn register, PDSCn register, and PISn register) apply to JTAG port registers. Note, however, that the JTAG port register base address differs from that of regular ports.

### Value after reset

The values after reset depend on the ports. For the values after reset, see the register descriptions in the following pages.

## 2.9.2 Pin Function Configuration

### 2.9.2.1 PMCn / JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

**Access:** PMCn: This register can be read or written in 16-bit units.  
JPMC0: This register can be read or written in 8-bit units.

**Address:** PMCn: <PORTn\_base> + 0400H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
JPMC0: <JPORT0\_base> + 0040H<sup>\*1</sup>

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC n_15	PMC n_14	PMC n_13	PMC n_12	PMC n_11	PMC n_10	PMC n_9	PMC n_8	PMC n_7	PMC n_6	PMC n_5	PMC n_4	PMC n_3	PMC n_2	PMC n_1	PMC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), and Table 2.61, Control Registers (P20).**

Table 2.18 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

**CAUTIONS** 只是设置为alternative mode，I/O是不能控制的，如果alternative function需要直接控制I/O，需要将PIPCn.PIPCN\_m bit设置为1

- I/O is not controlled by only setting alternative mode (PMCn.PMCn\_m bit = 1). If the alternative function requires direct I/O control, also set the PIPCN.PIPCN\_m bit to 1.
- If a port is to be used as an input pin in alternative mode, the signals from some pins will pass through a noise filter. These pins may require the setting of the FCLA0CTLm\_<name>, DNFA<name>CTL and the DNFA<name>EN register. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**, and **Section 2.13, Description of Port Noise Filter & Edge/Level Detection**.

#### NOTE

The control bits of the JTAG port mode control register (JPMC0) are JPMC0\_[7:0].

### 2.9.2.2 PMCSRn / JPMCSR0 — Port Mode Control Set/Reset Register

此寄存器提供一种额外的方法将控制数据写入PMC register

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits of PMCSRn act as a mask which specifies whether or not the value of PMCn.PMCn\_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

**Access:** PMCSRn: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000<sub>H</sub>. Reading bits 15 to 0 returns the value of register PMCn.

JPMCSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000<sub>H</sub>. Reading bits 7 to 0 returns the value of register JPMC0.

**Address:** PMCSRn: <PORTn\_base> + 0900<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)

JPMCSR0: <JPORT0\_base> + 0090<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMC SRn_31	PMC SRn_30	PMC SRn_29	PMC SRn_28	PMC SRn_27	PMC SRn_26	PMC SRn_25	PMC SRn_24	PMC SRn_23	PMC SRn_22	PMC SRn_21	PMC SRn_20	PMC SRn_19	PMC SRn_18	PMC SRn_17	PMC SRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMC SRn_15	PMC SRn_14	PMC SRn_13	PMC SRn_12	PMC SRn_11	PMC SRn_10	PMC SRn_9	PMC SRn_8	PMC SRn_7	PMC SRn_6	PMC SRn_5	PMC SRn_4	PMC SRn_3	PMC SRn_2	PMC SRn_1	PMC SRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device.

See the following tables in Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), and Table 2.61, Control Registers (P20).

Table 2.19 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMCSRn_m (PMCSRn_[15:0]) is written to PMCn_m. 0: PMCn_m is not affected by PMCSRn_m. 1: PMCn_m is PMCSRn_m. Example: If PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15.
15 to 0	PMCSRn_[15:0]	Data bits that specify the value of PMCn_m if PMCSRn_m of the corresponding upper bit (PMCSRn_[31:16]) is 1. 0: PMCn_m is 0. 1: PMCn_m is 1.

#### NOTE

The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0\_[31:0].

### 2.9.2.3 PIPCn — Port IP Control Register

决定Pn\_m pin的I/O方向是由port mode register控制还是由alternative function控制

This register specifies whether the I/O direction of the Pn\_m pin is controlled by the port mode register PMn.PMn\_m or by an alternative function.

如果Pn\_m pin工作在alternative mode，并且alternative function需要直接控制 I/O方向，那么 PIPCn.PIPCn\_m必须被置为1。这将I/O的控制权移交给 alternative function，并禁止PMn.PMn\_m的配置。

If the Pn\_m pin is operated in alternative mode (PMCn.PMCn\_m = 1) and the alternative function requires direct control of the I/O direction, then PIPCn.PIPCn\_m must be set to 1 as well. This transfers I/O control to the alternative function and overrules the PMn.PMn\_m setting.

Regarding the alternative functions for which the PIPC register must be set, see **Section 2.11, Port (Special I/O) Function Overview**

**Access:** This register can be read or written in 16-bit units.

**Address:** PIPCn: <PORTn\_base> + 4200H + n × 4 (n = 0, 10, 11)\*<sup>1</sup>

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.43, Control Registers (P0)**, **Table 2.53, Control Registers (P10)**, and **Table 2.55, Control Registers (P11)**.

**Table 2.20 PIPCn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

### 2.9.2.4 PMn / APMn / JPM0 — Port Mode Register

This register specifies whether the individual pins of the port group n are in input mode or in output mode. **此寄存器指定某个port group n中的单独的pin n在input状态还是在output状态**

**Access:** PMn, APMn: These registers can be read or written in 16-bit units.  
JPM0: This register can be read or written in 8-bit units.

**Address:** PMn: <PORTn\_base> + 0300<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
APMn: <PORTn\_base> + 03C8<sub>H</sub> + n × 4 (n = 0, 1)  
JPM0: <JPORT0\_base> + 0030<sub>H</sub><sup>\*1</sup>

**Value after reset:** FFFF<sub>H</sub><sup>\*2</sup>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 <sup>*3</sup>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JPO)**, **Table 2.43, Control Registers (P0)**, **Table 2.45, Control Registers (P1)**, **Table 2.47, Control Registers (P2)**, **Table 2.49, Control Registers (P8)**, **Table 2.51, Control Registers (P9)**, **Table 2.53, Control Registers (P10)**, **Table 2.55, Control Registers (P11)**, **Table 2.57, Control Registers (P12)**, **Table 2.59, Control Registers (P18)**, **Table 2.61, Control Registers (P20)**, **Table 2.63, Control Registers (AP0)**, and **Table 2.65, Control Registers (AP1)**.

Note 2. **The PM0 register is FFFE<sub>H</sub>.**

Note 3. The PM0\_0 bit is 0.

**Table 2.21 PMn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

#### NOTES

1. To use a port in input port mode (PMCn.PMCn\_m = 0 and PMn.PMn\_m = 1), the input buffer must be enabled (PIBCn.PIBCn\_m = 1). **要使用Input port mode, 必须使能Input buffer**
2. By default, PMn.PMn\_m specifies the I/O direction in port mode (PMCn.PMCn\_m = 0) and alternative mode (PMCn.PMCn\_m=1), since PIPCn.PIPCn\_m = 0 (I/O mode is controlled by PMn.PMn\_m) after reset.
3. The control bits of the analog port register (APMn) are APMn\_[15:0].
4. The control bits of the JTAG port mode register (JPM0) are JPM0\_[7:0].

### 2.9.2.5 PMSRn / APMSRn / JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method to write data to the PMn register.

The upper 16 bits of PMSRn act as a mask which specifies whether or not the value PMn.PMn\_m is set by the corresponding bit in the lower 16 bits of PMSRn.

**Access:** PMSRn, APMSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as  $0000_{H}$ .

Reading bits 15 to 0 returns the value of registers PMn and APMn.

JPMSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as  $000000_{H}$ . Reading bits 7 to 0 returns the value of register JPM0.

**Address:** PMSRn:  $<\text{PORTn}_\text{base}> + 0800_{H} + n \times 4$  ( $n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20$ )

APMSRn:  $<\text{PORTn}_\text{base}> + 08C8_{H} + n \times 4$  ( $n = 0, 1$ )

JPMSR0:  $<\text{JPORT0}_\text{base}> + 0080_{H}$ <sup>\*1</sup>

**Value after reset:**  $0000\ FFFF_{H}$ <sup>\*2</sup>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$1^{*3}$
	R/W															

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), Table 2.61, Control Registers (P20), Table 2.63, Control Registers (AP0), and Table 2.65, Control Registers (AP1).**

Note 2. The PMSR0 register is 0000 FFFE<sub>H</sub>.

Note 3. The PMSR0\_0 bit is 0.

**Table 2.22 PMSRn Register Contents**

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMSRn_m (PMSRn_[15:0]) is written to PMn_m. 0: PMn_m is not affected by PMSRn_m. 1: PMn_m is PMSRn_m.  Example: If PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15.
15 to 0	PMSRn_[15:0]	Data bits that specify the value of PMn_m if PMSRn_m of the corresponding upper bit (PMSRn_[31:16]) is 1. 0: PMn_m is 0. 1: PMn_m is 1.

#### NOTES

1. The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0\_[31:0].
2. The control bits of the analog port mode set/reset register (APMSRn) are APMSRn\_[31:0].

### 2.9.2.6 PIBCn / APIBCn / JPIBC0 / IPIBC0 — Port Input Buffer Control Register

In input port mode (PMCn.PMCn\_m = 0 and PMn.PMn\_m = 1), this register enables the port pin's input buffer. 在input port mode下，此寄存器使能port pin的input buffer

**Access:** PIBCn, APIBCn, IPIBC0: These registers can be read or written in 16-bit units.  
JPIBC0: This register can be read or written in 8-bit units.

**Address:** PIBCn: <PORTn\_base> + 4000<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
APIBCn: <PORTn\_base> + 40C8<sub>H</sub> + n × 4 (n = 0, 1)  
JPIBC0: <JPORT0\_base> + 0400<sub>H</sub>  
IPIBC0: <PORTn\_base> + 40F0<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBC n_15	PIBC n_14	PIBC n_13	PIBC n_12	PIBC n_11	PIBC n_10	PIBC n_9	PIBC n_8	PIBC n_7	PIBC n_6	PIBC n_5	PIBC n_4	PIBC n_3	PIBC n_2	PIBC n_1	PIBC n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Note 1.** The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0)**, **Table 2.43, Control Registers (P0)**, **Table 2.45, Control Registers (P1)**, **Table 2.47, Control Registers (P2)**, **Table 2.49, Control Registers (P8)**, **Table 2.51, Control Registers (P9)**, **Table 2.53, Control Registers (P10)**, **Table 2.55, Control Registers (P11)**, **Table 2.57, Control Registers (P12)**, **Table 2.59, Control Registers (P18)**, **Table 2.61, Control Registers (P20)**, **Table 2.63, Control Registers (AP0)**, **Table 2.65, Control Registers (AP1)**, and **Table 2.67, Control Registers (IP0)**.

**Table 2.23 PIBCn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer. 0: Input buffer disabled 1: Input buffer enabled
<b>NOTES</b>		
1. When the input buffer is disabled, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally.		
2. The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].		

#### CAUTION

Settings in this register are overruled in bidirectional mode (PBDCn.PBDCn\_m = 1).

设置此寄存器会禁止bidirectional mode (双向模式)

### 2.9.2.7 PFCn / JPFC0 — Port Function Control Register

**此寄存器和PFCEn和PFCAEn一起指定某pin 的alternative function**  
 This register, together with register PFCEn and PFCAEn, specifies an alternative function of the pins.

一部分alternative function直接控制 Pn\_m pin的I/O，对于这样的alternative function，

Some alternative functions directly control the I/O of the Pn\_m pin. For such alternative functions,  
**PIPCn.PIPCn\_m必须设置位1 并且I/O由peripheral function来选择**  
 PIPCN.PIPCN\_m must be set to 1 and the I/O is selected by the peripheral function.

**对于其他的alternative functions, input/output由PMn.PMn\_m来指定**  
 For other alternative functions, input/output must be specified by PMn.PMn\_m.

**Access:** PFCn: This register can be read or written in 16-bit units.  
 JPFC0: This register can be read or written in 8-bit units.

**Address:** PFCn: <PORTn\_base> + 0500H + n × 4 (n = 0, 1, 8, 9, 10, 11, 12, 18, 20)  
 JPFC0: <JPORT0\_base> + 0050H<sup>\*1</sup>

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFCn_15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PFCn_14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PFCn_13	R/W															
PFCn_12	R/W															
PFCn_11	R/W															
PFCn_10	R/W															
PFCn_9	R/W															
PFCn_8	R/W															
PFCn_7	R/W															
PFCn_6	R/W															
PFCn_5	R/W															
PFCn_4	R/W															
PFCn_3	R/W															
PFCn_2	R/W															
PFCn_1	R/W															
PFCn_0	R/W															

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device.  
 See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), and Table 2.61, Control Registers (P20).**

**Table 2.24 PFCn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of the pin. For details, see <b>Table 2.27, Setting Alternative Functions</b> .

#### NOTE

The control bits of the JTAG port function control register (JPFC0) are JPFC0\_[7:0].

### 2.9.2.8 PFCEn — Port Function Control Expansion Register

**此寄存器和PFCn和PFCAEn一起指定某pin 的alternative function**

This register, together with register PFCn and PFCAEn, specifies an alternative function of the pins.

Some alternative functions directly control the I/O of the Pn\_m pin. For such alternative functions, PIPCn.PIPCn\_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn\_m.

**Access:** This register can be read or written in 16-bit units.

**Address:** PFCEn: <PORTn\_base> + 0600<sub>H</sub> + n × 4 (n = 0, 8, 9, 10, 11, 12)<sup>\*1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCE n_15	PFCE n_14	PFCE n_13	PFCE n_12	PFCE n_11	PFCE n_10	PFCEn _9	PFCEn _8	PFCEn _7	PFCEn _6	PFCEn _5	PFCEn _4	PFCEn _3	PFCEn _2	PFCEn _1	PFCEn _0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.43, Control Registers (P0)**, **Table 2.49, Control Registers (P8)**, **Table 2.51, Control Registers (P9)**, **Table 2.53, Control Registers (P10)**, **Table 2.55, Control Registers (P11)**, and **Table 2.57, Control Registers (P12)**.

**Table 2.25 PFCEn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of the pin. For details, see <b>Table 2.27, Setting Alternative Functions</b> .

### 2.9.2.9 PFCAEn — Port Function Control Additional Expansion Register

**此寄存器和PFCn和PFCEn一起指定某pin 的al terna tive function**

This register selects the alternative peripheral functions together with PFCEn, PFCn registers.

Some alternative functions directly control the I/O of the Pn\_m pin. For such alternative functions, PIPCn.PIPCn\_m must be set to 1 and the I/O is specified by the peripheral function.

For other alternative functions, input/output must be specified by PMn.PMn\_m.

**Access:** This register can be read or written in 16-bit units.

**Address:** PFCAEn: <PORTn\_base> + 0A00H + n × 4 (n = 0, 10, 11)\*1

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAE n_15	PFCAE n_14	PFCAE n_13	PFCAE n_12	PFCAE n_11	PFCAE n_10	PFCAE n_9	PFCAE n_8	PFCAE n_7	PFCAE n_6	PFCAE n_5	PFCAE n_4	PFCAE n_3	PFCAE n_2	PFCAE n_1	PFCAE n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.43, Control Registers (P0)**, **Table 2.53, Control Registers (P10)**, and **Table 2.55, Control Registers (P11)**.

**Table 2.26 PFCAEn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies the alternative function of the pin. For details, see <b>Table 2.27, Setting Alternative Functions</b> .

**Table 2.27 Setting Alternative Functions**

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative input mode 1
			0	Alternative output mode 1
		1	1	Alternative input mode 2
			0	Alternative output mode 2
	1	0	1	Alternative input mode 3
			0	Alternative output mode 3
		1	1	Alternative input mode 4
			0	Alternative output mode 4
1	0	0	1	Alternative input mode 5
			0	Alternative output mode 5
		1	X	Setting prohibited
		1	X	Setting prohibited

**CAUTION**

- After selecting the alternative function by the PFCn\_m, PFCEn\_m, or PFCAEn\_m bit, set the PMCn\_m bit to "1".  
*对于此产品，一些function的I/O被赋予2个或者多个pins，但是特殊的pin function在某一时刻只能赋予一个pin*
- With this product, the I/O of some functions is assigned in two or more pins, but a specific pin function can only be set to one pin at a time. Setting the same pin function to two or more pins at the same time is prohibited.  
For example, if the a/b/c pin is used as b, the b/d/e pin cannot be used as b. In this case, the b/d/e pin must be configured as a pin function other than b.

**NOTE**

For more details on the assignment of each function, see **Sections 2.10.1 to 2.10.14**.

## 2.9.3 Pin Data Input/Output

### 2.9.3.1 PBDCn / APBDCn / JPBDC0 — Port Bidirection Control Register

**此寄存器使能output模式下的input buffer，并且将此port设置为bidirectional mode**

This register enables the input buffer in output mode and sets the port to bidirectional mode. In **bidirectional mode**, in **Pn\_m**上的信号电平可以从PPRn.PPRn\_m上读取

bidirectional mode, the level of the signal on a Pn\_m pin can be read from PPRn.PPRn\_m.

**Access:** PBDCn, APBDCn: These registers can be read or written in 16-bit units.  
JPBDC0: This register can be read or written in 8-bit units.

**Address:** PBDCn: <PORTn\_base> + 4100<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
APBDCn: <PORTn\_base> + 41C8<sub>H</sub> + n × 4 (n = 0, 1)  
JPBDC0: <JPORT0\_base> + 0410<sub>H</sub>\*<sup>1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), Table 2.61, Control Registers (P20), Table 2.63, Control Registers (AP0), and Table 2.65, Control Registers (AP1).**

**Table 2.28 PBDCn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PBDCn[15:0]	Enables/disables bidirectional mode of the corresponding pin. 0: Bidirectional mode disabled 1: Bidirectional mode enabled

#### CAUTION

When the Pn\_m port is used for the **alternative output function** (PMCn.PMCn\_m = 1, PMn.PMn\_m = 0), the level of the Pn\_m pin can be read from PPRn.PPRn\_m by enabling the bidirectional mode (PBDCn.PBDCn\_m = 1).

However, output of that alternative output function is input to the alternative input function of the same pin (the alternative input function set by PFCn.PFCn\_m, PFCEn.PFCEn\_m, and PFCAEn.PFCAEn\_m). If the alternative input function in question is being used by another pin, the alternative input function is not guaranteed.

#### NOTE

The control bits of the JTAG port bidirection control register (JPBDC0) are JPBDC0[7:0].

### 2.9.3.2 PPRn / APPRn / JPPR0 / IPPR0 — Port Pin Read Register

此寄存器反映了Pn\_m pin上的实际电平，不管它是Pn.Pn\_m bit上的值还是处于alternative output function的电平。This register reflects the actual level of the Pn\_m pin, whether it is the value of the Pn.Pn\_m bit or the level of an alternative output function.

**Access:** PPRn, APPRn, IPPR0: These registers are read-only registers that can be read in 16-bit units.  
JPPR0: This register is a read-only register that can be read in 8-bit units.

**Address:** PPRn: <PORTn\_base> + 0200H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
APPRn: <PORTn\_base> + 02C8H + n × 4 (n = 0, 1)  
JPPR0: <JPORT0\_base> + 0020H  
IPPR0: <PORTn\_base> + 02F0H<sup>1</sup>

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPRn_15	PPRn_14	PPRn_13	PPRn_12	PPRn_11	PPRn_10	PPRn_9	PPRn_8	PPRn_7	PPRn_6	PPRn_5	PPRn_4	PPRn_3	PPRn_2	PPRn_1	PPRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (J0)**, **Table 2.43, Control Registers (P0)**, **Table 2.45, Control Registers (P1)**, **Table 2.47, Control Registers (P2)**, **Table 2.49, Control Registers (P8)**, **Table 2.51, Control Registers (P9)**, **Table 2.53, Control Registers (P10)**, **Table 2.55, Control Registers (P11)**, **Table 2.57, Control Registers (P12)**, **Table 2.59, Control Registers (P18)**, **Table 2.61, Control Registers (P20)**, **Table 2.63, Control Registers (AP0)**, **Table 2.65, Control Registers (AP1)**, and **Table 2.67, Control Registers (IP0)**.

Table 2.29 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	The Pn_m pin, Pn.Pn_m value or alternative function output.

#### NOTES

- For the read values of the PPRn register, see **Section 2.7.4, Pin Data Input/Output**.
- The control bits of the JTAG port pin read register (JPPR0) are JPPR0\_[7:0].

### 2.9.3.3 Pn / APn / JP0 — Port Register

**此寄存器保有输出的 Pn.Pn\_m 数据，此数据是由在输出状态下的Pn\_m 输出的**  
 This register holds the Pn.Pn\_m data to be output via the related Pn\_m port in output port mode  
 (PMCn.PMCn\_m = 0 and PMn.PMn\_m = 0).

**Access:** Pn, APn: These registers can be read or written in 16-bit units.  
 JP0: This register can be read or written in 8-bit units.

**Address:** Pn: <PORTn\_base> + 0000<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
 APn: <PORTn\_base> + 00C8<sub>H</sub> + n × 4 (n = 0, 1)  
 JP0: <JP0\_base> + 0000<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pn_15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Value after reset	0	0	R/W													

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), Table 2.61, Control Registers (P20), Table 2.63, Control Registers (AP0), and Table 2.65, Control Registers (AP1).**

Table 2.30 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of the Pn_m pin (m = 0 to 15). 0: Outputs low level 1: Outputs high level

#### NOTE

The control bits of the JTAG port register (JP0) are JP0\_[7:0].

### 2.9.3.4 PNOTn / APNOTn / JPNOT0 — Port NOT Register

此寄存器就是允许port register Pn的Pn\_m bit反转，而不需要重新(直接)写入  
This register allows the Pn\_m bit of the port register Pn to be inverted without directly writing to Pn.

**Access:** PNOTn, APNOTn: These registers are write-only registers that can be written in 16-bit units. When read, 0000<sub>H</sub> is returned.

JPNOT0: This register is a write-only register that can be written in 8-bit units. When read, 00<sub>H</sub> is returned.

**Address:** PNOTn: <PORTn\_base> + 0700<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)

APNOTn: <PORTn\_base> + 07C8<sub>H</sub> + n × 4 (n = 0, 1)

JPNOT0: <JPOR0\_base> + 0070<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PNOTn_15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PNOTn_14	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Value after reset    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

R/W    W    W    W    W    W    W    W    W    W    W    W    W    W    W    W    W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device.  
See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (J0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), Table 2.61, Control Registers (P20), Table 2.63, Control Registers (AP0), and Table 2.65, Control Registers (AP1).**

Table 2.31 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted. 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted (Pn_m → Pn_m)

#### NOTE

The control bits of the JTAG port NOT register are JPNOT0\_[7:0].

### 2.9.3.5 PSRn / APSRn / JPSR0 — Port Set/Reset Register

**此寄存器提供一种alternative方法，将数据写入Pn Register** (注意此处是实际的数据区别于PMCSRn)

This register provides an alternative method to write data to the Pn register.

The upper 16 bits of PSRn act as a mask which specifies whether or not the value Pn.Pn\_m is set by the corresponding bit in the lower 16 bits of PSRn.

**Access:** PSRn, APSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000<sub>H</sub>. Reading bits 15 to 0 returns the value of registers Pn and APn.  
JPSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000<sub>H</sub>. Reading bits 7 to 0 returns the value of register JP0.

**Address:** PSRn: <PORTn\_base> + 0100<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
APSRn: <PORTn\_base> + 01C8<sub>H</sub> + n × 4 (n = 0, 1)  
JPSR0: <JP0\_base> + 0010<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSR n_31	PSR n_30	PSR n_29	PSR n_28	PSR n_27	PSR n_26	PSR n_25	PSR n_24	PSR n_23	PSR n_22	PSR n_21	PSR n_20	PSR n_19	PSR n_18	PSR n_17	PSR n_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSR n_15	PSR n_14	PSR n_13	PSR n_12	PSR n_11	PSR n_10	PSR n_9	PSR n_8	PSR n_7	PSR n_6	PSR n_5	PSR n_4	PSR n_3	PSR n_2	PSR n_1	PSR n_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), Table 2.61, Control Registers (P20), Table 2.63, Control Registers (AP0), and Table 2.65, Control Registers (AP1).**

Table 2.32 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit PSRn_m (PSRn_[15:0]) is written to Pn_m. 0: Pn_m is not affected by PSRn_m. 1: Pn_m is PSRn_m Example: If PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value if the corresponding upper bit (PSRn_[31:16]) PSRn_m is 1. 0: Pn_m = 0 1: Pn_m = 1

#### NOTE

The control bits of the JTAG port set/reset register (JPSR0) are JPSR0\_[31:0].

## 2.9.4 Configuration of Electrical Characteristics

### 2.9.4.1 PUn / JPU0 — Pull-Up Option Register

注意是在input下，  
output上下拉是没用的

This register specifies whether an internal pull-up resistor is connected to an input pin.

**Access:** PUn: This register can be read or written in 16-bit units.  
JPU0: This register can be read or written in 8-bit units.

**Address:** PUn: <PORTn\_base> + 4300<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
JPU0: <JPORT0\_base> + 0430<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), and Table 2.61, Control Registers (P20).**

Table 2.33 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether an internal pull-up resistor is connected to the corresponding pin. 0: No internal pull-up resistor connected 1: An internal pull-up resistor connected

#### NOTES

- If a pin is configured such that both an internal pull-up resistor (PUn.PUn\_m = 1) and pull-down resistor (PDn.PDn\_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected. **如果上拉和下拉同时设置，默认会使用下拉，而上拉不会采用。**
- The pull-up resistor has no effect when the pin is operated in output mode.**
- The control bits of the JTAG pull-up option register (JPU0) are JPU0\_[7:0].

### 2.9.4.2 PDn / JPD0 — Pull-Down Option Register

This register specifies whether to connect an internal pull-down resistor to an input pin.

**Access:** PDn: This register can be read or written in 16-bit units.  
JPD0: This register can be read or written in 8-bit units.

**Address:** PDn: <PORTn\_base> + 4400<sub>H</sub> + n × 4 (n = 0, 8, 9, 10, 11)  
JPD0: <JPORT0\_base> + 0440<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), and Table 2.55, Control Registers (P11)**.

**Table 2.34 PDn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin: 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

#### NOTES

- If a pin is configured such that both an internal pull-up resistor (PUn.PUn\_m = 1) and pull-down resistor (PDn.PDn\_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected. **如果上拉和下拉同时设置，默认会使用下拉，而上拉不会采用。**
- The internal pull-down resistor has no effect when the pin is operated in output mode.
- The control bits of the JTAG pull-down option register (JPD0) are JPD0\_[7:0].

### 2.9.4.3 PDSCn — Port Drive Strength Control Register

使用PPCMDn寄存器 正确的write顺序是必须的，这样才能更新此寄存器

This register specifies the output driver strength of the port pin. This function selects the fast mode (high drive strength) or slow mode (low drive strength) of the output buffer. The correct write sequence using the PPCMDn register is required in order to update this register. For details, see [Section 4](#),

**Write-Protected Registers.** Regarding the alternative functions for which the PDSC register needs to be set, see [Section 2.11.3.3, Output Buffer Control \(PDSC\)](#).

**Access:** This register can be read or written in 32-bit units.

**Address:** PDSCn: <PORTn\_base> + 4600<sub>H</sub> + n × 4 (n = 0, 10, 11, 12)<sup>\*1</sup>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSCn_15	PDSCn_14	PDSCn_13	PDSCn_12	PDSCn_11	PDSCn_10	PDSCn_9	PDSCn_8	PDSCn_7	PDSCn_6	PDSCn_5	PDSCn_4	PDSCn_3	PDSCn_2	PDSCn_1	PDSCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in [Section 2.10, Port \(General I/O\) Function Overview: Table 2.43, Control Registers \(P0\)](#), [Table 2.53, Control Registers \(P10\)](#), [Table 2.55, Control Registers \(P11\)](#), and [Table 2.57, Control Registers \(P12\)](#).

Table 2.35 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDSCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. 0: Lower drive strength (when the frequency output from the pin is 10 MHz or below) 1: High drive strength (when the frequency output from the pin is 40 MHz or less).

### 2.9.4.4 PODCn / JPODC0 — Port Open Drain Control Register

使用PPCMDn和JPPCMD0寄存器 正确的write顺序是必须的，这样才能更新此寄存器

此寄存器选择推挽还是漏极开路作为output buffer功能  
This register selects push-pull or open-drain as output buffer function. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see Section 4, Write-Protected Registers.

**Access:** PODCn, JPODC0: These registers can be read or written in 32-bit units.

**Address:** PODCn: <PORTn\_base> + 4500<sub>H</sub> + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)  
JPODC0: <JPOR0\_base> + 0450<sub>H</sub><sup>\*1</sup>

**Value after reset:** 0000 0000<sub>H</sub><sup>\*2</sup>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 <sup>*3</sup>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device.  
See the following tables in Section 2.10, Port (General I/O) Function Overview: Table 2.41, Control Registers (JP0), Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.49, Control Registers (P8), Table 2.51, Control Registers (P9), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), Table 2.59, Control Registers (P18), and Table 2.61, Control Registers (P20).

Note 2. The PODC0 register is 0000 0001<sub>H</sub>.

Note 3. The PODC0\_0 bit is 1.

Table 2.36 PODCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PODCn_[15:0]	Specifies the output buffer function. 0: Push-pull 推挽 1: Open-drain 漏极开路 (开漏)

#### NOTE

The control bits of the JTAG port open drain control register (JPODC0) are JPODC0\_[15:0].

### 2.9.4.5 PISn — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

**Access:** This register can be read or written in 16-bit units.

**Address:** PISn: <PORTn\_base> + 4700<sub>H</sub> + n × 4 (n = 0, 1, 2, 10, 11, 12, 20)<sup>\*1</sup>

**Value after reset:** FFFF<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PISn_15	PISn_14	PISn_13	PISn_12	PISn_11	PISn_10	PISn_9	PISn_8	PISn_7	PISn_6	PISn_5	PISn_4	PISn_3	PISn_2	PISn_1	PISn_0	
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview: Table 2.43, Control Registers (P0), Table 2.45, Control Registers (P1), Table 2.47, Control Registers (P2), Table 2.53, Control Registers (P10), Table 2.55, Control Registers (P11), Table 2.57, Control Registers (P12), and Table 2.61, Control Registers (P20)**.

**Table 2.37 PISn Register Contents**

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer characteristics: 0: Type 1 (SHMT1) 1: Type 2 (SHMT4)
<b>NOTE</b>		
Details of the definition of type 1 and type 2 are given in <b>Section 2.11.3.2, Input Buffer Control (PISn, JPISA0)</b> . Also see the data sheet for input buffer characteristics.		

### 2.9.4.6 JPISA0 — Port Input Buffer Selection Advanced Register

This register specifies the input buffer characteristics.

**Access:** This register can be read or written in 8-bit units.

**Address:** JPISA0: <JPORT0\_base> + 04A0<sub>H</sub><sup>\*1</sup>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	JPISA0_3	JPISA0_2	—	JPISA0_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device.  
See the following tables in **Section 2.10, Port (General I/O) Function Overview : Table 2.41, Control Registers (JP0)**.

**Table 2.38 JPISA0 Register Contents**

Bit Position	Bit Name	Function
7 to 4, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2, 0	JPISA0_[3, 2, 0]	Specifies the input buffer characteristics: 0: Type 2 (SHMT4) 1: Type 5 (TTL)

#### NOTE

Details of the definition of type 2 and type 5 are given in **Section 2.11.3.2, Input Buffer Control (PISn, JPISA0)**. Also see the data sheet for input buffer characteristics.

### 2.9.5 Port Register Protection

RH850/F1L has Port Protection Command Registers (PPCMDn) and Port Protection Status Registers (PPROTSn) which implement the Port Protection Cluster Function. For details on the registers, see **Section 4, Write-Protected Registers**.

## 2.9.6 Flowchart Example for Port Settings

Examples of the port settings are shown in the flowchart below.

### CAUTION

If the port is set to the PIPCn.PIPCn\_m bit = 0 and alternative output mode, the port might briefly enter alternative input mode. This will occur between when the PMCn.PMCn\_m bit is set to 1 and when the PMn.PMn\_m bit is set to 0. If an interrupt-related signal is specified as an alternate function of the port, the mode will temporarily become the alternative input mode, so either disable the interrupt in question, or specify the interrupt is ignored.

### 2.9.6.1 Batch Setting 批量配置

An example of specifying batch port group settings is shown in the flowchart below.

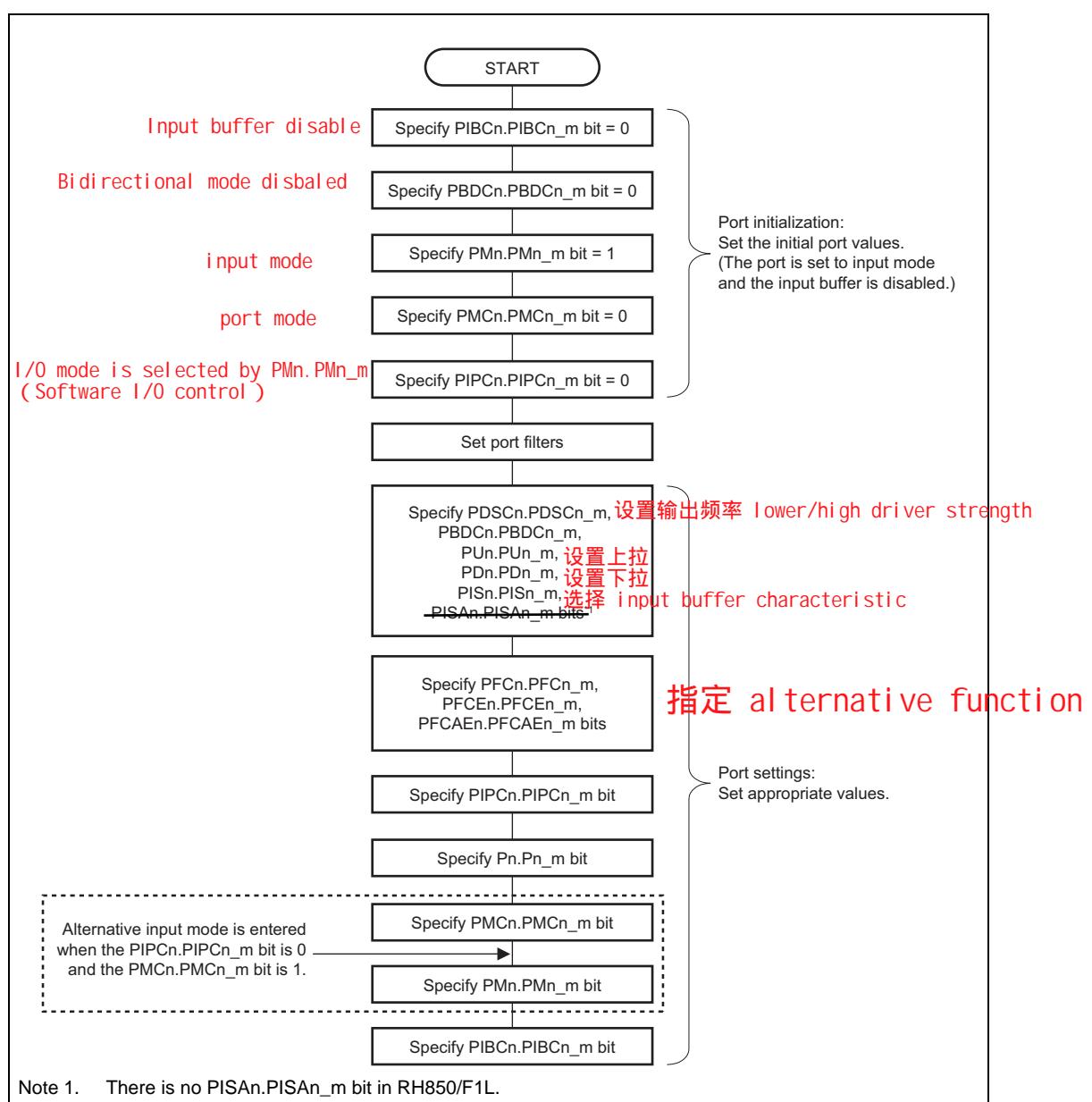


Figure 2.8 Example of Port Settings (When Specified in Batch)

### 2.9.6.2 Individual Settings 单个配置

An example of specifying individual port settings is shown in the flowchart below.

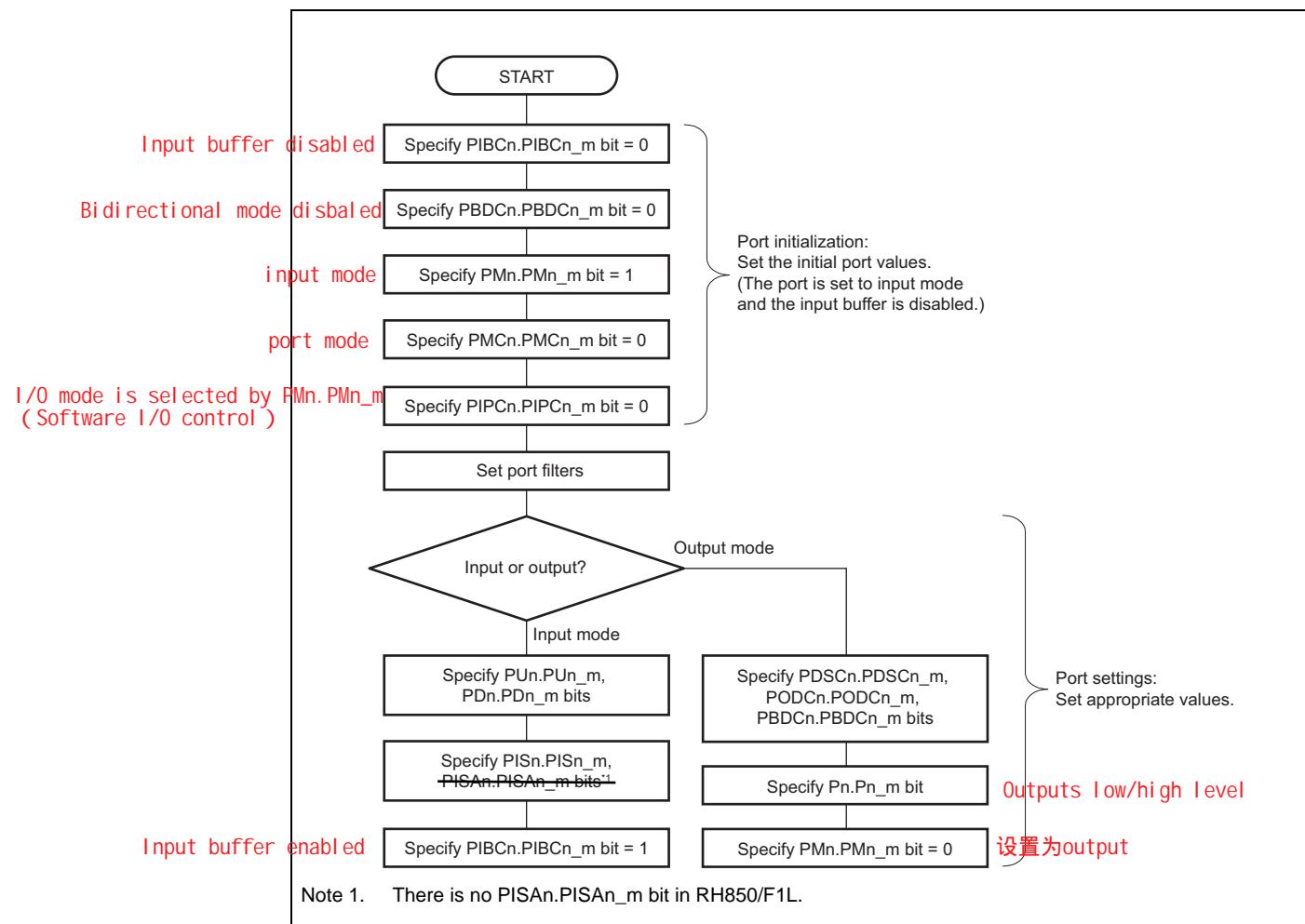


Figure 2.9 Example of Port Settings (in Port Mode)

I/O of Port  
 (a) With IP control 使用IP控制

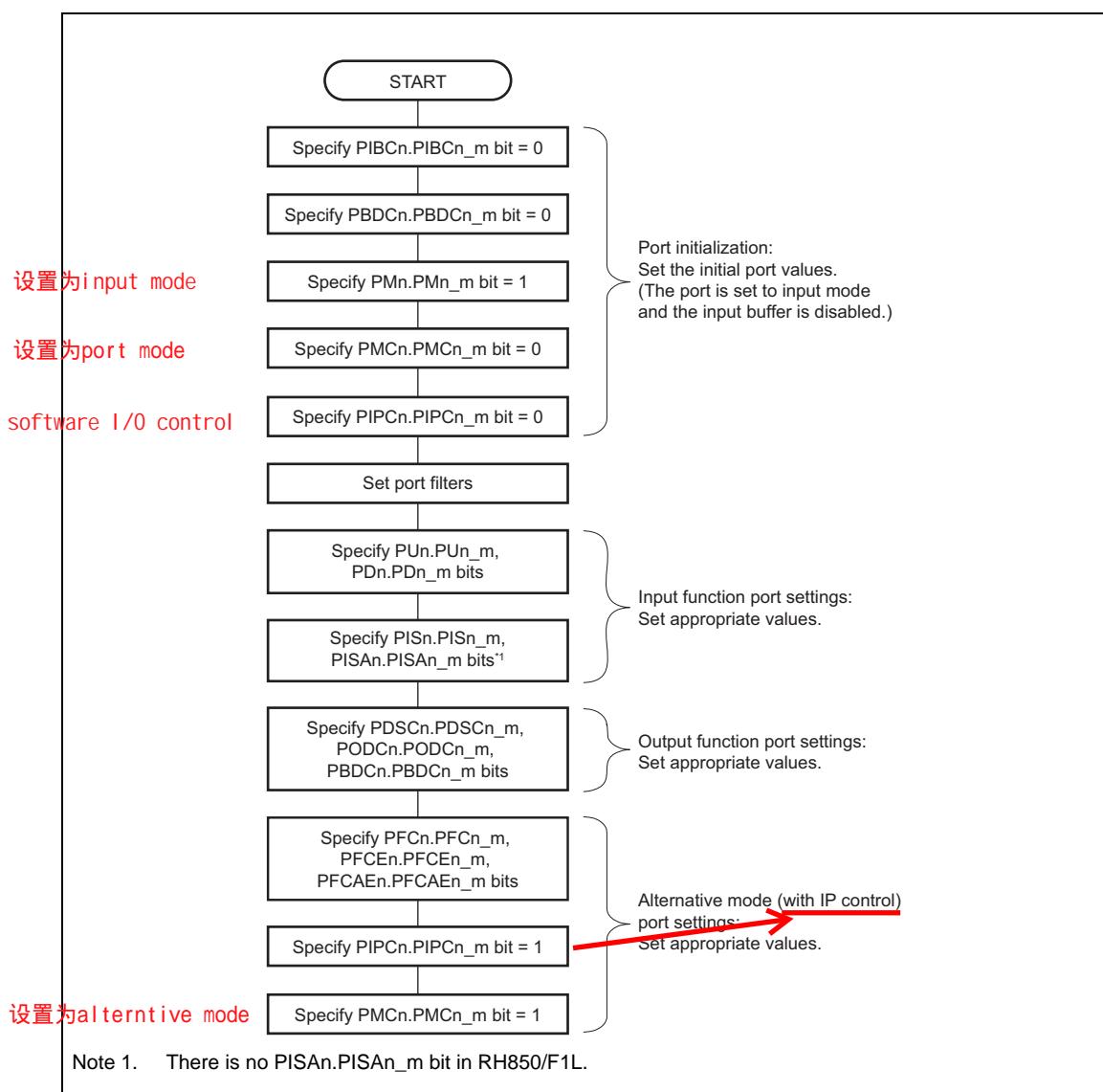


Figure 2.10 Example of Port Settings (in Alternative Mode) (1/2)

(b) Without IP control 不使用IP控制

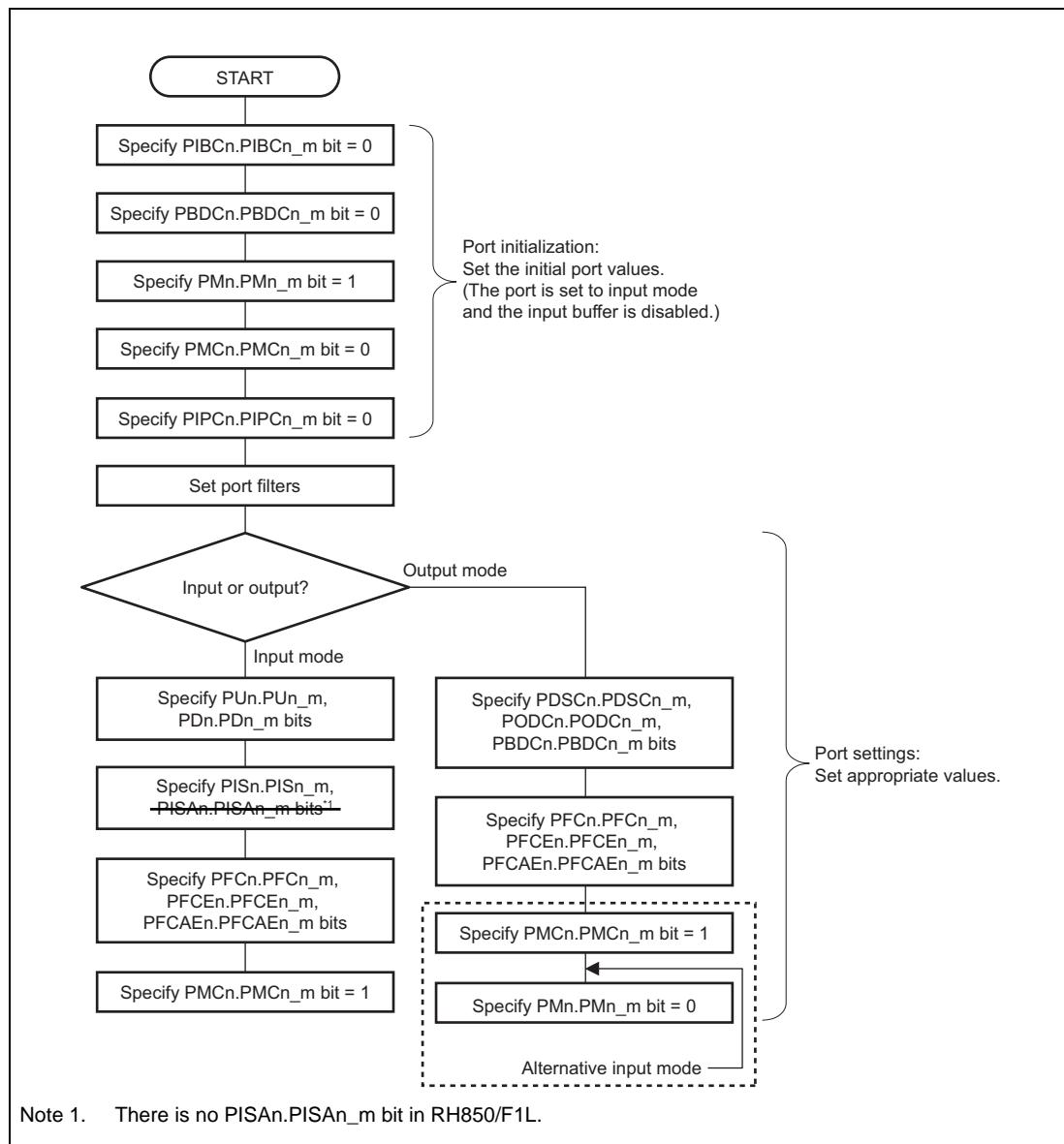


Figure 2.11 Example of Port Settings (in Alternative Mode) (2/2)

## 2.10 Port (General I/O) Function Overview

This section explains the port (general I/O) functions and all the functions assigned to the ports. See the following pages for details.

In addition, whether the port mode is alternative mode or not can be selected by PMCn register setting.

When PMCn.PMCn\_m = 1, alternative functions are selected by the PFCn, PFCEn, and PFCAEn registers.

Table 2.39 Port Function

Port	Pin Name	Size	Direction	Power Domain	Special Alternative Function	Device					
						48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
JTAG Port 0	JP0_0 - 5	6 bits	In/Out	AWO	JTAG, LPD	√	√	√	√	—	—
	JP0_0 - 6	7 bits				—	—	—	—	√	√
Port 0	P0_0 - 3	4 bits	In/Out	AWO	RESETOUT	√	—	—	—	—	—
	P0_0 - 6	7 bits				—	√	—	—	—	—
	P0_0 - 12	13 bits				—	—	√	—	—	—
	P0_0 - 14	15 bits				—	—	—	√	√	√
Port 1	P1_0 - 11	12 bits	In/Out	AWO		—	—	—	—	√	—
	P1_0 - 15	16 bits				—	—	—	—	—	√
Port 2	P2_0 - 6	7 bits	In/Out	AWO		—	—	—	—	—	√
Port 8	P8_0 - 1	2 bits	In/Out	AWO	ADCA0 (10-bit resolution)	√	—	—	—	—	—
	P8_0 - 6	7 bits				—	√	√	—	—	—
	P8_0 - 12	13 bits				—	—	—	√	√	√
Port 9	P9_0 - 1	2 bits	In/Out	ISO	ADCA0 (10-bit resolution)	√	—	—	—	—	—
	P9_0 - 3	4 bits				—	√	—	—	—	—
	P9_0 - 6	7 bits				—	—	√	√	√	√
Port 10	P10_0 - 10	11 bits	In/Out	ISO		√	—	—	—	—	—
	P10_0 - 14	15 bits				—	√	—	—	—	—
	P10_0 - 15	16 bits				—	—	√	√	√	√
Port 11	P11_0 - 4	5 bits	In/Out	ISO		—	—	√	—	—	—
	P11_0 - 7	8 bits				—	—	—	√	—	—
	P11_0 - 15	16 bits				—	—	—	—	√	√
Port 12	P12_0 - 2	3 bits	In/Out	ISO		—	—	—	—	√	—
	P12_0 - 5	6 bits				—	—	—	—	—	√
Port 18	P18_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	—	—	—	—	√	—
	P18_0 - 7	8 bits				—	—	—	—	—	√
Port 20	P20_4 - 5	2 bits	In/Out	ISO		—	—	—	—	√	—
	P20_0 - 5	6 bits				—	—	—	—	—	√
Analog Port 0	AP0_0 - 7	8 bits	In/Out	AWO	ADCA0 (12/10-bit resolution)	√	—	—	—	—	—
	AP0_0 - 9	10 bits				—	√	—	—	—	—
	AP0_0 - 10	11 bits				—	—	√	—	—	—
	AP0_0 - 15	16 bits				—	—	—	√	√	√
Analog Port 1	AP1_0 - 7	8 bits	In/Out	ISO	ADCA1 (12/10-bit resolution)	—	—	—	—	√	—
	AP1_0 - 15	16 bits				—	—	—	—	—	√
Input Port 0	IP0_0	1 bit	In	AWO	SOSC (XT2 pin)	—	—	—	—	√	√

## 2.10.1 JTAG Port 0 (JP0)

### 2.10.1.1 Alternative Function

The following alternative functions are available when JTAG port 0 is configured as a general-purpose I/O port by setting OPTJTAG[1:0] on the corresponding option byte to 00<sub>B</sub>.

**Table 2.40 JTAG Port 0 (JP0)**

Port Mode (JPMCO_m = 0)	Alternative Mode (JPMCO_m = 1)								Device	
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	Special Function
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
JP0_0	INTP0									DOUTDI/LPDI/ LPDIO
JP0_1	INTP1		TAUJ010	TAUJ000						DOUTDO/LPDO
JP0_2	INTP2		TAUJ011	TAUJ001						DOUTCK/ LPDCLK
JP0_3	INTP3	CSCXFOUT	TAUJ012	TAUJ002						DOUTMS
JP0_4										DOUTRST
JP0_5	NMI		TAUJ013	TAUJ003						DOURDY/ LPDCLKOUT
NMI	RTCAA0OUT	TAUJ013	TAUJ003							DOURDY/ LPDCLKOUT
JP0_6										EVT0

Note 1. Available in devices except for ones with 2-MB code flash memory.

Note 2. Only available in devices with 2-MB code flash memory.

### CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

## 2.10.1.2 Control Registers

Table 2.41 Control Registers (JP0)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins
JP0	JTAG port register 0	8	5-0	R/W	0000H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPSR0	JTAG port set/reset register 0	32	21-16, 5-0	R/W	0010H	0000 0000H	✓	✓	✓	—
			22-16, 6-0				—	—	—	✓
JPPR0	JTAG port pin read register 0	8	5-0	R	0020H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPM0	JTAG port mode register 0	8	5-0	R/W	0030H	FFH	✓	✓	✓	—
			6-0				—	—	—	✓
JPMC0	JTAG port mode control register 0	8	5, 3-0	R/W	0040H	00H	✓	✓	✓	✓
JPFC0	JTAG port function control register 0	8	5, 3-1	R/W	0050H	00H	✓	✓	✓	✓
JPNOT0	JTAG port NOT register 0	8	5-0	W	0070H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPMR0	JTAG port mode set/reset register 0	32	21-16, 5-0	R/W	0080H	0000 FFFFH	✓	✓	✓	—
			22-16, 6-0				—	—	—	✓
JPMCSR0	JTAG port mode control set/reset register 0	32	21, 19-16, 5, 3-0	R/W	0090H	0000 0000H	✓	✓	✓	✓
JPIBC0	JTAG port input buffer control register 0	8	5-0	R/W	0400H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPBDC0	JTAG port bidirection control register 0	8	5-0	R/W	0410H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPU0	Pull-up option register 0	8	5-0	R/W	0430H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPD0	Pull-down option register 0	8	5-0	R/W	0440H	00H	✓	✓	✓	—
			6-0				—	—	—	✓
JPODC0	JTAG port open drain control register 0	32	5-0	R/W	0450H	0000 0000H	✓	✓	✓	—
			6-0				—	—	—	✓
JPISA0	JTAG port input buffer selection advanced register 0	8	3, 2, 0	R/W	04A0H	00H	✓	✓	✓	✓
JPPRTS0	JTAG port protection status register 0	32	0	R	04B0H	0000 0000H	✓	✓	✓	✓
JPPCMD0	JTAG port protection command register 0	32	7-0	W	04C0H	xxxx xx00H	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.2 Port 0 (P0)

### 2.10.2.1 Alternative Functions

Table 2.42 Port 0 (P0) (1/2)

Alternative Mode (PMCO_m = 1)										Device							
Port Mode (PMCO_m = 0)	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P0_0	TAUD012	RAUD002	RLIN20RX	CAN0TX	PWGA100	CSIH0SSI	DPO				RESETOUT	✓	✓	✓	✓	✓	✓
P0_1	TAUD014	TAUD004	CAN0RX/ INTP0	RLIN20TX	INTP0	PWGA110	CSIH0SI	APO				✓	✓	✓	✓	✓	✓
P0_2	TAUD016	TAUD006		RLIN30TX		PWGA120	CSIH0SC	INTP1	DPO			✓	—	—	—	—	—
P0_3	TAUD018	TAUD008	CAN1RX/ INTP1	RLIN30TX		PWGA120	CSIH0SC	INTP1	DPO			—	✓	✓	✓	✓	✓
P0_4	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA100		SELDP0	DPIN8					✓	—	✓	✓	✓	✓
P0_5	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA100	CSIH1SI	SELDP0	DPIN8					—	—	✓	✓	✓	✓
P0_6	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1								—	✓	—	—	—	—
P0_7	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1		CSIH1SO						—	—	✓	✓	✓	✓
P0_8	INTP2		DPIN10	SELDP2								—	✓	—	—	—	—
	INTP2		DPIN10	SELDP2		CSIH1SC						—	—	✓	✓	✓	✓
	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RY1	CSIH1RYO						—	—	✓	—	—	—
	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RY1	CSIH1RYO	TAUB010	TAUB000	INTP3			—	—	*1	—	—	—
	RLIN21TX		DPIN6	CSCXFOUT	CSIH1RY1	CSIH1RYO	TAUB010	TAUB000	CAN3RX/ INTP3			—	—	*2	✓	✓	✓
	RLIN21TX		DPIN6									—	✓	—	—	—	—
	RLIN21TX		DPIN6									—	—	*1	—	—	—
	RLIN21TX		DPIN6									—	—	*2	✓	✓	✓

Table 2.42 Port 0 (P0) (2/2)

Port Mode (PMC0_m = 0)	Alternative Mode (PMC0_m = 1)										Device
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function	
P0_9	INTP12	CSIH1CSS0	DPIN7								—
	INTP12	CSIH1CSS0	DPIN7	RLIN22RX		TAUB014	TAUB004	INTP4		—	*1
	INTP12	CSIH1CSS0	DPIN7	RLIN22RX		TAUB014	TAUB004	CAN4RX/ INTP4		—	*2
P0_10	INTP3	CSIH1CSS1	DPIN11								—
	INTP3	CSIH1CSS1	DPIN11	RLIN22TX	TAUB016	TAUB006				—	*3
	INTP3	CSIH1CSS1	DPIN11	RLIN22TX	TAUB016	TAUB006	CAN4TX			—	*4
P0_11	RIC0SDA										—
	RIC0SDA										—
	RIC0SDA										—
P0_12	RIC0SCL										—
	RIC0SCL										—
	RIC0SCL										—
P0_13	RLIN32RX/ INTP12		INTP12	PWGA450	TAUB0110	TAUB010	CSIG0SI	RLIN26TX		—	—
	RLIN32RX/ INTP12		INTP12	PWGA450	TAUB0110	TAUB010	CSIG0SI	RLIN26TX		—	—
	RLIN32TX/ INTP12		INTP12	PWGA460	TAUB0112	TAUB012	CSIG0SO	INTP5		—	—
P0_14	RLIN32TX		INTP12	PWGA460	TAUB0112	TAUB012	CSIG0SO	CAN5RX/ INTP5		—	*1
	RLIN32TX		INTP12	PWGA470	TAUB0114	TAUB014	CSIG0SC			—	*2
	RLIN32TX		INTP12	PWGA470	TAUB0114	TAUB014	CSIG0SC	CAN5TX		—	*4

Note 1. Available except in F1L for Gateway.

Note 2. Only available in F1L for Gateway.

Note 3. Available in devices except for ones with 1.5- and 2-MB code flash memories.

Note 4. Only available in devices with 1.5- and 2-MB code flash memories.

**CAUTION**

The behavior and performance are not guaranteed when alternative functions are not assigned to the register. When the RESETOUT function is selected for the P0\_0 pin, the P0\_0 pin outputs a low-level as RESETOUT signal while a reset is asserted and continues to output low level after the reset is released. For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

## 2.10.2.2 Control Registers

Table 2.43 Control Registers (P0) (1/3)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*5</sup>			48 pins	64 pins	80 pins	100 pins
P0	Port register 0	16	3-0 6-0 12-0 14-0	R/W	0000 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PSR0	Port set/reset register 0	32	19-16, 3-0 22-16, 6-0 28-16, 12-0 30-16, 14-0	R/W	0100 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	✓	—
PPR0	Port pin read register 0	16	3-0 6-0 12-0 14-0	R	0200 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PM0	Port mode register 0	16	3-0 6-0 12-0 14-0	R/W	0300 <sub>H</sub>	FFFE <sub>H</sub>	✓	—	—	—
PMC0	Port mode control register 0	16	3-0 6-0 12-0 14-0	R/W	0400 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PFC0	Port function control register 0	16	3-0 6-0 11-0 14-0	R/W	0500 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PFCE0	Port function control expansion register 0	16	3-0 4-0 8-0 14-0	R/W	0600 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PNOT0	Port NOT register 0	16	3-0 6-0 12-0 14-0	W	0700 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—

Table 2.43 Control Registers (P0) (2/3)

Register	Function	Register Size	Effective Bit Position	R/W <sup>5</sup>	Offset Address	Value after Reset	Device			
							48 pins	64 pins	80 pins	100 pins
PMSSR0	Port mode set/reset register 0	32	19-16, 3-0	R/W	0800 <sub>H</sub>	0000 FFFF <sub>H</sub>	✓	—	—	—
			22-16, 6-0				—	✓	—	—
			28-16, 12-0				—	—	✓	—
			30-16, 14-0				—	—	✓	✓
PMCSR0	Port mode control set/reset register 0	32	19-16, 3-0	R/W	0900 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	—
			22-16, 6-0				—	✓	—	—
			28-16, 12-0				—	—	✓	—
			30-16, 14-0				—	—	✓	✓
PFCAE0	Port function control additional expansion register 0	16	3, 2	R/W	0A00 <sub>H</sub>	0000 <sub>H</sub>	✓	✓	—	—
			13, 9, 7, 3, 2				—	—	*1	—
			13, 9, 7, 3, 2				—	—	—	*3
			14, 13, 10-7, 3, 2				—	—	—	*4
PIBC0	Port input buffer control register 0	16	3-0	R/W	4000 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	✓	✓
PBDC0	Port bidirection control register 0	16	3-0	R/W	4100 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	✓	✓
PIPC0	Port I/P control register 0	16	3, 2	R/W	4200 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			6, 5, 3, 2				—	✓	—	—
			14, 13, 6, 5, 3, 2				—	—	✓	—
			14-0				—	—	✓	✓
PU0	Pull-up option register 0	16	3-0	R/W	4300 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	✓	✓
PD0	Pull-down option register 0	16	3-0	R/W	4400 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	✓
			14-0				—	—	✓	✓
PODC0	Port open drain control register 0	32	3-0	R/W	4500 <sub>H</sub>	0000 0001 <sub>H</sub>	✓	—	—	—
			6-0				—	✓	—	—
			12-0				—	—	✓	—
			14-0				—	—	✓	✓

Table 2.43 Control Registers (P0) (3/3)

Register	Function	Register Size	Effective Bit Position	R/W <sup>5</sup>	Offset Address	Value after Reset	Device			
							48 pins	64 pins	80 pins	100 pins
PDSC0	Port drive strength control register 0	32	3, 2	R/W	4600 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	✓	—	—
			7-5, 3, 2				—	—	✓	—
		16	14, 13, 7-5, 3, 2				—	—	✓	✓
			3, 1, 0	R/W	4700 <sub>H</sub>	FFFF <sub>H</sub>	✓	—	—	—
PIS0	Port input buffer selection register 0	16	6-0				—	✓	—	—
			12, 11, 7-0				—	—	✓	—
		32	13-11, 9, 7-0				—	—	✓	✓
			0	R	4B00 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	✓	✓	✓
PPROT0	Port protection status register 0	32	7-0	W			✓	✓	✓	✓
	Port protection command register 0		32				✓	✓	✓	✓

Note 1. Available except in F1L for Gateway.

Note 2. Only available in F1L for Gateway.

Note 3. Available in devices except for ones with 1.5- and 2-MB code flash memories.

Note 4. Only available in devices with 1.5- and 2-MB code flash memories.

Note 5. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.3 Port 1 (P1)

### 2.10.3.1 Alternative Functions

Table 2.44 Port 1 (P1)

Port Mode (PMC1_m = 0)	Alternative Mode (PMC1_m = 1)								Device
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
Input	Output	Input	Output	Input	Output	Input	Output	Special Function	
P1_0	RLIN33RX/ INTP13		INTP13						—
P1_1		RLIN33TX							—
P1_2	CAN3RX/ INTP3	INTP3							—
P1_3		CAN3TX	DPIN23						—
P1_4	RLIN35RX/ INTP15		INTP15						—
P1_5	ADCA1TRG0	RLIN35TX	DPIN17						—
P1_6	RLIN25RX		DPIN18						—
P1_7	ADCA1TRG1	RLIN25TX	DPIN19						—
P1_8	RLIN34RX/ INTP14		INTP14						—
P1_9		RLIN34TX	DPIN20						—
P1_10	RLIN24RX		DPIN21						—
P1_11	ADCA1TRG2	RLIN24TX	DPIN22						—
P1_12	CAN4RX/ INTP4		INTP4						—
P1_13		CAN4TX							—
P1_14	RLIN23RX								—
P1_15		RLIN23TX							—

### CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

## 2.10.3.2 Control Registers

Table 2.45 Control Registers (P1)

Register	Function			Effective Bit		Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>	48 pins			64 pins	80 pins	100 pins	144 pins
P1	Port register 1	16	11-0	R/W	0004 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PSR1	Port set/reset register 1	32	27-16, 11-0	R/W	0104 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓
			31-16, 15-0				—	—	—	—	✓
PPR1	Port pin read register 1	16	11-0	R	0204 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PM1	Port mode register 1	16	11-0	R/W	0304 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PMC1	Port mode control register 1	16	11-0	R/W	0404 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PFC1	Port function control register 1	16	11-2, 0	R/W	0504 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			12-2, 0				—	—	—	—	✓
PNOT1	Port NOT register 1	16	11-0	W	0704 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PMSR1	Port mode set/reset register 1	32	27-16, 11-0	R/W	0804 <sub>H</sub>	0000 FFFF <sub>H</sub>	—	—	—	—	✓
			31-16, 15-0				—	—	—	—	✓
PMCSR1	Port mode control set/reset register 1	32	27-16, 11-0	R/W	0904 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓
			31-16, 15-0				—	—	—	—	✓
PIBC1	Port input buffer control register 1	16	11-0	R/W	4004 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PBDC1	Port bidirection control register 1	16	11-0	R/W	4104 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PU1	Pull-up option register 1	16	11-0	R/W	4304 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PODC1	Port open drain control register 1	32	11-0	R/W	4504 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓
			15-0				—	—	—	—	✓
PIS1	Port input buffer selection register 1	16	10, 8, 6, 4, 2, 0	R/W	4704 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—	✓
			14, 12, 10, 8, 6, 4, 2, 0				—	—	—	—	✓
PPROTS1	Port protection status register 1	32	0	R	4B04 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓
PPCMD1	Port protection command register 1	32	7-0	W	4C04 <sub>H</sub>	xxxx xx00 <sub>H</sub>	—	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.4 Port 2 (P2)

### 2.10.4.1 Alternative Functions

Table 2.46 Port 2 (P2)

Port Mode (PMC2_m=0)	Alternative Mode (PMC2_m=1)								Device							
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
Input	Output	Input	Output	Input	Output	Input	Output	Input								
P2_0	RLIN27RX										—	—	—	—	—	—
P2_1	RLIN27TX										—	—	—	—	—	—
P2_2	RLIN28RX										—	—	—	—	—	—
P2_3	RLIN28TX										—	—	—	—	—	—
P2_4	RLIN29RX										—	—	—	—	—	—
P2_5	RLIN29TX										—	—	—	—	—	—
P2_6											—	—	—	—	—	—

**CAUTION**

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

## 2.10.4.2 Control Registers

Table 2.47 Control Registers (P2)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins
P2	Port register 2	16	6-0	R/W	0008 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PSR2	Port set/reset register 2	32	22-16, 6-0	R/W	0108 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PPR2	Port pin read register 2	16	6-0	R	0208 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PM2	Port mode register 2	16	6-0	R/W	0308 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—
PMC2	Port mode control register 2	16	5-0	R/W	0408 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PNOT2	Port NOT register 2	16	6-0	W	0708 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PMSR2	Port mode set/reset register 2	32	22-16, 6-0	R/W	0808 <sub>H</sub>	0000 FFFF <sub>H</sub>	—	—	—	—
PMCSR2	Port mode control set/reset register 2	32	21-16, 5-0	R/W	0908 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PIBC2	Port input buffer control register 2	16	6-0	R/W	4008 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PBDC2	Port bidirection control register 2	16	6-0	R/W	4108 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PU2	Pull-up option register 2	16	6-0	R/W	4308 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PODC2	Port open drain control register 2	32	6-0	R/W	4508 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PIS2	Port input buffer selection register 2	16	4, 2, 0	R/W	4708 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—
PPROTS2	Port protection status register 2	32	0	R	4B08 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PPCMD2	Port protection command register 2	32	7-0	W	4C08 <sub>H</sub>	xxxx xx00 <sub>H</sub>	—	—	—	—

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.5 Port 8 (P8)

### 2.10.5.1 Alternative Functions

Table 2.48 Port 8 (P8)

Port Mode (PMC8_m = 0)	Alternative Mode (PMC8_m = 1)								Device	
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Special Function
P8_0	TAUJ010	TAUJ000	DPIN2	INTP4	CSIH0CSS0					ADCA010S
	TAUJ010	TAUJ000	DPIN2	PWGA140	INTP4	CSIH0CSS0				ADCA010S
P8_1	TAPA0E50	TAUJ001	DPIN0	INTP5						ADCA011S
	TAPA0E50	TAUJ001	DPIN0	PWGA150	INTP5					ADCA011S
P8_2	TAUJ010	TAUJ001	DPIN0	PWGA150	INTP5	CSIH1CSS3				ADCA011S
	TAUJ010	TAUJ000	DPIN2	CSIH0CSS0	PWGA220	PWGA220				ADCA014S
P8_3	TAUJ010	TAUJ000	DPIN2	CSIH0CSS0	INTP6	PWGA220				ADCA014S
	TAUJ011	TAUJ001	DPIN3	CSIH0CSS1	PWGA230	PWGA230				ADCA015S
P8_4	TAUJ011	TAUJ001	DPIN3	CSIH0CSS1	INTP7	PWGA230				ADCA015S
	TAUJ012	TAUJ002	DPIN4	CSIH0CSS2						ADCA016S
P8_5	TAUJ012	TAUJ002	DPIN4	CSIH0CSS2	INTP8	PWGA360				ADCA016S
	TAUJ013	TAUJ003		CSIH0CSS3						ADCA016S
P8_6	TAUJ013	TAUJ003		CSIH0CSS3	PWGA370	PWGA370				ADCA017S
	TAUJ013	TAUJ003		CSIH0CSS3	INTP9	PWGA370				ADCA017S
NMI	NMI	CSIH0CSS4								ADCA018S
	NMI	CSIH0CSS4	PWGA380							ADCA018S
P8_7		CSIH0CSS4	PWGA380	RTCA0OUT						ADCA018S
P8_8		CSIH3CSS0	PWGA390							ADCA014S
P8_9		CSIH3CSS1	PWGA400							ADCA015S
P8_10		CSIH3CSS2	PWGA410							ADCA016S
P8_11		CSIH3CSS3	PWGA420							ADCA017S
P8_12		TAUJ112	DPIN15	PWGA430	CSIH1CSS4					ADCA018S
		TAUJ113	DPIN16	PWGA440	CSIH1CSS5					ADCA019S

### CAUTIONS

- The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
- Use special functions with their initial settings.

## 2.10.5.2 Control Registers

Table 2.49 Control Registers (P8) (1/2)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device					
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P8	Port register 8	16	1, 0 6-0 12-0	R/W	0020 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—	—	—
PSR8	Port set/reset register 8	32	17, 16, 1, 0 22-16, 6-0 28-16, 12-0	R/W	0120 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	✓	—	—	—
PPR8	Port pin read register 8	16	1, 0 6-0 12-0	R	0220 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	✓	✓	✓
PM8	Port mode register 8	16	1, 0 6-0 12-0	R/W	0320 <sub>H</sub>	FFFF <sub>H</sub>	✓	—	—	—	—	—
PMC8	Port mode control register 8	16	1, 0 6-0 12-0	R/W	0420 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	✓	✓	✓
PFC8	Port function control register 8	16	1, 0 5-0 12-0	R/W	0520 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	✓	—	—
PFCE8	Port function control expansion register 8	16	1, 0 3-0 4-0 12, 11, 5-0 12, 11, 6-0	R/W	0620 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	✓	—	—
PNOT8	Port NOT register 8	16	1, 0 6-0 12-0	W	0720 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—	—	—
PMCSR8	Port mode set/reset register 8	32	17, 16, 1, 0 22-16, 6-0 28-16, 12-0	R/W	0820 <sub>H</sub>	0000 FFFF <sub>H</sub>	✓	—	—	✓	—	—
PMCSR8	Port mode control set/reset register 8	32	17, 16, 1, 0 22-16, 6-0 28-16, 12-0	R/W	0920 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	✓	—	—

Table 2.49 Control Registers (P8) (2/2)

Register	Function	Register Size		Effective Bit Position		R/W <sup>1</sup>	Offset Address	Value after Reset	Device			
		Position	48 pins	64 pins	80 pins				120	144	176	pins
PIBC8	Port input buffer control register 8	16	1, 0 6-0	R/W 4020 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—	—	—	—
PBDC8	Port bidirection control register 8	16	1, 0 6-0	R/W 4120 <sub>H</sub>	0000 <sub>H</sub>	✓	—	✓	—	—	—	✓
PU8	Pull-up option register 8	16	1, 0 6-0	R/W 4320 <sub>H</sub>	0000 <sub>H</sub>	✓	—	✓	—	—	—	✓
PD8	Pull-down option register 8	16	1, 0 6-0	R/W 4420 <sub>H</sub>	0000 <sub>H</sub>	✓	—	✓	—	—	—	✓
PODC8	Port open drain control register 8	32	1, 0 6-0	R/W 4520 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	✓	✓	✓	✓	✓
PPROTS8	Port protection status register 8	32	0 12-0	R 4B20 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	✓	✓	✓	✓
PPCMD8	Port protection command register 8	32	7-0	W 4C20 <sub>H</sub>	xxxx xx00 <sub>H</sub>	✓	✓	✓	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.6 Port 9 (P9)

### 2.10.6.1 Alternative Functions

Table 2.50 Port 9 (P9)

Port Mode (PMC9_m = 0)	Alternative Mode (PMC9_m = 1)								Device	
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P9_0	NMI	PWGA8O	TAUD010	TAUD000	ADCA01TRG0	KR014			ADCA012S	✓
	NMI	PWGA8O	TAUD010	TAUD000	ADCA01TRG0	CSIH2CSS0	KR014		ADCA012S	—
P9_1	INTP11	PWGA9O	TAUD012	TAUD002	KR015				ADCA013S	✓
	INTP11	PWGA9O	TAUD012	TAUD002	KR015	CSIH2CSS1			ADCA013S	—
P9_2	KR016	PWGA200	TAPA0ES0						ADCA019S	—
	KR016	PWGA200	TAPA0ES0	CSIH2CSS2					ADCA019S	—
P9_3	KR017	PWGA210							ADCA0110S	—
	KR017	PWGA210		CSIH2CSS3					ADCA0110S	—
P9_4		CSIH0CSS5							ADCA0111S	—
		CSIH0CSS5		PWGA330	TAUJ110	TAUJ100			ADCA0111S	—
P9_5		CSIH0CSS6							ADCA0112S	—
		CSIH0CSS6							ADCA0112S	—
P9_6		CSIH0CSS7		PWGA340	TAUJ111	TAUJ101			ADCA0113S	—
		CSIH0CSS7							ADCA0113S	—
				PWGA350					ADCA0113S	—

### CAUTIONS

- 1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
- 2. Use special functions with their initial settings.

## 2.10.6.2 Control Registers

Table 2.51 Control Registers (P9) (1/2)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins
P9	Port register 9	16	1, 0 3-0 6-0	R/W	0024 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PSR9	Port set/reset register 9	32	17, 16, 1, 0 19-16, 3-0 22-16, 6-0	R/W	0124 <sub>H</sub>	00000000 <sub>H</sub>	✓	—	✓	✓
PPR9	Port pin read register 9	16	1, 0 3-0 6-0	R	0224 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PM9	Port mode register 9	16	1, 0 3-0 6-0	R/W	0324 <sub>H</sub>	FFFF <sub>H</sub>	✓	—	—	—
PMC9	Port mode control register 9	16	1, 0 3-0 6-0	R/W	0424 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PFC9	Port function control register 9	16	1, 0 2-0 3-0 6-0	R/W	0524 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PFCE9	Port function control expansion register 9	16	1, 0 5, 4, 1, 0	R/W	0624 <sub>H</sub>	0000 <sub>H</sub>	✓	✓	—	—
PNOT9	Port NOT register 9	16	1, 0 3-0 6-0	W	0724 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PMCSR9	Port mode set/reset register 9	32	17, 16, 1, 0 19-16, 3-0 22-16, 6-0	R/W	0824 <sub>H</sub>	0000 FFFF <sub>H</sub>	✓	—	✓	✓
PMCSR9	Port mode control set/reset register 9	32	17, 16, 1, 0 19-16, 3-0 22-16, 6-0	R/W	0924 <sub>H</sub>	00000000 <sub>H</sub>	✓	—	—	—
PIBC9	Port input buffer control register 9	16	1, 0 3-0 6-0	R/W	4024 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PBDC9	Port bidirection control register 9	16	1, 0 3-0 6-0	R/W	4124 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—

Table 2.51 Control Registers (P9) (2/2)

Register	Function	Register Size	Position	Effective Bit		Value after Reset	Device			
				R/W <sup>1</sup>	Offset Address		48 pins	64 pins	80 pins	100 pins
PU9	Pull-up option register 9	16	1, 0	R/W	4324 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			3-0	—	—	—	✓	—	—	—
			6-0	—	—	—	✓	✓	✓	✓
PD9	Pull-down option register 9	16	1, 0	R/W	4424 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			3-0	—	—	—	✓	—	—	—
			6-0	—	—	—	✓	—	—	—
PODC9	Port open drain control register 9	32	1, 0	R/W	4524 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	—
			3-0	—	—	—	✓	—	—	—
			6-0	—	—	—	✓	—	—	—
PPROTS9	Port protection status register 9	32	0	R	4B24 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	✓	✓	✓
PPCMD9	Port protection command register 9	32	7-0	W	4C24 <sub>H</sub>	xxxx xx00 <sub>H</sub>	✓	✓	✓	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.7 Port 10 (P10)

### 2.10.7.1 Alternative Functions

Table 2.52 Port 10 (P10) (1/2)

Port Mode (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)										Device						
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output								
P10_0	TAUD01	TAUD01	CAN0RX/ INTP0	CSCXFOUT		PWGA00		TAPA0UP			✓	✓	—	—	—	—	
	TAUD01	TAUD01	CAN0RX/ INTP0	CSCXFOUT		PWGA00		TAPA0UP	CSIH1SI		—	—	✓	✓	✓	—	
	TAUD01	TAUD01	CAN0RX/ INTP0	CSCXFOUT		PWGA00		TAPA0UP	CSIH1SI	MEMC0 A19	—	—	—	—	—	✓	
P10_1	TAUD03	TAUD03	CAN0TX	CAN0TX		PWGA10		TAPA0UN			MODE0	✓	✓	—	—	—	—
	TAUD03	TAUD03	CAN0TX	CAN0TX		PWGA10		TAPA0UN	CSIH1SC	MODE0	—	✓	✓	✓	✓	✓	
P10_2	TAUD05	TAUD05	RICOSDA	KR010	PWGA20	ADCA0TRG0	TAPA0VP			MODE1	✓	✓	—	—	—	—	
	TAUD05	TAUD05	RICOSDA	KR010	PWGA20	ADCA0TRG0	TAPA0VP			MODE1	—	—	✓	✓	✓	✓	
P10_3	TAUD07	TAUD07	RICOSCL	KR011	PWGA30	ADCA0TRG1	TAPA0VN			✓	✓	—	—	—	—	—	
	TAUD07	TAUD07	RICOSCL	KR011	PWGA30	ADCA0TRG1	TAPA0VN			✓	✓	—	✓	✓	✓	✓	
P10_4	TAUD09	TAUD09	RLIN21RX	KR012	ADCA0SEL0	ADCA0TRG2	TAPA0WP			✓	✓	✓	✓	✓	✓	✓	
	TAUD09	TAUD09	RLIN21TX	KR013	ADCA0SEL1		TAPA0WN			CSIG0RYO	CSIG0RYO	—	—	—	—	—	
P10_5	TAUD011	TAUD011	CSIG0SO	ENCA0TIN0	ADCA0SEL2					CSIG1SS1	CSIG1SS1	—	—	—	—	—	
	TAUD013	TAUD013	CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1				MEMC0CLK	MEMC0CLK	—	—	—	—	—	
P10_6	TAUD013	TAUD013	CSIG0SO	ENCA0TIN0	ADCA0SEL2					CSIG0SS1	CSIG0SS1	✓	✓	✓	✓	✓	
	TAUD013	TAUD013	CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1				CSIG0RYO	CSIG0RYO	✓	✓	✓	✓	✓	
P10_7	TAUD015	TAUD015	CSIG0SC	ENCA0TIN1	PWGA40					MEMC0AD0	MEMC0AD0	—	—	—	—	✓	
	TAUD015	TAUD015	CSIG0SC	ENCA0TIN1	PWGA40					CAN1TX	CAN1TX	—	✓	✓	✓	✓	
	TAUD015	TAUD015	CSIG0SC	ENCA0TIN1	PWGA40					MEMC0AD1	MEMC0AD1	—	—	—	—	✓	
P10_8	TAUD010	TAUD010	CSIG0SI	ENCA0EC	PWGA50					FLMD1	FLMD1	✓	✓	✓	✓	✓	
	TAUD010	TAUD010	CSIG0SI	ENCA0EC	PWGA50	MEMC0AD2				FLMD1	FLMD1	—	—	—	—	✓	
P10_9	TAUD012	TAUD012	RLIN30RX/ INTP10	ENCA0E0	PWGA60	CSIH0RYO				MEMC0AD3	MEMC0AD3	—	—	—	—	✓	
	TAUD012	TAUD012	RLIN30RX/ INTP10	ENCA0E0	PWGA60	CSIH0RYO				MEMC0AD3	MEMC0AD3	—	—	—	—	✓	

Table 2.52 Port 10 (P10) (2/2)

Port Mode (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)										Device
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function	
P10_10	TAUD014	TAUD014	RLIN30TX	ENCA0E1	PWGA7O	CSIH0CSS1				✓	✓
P10_11	TAUD014	TAUD014	RLIN30TX	ENCA0E1	PWGA7O	MEMC0AD4				—	✓
	PWGA16O	PWGA16O	RLIN31RX/ INTP11							✓	—
	PWGA16O	PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0				—	—
	PWGA16O	PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0	TAUB011	TAUB001		—	—
P10_12	PWGA16O	PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0	TAUB011	TAUB001	MEMC0AD5		—
	PWGA17O	PWGA17O	RLIN31TX							✓	—
	PWGA17O	PWGA17O	RLIN31TX			CSIH1CSS1				—	—
	PWGA17O	PWGA17O	RLIN31TX			CSIH1CSS1	TAUB013	TAUB003		—	—
P10_13	CSIH0SSI	PWGA18O	RLIN32RX/ INTP12						MEMC0AD6		—
CSIH0SSI	PWGA18O	PWGA18O	RLIN32RX/ INTP12							—	—
CSIH0SSI	PWGA18O	PWGA18O	RLIN32RX/ INTP12							—	—
CSIH0SSI	PWGA19O	PWGA19O	RLIN32TX							—	—
P10_14	PWGA19O	PWGA19O	RLIN32TX							—	—
	PWGA19O	PWGA19O	RLIN32TX			CSIH3SSI		TAUB017	TAUB007		—
ADCA1TRG0	PWGA19O	PWGA19O	RLIN32TX			CSIH3SSI		TAUB017	TAUB007		—
ADCA1TRG0	PWGA19O	PWGA19O	RLIN32TX			CSIH3SSI		TAUB017	TAUB007		—
P10_15	CSIH3RY1	CSIH3RYO	PWGA24O	RLIN22RX						—	—
CSIH3RY1	CSIH3RYO	CSIH3RYO	PWGA24O	RLIN22RX						—	—
									MEMC0RD		—

**CAUTION**

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

## 2.10.7.2 Control Registers

Table 2.53 Control Registers (P10) (1/3)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins
P10	Port register 10	16	10-0 14-0 15-0	R/W	0028 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PSR10	Port set/reset register 10	32	26-16, 10-0 30-16, 14-0 31-16, 15-0	R/W	0128 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	—
PPR10	Port pin read register 10	16	10-0 14-0 15-0	R	0228 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PM10	Port mode register 10	16	10-0 14-0 15-0	R/W	0328 <sub>H</sub>	FFFF <sub>H</sub>	✓	—	—	—
PMC10	Port mode control register 10	16	10-0 14-0 15-0	R/W	0428 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PFC10	Port function control register 10	16	10-0 12-0 14-0 15-0	R/W	0528 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PFCE10	Port function control expansion register 10	16	10-0 14-0 15-0	R/W	0628 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PNOT10	Port NOT register 10	16	10-0 14-0 15-0	W	0728 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
PMSR10	Port mode set/reset register 10	32	26-16, 10-0 30-16, 14-0 31-16, 15-0	R/W	0828 <sub>H</sub>	0000 FFFF <sub>H</sub>	✓	—	—	—

Table 2.53 Control Registers (P10) (2/3)

Register	Function	Register Size	Effective Bit Position	R/W <sup>1</sup>	Offset Address	Value after Reset	Device			
							48 pins	64 pins	80 pins	100 pins
PMCSR10	Port mode control set/reset register 10	32	26-16, 10-0	R/W	0928 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	—
			30-16, 14-0				—	✓	—	—
			31-16, 15-0				—	—	✓	✓
PFCAE10	Port function control additional expansion register 10	16	5, 4	R/W	0A28 <sub>H</sub>	0000 <sub>H</sub>	✓	✓	—	—
			5-0				—	—	✓	✓
			15-9, 7-0				—	—	—	✓
PIBC10	Port input buffer control register 10	16	10-0	R/W	4028 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			14-0				—	—	✓	✓
			15-0				—	—	—	—
PBDC10	Port bidirection control register 10	16	10-0	R/W	4128 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			14-0				—	—	✓	✓
			15-0				—	—	✓	✓
PIPC10	Port IP control register 10	16	7-0	R/W	4228 <sub>H</sub>	0000 <sub>H</sub>	✓	✓	✓	✓
			14-0				—	—	—	✓
			15-0				—	—	—	—
PU10	Pull-up option register 10	16	10-0	R/W	4328 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PD10	Pull-down option register 10	16	10-0	R/W	4428 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PODC10	Port open drain control register 10	32	10-0	R/W	4528 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	—
			14-0				—	✓	—	—
			15-0				—	—	✓	✓
PDSC10	Port drive strength control register 10	32	7-0	R/W	4628 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	✓	✓	✓
			15-0				—	—	—	✓

Table 2.53 Control Registers (P10) (3/3)

Register	Function	Register Size	Effective Bit Position	R/W <sup>*1</sup>	Offset Address	Value after Reset	Device					
							48 pins	64 pins	80 pins	100 pins		
PIS10	Port input buffer selection register 10	16	9, 4-2, 0	R/W	4728 <sub>H</sub>	FFFF <sub>H</sub>	✓	—	—	—		
			11, 9, 6, 4-2, 0	R/W			—	✓	—	—		
			13, 11, 9, 6, 4-2, 0				—	—	✓	—		
			15, 13, 11, 9, 6, 4-2, 0				—	—	—	✓		
PPROTS10	Port protection status register 10	32	0	R	4B28 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	✓	✓	✓		
PPCMD10	Port protection command register 10	32	7-0	W	4C28 <sub>H</sub>	xxxx xx00 <sub>H</sub>	✓	✓	✓	✓		

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.8 Port 11 (P11)

### 2.10.8.1 Alternative Functions

Table 2.54 Port 11 (P11) (1/2)

Alternative Mode (PMC11_m = 1)										Device							
Port mode (PMC11_m = 0)	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P11_0	CSIH2RYI	CSIH2RYO										—	—	✓	—	—	
	CSIH2RYI	CSIH2RYO	PWGA25O		RLIN22TX	TAUB011	TAUB0011					—	—	✓	—	—	
	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O	RLIN22TX	TAUB011	TAUB0011					—	—	✓	—	—	
	CSIH2RYI	CSIH2RYO	ADCA11TRG2	PWGA25O	RLIN22TX	TAUB011	TAUB0011					—	—	✓	—	✓	
	CSIH2SSI		RLIN20RX									—	—	✓	—	—	
P11_1	CSIH2SSI		RLIN20RX		PWGA26O	TAUB013	TAUB0013					—	—	✓	—	—	
	CSIH2SSI		RLIN20RX		PWGA26O	TAUB013	TAUB0013	MEMC0AD9				—	—	✓	—	✓	
	CSIH2SO		RLIN20TX		PWGA27O	TAUB015	TAUB0015					—	—	✓	—	—	
	CSIH2SO		RLIN20TX		PWGA27O	TAUB015	TAUB0015	MEMC0AD10				—	—	✓	—	—	
	CSIH2SO		RLIN20TX		PWGA27O	TAUB015	TAUB0015	MEMC0AD10				—	—	✓	—	✓	
P11_2	CSIH2SC											—	—	✓	—	—	
	CSIH2SC	INTP3	PWGA28O									—	—	✓	—	—	
	CSIH2SC	CAN3RX/ INTP3	PWGA28O									—	—	*1	—	—	
	CSIH2SC	CAN3RX/ INTP3	PWGA28O	TAUB111	TAUB101	MEMC0AD11						—	—	*2	✓	—	
	CSIH2SI											—	—	✓	—	—	
P11_3	CSIH2SI		PWGA29O									—	—	✓	—	—	
	CSIH2SI	CAN3TX	PWGA29O									—	—	✓	—	—	
	CSIH2SI	CAN3TX	PWGA29O									—	—	✓	—	—	
	CSIH2SI	CAN3TX	PWGA29O	TAUB103	TAUB103	MEMC0AD12						—	—	✓	—	✓	
	INTP5	RLIN33TX	PWGA30O	CSIH3SI								—	—	✓	—	✓	
P11_5	CAN5RX/ INTP5	RLIN33TX	PWGA30O	CSIH3SI								—	—	*1	*3	—	
	CAN5RX/ INTP5	RLIN33TX	PWGA30O	CSIH3SI								—	—	*2	*4	—	
	CAN5RX/ INTP5	RLIN33TX	PWGA30O	CSIH3SI	TAUB115	TAUB105	MEMC0AD13					—	—	✓	—	✓	

Table 2.54 Port 11 (P11) (2/2)

Port mode (PMCC11_m = 0)	Alternative Mode (PMCC11_m = 1)										Device
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function	
P11_6	RLIN33RX/ INTP13			PWGA310		CSIH3SO				—	— *1 —
	RLIN33RX/ INTP13	CAN5TX		PWGA310		CSIH3SO				—	*2 —
	RLIN33RX/ INTP13			ADCA1TRG1	PWGA310	CSIH3SO				—	*3 —
	RLIN33RX/ INTP13	CAN5TX		ADCA1TRG1	PWGA310	CSIH3SO				—	*4 —
	RLIN33RX/ INTP13			ADCA1TRG1	PWGA310	CSIH3SO				—	
P11_7	INTP5			PWGA320	CSIH3SC					—	— √ —
	INTP5			PWGA320	CSIH3SC					—	— √ —
	CSI(G1SSI)	RLIN35TX		PWGA480		TAUB119	TAUB109	MEMC0AD15		—	— √ —
	CSI(G1SSI)	RLIN35TX		PWGA480		TAUB111	TAUB1011	MEMC0CS0		—	— √ —
P11_9		CSI1SO	RLIN35RX/ INTP15	PWGA490						—	— √ —
		CSI1SO	RLIN35RX/ INTP15	PWGA490	TAUB113	TAUB1013		MEMC0CS1		—	— √ —
P11_10		CSIG1SC		PWGA500						—	— √ —
		CSIG1SC	RLIN25TX	PWGA500	TAUB115	TAUB1015		MEMC0CS2		—	— √ —
P11_11	CSIG1SI	RLIN25TX		PWGA510		TAUB110	TAUB100	MEMC0CS3		—	— √ —
	CSIG1SI			PWGA510						—	— √ —
P11_12	RLIN25RX			PWGA520						—	— √ —
	RLIN25RX			PWGA520						—	— √ —
P11_13	RLIN24RX			PWGA530						—	— √ —
	RLIN24RX			PWGA530						—	— √ —
P11_14		RLIN24TX		PWGA530	TAUB114	TAUB104	MEMC0WAIT			—	— √ —
		RLIN24TX		PWGA540		TAUB116	TAUB106	MEMC0BEN0		—	— √ —
P11_15	CAN2RX/ INTP2	CSIH2CSS4		PWGA550				MEMC0BEN1		—	— √ —
	CAN2RX/ INTP2	CSIH2CSS4		PWGA550						—	— √ —
				PWGA550	TAUB118	TAUB108	MEMC0ASTB			—	— √ —

Note 1. Available except in F1L for Gateway.

Note 2. Only available in F1L for Gateway.

- Note 3. Available in devices except for ones with 1.5- and 2-MB code flash memories.
- Note 4. Only available in devices with 1.5- and 2-MB code flash memories.

---

**CAUTION**

---

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

---

## 2.10.8.2 Control Registers

**Table 2.55 Control Registers (P11) (1/2)**

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*5</sup>			48 pins	64 pins	80 pins	100 pins
P11	Port register 11	16	4-0 7-0 15-0	R/W	002C <sub>H</sub>	0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PSR11	Port set/reset register 11	32	20-16, 4-0 23-16, 7-0 31-16, 15-0	R/W	012C <sub>H</sub>	0000 0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PPR11	Port pin read register 11	16	4-0 7-0 15-0	R	022C <sub>H</sub>	0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PM11	Port mode register 11	16	4-0 7-0 15-0	R/W	032C <sub>H</sub>	FFFF <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PMC11	Port mode control register 11	16	4-0 7-0 15-0	R/W	042C <sub>H</sub>	0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PFC11	Port function control register 11	16	2, 1 7-0 15-0	R/W	052C <sub>H</sub>	0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PFCE11	Port function control expansion register 11	16	7-5, 2-0 15-0	R/W	062C <sub>H</sub>	0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PNOT11	Port NOT register 11	16	4-0 7-0 15-0	W	072C <sub>H</sub>	0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PMCSR11	Port mode set/reset register 11	32	20-16, 4-0 23-16, 7-0 31-16, 15-0	R/W	082C <sub>H</sub>	0000 FFFF <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —
PMCSR11	Port mode control set/reset register 11	32	20-16, 4-0 23-16, 7-0 31-16, 15-0	R/W	092C <sub>H</sub>	0000 0000 <sub>H</sub>	— — — — —	— — — — —	— — — — —	— — — — —

Table 2.55 Control Registers (P11) (2/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W <sup>*5</sup>			48 pins	64 pins	80 pins
PFCAE11	Port function control additional expansion register 11	16	7-5, 2-0	R/W	0A2C <sub>H</sub>	0000 <sub>H</sub>	—	—	—
PIBC11	Port input buffer control register 11	16	4-0	R/W	402C <sub>H</sub>	0000 <sub>H</sub>	—	✓	—
			7-0			—	—	✓	—
			15-0			—	—	✓	✓
PBDC11	Port bidirection control register 11	16	4-0	R/W	412C <sub>H</sub>	0000 <sub>H</sub>	—	✓	—
			7-0			—	—	✓	—
			15-0			—	—	✓	✓
PIPC11	Port IP control register 11	16	3, 2	R/W	422C <sub>H</sub>	0000 <sub>H</sub>	—	✓	—
			7, 6, 3, 2			—	—	✓	—
			10, 9, 7, 6, 3, 2			—	—	✓	—
			10, 9, 7-1			—	—	✓	—
PU11	Pull-up option register 11	16	4-0	R/W	432C <sub>H</sub>	0000 <sub>H</sub>	—	✓	—
			7-0			—	—	✓	—
			15-0			—	—	✓	—
PD11	Pull-down option register 11	16	4-0	R/W	442C <sub>H</sub>	0000 <sub>H</sub>	—	✓	✓
PODC11	Port open drain control register 11	32	4-0	R/W	452C <sub>H</sub>	0000 0000 <sub>H</sub>	—	✓	—
			7-0			—	—	✓	—
			15-0			—	—	✓	✓
PDSC11	Port drive strength control register 11	32	3, 2	R/W	462C <sub>H</sub>	0000 0000 <sub>H</sub>	—	✓	—
			7, 6, 3, 2			—	—	✓	—
			10, 9, 7, 6, 3, 2			—	—	✓	—
			15-13, 11-0			—	—	✓	—
PIS11	Port input buffer selection register 11	16	1	R/W	472C <sub>H</sub>	FFFF <sub>H</sub>	—	✓	—
			6, 1			—	—	*1	—
			6, 5, 3, 1			—	—	*2	—
			15, 13, 12, 9, 6, 3, 1			—	—	*3	—
			15, 13, 12, 9, 6, 5, 3, 1			—	—	*4	✓
PPROTS11	Port protection status register 11	32	0	R	4B2C <sub>H</sub>	0000 0000 <sub>H</sub>	—	✓	✓
PPCMD11	Port protection command register 11	32	7-0	W	4C2C <sub>H</sub>	xxx xx00 <sub>H</sub>	—	✓	✓

Note 1. Available except in F1L for Gateway.

- Note 2. Only available in F1L for Gateway.
- Note 3. Available in devices except for ones with 1.5- and 2-MB code flash memories.
- Note 4. Only available in devices with 1.5- and 2-MB code flash memories.
- Note 5. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.9 Port 12 (P12)

### 2.10.9.1 Alternative Functions

Table 2.56 Port 12 (P12)

Port Mode (PMC12_m =0)	Alternative Mode (PMC12_m =1)								Device								
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output								
P12_0	CAN2TX	PWGA560										—	—	—	—	—	—
	CAN2TX	PWGA560	TAUB1010									—	—	—	—	—	✓
P12_1	RLIN34RX/ INTP14	CSIH2CSS5	PWGA570									—	—	—	—	✓	—
	RLIN34RX/ INTP14	CSIH2CSS5	PWGA570	TAUB1012								—	—	—	—	✓	—
P12_2	RLIN34TX	PWGA580										—	—	—	—	✓	—
	RLIN34TX	PWGA580	TAUB1014									—	—	—	—	—	✓
P12_3	RLIN27RX	PWGA680										—	—	—	—	—	✓
	RLIN27TX	PWGA690										—	—	—	—	—	✓
P12_4		PWGA700										—	—	—	—	—	✓
P12_5												—	—	—	—	—	✓

### CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

## 2.10.9.2 Control Registers

Table 2.57 Control Registers (P12) (1/2)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins
P12	Port register 12	16	2-0 5-0	R/W	0030 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PSR12	Port set/reset register 12	32	18-16, 2-0 21-16, 5-0	R/W	0130 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PPR12	Port pin read register 12	16	2-0 5-0	R	0230 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PM12	Port mode register 12	16	2-0 5-0	R/W	0330 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—
PMC12	Port mode control register 12	16	2-0 5-0	R/W	0430 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PFC12	Port function control register 12	16	2-0 4-0	R/W	0530 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PFCE12	Port function control expansion register 12	16	2-0	R/W	0630 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PNOT12	Port NOT register 12	16	2-0 5-0	W	0730 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PMSR12	Port mode set/reset register 12	32	18-16, 2-0 21-16, 5-0	R/W	0830 <sub>H</sub>	0000 FFFF <sub>H</sub>	—	—	—	—
PMCSR12	Port mode control set/reset register 12	32	18-16, 2-0 21-16, 5-0	R/W	0930 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PIBC12	Port input buffer control register 12	16	2-0 5-0	R/W	4030 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PBDC12	Port bidirection control register 12	16	2-0 5-0	R/W	4130 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PU12	Pull-up option register 12	16	2-0 5-0	R/W	4330 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—
PODC12	Port open drain control register 12	32	2-0 5-0	R/W	4530 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PDSC12	Port drive strength control register 12	32	2-0	R/W	4630 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—

Table 2.57 Control Registers (P12) (2/2)

Register	Function	Effective Bit		R/W <sup>a</sup>	Offset Address	Value after Reset	Device			
		Register Size	Position				48 pins	64 pins	80 pins	100 pins
PIS12	Port input buffer selection register 12	16	1 3,1	R/W	4730 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—
PPROTS12	Port protection status register 12	32	0	R	4B30 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PPCMD12	Port protection command register 12	32	7-0	W	4C30 <sub>H</sub>	xxxx xx00 <sub>H</sub>	—	—	—	—

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.10 Port 18 (P18)

### 2.10.10.1 Alternative Functions

Table 2.58 Port 18 (P18)

Port Mode (PMC18_m = 0)	Alternative Mode (PMC18_m = 1)										Device
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function
P18_0	CSIG1RYI	CSIG1RYO	PWGA610								ADCA110S
P18_1		PWGA620									ADCA111S
P18_2		PWGA630									ADCA112S
P18_3		PWGA710									ADCA113S
P18_4		CSIH1CSS4									ADCA113S
P18_5		CSIH1CSS5									ADCA114S
P18_6											ADCA115S
P18_7											ADCA116S
											ADCA117S

### CAUTIONS

- 1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
- 2. Use special functions with their initial settings.

## 2.10.10.2 Control Registers

Table 2.59 Control Registers (P18)

Register	Function			Effective Bit		Offset Address	Value after Reset	Device				
		Register Size	Position	R/W <sup>*1</sup>	48 pins			64 pins	80 pins	100 pins	144 pins	176 pins
P18	Port register 18	16	3-0 7-0	R/W	0048 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PSR18	Port set/reset register 18	32	19-16, 3-0 23-16, 7-0	R/W	0148 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓	—
PPR18	Port pin read register 18	16	3-0 7-0	R	0248 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PM18	Port mode register 18	16	3-0 7-0	R/W	0348 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—	✓	—
PMC18	Port mode control register 18	16	2-0 5-0	R/W	0448 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PFC18	Port function control register 18	16	0	R/W	0548 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PNOT18	Port NOT register 18	16	3-0 7-0	W	0748 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PMSR18	Port mode set/reset register 18	32	19-16, 3-0 23-16, 7-0	R/W	0848 <sub>H</sub>	0000 FFFF <sub>H</sub>	—	—	—	—	✓	—
PMCSR18	Port mode control set/reset register 18	32	18-16, 2-0 21-16, 5-0	R/W	0948 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓	—
PIBC18	Port input buffer control register 18	16	3-0 7-0	R/W	4048 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PBDC18	Port bidirection control register 18	16	3-0 7-0	R/W	4148 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
PU18	Pull-up option register 18	16	3-0 7-0	R/W	4348 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓	—
POD18	Port open drain control register 18	32	3-0 7-0	R/W	4548 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓	—
PPROTS18	Port protection status register 18	32	0	R	4B48 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓	—
PPCMD18	Port protection command register 18	32	7-0	W	4C48 <sub>H</sub>	xxxx xx00 <sub>H</sub>	—	—	—	—	✓	—

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.11 Port 20 (P20)

### 2.10.11.1 Alternative Functions

Table 2.60 Port 20 (P20)

Port Mode (PMC20_m = 0)	Alternative Mode (PMC20_m = 1)								Device	
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative	
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
P20_0	RLIN26RX			PWGA640						—
P20_1		RLIN26TX		PWGA650						—
P20_2	CAN4RX/ INTP4			PWGA660						—
P20_3		CAN4TX		PWGA670						—
P20_4	RLIN23RX			PWGA590						—
P20_5		RLIN23TX		PWGA600						—

#### CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

## 2.10.11.2 Control Registers

Table 2.61 Control Registers (P20) (1/2)

Register	Function			Effective Bit		Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>	48 pins			64 pins	80 pins	100 pins	144 pins
P20	Port register 20	16	5, 4 5-0	R/W	0050 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PSR20	Port set/reset register 20	32	21-20, 5, 4 21-16, 5-0	R/W	0150 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓
PPR20	Port pin read register 20	16	5, 4 5-0	R	0250 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PM20	Port mode register 20	16	5, 4 5-0	R/W	0350 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—	✓
PMC20	Port mode control register 20	16	5, 4 5-0	R/W	0450 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PFC20	Port function control register 20	16	5, 4 5-0	R/W	0550 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PNOT20	Port NOT register 20	16	5, 4 5-0	W	0750 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PMCSR20	Port mode set/reset register 20	32	21-20, 5, 4 21-16, 5-0	R/W	0850 <sub>H</sub>	0000 FFFF <sub>H</sub>	—	—	—	—	✓
PMCSR20	Port mode control set/reset register 20	32	21-20, 5, 4 21-16, 5-0	R/W	0950 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓
PIBC20	Port input buffer control register 20	16	5, 4 5-0	R/W	4050 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PBDC20	Port bidirection control register 20	16	5, 4 5-0	R/W	4150 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PU20	Pull-up option register 20	16	5, 4 5-0	R/W	4350 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	✓
PODC20	Port open drain control register 20	32	5, 4 5-0	R/W	4550 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—	✓

Table 2.61 Control Registers (P20) (2/2)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>a</sup>			48 pins	64 pins	80 pins	100 pins
PIS20	Port input buffer selection register 20	16	4	R/W	4750 <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	—
			4, 2, 0				—	—	—	—
PPROTS20	Port protection status register 20	32	0	R	4B50 <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	—
PPCMD20	Port protection command register 20	32	7-0	W	4C50 <sub>H</sub>	xxxx xx00 <sub>H</sub>	—	—	—	—

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.12 Analog Port 0 (AP0)

### 2.10.12.1 Alternative Functions

Table 2.62 Analog Port 0 (AP0)

Port Mode	Alternative Mode						Device									
	1st Alternative		2nd Alternative		3rd Alternative											
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
AP0_0										ADCA010	✓	✓	✓	✓	✓	✓
AP0_1										ADCA011	✓	✓	✓	✓	✓	✓
AP0_2										ADCA012	✓	✓	✓	✓	✓	✓
AP0_3										ADCA013	✓	✓	✓	✓	✓	✓
AP0_4										ADCA014	✓	✓	✓	✓	✓	✓
AP0_5										ADCA015	✓	✓	✓	✓	✓	✓
AP0_6										ADCA016	✓	✓	✓	✓	✓	✓
AP0_7										ADCA017	✓	✓	✓	✓	✓	✓
AP0_8										ADCA018	—	✓	✓	✓	✓	✓
AP0_9										ADCA019	—	✓	✓	✓	✓	✓
AP0_10										ADCA010	—	—	✓	✓	✓	✓
AP0_11										ADCA011	—	—	—	✓	✓	✓
AP0_12										ADCA012	—	—	✓	✓	✓	✓
AP0_13										ADCA013	—	—	✓	✓	✓	✓
AP0_14										ADCA014	—	—	✓	✓	✓	✓
AP0_15										ADCA015	—	—	✓	✓	✓	✓

### CAUTION

Use special functions with their initial settings.

## 2.10.12.2 Control Registers

Table 2.63 Control Registers (AP0) (1/2)

Register	Function	Register Size	Effective Bit			Value after Reset	Device			
			Position	R/W <sup>*1</sup>	Offset Address		48 pins	64 pins	80 pins	100 pins
AP0	Analog port register 0	16	7-0	R/W	00C8 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	✓	✓
APSRO	Analog port set/reset register 0	32	23-16, 7-0	R/W	01C8 <sub>H</sub>	0000 0000 <sub>H</sub>	✓	—	—	—
			25-16, 9-0				—	✓	—	—
			26-16, 10-0				—	✓	—	—
			31-16, 15-0				—	—	✓	✓
APP0	Analog port pin read register 0	16	7-0	R	02C8 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	✓	✓
APMO	Analog port mode register 0	16	7-0	R/W	03C8 <sub>H</sub>	FFFF <sub>H</sub>	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	✓	✓
APNOT0	Analog port NOT register 0	16	7-0	W	07C8 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	✓	✓
APMSR0	Analog port mode set/reset register 0	32	23-16, 7-0	R/W	08C8 <sub>H</sub>	0000 FFFF <sub>H</sub>	✓	—	—	—
			25-16, 9-0				—	✓	—	—
			26-16, 10-0				—	—	✓	—
			31-16, 15-0				—	—	✓	✓
APIBC0	Analog port input buffer control register 0	16	7-0	R/W	40C8 <sub>H</sub>	0000 <sub>H</sub>	✓	—	—	—
			9-0				—	✓	—	—
			10-0				—	—	✓	—
			15-0				—	—	✓	✓

Table 2.63 Control Registers (AP0) (2/2)

Register	Function	Register Size	Effective Bit Position	Device								
				R/W <sup>*1</sup>	Offset Address	Value after Reset	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
APBDC0	Analog port bidirection control register 0	16	7-0 9-0 10-0 15-0	R/W	41C8H	0000H	✓	—	—	—	—	—

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.13 Analog Port 1 (AP1)

### 2.10.13.1 Alternative Functions

Table 2.64 Analog Port 1 (AP1)

Port Mode	Alternative Mode						Device									
	1st Alternative		2nd Alternative		3rd Alternative											
Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Special Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
AP1_0										ADCA110	—	—	—	—	—	—
AP1_1										ADCA111	—	—	—	—	—	✓
AP1_2										ADCA112	—	—	—	—	—	✓
AP1_3										ADCA113	—	—	—	—	—	✓
AP1_4										ADCA114	—	—	—	—	—	✓
AP1_5										ADCA115	—	—	—	—	—	✓
AP1_6										ADCA116	—	—	—	—	—	✓
AP1_7										ADCA117	—	—	—	—	—	✓
AP1_8										ADCA118	—	—	—	—	—	✓
AP1_9										ADCA119	—	—	—	—	—	✓
AP1_10										ADCA1110	—	—	—	—	—	✓
AP1_11										ADCA1111	—	—	—	—	—	✓
AP1_12										ADCA1112	—	—	—	—	—	✓
AP1_13										ADCA1113	—	—	—	—	—	✓
AP1_14										ADCA1114	—	—	—	—	—	✓
AP1_15										ADCA1115	—	—	—	—	—	✓

### CAUTION

Use special functions with their initial settings.

## 2.10.13.2 Control Registers

Table 2.65 Control Registers (AP1)

Register	Function	Effective Bit			Offset Address	Value after Reset	Device			
		Register Size	Position	R/W <sup>*1</sup>			48 pins	64 pins	80 pins	100 pins
AP1	Analog port register 1	16	7-0	R/W	00CC <sub>H</sub>	0000 <sub>H</sub>	—	—	—	✓
			15-0				—	—	—	✓
APSR1	Analog port set/reset register 1	32	23-16, 7-0	R/W	01CC <sub>H</sub>	0000 0000 <sub>H</sub>	—	—	—	✓
			31-16, 15-0				—	—	—	✓
APPR1	Analog port pin read register 1	16	7-0	R	02CC <sub>H</sub>	0000 <sub>H</sub>	—	—	—	✓
			15-0				—	—	—	✓
APM1	Analog port mode register 1	16	7-0	R/W	03CC <sub>H</sub>	FFFF <sub>H</sub>	—	—	—	✓
			15-0				—	—	—	✓
APNOT1	Analog port NOT register 1	16	7-0	W	07CC <sub>H</sub>	0000 <sub>H</sub>	—	—	—	✓
			15-0				—	—	—	✓
APMSR1	Analog port mode Set/reset register 1	32	23-16, 7-0	R/W	08CC <sub>H</sub>	0000 FFFF <sub>H</sub>	—	—	—	✓
			31-16, 15-0				—	—	—	✓
APIBC1	Analog port input buffer control register 1	16	7-0	R/W	40CC <sub>H</sub>	0000 <sub>H</sub>	—	—	—	✓
			15-0				—	—	—	✓
APBDC1	Analog port bidirection control register 1	16	7-0	R/W	41CC <sub>H</sub>	0000 <sub>H</sub>	—	—	—	✓
			15-0				—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

## 2.10.14 Input Port 0 (IP0)

### 2.10.14.1 Alternative Functions

Table 2.66 Input Port 0 (IP0)

Port Mode	Alternative Mode								Device	
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative			
	Input	Output	Input	Output	Input	Output	Input	Output		
IP0_0									XT2	

## 2.10.14.2 Control Registers

Table 2.67 Control Registers (IP0)

Register	Function	Register Size	Effective Bit Position	R/W <sup>*1</sup>	Offset Address	Value after Reset	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Device
IPPRO	Input port pin read register 0	16	0	R	02F0 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	—	—	✓
IPIBC0	Port input buffer control register 0	16	0	R/W	40F0 <sub>H</sub>	0000 <sub>H</sub>	—	—	—	—	—	—	✓

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.  
When writing to unused bits, write the value after reset.

### CAUTION

When the IP0\_0/XT2 pin is used as an input port, set the IPIBC0\_0 bit to 1 while stopping the SOSC operation.

For details on the settings for SOSC operations, see **Section 10.4.2.6, SOSCE — SubOSC Enable Register**. When the IP0\_0/XT2 pin is used for the SubOSC (SOSC) not as an input port, set the IPIBC0\_0 bit to 0.

## 2.11 Port (Special I/O) Function Overview

This section describes the port (special I/O) functions.

### 2.11.1 Special I/O after Reset

The special port function after reset release is shown below.

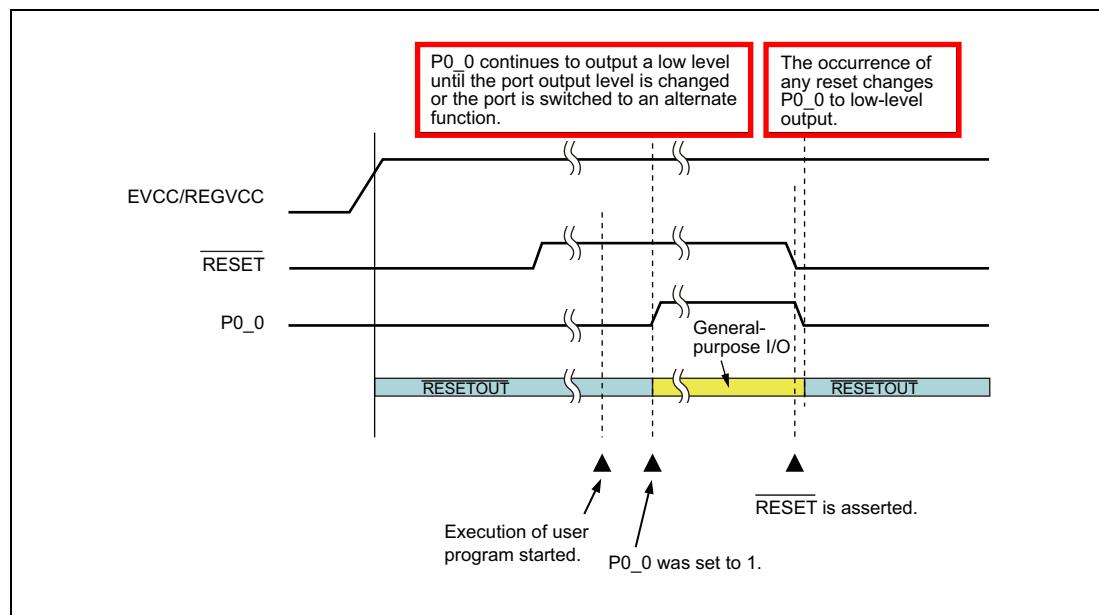
#### 2.11.1.1 P0\_0: RESETOUT

The P0\_0 pin outputs low level while a reset is asserted and continues to output low level after the reset is released due to the initial value of the registers as follows:

- P0.P0\_0 = 0: Low level
- PM0.PM0\_0 = 0: Output port
- PODC0.PODC0\_0 = 1: Open-drain output

Therefore, this pin can be used as RESETOUT function.

When the P0\_0 pin setting is updated with another value, the pin operates by new setting.



**Figure 2.12 P0\_0 Pin (RESETOUT Signal) Operation While a Reset Is Asserted and Released**

### 2.11.1.2 JP0\_0 to JP0\_6: Debug Interface

If the OPJTAG[1:0] setting is the combination below, the pins of the JTAG port group can be used as a debug interface after reset release.

**Table 2.68 Debug Interface**

OPJTAG1	OPJTAG0	Mode	JP0_0	JP0_1	JP0_2	JP0_3	JP0_4	JP0_5	JP0_6
1	1	Nexus I/F	DCUTDI input	DCUTDO output	DCUTCK input	DCUTMS input	DCUTRST input	DCURDY output	EVTO <sup>*1</sup> output
0	1	LPD (4-pin)	LPDI input	LPDO output	LPDCLK input	Port/ alternative function	Port/ alternative function	LPDCLK OUT output	Port/ alternative function
1	0	LPD (1-pin)	LPDIO input/ output	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function

Consequently, the port and alternative functions on these pins cannot be used while the debugger is connected.

**Note 1.** The EVTO pin is only available in devices with 2-MB code flash memory.

#### NOTE

For the OPJTAG[1:0] settings, see **Section 35.11.2, OPBT0 — Option Byte 0**.

### 2.11.1.3 FPDR(JP0\_0), FPDT(JP0\_1), FPCK(JP0\_2): Flash Programmer

These pins are used for connecting a flash programmer. See *Flash Programmer's Manual* for details.

### 2.11.1.4 Mode Pins

The FLMD0 pin in combination with the P10\_8: FLMD1 pin can select serial programming mode. In addition, the P10\_1: MODE0 pin in combination with the P10\_2: MODE1 pin can select boundary scan mode. For details on the mode selection, see **Section 5, Operating Mode**.

### 2.11.1.5 IP0\_0: XT2

This pin is the SubOSC (SOSC) input pin. When the IPIBC0\_0 bit = 1, the IP0\_0/XT2 pin is used as an input port. If you make this setting, stop SOSC operation at the same time.

## 2.11.2 A/D Input Alternative I/O

The following ports are permanently connected to A/D input functions. (However, an analog input to the A/D is controlled by the A/D module.)

**Table 2.69 A/D Input Alternative Pins (1/2)**

Port	A/D Input	Device					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P8_0	ADCA0I0S	✓	✓	✓	✓	✓	✓
P8_1	ADCA0I1S	✓	✓	✓	✓	✓	✓
P8_2	ADCA0I4S	—	✓	✓	✓	✓	✓
P8_3	ADCA0I5S	—	✓	✓	✓	✓	✓
P8_4	ADCA0I6S	—	✓	✓	✓	✓	✓
P8_5	ADCA0I7S	—	✓	✓	✓	✓	✓
P8_6	ADCA0I8S	—	✓	✓	✓	✓	✓
P8_7	ADCA0I14S	—	—	—	✓	✓	✓
P8_8	ADCA0I15S	—	—	—	✓	✓	✓
P8_9	ADCA0I16S	—	—	—	✓	✓	✓
P8_10	ADCA0I17S	—	—	—	✓	✓	✓
P8_11	ADCA0I18S	—	—	—	✓	✓	✓
P8_12	ADCA0I19S	—	—	—	✓	✓	✓
P9_0	ADCA0I2S	✓	✓	✓	✓	✓	✓
P9_1	ADCA0I3S	✓	✓	✓	✓	✓	✓
P9_2	ADCA0I9S	—	✓	✓	✓	✓	✓
P9_3	ADCA0I10S	—	✓	✓	✓	✓	✓
P9_4	ADCA0I11S	—	—	✓	✓	✓	✓
P9_5	ADCA0I12S	—	—	✓	✓	✓	✓
P9_6	ADCA0I13S	—	—	✓	✓	✓	✓
P18_0	ADCA1I0S	—	—	—	—	✓	✓
P18_1	ADCA1I1S	—	—	—	—	✓	✓
P18_2	ADCA1I2S	—	—	—	—	✓	✓
P18_3	ADCA1I3S	—	—	—	—	✓	✓
P18_4	ADCA1I4S	—	—	—	—	—	✓
P18_5	ADCA1I5S	—	—	—	—	—	✓
P18_6	ADCA1I6S	—	—	—	—	—	✓
P18_7	ADCA1I7S	—	—	—	—	—	✓
AP0_0	ADCA0I0	✓	✓	✓	✓	✓	✓
AP0_1	ADCA0I1	✓	✓	✓	✓	✓	✓
AP0_2	ADCA0I2	✓	✓	✓	✓	✓	✓
AP0_3	ADCA0I3	✓	✓	✓	✓	✓	✓
AP0_4	ADCA0I4	✓	✓	✓	✓	✓	✓
AP0_5	ADCA0I5	✓	✓	✓	✓	✓	✓
AP0_6	ADCA0I6	✓	✓	✓	✓	✓	✓
AP0_7	ADCA0I7	✓	✓	✓	✓	✓	✓
AP0_8	ADCA0I8	—	✓	✓	✓	✓	✓
AP0_9	ADCA0I9	—	✓	✓	✓	✓	✓

Table 2.69 A/D Input Alternative Pins (2/2)

Port	A/D Input	Device					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
AP0_10	ADCA0I10	—	—	✓	✓	✓	✓
AP0_11	ADCA0I11	—	—	—	✓	✓	✓
AP0_12	ADCA0I12	—	—	—	✓	✓	✓
AP0_13	ADCA0I13	—	—	—	✓	✓	✓
AP0_14	ADCA0I14	—	—	—	✓	✓	✓
AP0_15	ADCA0I15	—	—	—	✓	✓	✓
AP1_0	ADCA1I0	—	—	—	—	✓	✓
AP1_1	ADCA1I1	—	—	—	—	✓	✓
AP1_2	ADCA1I2	—	—	—	—	✓	✓
AP1_3	ADCA1I3	—	—	—	—	✓	✓
AP1_4	ADCA1I4	—	—	—	—	✓	✓
AP1_5	ADCA1I5	—	—	—	—	✓	✓
AP1_6	ADCA1I6	—	—	—	—	✓	✓
AP1_7	ADCA1I7	—	—	—	—	✓	✓
AP1_8	ADCA1I8	—	—	—	—	—	✓
AP1_9	ADCA1I9	—	—	—	—	—	✓
AP1_10	ADCA1I10	—	—	—	—	—	✓
AP1_11	ADCA1I11	—	—	—	—	—	✓
AP1_12	ADCA1I12	—	—	—	—	—	✓
AP1_13	ADCA1I13	—	—	—	—	—	✓
AP1_14	ADCA1I14	—	—	—	—	—	✓
AP1_15	ADCA1I15	—	—	—	—	—	✓

## 2.11.3 Special I/O Control

### 2.11.3.1 Direct I/O Control (PIPC)

Some alternative functions take over the input and output control of the ports.

The following table lists all alternative functions where PIPCn.PIPCn\_m must be set to 1. For details, see **Section 2.9.2.3, PIPCn — Port IP Control Register**.

**Table 2.70 Alternative Modes that Require Setting PIPCn.PIPCn\_m = 1 (1/2)**

Function	Alternative Functions Name	Port Name	Device						Reference Section
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	
MEMC	MEMC0AD0	P10_6	—	—	—	—	—	✓	Section 13
	MEMC0AD1	P10_7	—	—	—	—	—	✓	
	MEMC0AD2	P10_8	—	—	—	—	—	✓	
	MEMC0AD3	P10_9	—	—	—	—	—	✓	
	MEMC0AD4	P10_10	—	—	—	—	—	✓	
	MEMC0AD5	P10_11	—	—	—	—	—	✓	
	MEMC0AD6	P10_12	—	—	—	—	—	✓	
	MEMC0AD7	P10_13	—	—	—	—	—	✓	
	MEMC0AD8	P10_14	—	—	—	—	—	✓	
	MEMC0AD9	P11_1	—	—	—	—	—	✓	
	MEMC0AD10	P11_2	—	—	—	—	—	✓	
	MEMC0AD11	P11_3	—	—	—	—	—	✓	
	MEMC0AD12	P11_4	—	—	—	—	—	✓	
	MEMC0AD13	P11_5	—	—	—	—	—	✓	
	MEMC0AD14	P11_6	—	—	—	—	—	✓	
	MEMC0AD15	P11_7	—	—	—	—	—	✓	
TAPA	TAPA0UP	P10_0	✓	✓	✓	✓	✓	✓	Section 27
	TAPA0UN	P10_1	✓	✓	✓	✓	✓	✓	
	TAPA0VP	P10_2	✓	✓	✓	✓	✓	✓	
	TAPA0VN	P10_3	✓	✓	✓	✓	✓	✓	
	TAPA0WP	P10_4	✓	✓	✓	✓	✓	✓	
	TAPA0WN	P10_5	✓	✓	✓	✓	✓	✓	
CSIG	CSIG0SO	P0_13	—	—	—	✓	✓	✓	Section 14
		P10_6	✓	✓	✓	✓	✓	✓	
	CSIG0SC	P0_14	—	—	—	✓	✓	✓	
		P10_7	✓	✓	✓	✓	✓	✓	
	CSIG1SO	P11_9	—	—	—	—	✓	✓	
	CSIG1SC	P11_10	—	—	—	—	✓	✓	

Table 2.70 Alternative Modes that Require Setting PIPCn.PIPCn\_m = 1 (2/2)

Function	Alternative Functions Name	Port Name	Device						Reference Section
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	
CSIH	CSIH0SO	P0_3	✓	✓	✓	✓	✓	✓	Section 15
	CSIH0SC	P0_2	✓	✓	✓	✓	✓	✓	
	CSIH1SO	P0_5	—	—	✓	✓	✓	✓	
		P10_2	—	—	✓	✓	✓	✓	
	CSIH1SC	P0_6	—	—	✓	✓	✓	✓	
		P10_1	—	—	✓	✓	✓	✓	
	CSIH2SO	P11_2	—	—	✓	✓	✓	✓	
	CSIH2SC	P11_3	—	—	✓	✓	✓	✓	
CSIH3	CSIH3SO	P11_6	—	—	—	✓	✓	✓	Section 15
	CSIH3SC	P11_7	—	—	—	✓	✓	✓	

### 2.11.3.2 Input Buffer Control (PISn, JPISA0)

The port input buffer characteristics (type 1/type 2) of this device can be selected using the PISn register. The applicable pins are shown in the following table. The pins not listed in the table only support type 2 (SHMT4).

The JTAG port input buffer characteristics (type 2/type 5) of this device can be selected using the JPISA0 register. The applicable pins are shown in **Table 2.72**.

**Table 2.71 Port Input Buffer Characteristics Selection (1/2)**

Port Name	Input Buffer Selection			Device					
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	Type 5	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P0_0	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P0_1	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P0_2	SHMT1	SHMT4	—	—	✓	✓	✓	✓	✓
P0_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P0_4	SHMT1	SHMT4	—	—	✓	✓	✓	✓	✓
P0_5	SHMT1	SHMT4	—	—	✓	✓	✓	✓	✓
P0_6	SHMT1	SHMT4	—	—	✓	✓	✓	✓	✓
P0_7	SHMT1	SHMT4	—	—	—	✓	✓	✓	✓
P0_9	SHMT1	SHMT4	—	—	—	—	✓	✓	✓
P0_11	SHMT1	SHMT4	—	—	—	✓	✓	✓	✓
P0_12	SHMT1	SHMT4	—	—	—	✓	✓	✓	✓
P0_13	SHMT1	SHMT4	—	—	—	—	✓	✓	✓
P1_0	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P1_2	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P1_4	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P1_6	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P1_8	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P1_10	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P1_12	SHMT1	SHMT4	—	—	—	—	—	—	✓
P1_14	SHMT1	SHMT4	—	—	—	—	—	—	✓
P2_0	SHMT1	SHMT4	—	—	—	—	—	—	✓
P2_2	SHMT1	SHMT4	—	—	—	—	—	—	✓
P2_4	SHMT1	SHMT4	—	—	—	—	—	—	✓
P10_0	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P10_2	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P10_3	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P10_4	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P10_6	SHMT1	SHMT4	—	—	✓	✓	✓	✓	✓
P10_9	SHMT1	SHMT4	—	✓	✓	✓	✓	✓	✓
P10_11	SHMT1	SHMT4	—	—	✓	✓	✓	✓	✓
P10_13	SHMT1	SHMT4	—	—	—	✓	✓	✓	✓
P10_15	SHMT1	SHMT4	—	—	—	—	✓	✓	✓
P11_1	SHMT1	SHMT4	—	—	—	✓	✓	✓	✓
P11_3	SHMT1	SHMT4	—	—	—	—	*1	✓	✓
P11_5	SHMT1	SHMT4	—	—	—	—	*1	*2	✓

**Table 2.71 Port Input Buffer Characteristics Selection (2/2)**

Port Name	Input Buffer Selection			Device					
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	Type 5	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P11_6	SHMT1	SHMT4	—	—	—	—	✓	✓	✓
P11_9	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P11_12	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P11_13	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P11_15	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P12_1	SHMT1	SHMT4	—	—	—	—	—	✓	✓
P12_3	SHMT1	SHMT4	—	—	—	—	—	—	✓
P20_0	SHMT1	SHMT4	—	—	—	—	—	—	✓
P20_2	SHMT1	SHMT4	—	—	—	—	—	—	✓
P20_4	SHMT1	SHMT4	—	—	—	—	—	✓	✓

Note 1. Only available in F1L for Gateway.

Note 2. Only available in devices with 1.5- and 2-MB code flash memories.

**Table 2.72 JTAG Port Input Buffer Characteristics Selection**

Port Name	Input Buffer Selection			Devices					
	Type 1	Type 2 (JPISA0_m = 0)	Type 5 (JPISA0_m = 1)	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
JP0_0	—	SHMT4	TTL <sup>*1,*2,*3,*4</sup>	✓	✓	✓	✓	✓	✓
JP0_2	—	SHMT4	TTL <sup>*1,*2,*3</sup>	✓	✓	✓	✓	✓	✓
JP0_3	—	SHMT4	TTL <sup>*1,*2</sup>	✓	✓	✓	✓	✓	✓
JP0_4	—	SHMT4	— <sup>*1,*2</sup>	✓	✓	✓	✓	✓	✓

Note 1. TTL is selected for Boundary scan mode without JPISA0 register setting.

Note 2. TTL is selected for Nexus in normal operating mode without JPISA0 register setting.

Note 3. TTL is selected for LPD (4-pin) in normal operating mode without JPISA0 register setting.

Note 4. TTL is selected for LPD (1-pin) in normal operating mode without JPISA0 register setting.

## NOTES

- For the SHMT1, SHMT4, and TTL pin characteristics, see the data sheet.
- For the input buffer after reset, type 2 (SHMT4) is selected.

### 2.11.3.3 Output Buffer Control (PDSC)

The port output driver strength (slow mode/fast mode) can be selected using the PDSCn register. The applicable pins are shown in the following table. Only slow mode is supported for ports other than those listed below.

**Table 2.73 Pins with Selectable Output Driving Ability (1/2)**

Port Name	Device					
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P0_2* <sup>1</sup>	√	√	√	√	√	√
P0_3* <sup>1</sup>	√	√	√	√	√	√
P0_5* <sup>2</sup>	—	—	√	√	√	√
P0_6* <sup>2</sup>	—	—	√	√	√	√
P0_7	—	—	√	√	√	√
P0_13	—	—	—	√	√	√
P0_14	—	—	—	√	√	√
P10_0	√	√	√	√	√	√
P10_1* <sup>2</sup>	√	√	√	√	√	√
P10_2* <sup>2</sup>	√	√	√	√	√	√
P10_3	√	√	√	√	√	√
P10_4	√	√	√	√	√	√
P10_5	√	√	√	√	√	√
P10_6	√	√	√	√	√	√
P10_7	√	√	√	√	√	√
P10_8	—	—	—	—	—	√
P10_9	—	—	—	—	—	√
P10_10	—	—	—	—	—	√
P10_11	—	—	—	—	—	√
P10_12	—	—	—	—	—	√
P10_13	—	—	—	—	—	√
P10_14	—	—	—	—	—	√
P10_15	—	—	—	—	—	√
P11_0	—	—	—	—	—	√
P11_1	—	—	—	—	—	√
P11_2* <sup>2</sup>	—	—	√	√	√	√
P11_3* <sup>2</sup>	—	—	√	√	√	√
P11_4	—	—	—	—	—	√
P11_5	—	—	—	—	—	√
P11_6* <sup>2</sup>	—	—	—	√	√	√
P11_7* <sup>2</sup>	—	—	—	√	√	√
P11_8	—	—	—	—	—	√
P11_9	—	—	—	—	√	√
P11_10	—	—	—	—	√	√
P11_11	—	—	—	—	—	√
P11_13	—	—	—	—	—	√
P11_14	—	—	—	—	—	√
P11_15	—	—	—	—	—	√

**Table 2.73 Pins with Selectable Output Driving Ability (2/2)**

Port Name	Device					
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P12_0	—	—	—	—	✓	✓
P12_1	—	—	—	—	✓	✓
P12_2	—	—	—	—	✓	✓

Note 1. Set fast mode if the load capacitance of CSIH is 100pF.

Note 2. Set fast mode if the load capacitance of CSIH is 50 pF.

## 2.12 Noise Filter & Edge/Level Detector

### 去除噪声和毛刺

The input signals at some pins are passed through a filter to remove noise and glitches. The RH850/F1L supports both analog and digital filters. 支持模拟和数字滤波器  
在信号经过滤波器的时候，其同样支持边沿和电平探测功能

It also supports the function for edge and level detection after the signals have passed through a filter.

The first part of this section provides an overview of port input pins that are equipped with a filter and the filter type, noise filter & edge/level detection control registers and bits, and register addresses.

For details on the digital/analog filter function and noise filter & edge/level detection control registers, see **Section 2.13, Description of Port Noise Filter & Edge/Level Detection**.

### NOTE

In this section, <name> in the noise filter control register represents the peripheral function connected to a filter.

### 2.12.1 Port Filter Assignment

A list of the input pins that incorporate an analog or digital filter is provided below.

#### 2.12.1.1 Input Pins that Incorporate Analog Filter Type A

##### 包含模拟过滤器和边沿/电平检测功能

The input pins of analog filter type A incorporate an analog filter and edge/level detection function. Edge/level detection is controlled by the following registers.

- Filter control register FCLA0CTLm\_<name> ( $m = 0$  to  $7$ )

A dedicated FCLA0CTLm\_<name> register is provided for each pin in a port that incorporates an analog filter.

Table 2.74 Input Pins that Incorporate Analog Filter Type A

Input Signal	FCLA0CTL Register Configuration		Device					
	Register	Address	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
NMI	FCLA0CTL0_NMI	FFC3 4000H	✓	✓	✓	✓	✓	✓
INTP0	FCLA0CTL0_INTPL	FFC3 4020H	✓	✓	✓	✓	✓	✓
INTP1	FCLA0CTL1_INTPL	FFC3 4024H	✓	✓	✓	✓	✓	✓
INTP2	FCLA0CTL2_INTPL	FFC3 4028H	✓	✓	✓	✓	✓	✓
INTP3	FCLA0CTL3_INTPL	FFC3 402CH	✓	✓	✓	✓	✓	✓
INTP4	FCLA0CTL4_INTPL	FFC3 4030H	✓	✓	✓	✓	✓	✓
INTP5	FCLA0CTL5_INTPL	FFC3 4034H	✓	✓	✓	✓	✓	✓
INTP6	FCLA0CTL6_INTPL	FFC3 4038H	—	—	✓	✓	✓	✓
INTP7	FCLA0CTL7_INTPL	FFC3 403CH	—	—	✓	✓	✓	✓
INTP8	FCLA0CTL0_INTPH	FFC3 4040H	—	—	✓	✓	✓	✓
INTP9	FCLA0CTL1_INTPH	FFC3 4044H	—	—	—	—	✓	✓
INTP10	FCLA0CTL2_INTPH	FFC3 4048H	✓	✓	✓	✓	✓	✓
INTP11	FCLA0CTL3_INTPH	FFC3 404CH	✓	✓	✓	✓	✓	✓
INTP12	FCLA0CTL4_INTPH	FFC3 4050H	—	—	✓	✓	✓	✓
INTP13	FCLA0CTL5_INTPH	FFC3 4054H	—	—	—	✓	✓	✓
INTP14	FCLA0CTL6_INTPH	FFC3 4058H	—	—	—	—	✓	✓
INTP15	FCLA0CTL7_INTPH	FFC3 405CH	—	—	—	—	✓	✓

### 2.12.1.2 Input Pins that Incorporate Analog Filter Type B

The input pins of analog filter type B incorporate an analog filter. Edge/level detection is controlled by the registers for individual peripheral functions.

**Table 2.75 Input Pins that Incorporate Analog Filter Type B**

Input Signal	Edge/Level Detection	Device					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
TAUJ0I0	Edge detection* <sup>1</sup>	✓	✓	✓	✓	✓	✓
TAUJ0I1	Edge detection* <sup>1</sup>	✓	✓	✓	✓	✓	✓
TAUJ0I2	Edge detection* <sup>1</sup>	✓	✓	✓	✓	✓	✓
TAUJ0I3	Edge detection* <sup>1</sup>	✓	✓	✓	✓	✓	✓
TAUJ1I0	Edge detection* <sup>1</sup>	—	—	—	✓	✓	✓
TAUJ1I1	Edge detection* <sup>1</sup>	—	—	—	✓	✓	✓
TAUJ1I2	Edge detection* <sup>1</sup>	—	—	—	✓	✓	✓
TAUJ1I3	Edge detection* <sup>1</sup>	—	—	—	✓	✓	✓
TAPA0ESO	Edge detection* <sup>2</sup>	✓	✓	✓	✓	✓	✓
KR0I0	Low level detection	✓	✓	✓	✓	✓	✓
KR0I1	Low level detection	✓	✓	✓	✓	✓	✓
KR0I2	Low level detection	✓	✓	✓	✓	✓	✓
KR0I3	Low level detection	✓	✓	✓	✓	✓	✓
KR0I4	Low level detection	✓	✓	✓	✓	✓	✓
KR0I5	Low level detection	✓	✓	✓	✓	✓	✓
KR0I6	Low level detection	—	✓	✓	✓	✓	✓
KR0I7	Low level detection	—	✓	✓	✓	✓	✓

Note 1. For details on edge detection for TAUJ, see **Section 24.3.3.4, TAUJnCMURm — TAUJn Channel Mode User Register**.

Note 2. For details on edge detection for TAPA, see **Section 27.3.2, TAPAnCTL0 — TAPA Control Register 0**.

### 2.12.1.3 Input Pins that Incorporate Analog Filter Type C

The input pins of analog filter type C only incorporate an analog filter function.

仅仅包含一个模拟过滤器功能

**Table 2.76 Input Pins that Incorporate Analog Filter Type C**

Input Signal
FLMD0
FLMD1
MODE0
MODE1
RESET
DCUTDI
DCUTRST

数字滤波器类型D

#### 2.12.1.4 Input Pins that Incorporate Digital Filter Type D

The input pins of digital filter type D incorporate a digital filter and edge detection function. The digital filter and edge detection are controlled by the following registers.

- Filter control register FCLA0CTLm\_<name> ( $m = 0$ )  
Each port with a digital filter has a special FCLA0CTLm\_<name> register.
- Digital noise elimination control register DNFA<name>CTL  
Each DNFA<name>CTL control register controls digital filter processing for three input signals per group.
- Digital noise elimination enable register DNFA<name>EN  
The setting of the DNFA<name>ENL[2:0] bits in DNFA<name>EN enables or disables digital noise elimination for three input signals per group.

**Table 2.77** Input Pins that Incorporate Digital Filter Type D

Input Pin	Device				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register		Filter Control Register			
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	Control Register	Address	Control Register	Address	Control Register	Address
ADCA0TRG0	✓	✓	✓	✓	✓	✓	DNFA ADCTL0CTL	FFC3 00A0H	DNFA ADCTL0EN (DNFAA DCTL0ENL)	FFC3 00A4H (FFC3 00ACCH)	FCLA0CTL0 _ADC0	FFC3 4060H
ADCA0TRG1	✓	✓	✓	✓	✓	✓			DNFAADCTL0 ENL1		FCLA0CTL1 _ADC0	FFC3 4064H
ADCA0TRG2	✓	✓	✓	✓	✓	✓			DNFAADCTL0 ENL2		FCLA0CTL2 _ADC0	FFC3 4068H
ADCA1TRG0	—	—	—	—	✓	✓	DNFA ADCTL1CTL	FFC3 00C0H	DNFA ADCTL1EN (DNFAA DCTL1ENL)	FFC3 00C4H (FFC3 00CCCH)	FCLA0CTL0 _ADC1	FFC3 4080H
ADCA1TRG1	—	—	—	—	✓	✓			DNFAADCTL1 ENL1		FCLA0CTL1 _ADC1	FFC3 4084H
ADCA1TRG2	—	—	—	—	✓	✓			DNFAADCTL1 ENL2		FCLA0CTL2 _ADC1	FFC3 4088H

### 2.12.1.5 Input Pins that Incorporate Digital Filter Type E

The input pins of digital filter type E incorporate a digital filter. The digital filter is controlled by the following registers. Edge detection is controlled by the registers for individual peripheral functions.

- Digital noise elimination control register DNFA<name>CTL  
Each DNFA<name>CTL control register controls digital filter processing for up to 16 input signals per group.
- Digital noise elimination enable register DNFA<name>EN  
The setting of the DNFA<name>ENL[7:0] and DNFA<name>ENH[7:0] bits in DNFA<name>EN enables or disables digital noise elimination for up to 16 input signals per group.

Table 2.78 Input Pins that Incorporate Digital Filter Type E (1/3)

Input Pin	Device				Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register		Edge Detection Register Name		
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	Control Register	Address	Control Register	Control Bit	Address
TAUD010	✓	✓	✓	✓	✓	✓	DNFATAUD01CTL	FFC3 0000 <sub>H</sub>	DNFATAUD01ENL0	DNFATAUD01ENL0	FFC3 0004 <sub>H</sub> (FFC3 0008 <sub>H</sub> / FFC3 000C <sub>H</sub> )
TAUD011	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL1	DNFATAUD01ENL1	
TAUD012	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL2	DNFATAUD01ENL2	
TAUD013	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL3	DNFATAUD01ENL3	
TAUD014	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL4	DNFATAUD01ENL4	
TAUD015	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL5	DNFATAUD01ENL5	
TAUD016	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL6	DNFATAUD01ENL6	
TAUD017	✓	✓	✓	✓	✓	✓			DNFATAUD01ENL7	DNFATAUD01ENL7	
TAUD018	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH0	DNFATAUD01ENH0	
TAUD019	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH1	DNFATAUD01ENH1	
TAUD010	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH2	DNFATAUD01ENH2	
TAUD011	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH3	DNFATAUD01ENH3	
TAUD012	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH4	DNFATAUD01ENH4	
TAUD013	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH5	DNFATAUD01ENH5	
TAUD014	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH6	DNFATAUD01ENH6	
TAUD015	✓	✓	✓	✓	✓	✓			DNFATAUD01ENH7	DNFATAUD01ENH7	

Table 2.78 Input Pins that Incorporate Digital Filter Type E (2/3)

Input Pin	Device				Digital Noise Elimination Control Register			Digital Noise Elimination Enable Register			Edge Detection Register Name *2
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	Control Register	Address	Control Register	Control Bit	
TAUB010	—	—	—	—	✓	✓	DNFA TAUB0ICCTL	FFC3 0020 <sub>H</sub>	DNFA TAUB0IEN	DNFATAUB0IENL0	FFC3 0024 <sub>H</sub> (FFC3 0028 <sub>H</sub> FFC3 002C <sub>H</sub> )
TAUB011	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL1	
TAUB012	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL2	
TAUB013	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL3	
TAUB014	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL4	
TAUB015	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL5	
TAUB016	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL6	
TAUB017	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENL7	
TAUB018	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH0	
TAUB019	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH1	
TAUB010	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH2	
TAUB011	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH3	
TAUB012	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH4	
TAUB013	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH5	
TAUB014	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH6	
TAUB015	—	—	—	—	✓	✓			DNFA TAUB0IEN	DNFATAUB0IENH7	

Table 2.78 Input Pins that Incorporate Digital Filter Type E (3/3)

Input Pin	Device				Digital Noise Elimination Control Register				Digital Noise Elimination Enable Register				Edge Detection Register Name
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	Control Register	Address	Control Register	Control Bit	Address		
TAUB110	—	—	—	—	—	—	DNFA TAUB1ICTL	FFC3 0040 <sub>H</sub>	DNFA TAUB1IE	(DNFA TAUB1ENH/DNFA TAUB1ENL)	FFC3 0044 <sub>H</sub> (FFC3 0048 <sub>H</sub> FFC3 004C <sub>H</sub> )	*2	
TAUB111	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 1			
TAUB112	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 2			
TAUB113	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 3			
TAUB114	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 4			
TAUB115	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 5			
TAUB116	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 6			
TAUB117	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENL 7			
TAUB118	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 0			
TAUB119	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 1			
TAUB110	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 2			
TAUB111	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 3			
TAUB112	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 4			
TAUB113	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 5			
TAUB114	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 6			
TAUB115	—	—	—	—	—	—			DNFA TAUB1IE	DNFA TAUB1ENH 7			
ENCAGTINO	✓	✓	✓	✓	✓	✓	DNFA ENCA0ICCTL	FFC3 0060 <sub>H</sub>	DNFA ENCA0IE	(DNFA ENCA0IE)	FFC3 0064 <sub>H</sub> (FFC3 006C <sub>H</sub> )	*3	
ENCAGTIN1	✓	✓	✓	✓	✓	✓			DNFA ENCA0IE	DNFA ENCA0IE 1			
ENCAGE0	✓	✓	✓	✓	✓	✓			DNFA ENCA0IE	DNFA ENCA0IE 2			
ENCAGE1	✓	✓	✓	✓	✓	✓			DNFA ENCA0IE	DNFA ENCA0IE 3			
ENCAGEC	✓	✓	✓	✓	✓	✓			DNFA ENCA0IE	DNFA ENCA0IE 4			

Note 1. For the setting for TAUD edge detection, see Section 23.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register.

Note 2. For the setting for TAUB edge detection, see Section 22.3.3.4, TAUBnCMURm — TAUBn Channel Mode User Register

Note 3. For the setting for ENCA edge detection, see Section 26.3.3, ENCA<sub>n</sub>O<sub>C0</sub> — ENCA<sub>n</sub> I/O Control Register 0.

## 2.12.2 Clock Supply for Port Filters

The following table shows the clock supply for each filter type in each port domain.

Table 2.79 Clock Supply for Port Filters

Peripheral Function	Port Domain* <sup>1</sup>	Filter Type	Filter Clock	Setting Register	
				Source Clock Selection	Clock Selection
ADCA0	Always-On area	Digital filter type D	DNFATCKI	CKSC_AADCAS_CTL	CKSC_AADCAD_CTL
ADCA1	Isolated area	Digital filter type D	DNFATCKI	CKSC_IADCAS_CTL	CKSC_IADCAD_CTL
TAUD0	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
TAUB0	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
TAUB1	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
ENCA0	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—

Note 1. Power supply domain

### NOTE

For the Setting Register, see **Section 10.4.3, Clock Selector Control Register**.

## 2.13 Description of Port Noise Filter & Edge/Level Detection

External signals pass through different types of filters according to the use of each external input signal.

### NOTE

In this section, <name> in the noise filter control register represents the peripheral function connected to a filter.

### 2.13.1 Overview

#### 2.13.1.1 Analog Filter Types

Analog filters have fixed characteristics.

- Type A: An analog filter with edge detection or level detection.  
Used for external interrupt signals.
- Type B: An analog filter  
Edge detection is performed by each peripheral function. Used for the timer input signals, asynchronous Hi-Z control input signals, and key return input signals.
- Type C: An analog filter only  
Used for the external  $\overline{\text{RESET}}$  input and mode signals.

#### 2.13.1.2 Digital Filter Types

The digital filter characteristics can be adjusted to suit the application.

- Type D: A digital filter with edge detection.  
Used for the A/D converter external trigger pin.
- Type E: A digital filter. Edge detection is performed by each peripheral function.  
Used for the timer input signals and encoder input signals.

## 2.13.2 Analog Filters

### 2.13.2.1 Analog Filter Characteristic

See the data sheet for the input conditions for signals input to pins that incorporate an analog filter.

### 2.13.2.2 Analog Filter Control Registers

A dedicated FCLA0CTLm\_<name> register or control register in the peripheral macro is provided for input pins that incorporate an analog filter.

The assignment of the input signals to the control registers and their addresses are given in **Table 2.74, Input Pins that Incorporate Analog Filter Type A**, in **Section 2.12.1, Port Filter Assignment**.

### 2.13.2.3 Analog Filter in Standby Mode

Analog filters for the function of waking-up from the DeepSTOP mode are located in the always-on area (AWO). Analog filters in the AWO area always operate.

The analog filter in standby mode and its wake-up capability depend on the filter types. See the description of the analog filter types below.

#### (1) Analog Filter Type A

A block diagram of analog filter type A is shown below.

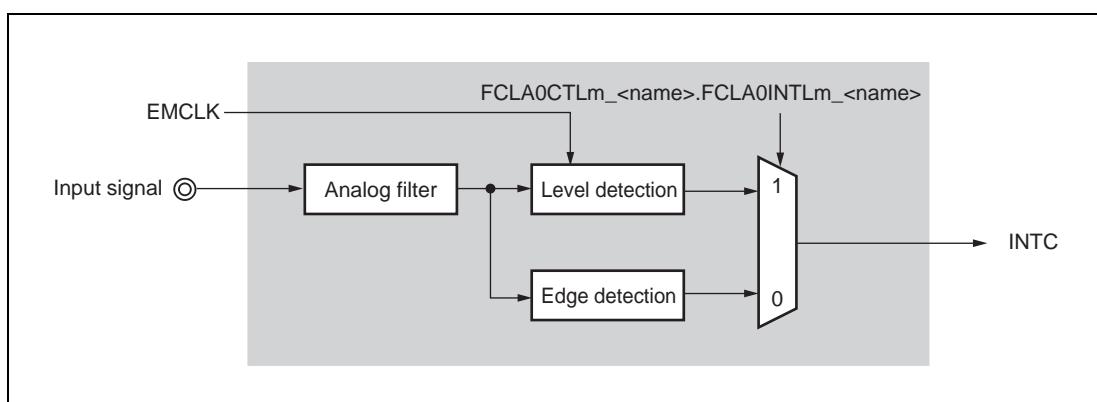


Figure 2.13 Block Diagram of Analog Filter Type A

After passing an external signal through the filter to eliminate noise and glitches, an output signal is generated according to whether an event is detected; that is whether a specified level is detected or whether a change in the level (an edge) occurs.

Whether a level or an edge is detected is selected by the control bit `FCLA0CTLm_<name>.FCLA0INTLm_<name>`.

- FCLA0INTLm\_<name> bit = 0: Edge detection

Whether a rising or falling edge is detected can be specified by setting the `FCLA0CTLm_<name>.FCLA0INTRm_<name>` and `FCLA0CTLm_<name>.FCLA0INTFm_<name>` bits.

- FCLA0INTLm\_<name> bit = 1: Level detection

The detection of a high level or low level can be specified by setting `FCLA0CTLm_<name>.FCLA0INTRm_<name>` bit.

The table below summarizes the detection conditions of the analog filter.

**Table 2.80 Analog Filter Event Detection Conditions**

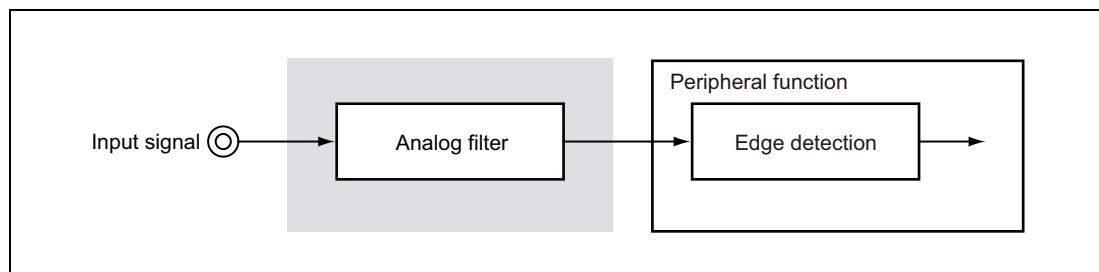
FCLA0INTLm_<name>	FCLA0INTFm_<name>	FCLA0INTRm_<name>	Edge Detection	Level Detection
0	0	0	No edge detected	Disabled
	0	1	Rising edge	
	1	0	Falling edge	
	1	1	Rising and falling edges	
1	X	0	Disabled	Low level
	X	1		High level

### Analog filter type A in Standby mode

The output signal of an analog filter type A can always be used as a standby mode wake-up signal.

#### (2) Analog filter type B

A block diagram of analog filter type B is shown below.



**Figure 2.14 Block Diagram of Analog Filter Type B**

### Analog filter type B in Standby mode

The output signal of an analog filter type B can always be used as a standby mode wake-up signal.

### (3) Analog filter type C

A block diagram of analog filter type C is shown below.

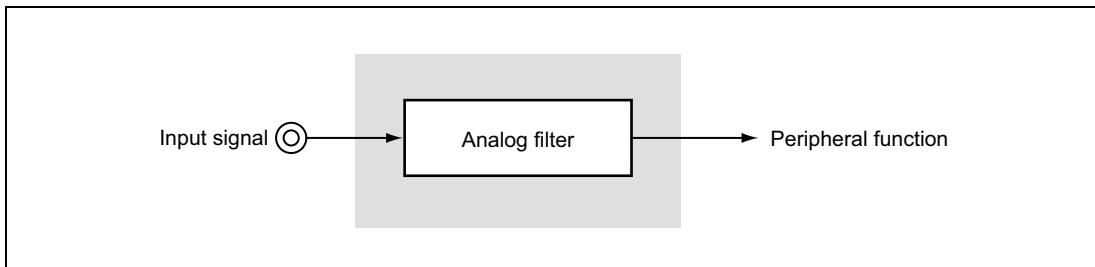


Figure 2.15 Block Diagram of Analog Filter Type C

The generated signals are always input signals that have passed through an analog filter.

#### Analog filter type C in Standby mode

Pins equipped with type C analog filters in this product do not support the input of event signals to trigger wake-up from standby.

## 2.13.3 Digital Filters

### 2.13.3.1 Digital Filter Characteristic

The digital filters allow the filter characteristics to be adjusted according to the needs of the application.

The input signal is sampled with the sampling frequency  $f_s$ .

If a specified number of successive samples yield the same (high or low) level, the signal level is judged as valid and the filter output signal is set accordingly.

If an external signal level change is detected within the specified number of samples (same level samples), the signal level is judged as noise and the filter output signal does not change.

The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.

Both parameters can be specified:

- $\text{DNFA}_{\text{name}}\text{CTL.DNFA}_{\text{name}}\text{PRS}[2:0]$  select the sampling frequency based on  $f_s = f_{\text{DNFATCKI}} / 2^{\text{DNFA}_{\text{name}}\text{PRS}[2:0]}$   
where  $f_{\text{DNFATCKI}}$  is the frequency of the DNFATCKI clock.
- $\text{DNFA}_{\text{name}}\text{CTL.DNFA}_{\text{name}}\text{NFSTS}[1:0]$  determines the number of same level samples (2 to 5):

$$s = \text{DNFA}_{\text{name}}\text{NFSTS}[1:0] + 2$$

External signal pulse shorter than the following are suppressed at all times.

$$s \times 1/f_s$$

External signal pulses longer than the following are always judged as valid and are passed on to the filter output.

$$(s + 1) \times 1/f_s$$

External signal pulses in the following range may be suppressed or judged as valid.

$$s \times 1/f_s \text{ to } (s + 1) \times 1/f_s$$

The filter operation is illustrated in the figure below with  $\text{DNFA}_{\text{name}}\text{NFSTS}[1:0] = 01_B$ , i.e.  $s = 3$  same level samples.

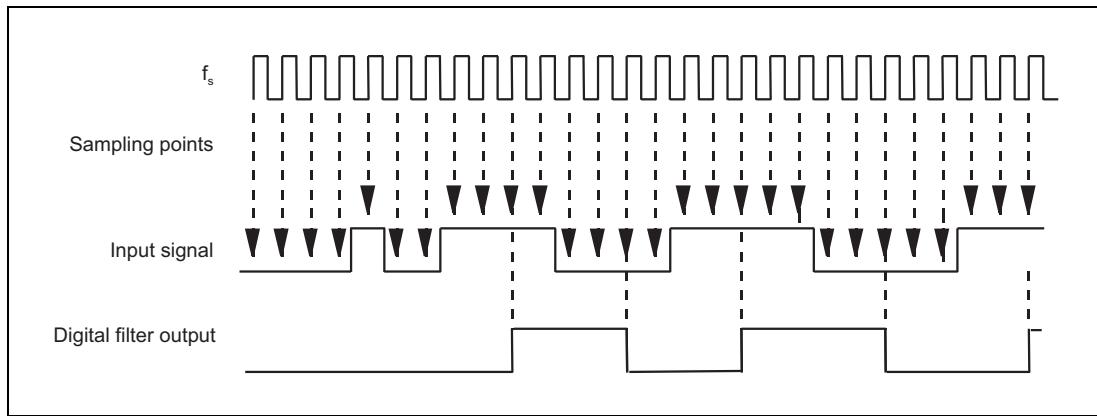


Figure 2.16 Digital Filter Function

### 2.13.3.2 Digital Filter Groups

The input signals equipped with digital filters are ordered in groups of up to 16 signals.

The digital filter characteristics, specified by  $\text{DNFA<name>} \text{CTL.DNFA<name>} \text{PRS[2:0]}$  and  $\text{DNFA<name>} \text{NFSTS[1:0]}$  apply to the signals group.

However, the digital filter for each signal can be enabled or disabled separately by  $\text{DNFA<name>} \text{EN.DNFA<name>} \text{ENLm}$  ( $m = 0$  to  $7$ ) and  $\text{DNFA<name>} \text{EN.DNFA<name>} \text{ENHm}$  ( $m = 0$  to  $7$ ).

#### CAUTIONS

- When the output signal from the digital filter is input to an alternative function, allow at least the following interval to elapse after the digital filter is enabled ( $\text{DNFA<name>} \text{EN.DNFA<name>} \text{ENLm}$  ( $m = 0$  to  $7$ ) = 1 and  $\text{DNFA<name>} \text{EN.DNFA<name>} \text{ENHm}$  ( $m = 0$  to  $7$ ) = 1) for the port pin to switch to the alternative function.

$$s = \text{DNFA<name>} \text{NFSTS[1:0]} + 2$$

$$s \times 1/f_s + 2 \times 1/f_{\text{DNFATCKI}}$$

- When a digital filter's output signal is used as an interrupt signal, only enable the digital filter ( $\text{DNFA<name>} \text{EN.DNFA<name>} \text{ENLm}$  ( $m = 0$  to  $7$ ) = 1 and  $\text{DNFA<name>} \text{EN.DNFA<name>} \text{ENHm}$  ( $m = 0$  to  $7$ ) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.

$$s \times 1/f_s + 3 \times 1/f_{\text{DNFATCKI}}$$

### 2.13.3.3 Digital Filters in Standby Mode

Digital filters for the function of waking-up from the DeepSTOP mode are located in the always-on area (AWO). Digital filters on the Always-On-Area (AWO) are always operating.

Digital noise elimination requires the clock supply DNFATCKI to operate.

Pins equipped with digital filters in this product do not support the input of event signals to trigger wake-up from standby.

### 2.13.3.4 Digital Filter Control Registers

For each group consisting of up to 16 digital filters, the digital noise elimination control register DNFA<name>CTL and digital noise elimination enable register DNFA<name>EN are used to set all the filters in the same group (<name> = peripheral function group).

The DNFA<name>CTL register specifies the characteristics of the digital noise elimination filter for the digital filter of <name>.

The DNFA<name>EN register enables/disables each filter by setting the corresponding bit in DNFA<name>EN.DNFA<name>ENLm ( $m = 0$  to  $7$ ) and DNFA<name>EN.DNFA<name>ENHm ( $m = 0$  to  $7$ ).

The edge detection setup is done via the filter dedicated control register and the registers for individual peripheral functions.

The FCLA0CTL0\_ADCn registers are ordered in groups of 3 registers with the same index n. The register index n is 0 or 1.

The assignment of the input signals to the control registers and their addresses are given in **Table 2.81, Output Options for Digital Filter Type D** in **Section 2.12.1, Port Filter Assignment**.

#### CAUTION

**Do not change any control register settings while the corresponding digital filter is enabled by DNFA<name>EN.DNFA<name>ENLm ( $m = 0$  to  $7$ ) = 1 and DNFA<name>EN.DNFA<name>ENHm ( $m = 0$  to  $7$ ) = 1. Otherwise an unintended filter output may be generated.**

#### (1) Digital filter type D

A block diagram of digital filter type D is shown below.

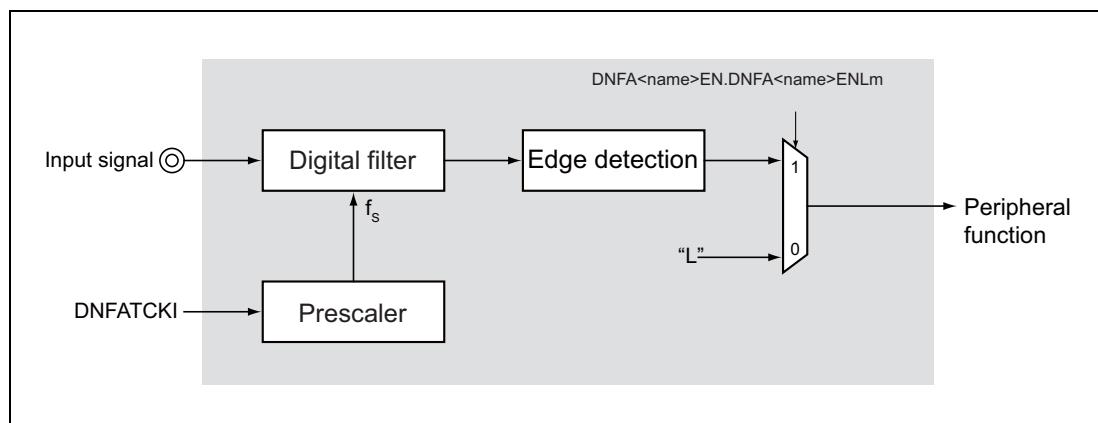


Figure 2.17 Block Diagram of Digital Filter Type D

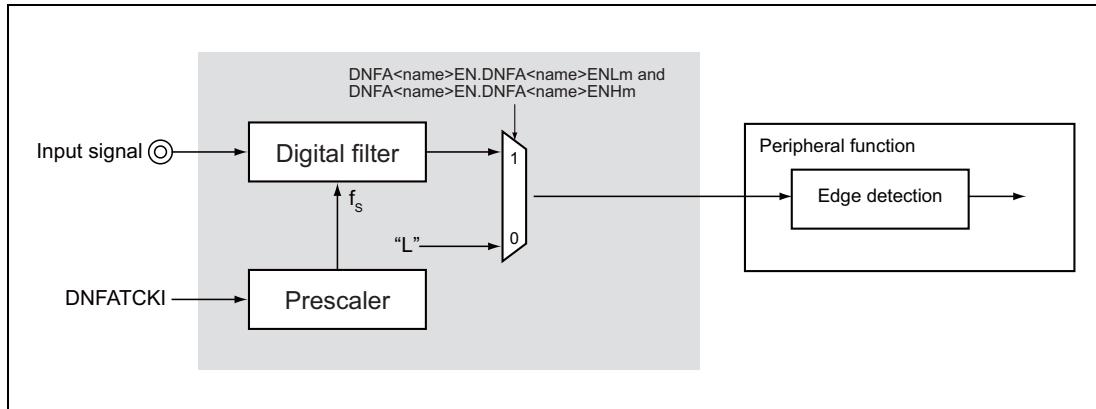
The generated signal depends on the register setting, as shown in the following table.

Table 2.81 Output Options for Digital Filter Type D

DNFA<name>EN.DNFA<name>ENLm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

## (2) Digital filter type E

A block diagram of digital filter type E is shown below.



**Figure 2.18 Block Diagram of Digital Filter Type E**

The generated signal depends on the register setting, as shown in the following table.

**Table 2.82 Output Options for Digital Filter Type E**

DNFA<name>EN.DNFA<name>ENLm and DNFA<name>EN.DNFA<name>ENHm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

### 2.13.4 Filter Control Registers

The analog and digital filters are controlled and operated by the following registers:

**Table 2.83 List of Filter Registers**

Register Name	Symbol	Address
Filter control register m	FCLA0CTLm_<name>	The addresses are shown in the tables in <b>Section 2.12.1, Port Filter Assignment</b> .
Digital noise elimination control register	DNFA<name>CTL	
Digital noise elimination enable register	DNFA<name>EN	

### 2.13.4.1 FCLA0CTLm\_<name> — Filter Control Register

This register controls the analog and digital filter operation.

**Access:** This register can be read or written in 8-bit units.

**Address:** The allocation of input signals to FCLA0CTLm\_<name> registers and the address of each register are shown in the tables in **Section 2.12.1, Port Filter Assignment**.

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	FCLA0INTLm_<name>	FCLA0INTFm_<name>	FCLA0INTRm_<name>
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

**Table 2.84 FCLA0CTLm\_<name> Register Contents**

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	FCLA0INTLm_<name>	Detection Mode Selection 0: Edge detection 1: Level detection  <b>Note:</b> This bit is only valid for analog filter type A.
1	FCLA0INTFm_<name>	<ul style="list-style-type: none"> <li>In level detection mode (FCLA0INTLm_&lt;name&gt; = 1): This bit has no effect.</li> <li>In edge detection mode (FCLA0INTLm_&lt;name&gt; = 0): Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled</li> </ul> <b>Note:</b> This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.
0	FCLA0INTRm_<name>	<ul style="list-style-type: none"> <li>In level detection mode (FCLA0INTLm_&lt;name&gt; = 1): Detected level selection 0: Low level detection 1: High level detection</li> <li>In edge detection mode (FCLA0INTLm_&lt;name&gt; = 0): Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled</li> </ul> <b>Note:</b> This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.

#### CAUTION

Digital filter type D: Always set bit 2 to "0".

### 2.13.4.2 DNFA<name>CTL — Digital Noise Elimination Control Register

This register is used to specify the filter characteristics of the digital noise elimination filter.

#### NOTE

This register is only valid for digital filter type D and digital filter type E.

**Access:** This register can be read or written in 8-bit units.

**Address:** For the correspondence between the DNFA<name>CTL register and input signals, and the addresses of individual registers, see **Table 2.77, Input Pins that Incorporate Digital Filter Type D** and **Table 2.78, Input Pins that Incorporate Digital Filter Type E** in **Section 2.12.1, Port Filter Assignment**.

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	DNFA<name>NFSTS[1:0]		—	—		DNFA<name>PRS[2:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

**Table 2.85 DNFA<name>CTL Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 5	DNFA<name> NFSTS[1:0]	The DNFA<name>NFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid.
	DNFA<name>NFSTS[1:0]	Number of Samples
	00 <sub>B</sub>	2
	01 <sub>B</sub>	3
	10 <sub>B</sub>	4
	11 <sub>B</sub>	5
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DNFA<name> PRS[2:0]	Digital filter sampling clock selection
	DNFA<name>PRS[2:0]	Sampling Clock Frequency
	000 <sub>B</sub>	DNFATCKI/1
	001 <sub>B</sub>	DNFATCKI/2
	010 <sub>B</sub>	DNFATCKI/4
	011 <sub>B</sub>	DNFATCKI/8
	100 <sub>B</sub>	DNFATCKI/16
	101 <sub>B</sub>	DNFATCKI/32
	110 <sub>B</sub>	DNFATCKI/64
	111 <sub>B</sub>	DNFATCKI/128

### 2.13.4.3 DNFA<name>EN — Digital Noise Elimination Enable Register

This register enables and disables digital noise elimination for a specified input signal.

#### NOTE

This register is only valid for digital filter type D and digital filter type E.

**Access:** This register can be read or written in 16-bit units.

The higher- and lower-order bytes (DNFA<name>ENH[7:0] and DNFA<name>ENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFA<name>ENH and DNFA<name>ENL.

**Address:** For the correspondence between the DNFA<name>EN register and input signals, and the addresses of individual registers, see **Table 2.78, Input Pins that Incorporate Digital Filter Type E** and **Table 2.77, Input Pins that Incorporate Digital Filter Type D** in **Section 2.12.1, Port Filter Assignment**.

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNFA<name> ENH7	DNFA<name> ENH6	DNFA<name> ENH5	DNFA<name> ENH4	DNFA<name> ENH3	DNFA<name> ENH2	DNFA<name> ENH1	DNFA<name> ENH0	DNFA<name> ENL7	DNFA<name> ENL6	DNFA<name> ENL5	DNFA<name> ENL4	DNFA<name> ENL3	DNFA<name> ENL2	DNFA<name> ENL1	DNFA<name> ENL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 2.86 DNFA<name>EN Register Contents**

Bit Position	Bit Name	Function
15 to 0	DNFA<name> ENH[7:0]	Digital Noise Elimination Enable/Disable Control 0: Digital noise elimination disabled 1: Digital noise elimination enabled

## Section 3 CPU System

This section describes the CPU-related items included in the *RH850G3K User's Manual: Software* that require specific attention when using the RH850/F1L, as well as CPU-related items that apply specifically to the RH850/F1L and are therefore not included in the *RH850G3K User's Manual: Software*.

### 3.1 Overview

#### 3.1.1 Architectural Features

The CPU incorporated in the RH850/F1L has a Harvard architecture and uses a RISC instruction set.

This CPU has a 5-stage pipeline and can execute almost all instructions in a single clock cycle.

The CPU also uses a 32-bit hardware multiplier, enabling high-speed multiplication processing. A range of instructions such as saturating and bit manipulation instructions are also supported.

##### 3.1.1.1 CPU

- Core: G3K CPU × 1  
Instruction execution time: 12.5 ns (when operating at 80 MHz)  
10.4 ns (when operating at 96 MHz)
- 5-stage pipeline
- CPU operating modes: User mode and supervisor mode.

##### 3.1.1.2 Coprocessor 没有协处理器

- There is no coprocessor (floating-point coprocessor or fixed-point SIMD coprocessor).

##### 3.1.1.3 Exceptions and Interrupts

- Interrupts of the V850E1 CPU and V850E2 CPU are handled as types of exceptions.
  - Interrupt offset addresses are generated by the CPU (not output from the interrupt controller).
  - FE-level interrupts (FENMI and FEINT) and EI-level interrupts (INT) are input from the interrupt controller.
- The direct vector and table reference methods are both supported.  
直接向量和表参考方法
- Up to 8 priority levels can be assigned to interrupts.  
8个优先级

### 3.1.1.4 Memory Management

- Memory space
  - Program area: Linear 128 MB area
  - Data area: Linear 128 MB area
- Processor protection functions
  - Memory Protection Unit (MPU)  
The MPU protects the CPU's memory space against execution or data manipulation that is not allowed for user programs (up to 4 memory protection areas).
  - There is no memory protection setting check function.
- There is no memory management unit (MMU).
- Cache  
There is no cache operation function (cache instructions are not supported).

### 3.1.1.5 System Register Protection

System register accesses can be protected by access permission (user mode or supervisor mode).

### 3.1.1.6 Functions Not Supported

The following functions are not supported by the RH850/F1L:

- Hardware multithreading function
- CPU virtualization support function
- Multiprocessing function

## 3.1.2 Power Supply and Clock

The CPU subsystem is located on the Isolated-Area (ISO) and is supplied with the CPUCLK clock.

### 3.1.3 RH850/F1L CPU Subsystem

The following figure shows a block diagram of the RH850/F1L CPU subsystem.

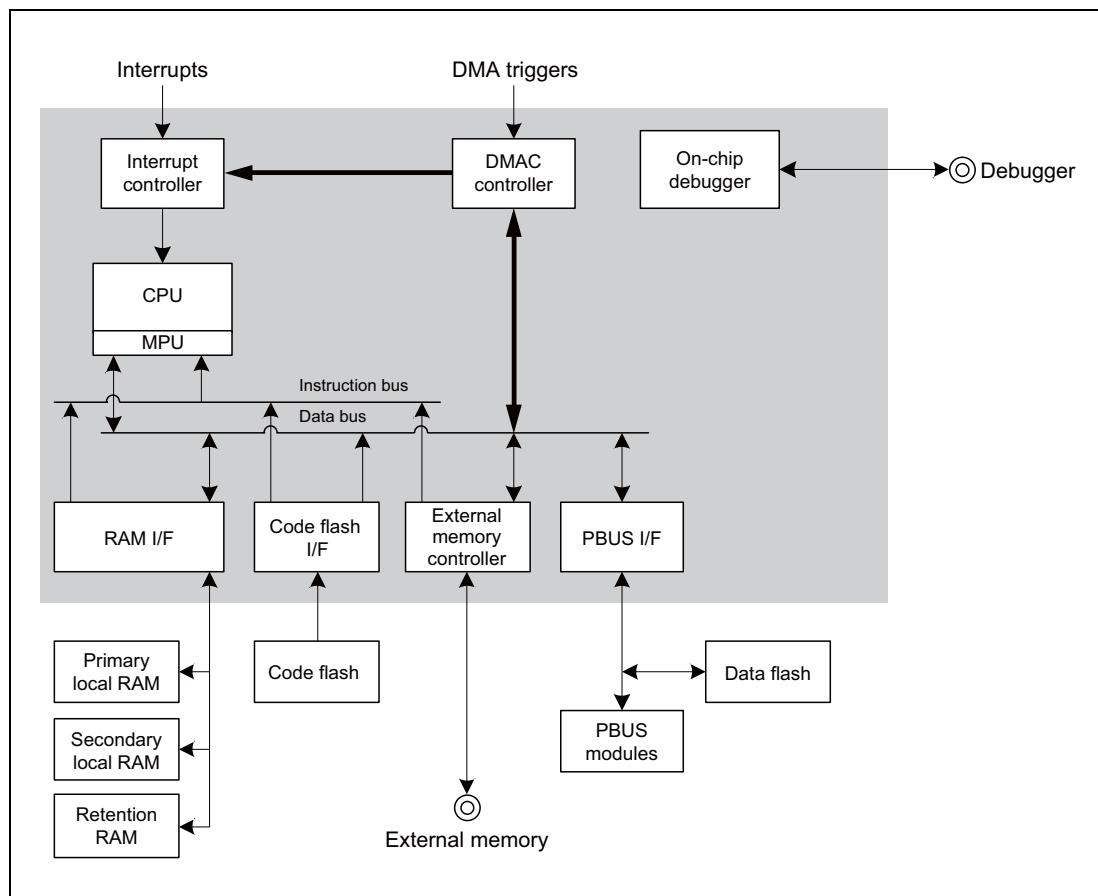


Figure 3.1 RH850/F1L CPU Subsystem

All buses used for data transfer among many peripheral devices are controlled by two master devices.

- CPU
- DMA controller (DMAC)

Table 3.1 RH850/F1L CPU Subsystem Data / Instruction Buses

Master	Bus	Code Flash	Primary Local RAM/ Secondary Local RAM/ Retention RAM	Data Flash	PBUS I/F
CPU	Instruction (32 bits)	R	R	—	—
	Data (32 bits)	R	R/W	R/W <sup>*1</sup>	R/W
DMA controller	DMA data bus (32 bits)	R	R/W	R	R/W

Note 1. Cannot be written by the store instruction.

## 3.2 Processor Model

This section describes the items included in Section 2 of the *RH850G3K User's Manual: Software* that require specific attention when using the RH850/F1L.

### 3.2.1 CPU Operating Mode

The RH850/F1L only supports two CPU operating modes. Resources are managed by a 2-layer control system consisting of a supervisor (SV) layer and a user layer.

All instructions can be executed in supervisor mode. The instructions that can be executed in user mode are limited, and an exception occurs if an unauthorized instruction is executed.

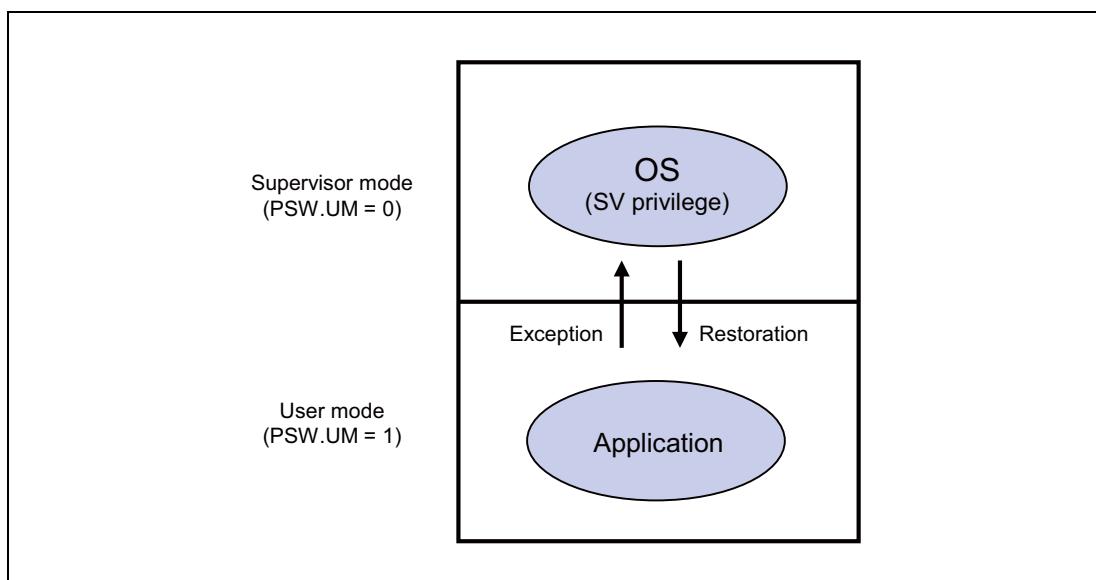


Figure 3.2 CPU Operating Modes

### 3.2.2 Hardware Thread

只是单线程运行

The RH850/F1L only supports single-thread processing. If the HALT instruction is executed, therefore, the CPU stops operating.

Also, when the SNOOZE instruction is executed, the thread that was stopped by the SNOOZE instruction starts operating again after 32 clock cycles have elapsed.

32个时钟周期以后继续执行指令

### 3.2.3 CPU Data Address and Physical Program Address Space

This section describes the CPU's address space, i.e. size and addresses of CPU address space and the physical address space.

The address range of data space and program space together with their wraparound properties are presented.

The CPU supports the following address space.

- 128 MB CPU data address space

128 MB of a 4 GB (max.) address space can be accessed by using the 32-bit general-purpose registers. **4G中的128M是可以通过32bit的通用寄存器访问的**

- 128 MB physical program address space

The CPU provides 128 MB of physical address space to access instruction codes in the program memory. This means that up to 128 MB external or internal memory is accessible.

**高达128M的外部或者内部存储空间是可以被访问的**

### 3.2.3.1 Program Space and Data Space

The figure below shows the assignment of data space and program space in the CPU address space.

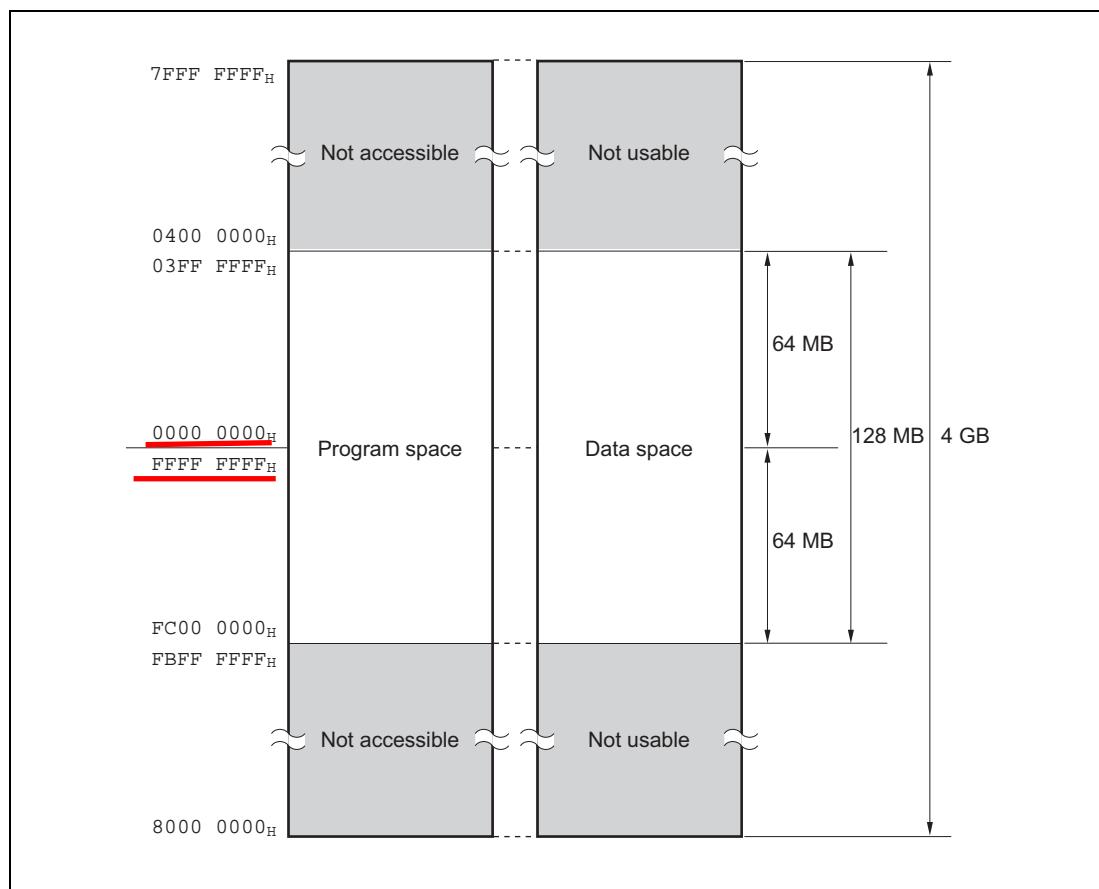


Figure 3.3 CPU Address Space

自己的理解：26位为1，高5bit t均为1，26位为0，高5位均为0

对于指令地址的编址，作为符号扩展位的第26bit t自动被指定为存有指令地址寄存器的高5位

For instruction address addressing, a value whose 26th bit is sign-extended is automatically specified for the higher-order 5 bits of the register holding the instruction address. Therefore, the addressable range is 0000 0000<sub>H</sub> to 03FF FFFE<sub>H</sub> and FC00 0000<sub>H</sub> to FFFF FFFE<sub>H</sub> (and the least significant bit is always 0). Be sure to place the instructions and the tables referenced using the SWITCH, CALLT, and SYSCALL instructions in the instruction addressable address range.

The first address in the table when the SWITCH instruction straddles the boundary of a 128 MB space (the next address of the SWITCH instruction) is handled as follows:

- If the address of the SWITCH instruction is PC = 03FF FFFE<sub>H</sub>, the first address in the table is calculated as (PC + 2) = 0400 0000<sub>H</sub>.
- If the address of the SWITCH instruction is PC = 0400 0000<sub>H</sub> (physical address = FC00 0000<sub>H</sub>), the first address in the table is calculated as (PC + 2) = FC00 0002<sub>H</sub>.

### 3.2.3.2 Wrap-Around of Data Space

If an operand address calculation exceeds 32 bits, only the lower 32 bits of the result are considered.

Therefore, the addresses  $0000\ 0000_H$  and  $FFFF\ FFFF_H$  are contiguous addresses.

This results in a wrap-around of the data space:

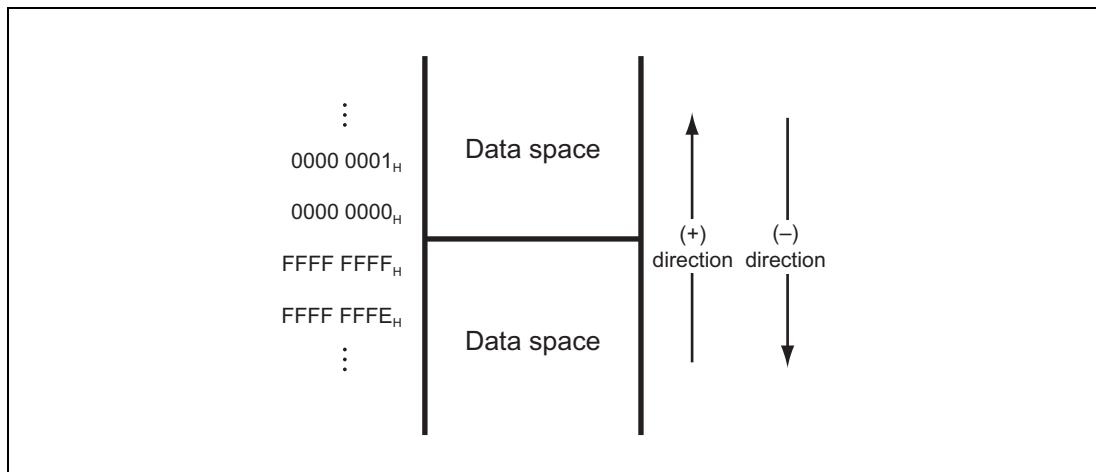


Figure 3.4 Wrap-Around of Data Space

### 3.2.3.3 Wrap-Around of Program Space

If an instruction address calculation exceeds 26 bits, only the lower 26 bits of the result are considered (the value resulting from sign-extending bit 26 is set to the higher-order 5 bits). Therefore, the addresses  $03FF\ FFFF_H$  and  $FC00\ 0000_H$ , and  $FFFF\ FFFF_H$  and  $0000\ 0000_H$  are contiguous addresses. This results in a wrap-around of the program space.

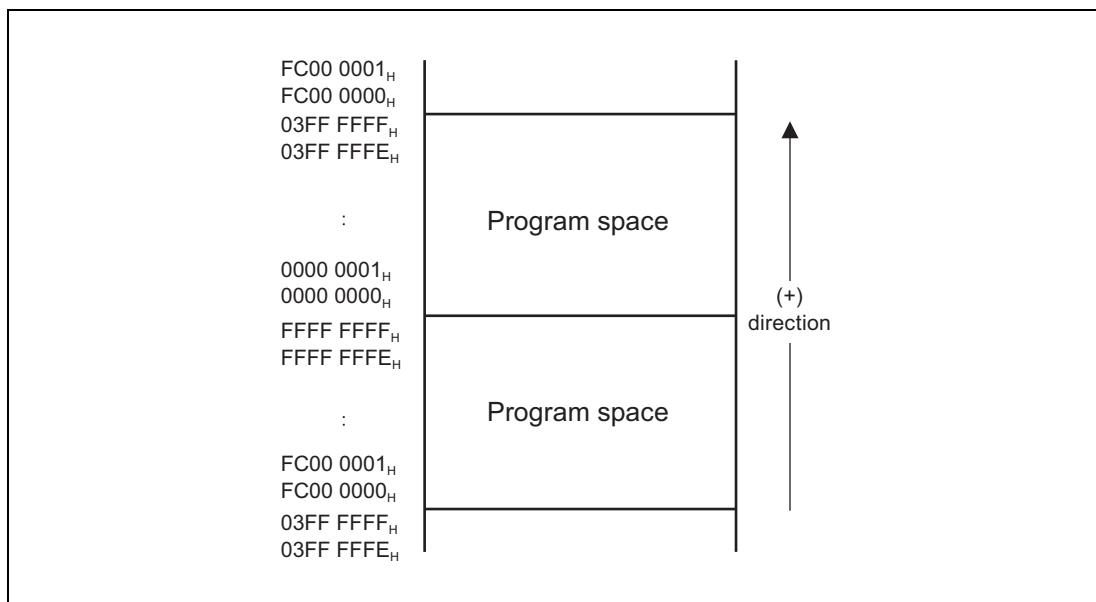
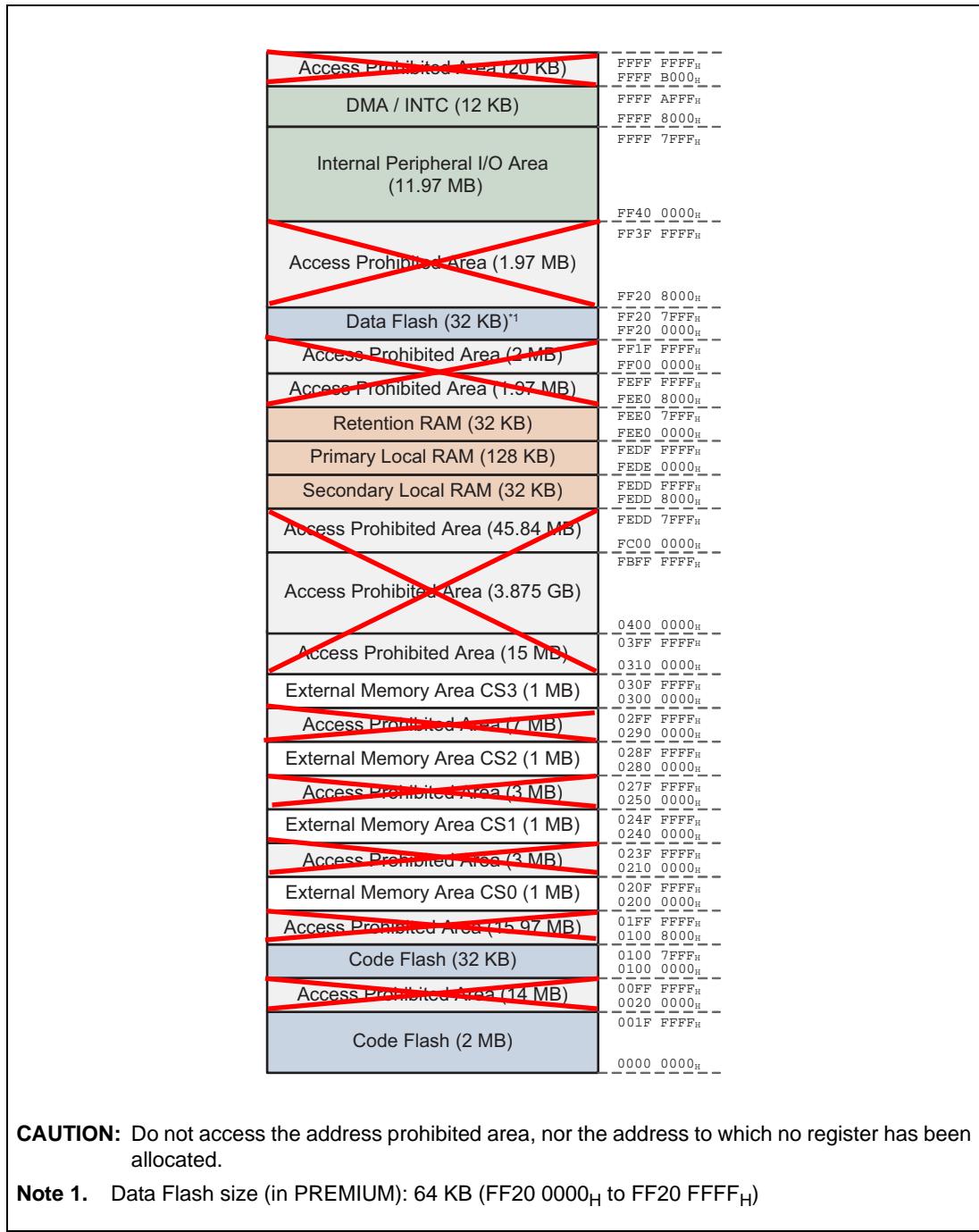


Figure 3.5 Wrap-Around of Program Space

### 3.2.4 RH850/F1L Memory Map

This section describes the CPU memory map, DMA address map, and specific memory space.



**Figure 3.6 RH850/F1L Memory Map (Example of the product with 176 pins and code flash (2 MB))**

The size, and thus the address range, on the code flash and RAM differs depending on the product. The external memory area is only included in F1L devices with 176 pins.

### 3.2.4.1 Code Flash Area

The following table lists the code flash sizes and address ranges.

**Table 3.2 Code Flash Memory Area**

Code Flash Size	Address Range	Product Part Name					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	0000 0000 <sub>H</sub> - 0003 FFFF <sub>H</sub>	R7F7010083AFP R7F7010084AFP	R7F7010113AFP R7F7010114AFP	R7F7010163AFP R7F7010164AFP	R7F7010213AFP R7F7010214AFP	—	—
384 KB	0000 0000 <sub>H</sub> - 0005 FFFF <sub>H</sub>	R7F7010093AFP R7F7010094AFP	R7F7010123AFP R7F7010124AFP	R7F7010173AFP R7F7010174AFP	R7F7010223AFP R7F7010224AFP	—	—
512 KB	0000 0000 <sub>H</sub> - 0007 FFFF <sub>H</sub>	R7F7010103AFP R7F7010104AFP	R7F7010133AFP R7F7010134AFP	R7F7010183AFP R7F7010184AFP	R7F7010233AFP R7F7010234AFP R7F7010023AFP R7F7010024AFP	—	—
768 KB	0000 0000 <sub>H</sub> - 000B FFFF <sub>H</sub>	—	R7F7010143AFP R7F7010144AFP	R7F7010193AFP R7F7010194AFP	R7F7010243AFP R7F7010244AFP	R7F7010283AFP R7F7010284AFP	R7F7010323AFP R7F7010324AFP
1 MB	0000 0000 <sub>H</sub> - 000F FFFF <sub>H</sub>	—	R7F7010153AFP R7F7010154AFP	R7F7010203AFP R7F7010204AFP	R7F7010253AFP R7F7010254AFP R7F7010033AFP R7F7010034AFP	R7F7010293AFP R7F7010294AFP	R7F7010333AFP R7F7010334AFP
1.5 MB	0000 0000 <sub>H</sub> - 0017 FFFF <sub>H</sub>	—	—	—	—	R7F7010303AFP R7F7010304AFP	R7F7010343AFP R7F7010344AFP
2 MB	0000 0000 <sub>H</sub> - 001F FFFF <sub>H</sub>	—	—	—	—	R7F7010483AFP R7F7010484AFP	R7F7010523AFP R7F7010524AFP
						R7F7010543AFP R7F7010544AFP	R7F7010563AFP R7F7010564AFP
						R7F7010063AFP R7F7010064AFP R7F7010493AFP R7F7010494AFP	R7F7010073AFP R7F7010074AFP R7F7010533AFP R7F7010534AFP
						R7F7010553AFP R7F7010554AFP	R7F7010573AFP R7F7010574AFP

### 3.2.4.2 Data Flash Area

The following table lists the data flash sizes and address ranges.

**Table 3.3 Data Flash Area**

Code Flash Size	Data Flash Size	Address Range	Product Part Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	R7F7010083AFP R7F7010084AFP	R7F7010113AFP R7F7010114AFP	R7F7010163AFP R7F7010164AFP	R7F7010213AFP R7F7010214AFP	—	—
384 KB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	R7F7010093AFP R7F7010094AFP	R7F7010123AFP R7F7010124AFP	R7F7010173AFP R7F7010174AFP	R7F7010223AFP R7F7010224AFP	—	—
512 KB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	R7F7010103AFP R7F7010104AFP	R7F7010133AFP R7F7010134AFP	R7F7010183AFP R7F7010184AFP	R7F7010233AFP R7F7010234AFP R7F7010023AFP R7F7010024AFP	—	—
768 KB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	—	R7F7010143AFP R7F7010144AFP	R7F7010193AFP R7F7010194AFP	R7F7010243AFP R7F7010244AFP	R7F7010283AFP R7F7010284AFP	R7F7010323AFP R7F7010324AFP
				R7F7010403AFP R7F7010404AFP	R7F7010423AFP R7F7010424AFP	R7F7010443AFP R7F7010444AFP	R7F7010463AFP R7F7010464AFP	R7F7010503AFP R7F7010504AFP
1 MB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	—	R7F7010153AFP R7F7010154AFP	R7F7010203AFP R7F7010204AFP	R7F7010253AFP R7F7010254AFP R7F7010033AFP R7F7010034AFP	R7F7010293AFP R7F7010294AFP	R7F7010333AFP R7F7010334AFP
				R7F7010413AFP R7F7010414AFP	R7F7010433AFP R7F7010434AFP	R7F7010453AFP R7F7010454AFP	R7F7010473AFP R7F7010474AFP	R7F7010513AFP R7F7010514AFP
1.5 MB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	—	—	—	—	R7F7010303AFP R7F7010304AFP	R7F7010343AFP R7F7010344AFP
				—	—	—	R7F7010483AFP R7F7010484AFP	R7F7010523AFP R7F7010524AFP
2 MB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	—	—	—	—	R7F7010063AFP R7F7010064AFP R7F7010493AFP R7F7010494AFP	R7F7010073AFP R7F7010074AFP R7F7010533AFP R7F7010534AFP
	64 KB	FF20 0000 <sub>H</sub> - FF20 FFFF <sub>H</sub>	—	—	—	—	R7F7010543AFP R7F7010544AFP	R7F7010563AFP R7F7010564AFP
2 MB	32 KB	FF20 0000 <sub>H</sub> - FF20 7FFF <sub>H</sub>	—	—	—	—	R7F7010553AFP R7F7010554AFP	R7F7010573AFP R7F7010574AFP
	64 KB	FF20 0000 <sub>H</sub> - FF20 FFFF <sub>H</sub>	—	—	—	—	R7F7010553AFP R7F7010554AFP	R7F7010573AFP R7F7010574AFP

### 3.2.4.3 Primary Local RAM Area

在进入Deep STOP或者任何形式的重启之后，RAM中的数据不会保存（傻子都知道）

Values are not retained in this area in DeepSTOP mode or after any type of reset.

The following table lists the primary local RAM sizes and address ranges.

Table 3.4 Primary Local RAM Area

Code Flash Size	Primary Local RAM Size	Address Range	Product Part Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	—	—	R7F7010083AFP R7F7010084AFP	R7F7010113AFP R7F7010114AFP	R7F7010163AFP R7F7010164AFP	R7F7010213AFP R7F7010214AFP	—	—
384 KB	16 KB	FEDF C000 <sub>H</sub> - FEDF FFFF <sub>H</sub>	R7F7010093AFP R7F7010094AFP	R7F7010123AFP R7F7010124AFP	R7F7010173AFP R7F7010174AFP	R7F7010223AFP R7F7010224AFP	—	—
512 KB	32 KB	FEDF 8000 <sub>H</sub> - FEDF FFFF <sub>H</sub>	R7F7010103AFP R7F7010104AFP	R7F7010133AFP R7F7010134AFP	R7F7010183AFP R7F7010184AFP	R7F7010233AFP R7F7010234AFP R7F7010023AFP R7F7010024AFP	—	—
768 KB	64 KB	FEDF 0000 <sub>H</sub> - FEDF FFFF <sub>H</sub>	—	R7F7010143AFP R7F7010144AFP	R7F7010193AFP R7F7010194AFP	R7F7010243AFP R7F7010244AFP	R7F7010283AFP R7F7010284AFP	R7F7010323AFP R7F7010324AFP
				R7F7010403AFP R7F7010404AFP	R7F7010423AFP R7F7010424AFP	R7F7010443AFP R7F7010444AFP	R7F7010463AFP R7F7010464AFP	R7F7010503AFP R7F7010504AFP
1 MB	96 KB	FEDE 8000 <sub>H</sub> - FEDF FFFF <sub>H</sub>	—	R7F7010153AFP R7F7010154AFP	R7F7010203AFP R7F7010204AFP	R7F7010253AFP R7F7010254AFP R7F7010033AFP R7F7010034AFP	R7F7010293AFP R7F7010294AFP	R7F7010333AFP R7F7010334AFP
				R7F7010413AFP R7F7010414AFP	R7F7010433AFP R7F7010434AFP	R7F7010453AFP R7F7010454AFP	R7F7010473AFP R7F7010474AFP	R7F7010513AFP R7F7010514AFP
1.5 MB	128 KB	FEDE 0000 <sub>H</sub> - FEDF FFFF <sub>H</sub>	—	—	—	—	R7F7010303AFP R7F7010304AFP	R7F7010343AFP R7F7010344AFP
							R7F7010483AFP R7F7010484AFP	R7F7010523AFP R7F7010524AFP
							R7F7010543AFP R7F7010544AFP	R7F7010563AFP R7F7010564AFP
2 MB	128 KB	FEDE 0000 <sub>H</sub> - FEDF FFFF <sub>H</sub>	—	—	—	—	R7F7010063AFP R7F7010064AFP R7F7010493AFP R7F7010494AFP	R7F7010073AFP R7F7010074AFP R7F7010533AFP R7F7010534AFP
							R7F7010553AFP R7F7010554AFP	R7F7010573AFP R7F7010574AFP

#### CAUTION

Instruction fetching and data access from an un-initialized local RAM area may generate an ECC error interrupt or a SYSERR exception. Before reading from a local RAM area, initialize the entire local RAM area.

### 3.2.4.4 Secondary Local RAM Area

Values are not retained in this area in DeepSTOP mode or after any type of reset.

The following table lists the secondary local RAM sizes and address ranges.

**Table 3.5 Secondary Local RAM Area**

Code Flash Size	Secondary Local RAM Size	Address Range	Product Part Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	—	—	R7F7010083AFP R7F7010084AFP	R7F7010113AFP R7F7010114AFP	R7F7010163AFP R7F7010164AFP	R7F7010213AFP R7F7010214AFP	—	—
384 KB	—	—	R7F7010093AFP R7F7010094AFP	R7F7010123AFP R7F7010124AFP	R7F7010173AFP R7F7010174AFP	R7F7010223AFP R7F7010224AFP	—	—
512 KB	—	—	R7F7010103AFP R7F7010104AFP	R7F7010133AFP R7F7010134AFP	R7F7010183AFP R7F7010184AFP	R7F7010233AFP R7F7010234AFP R7F7010023AFP R7F7010024AFP	—	—
768 KB	—	—	—	R7F7010143AFP R7F7010144AFP	R7F7010193AFP R7F7010194AFP	R7F7010243AFP R7F7010244AFP	R7F7010283AFP R7F7010284AFP	R7F7010323AFP R7F7010324AFP
1 MB	—	—	—	R7F7010153AFP R7F7010154AFP	R7F7010203AFP R7F7010204AFP	R7F7010253AFP R7F7010254AFP R7F7010033AFP R7F7010034AFP	R7F7010293AFP R7F7010294AFP	R7F7010333AFP R7F7010334AFP
1.5 MB	—	—	—	—	—	—	R7F7010303AFP R7F7010304AFP	R7F7010343AFP R7F7010344AFP
							R7F7010483AFP R7F7010484AFP	R7F7010523AFP R7F7010524AFP
							R7F7010543AFP R7F7010544AFP	R7F7010563AFP R7F7010564AFP
2 MB	32 KB	FEDD 8000 <sub>H</sub> - FEDD FFFF <sub>H</sub>	—	—	—	—	R7F7010063AFP R7F7010064AFP R7F7010493AFP R7F7010494AFP	R7F7010073AFP R7F7010074AFP R7F7010533AFP R7F7010534AFP
							R7F7010553AFP R7F7010554AFP	R7F7010573AFP R7F7010574AFP

#### CAUTION

**Instruction fetching and data access from an un-initialized local RAM area may generate an ECC error interrupt or a SYSERR exception. Before reading from a local RAM area, initialize the entire local RAM area.**

### 3.2.4.5 Retention RAM Area

Values are retained in this area in DeepSTOP mode and after all resets. The values are retained when the power supply voltage is at or above RAM retention voltage even when the power supply voltage falls below the POC voltage.

The following table lists the Retention RAM sizes and address ranges.

**Table 3.6 Retention RAM Area**

Code Flash Size	Retention RAM Size	Address Range	Product Part Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	R7F7010083AFP R7F7010084AFP	R7F7010113AFP R7F7010114AFP	R7F7010163AFP R7F7010164AFP	R7F7010213AFP R7F7010214AFP	—	—
384 KB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	R7F7010093AFP R7F7010094AFP	R7F7010123AFP R7F7010124AFP	R7F7010173AFP R7F7010174AFP	R7F7010223AFP R7F7010224AFP	—	—
512 KB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	R7F7010103AFP R7F7010104AFP	R7F7010133AFP R7F7010134AFP	R7F7010183AFP R7F7010184AFP	R7F7010233AFP R7F7010234AFP R7F7010023AFP R7F7010024AFP	—	—
768 KB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	—	R7F7010143AFP R7F7010144AFP	R7F7010193AFP R7F7010194AFP	R7F7010243AFP R7F7010244AFP	R7F7010283AFP R7F7010284AFP	R7F7010323AFP R7F7010324AFP
				R7F7010403AFP R7F7010404AFP	R7F7010423AFP R7F7010424AFP	R7F7010443AFP R7F7010444AFP	R7F7010463AFP R7F7010464AFP	R7F7010503AFP R7F7010504AFP
1 MB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	—	R7F7010153AFP R7F7010154AFP	R7F7010203AFP R7F7010204AFP	R7F7010253AFP R7F7010254AFP R7F7010033AFP R7F7010034AFP	R7F7010293AFP R7F7010294AFP	R7F7010333AFP R7F7010334AFP
				R7F7010413AFP R7F7010414AFP	R7F7010433AFP R7F7010434AFP	R7F7010453AFP R7F7010454AFP	R7F7010473AFP R7F7010474AFP	R7F7010513AFP R7F7010514AFP
1.5 MB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	—	—	—	—	R7F7010303AFP R7F7010304AFP	R7F7010343AFP R7F7010344AFP
							R7F7010483AFP R7F7010484AFP	R7F7010523AFP R7F7010524AFP
							R7F7010543AFP R7F7010544AFP	R7F7010563AFP R7F7010564AFP
2 MB	32 KB	FEE0 0000 <sub>H</sub> - FEE0 7FFF <sub>H</sub>	—	—	—	—	R7F7010063AFP R7F7010064AFP R7F7010493AFP R7F7010494AFP	R7F7010073AFP R7F7010074AFP R7F7010533AFP R7F7010534AFP
							R7F7010553AFP R7F7010554AFP	R7F7010573AFP R7F7010574AFP

#### CAUTION

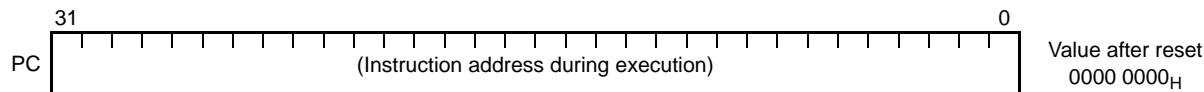
Instruction fetching and data access from an un-initialized local RAM area may generate an ECC error interrupt or a SYSERR exception. Before reading from a local RAM area, initialize the entire local RAM area.

### 3.3 Register Reference

This section describes the items included in Section 3 of the *RH850G3K User's Manual: Software* that require specific attention when using the RH850/F1L.

#### 3.3.1 PC — Program Counter

The PC retains the address of the instruction being executed. Bit 0 is fixed to 0, and branching to an odd number address is disabled.



##### CAUTION

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27.

### 3.3.2 Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

System registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

**Table 3.7 Basic System Registers**

Register No. (regID, selID)	Symbol	Function	Changes from the RH850G3K User's Manual: Software	Access Permission
SR0, 0	EIPC <sup>*1</sup>	Status save registers when acknowledging EI level exception	Some changes in specifications	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	Some changes in specifications	SV
SR2, 0	FEPC <sup>*1</sup>	Status save registers when acknowledging FE level exception	Some changes in specifications	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	Some changes in specifications	SV
SR 5, 0	PSW	Program status word	Some changes in specifications	<sup>*2</sup>
SR 13, 0	EIIC	EI level exception cause	No change	SV
SR 14, 0	FEIC	FE level exception cause	No change	SV
SR 16, 0	CTPC <sup>*1</sup>	CALLT execution status save register	Some changes in specifications	UM
SR 17, 0	CTPSW	CALLT execution status save register	No change	UM
SR 20, 0	CTBP <sup>*1</sup>	CALLT base pointer	Some changes in specifications	UM
SR 28, 0	EIWR	EI level exception working register	No change	SV
SR 29, 0	FEWR	FE level exception working register	No change	SV
SR 31, 0	BSEL	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR0, 1	MCFG0	Machine configuration 0	Some changes in specifications	SV
SR2, 1	RBASE <sup>*1</sup>	Reset vector base address	Some changes in specifications	SV
SR3, 1	EBASE <sup>*1</sup>	Exception handler vector address	Some changes in specifications	SV
SR4, 1	INTBP <sup>*1</sup>	Base address of the interrupt handler table	Some changes in specifications	SV
SR 5, 1	MCTL	CPU control	Some changes in specifications	SV
SR 6, 1	PID	Processor ID	Some changes in specifications	SV
SR 11, 1	SCCFG	SYSCALL operation setting	No change	SV
SR 12, 1	SCBP <sup>*1</sup>	SYSCALL base pointer	Some changes in specifications	SV
SR0, 2	HTCFG0	Thread configuration	Some changes in specifications	SV
SR6, 2	MEA	Memory error address	No change	SV
SR7, 2	ASID	Address space ID	No change	SV
SR8, 2	MEI	Memory error information	Some changes in specifications	SV

Note 1. For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27.

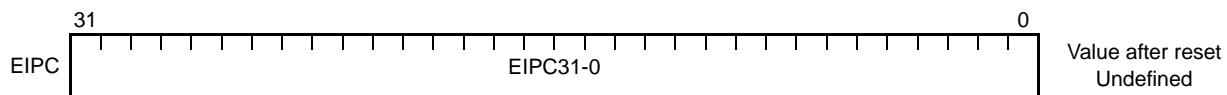
Note 2. The access permission differs depending on the bit. For details, see **Table 3.12, Access Permission for PSW Register**.

### 3.3.2.1 EIPC — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see *Section 4.1.3, Types of exceptions*, in the *RH850G3K User's Manual: Software*).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address cannot be specified.



**Table 3.8 EIPC Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	EIPC31-1	These bits indicate the PC saved when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the PC saved when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0.	R/W	Undefined

**CAUTION**

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of EIPC.

### 3.3.2.2 EIPSW — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

EIPSW	31	30	29	20	19	18	16	15	14	12	11	10	9	8	7	6	5	4	3	2	1	0	Value after reset 0000 0020H								
	0	U	M	0	0	0	0	0	0	CU2-0	E	B	V	0	0	0	Debug	0	N	P	E	I	S	A	T	C	Y	O	V	S	Z

Table 3.9 EIPSW Register Contents

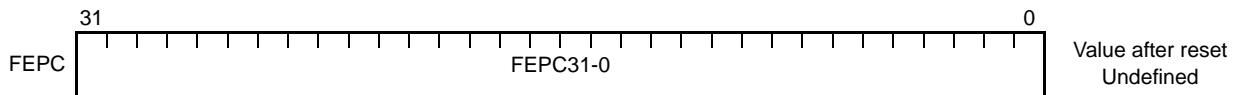
Bit Position	Bit Name	Function	R/W	Value After Reset
31	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18 to 16	CU2-0	The RH850/F1L does not support these bits.	R	0
15	EBV	This bit stores the PSW.EBV bit setting when an EI level exception is acknowledged.	R/W	0
14 to 12	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
11 to 9	Debug	These bits store the PSW.Debug field when an EI level exception is acknowledged.	R	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

### 3.3.2.3 FEPC — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see *Section 4.1.3, Types of exceptions*, in the *RH850G3K User's Manual: Software*).

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address cannot be specified.



**Table 3.10 FEPC Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	FEPC31-1	These bits indicate the PC saved when an FE level exception is acknowledged.	R/W	Undefined
0	FEPC0	This bit indicates the PC saved when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0.	R/W	Undefined

#### CAUTION

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of FEPC.

### 3.3.2.4 FEPSW — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

	31	30	29	20	19	18	16	15	14	12	11	9	8	7	6	5	4	3	2	1	0										
FEPSW	0	U	M	0	0	0	0	0	0	CU2-0	E	B	V	0	0	0	Debug	0	N	P	E	I	S	A	T	C	O	V	S	Z	Value after reset 0000 0020H

Table 3.11 FEPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18 to 16	CU2-0	The RH850/F1L does not support these bits.	R	0
15	EBV	This bit stores the PSW.EBV bit setting when an FE level exception is acknowledged.	R/W	0
14 to 12	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
11 to 9	Debug	These bits store the PSW.Debug field when an FE level exception is acknowledged.	R	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

### 3.3.2.5 PSW — Program Status Word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by conditional instructions (Bcond, CMOV, etc.)).

#### CAUTIONS

- When the LDSR instruction is used to change the contents of bits 7 to 0 in this register, the changed contents become valid immediately after the completion of execution of the LDSR instruction.**
- The access permission for the PSW register differs with each bit. All bits can be read, but some bits can only be written under certain conditions. See Table 3.12, Access Permission for PSW Register for the access permission for each bit.**

Table 3.12 Access Permission for PSW Register

Bit	Access Permission When Reading		Access Permission When Writing
30	UM	UM	SV* <sup>1</sup>
15	EBV		SV* <sup>1</sup>
7	NP		SV* <sup>1</sup>
6	EP		SV* <sup>1</sup>
5	ID		SV* <sup>1</sup>
4	SAT		UM
3	CY		UM
2	OV		UM
1	S		UM
0	Z		UM

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

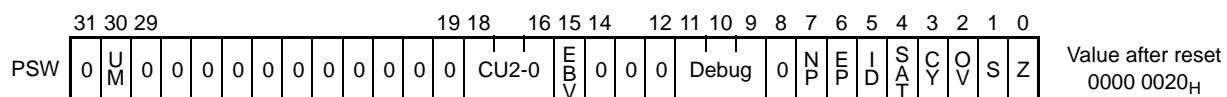


Table 3.13 PSW Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
30	UM	This bit indicates that the CPU is in the user mode (the UM mode) 0: Supervisor mode 1: User mode	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18 to 16	CU2-0	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	000
15	EBV	This bit indicates the reset vector and exception vector operation. For details, see the RBASE register in <b>Section 3.3.2.10, RBASE — Reset Vector Base Address</b> and EBASE register in <b>Section 3.3.2.11, EBASE — Exception Handler Vector Address</b> .	R/W	0
14 to 12	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
11 to 9	Debug	These bits are used in debugging function for the development tool. In normal cases, set these bits to "0".	R	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

Table 3.13 PSW Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
7	NP	This bit indicates that an FE-level exception is being processed. When an FE level exception is acknowledged, this bit is set to 1, which prohibits occurrence of multiple exceptions.*1 0: FE level exception handling is not in progress. 1: FE level exception handling is in progress.	R/W	0
6	EP	This bit indicates that an exception other than an interrupt is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An interrupt is being serviced. 1: An exception other than an interrupt is being serviced.	R/W	0
5	ID	This bit indicates enabling or disabling of the acknowledgment of EI-level exceptions. It is set to 1 when an EI level exception or FE level exception is acknowledged, disabling the acknowledgment of further EI-level exception.*1 This bit is also used to disable the acknowledgment of EI level exceptions in a critical section during ordinary program execution or interrupt servicing. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. Changes to the value of the ID bit by the EI or DI instruction will be effective from the next instruction. 0: The acknowledgment of an EI level exception is enabled. 1: The acknowledgment of an EI level exception is disabled.	R/W	1
4	SAT*2	This bit indicates that the operation result is saturated because the result of a saturated operation instruction operation has overflowed. This is a cumulative flag, so when the operation result of the saturated operation instruction becomes saturated, this bit is set to 1, but it is not cleared to 0 when the operation result for a subsequent instruction is not saturated. This bit is cleared to 0 by the LDSR instruction. This bit is neither set to 1 nor cleared to 0 when an arithmetic operation instruction is executed. 0: Not saturated 1: Saturated	R/W	0
3	CY	This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry and borrow have not occurred. 1: Carry or borrow has occurred.	R/W	0
2	OV*2	This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred.	R/W	0
1	S*2	This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative.	R/W	0
0	Z	This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0.	R/W	0

Note 1. Exceptions from certain sources are accepted regardless of the states of the ID and NP bits. For details, see **Section 3.6.4, Overwriting Context upon Acknowledgment of Multiple Exceptions**.

Note 2. The operation result of the saturation processing is determined in accordance with the contents of the OV flag and S flag during a saturated operation. The SAT flag is set to 1, only when the OV flag is set to 1 in a saturated operation.

Operation result status	Flag status			Operation result after saturation processing
	SAT	OV	S	
Exceeded positive maximum value	1	1	0	7FFF FFFF <sub>H</sub>
Exceeded negative maximum value	1	1	1	8000 0000 <sub>H</sub>
Positive (maximum value not exceeded)	Value prior to operation is retained.	0	0	Operation result itself
Negative (maximum value not exceeded)			1	

### 3.3.2.6 CTPC — Status Save Register when Executing CALLT

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC.

Be sure to set an even-numbered address to the CTPC register. An odd-numbered address cannot be specified.

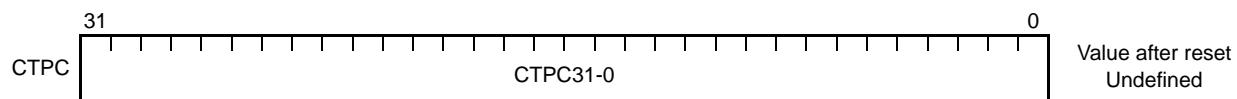


Table 3.14 CTPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	CTPC31-1	These bits indicate the PC of the instruction after the CALLT instruction.	R/W	Undefined
0	CTPC0	This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0.	R/W	Undefined

#### CAUTION

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of CTPC.

### 3.3.2.7 CTBP — CALLT Base Pointer

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a halfword address.

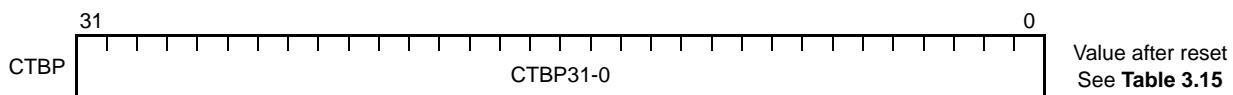


Table 3.15 CTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	CTBP31-1	These bits indicate the base pointer address of the CALLT instruction. These bits indicate the first address in the table used by the CALLT instruction.	R/W	Undefined
0	CTBP0	This bit indicates the base pointer address of the CALLT instruction. This bit indicates the first address in the table used by the CALLT instruction. Always set this bit to 0.	R	0

#### CAUTION

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of CTBP.

### 3.3.2.8 HTCFG0 — Thread Configuration

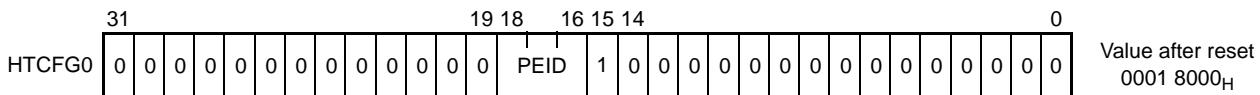
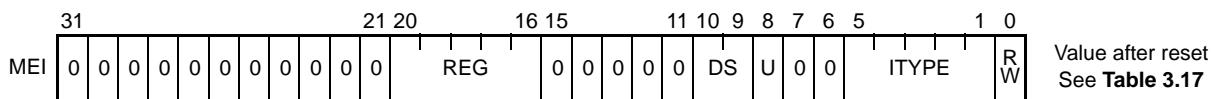


Table 3.16 HTCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18 to 16	PEID	These bits indicate the processor element number.	R	001 <sub>B</sub>
15	—	(Be sure to set this bit to 1. A value of 1 will be returned when read.)	R	1
14 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

### 3.3.2.9 MEI — Memory Error Information

This register is used to store information about the instruction that caused the exception when a misaligned (MAE) or memory protection (MDP) exception occurs. This information is used during emulation.



**Table 3.17 MEI Register Contents**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>	<b>R/W</b>	<b>Value After Reset</b>
31 to 21	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
20 to 16	REG4-0	These bits indicate the number of the source or destination register accessed by the instruction that caused the exception. For details, see <b>Table 3.18</b> .	R/W	Undefined
15 to 11	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
10, 9	DS	These bits indicate the data type of the instruction that caused the exception.*1 0: Byte (8 bits) 1: Halfword (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) For details, see <b>Table 3.18</b> .	R/W	Undefined
8	U	These bits indicate the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see <b>Table 3.18</b> .	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
5 to 1	ITYPE4-0	These bits indicate the instruction that caused the exception. For details, see <b>Table 3.18</b> .	R/W	Undefined
0	RW	This bit indicates whether the operation of the instruction that caused the exception was read (Load-memory) or write (Store-memory) 0: Read (Load-memory) 1: Write (Store-memory) For details, see <b>Table 3.18</b> .	R/W	Undefined

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

**Table 3.18 Instructions Causing Exceptions and Values of MEI Register (1/2)**

<b>Instruction</b>	<b>REG</b>	<b>DS</b>	<b>U</b>	<b>RW</b>	<b>ITYPE</b>
SLD.B	dst	0 (byte)	0 (signed)	0 (read)	00000 <sub>B</sub>
SLD.BU	dst	0 (byte)	1 (unsigned)	0 (read)	00000 <sub>B</sub>
SLD.H	dst	1 (halfword)	0 (signed)	0 (read)	00000 <sub>B</sub>
SLD.HU	dst	1 (halfword)	1 (unsigned)	0 (read)	00000 <sub>B</sub>
SLD.W	dst	2 (word)	0 (signed)	0 (read)	00000 <sub>B</sub>
SST.B	src	0 (byte)	0 (signed)	1 (write)	00000 <sub>B</sub>
SST.H	src	1 (halfword)	0 (signed)	1 (write)	00000 <sub>B</sub>
SST.W	src	2 (word)	0 (signed)	1 (write)	00000 <sub>B</sub>
LD.B (disp16)	dst	0 (byte)	0 (signed)	0 (read)	00001 <sub>B</sub>
LD.BU (disp16)	dst	0 (byte)	1 (unsigned)	0 (read)	00001 <sub>B</sub>
LD.H (disp16)	dst	1 (halfword)	0 (signed)	0 (read)	00001 <sub>B</sub>
LD.HU (disp16)	dst	1 (halfword)	1 (unsigned)	0 (read)	00001 <sub>B</sub>
LD.W (disp16)	dst	2 (word)	0 (signed)	0 (read)	00001 <sub>B</sub>
ST.B (disp16)	src	0 (byte)	0 (signed)	1 (write)	00001 <sub>B</sub>
ST.H (disp16)	src	1 (halfword)	0 (signed)	1 (write)	00001 <sub>B</sub>
ST.W (disp16)	src	2 (word)	0 (signed)	1 (write)	00001 <sub>B</sub>
LD.B (disp23)	dst	0 (byte)	0 (signed)	0 (read)	00010 <sub>B</sub>
LD.BU (disp23)	dst	0 (byte)	1 (unsigned)	0 (read)	00010 <sub>B</sub>
LD.H (disp23)	dst	1 (halfword)	0 (signed)	0 (read)	00010 <sub>B</sub>
LD.HU (disp23)	dst	1 (halfword)	1 (unsigned)	0 (read)	00010 <sub>B</sub>
LD.W (disp23)	dst	2 (word)	0 (signed)	0 (read)	00010 <sub>B</sub>
ST.B (disp23)	src	0 (byte)	0 (signed)	1 (write)	00010 <sub>B</sub>

**Table 3.18 Instructions Causing Exceptions and Values of MEI Register (2/2)**

<b>Instruction</b>	<b>REG</b>	<b>DS</b>	<b>U</b>	<b>RW</b>	<b>ITYPE</b>
ST.H (disp23)	src	1 (halfword)	0 (signed)	1 (write)	00010 <sub>B</sub>
ST.W (disp23)	src	2 (word)	0 (signed)	1 (write)	00010 <sub>B</sub>
LD.DW (disp23)	dst	3 (double-word)	0 (signed)	0 (read)	00010 <sub>B</sub>
ST.DW (disp23)	src	3 (double-word)	0 (signed)	1 (write)	00010 <sub>B</sub>
LDL.W	dst	2 (word)	0 (signed)	0 (read)	00111 <sub>B</sub>
STC.W	src	2 (word)	0 (signed)	1 (write)	00111 <sub>B</sub>
CAXI	dst	2 (word)	1 (unsigned)	0 (read) <sup>*1</sup>	01000 <sub>B</sub>
SET1	—	0 (byte)	1 (unsigned)	0 (read) <sup>*1</sup>	01001 <sub>B</sub>
CLR1	—	0 (byte)	1 (unsigned)	0 (read) <sup>*1</sup>	01001 <sub>B</sub>
NOT1	—	0 (byte)	1 (unsigned)	0 (read) <sup>*1</sup>	01001 <sub>B</sub>
TST1	—	0 (byte)	1 (unsigned)	0 (read)	01001 <sub>B</sub>
PREPARE	—	2 (word)	1 (unsigned)	1 (write)	01100 <sub>B</sub>
DISPOSE	—	2 (word)	1 (unsigned)	0 (read)	01100 <sub>B</sub>
PUSHSP	—	2 (word)	1 (unsigned)	1 (write)	01101 <sub>B</sub>
POPSP	—	2 (word)	1 (unsigned)	0 (read)	01101 <sub>B</sub>
SWITCH	—	1 (halfword)	0 (signed)	0 (read)	10000 <sub>B</sub>
CALLT	—	1 (halfword)	1 (unsigned)	0 (read)	10001 <sub>B</sub>
SYSCALL	—	2 (word)	1 (unsigned)	0 (read)	10010 <sub>B</sub>
Interrupt (table reference) <sup>*2</sup>	—	2 (word)	1 (unsigned)	0 (read)	10101 <sub>B</sub>

Note 1. This exception occurs when the instruction executes a read access.

Note 2. When reading the interrupt vector by using the table reference method

#### NOTE

dst: destination register number, src: source register number

#### Misaligned Exception

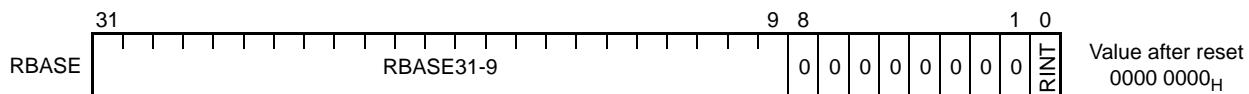
Access to data at an address that does not correspond to the required boundary for the unit of access leads to a misaligned exception. For example, reading or writing a word from or to an address that is not a multiple of 4 leads to an exception of this type. However, if the type is double-word, access at any word-form boundary does not lead to an exception. When MCTL.MA is 1, access proceeds without generating misaligned exceptions.

However, even if MCTL.MA is 1, the double-word access to an address not the word-form, leads to a misaligned exception.

### 3.3.2.10 RBASE — Reset Vector Base Address

如果PSW.EBV bit为0，RBASE作为异常向量地址

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this vector address is also used as the exception vector address.



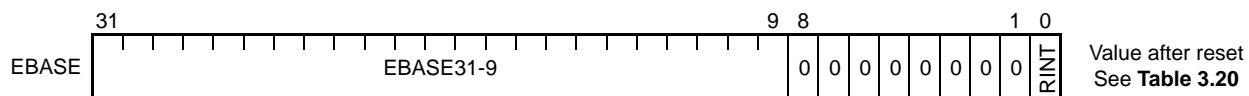
**Table 3.19 RBASE Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	RBASE31-9	These bits indicate the reset vector when there is a reset. When PSW.EBV = 0, this address is also used as the exception vector. The RBASE8-0 bits are implicitly 0.	R	0000 0000 0000 0000 0000 000B <sup>*1</sup>
8 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	RINT	When the RINT bit is set, the offset address of the exception handler for interrupt processing is reduced. This bit is valid when PSW.EBV = 0. <small>只有在PSW.EBV = 0的时候此位才有效</small>	R	0

Note 1. When the reset vector is changed by self-programming, the address will be changed.

### 3.3.2.11 EBASE — Exception Handler Vector Address

This register indicates the exception handler vector address. This register is valid when the PSW.EBV bit is 1.  
如果PSW.EBV bit为0，EBASE作为异常向量地址



**Table 3.20 EBASE Register Contents**

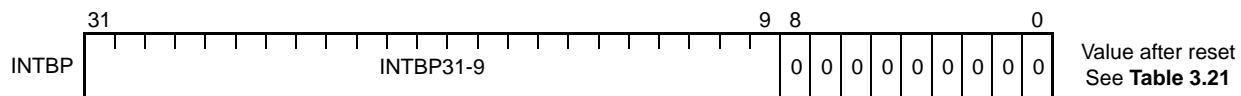
Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	EBase31-9	The exception handler routine address is changed to the address resulting from adding the offset address of each exception to the base address specified for this register. The EBase8-0 bits are implicitly 0.	R/W	Undefined
8 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	RINT	When the RINT bit is set, the offset address of the exception handler used during interrupt servicing is decremented.	R/W	Undefined

## **CAUTION**

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of EBASE.

### 3.3.2.12 INTBP — Base Address of the Interrupt Handler Table

This register indicates the base address of the address table when the table reference method is selected as the interrupt handler address selection method.



**Table 3.21** INTBP Register Contents

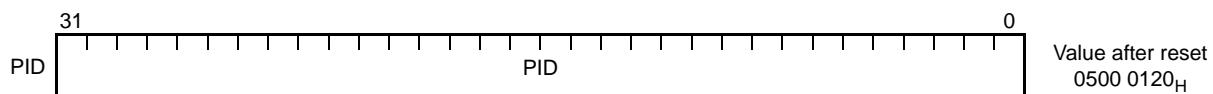
Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	INTBP31-9	<p>These bits indicate the base pointer address for an interrupt prescribed in the expanded specifications.</p> <p>The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupt prescribed by the expanded specifications (EIINT0 to 287) is acknowledged.</p> <p>The INTBP8-0 bits are implicitly 0.</p>	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

## **CAUTION**

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of INTBP.

### 3.3.2.13 PID – Processor ID

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.



**Table 3.22** PID Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 24	PID	Architecture identifier This identifier indicates the architecture of the processor.	R	05H
23 to 8		Function identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bits 23 to 9 Reserved Bit 8 Memory protection unit (MPU) function	R	0001H
7 to 0		Version identifier This identifier indicates the version of the processor.	R	20H

### 3.3.2.14 SCBP — SYSCALL Base Pointer

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

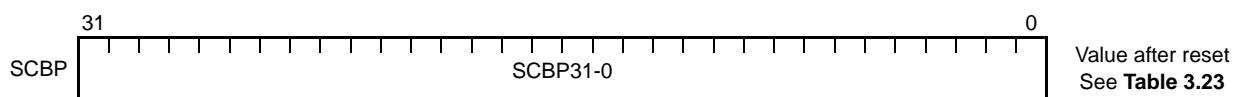


Table 3.23 SCBP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	SCBP31-2	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the first address in the table used by the SYSCALL instruction.	R/W	Undefined
1, 0	SCBP1-0	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the first address in the table used by the SYSCALL instruction. Always set this bit to 0.	R	0

#### CAUTION

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of SCBP.

### 3.3.2.15 MCFG0 — Machine Configuration 0

This register indicates the CPU configuration.

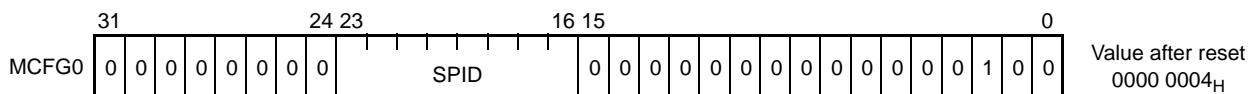


Table 3.24 MCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 24	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
23 to 16	SPID	Not supported by the RH850/F1L. Be sure to clear to 0.	R	0000 0000
15 to 3	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
2	—	(Reserved for future expansion. Be sure to clear to 1.)	R	1
1, 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

### **3.3.2.16 MCTL — Machine Control**

This register is used to control the CPU.

**Table 3.25 MCTL Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
30 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1	MA	This bit is used to control the misaligned access. 0: In the event of a misaligned access, an exception is always generated.* <sup>1</sup> 1: The correct operation is controlled by hardware.* <sup>2</sup>	R/W	0
0	UIC	This bit is used to control the interrupt enable/disable operation in the user mode. When this bit is set to 1, executing the EI/DI instruction becomes possible.	R/W	0

Note 1. Excluding LD.DW, and ST.DW for word boundary allocation.

Note 2. Exception still occurs in case of LD.DW or ST.DW for misaligned access except word boundary allocation.

### 3.3.3 Interrupt Function Registers

**Table 3.26** Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Changes from the RH850G3K User's Manual: Software	Access Permission
SR10, 2	ISPR	Priority of interrupt being serviced	Some changes in specifications	SV
SR11, 2	PMR	Interrupt priority masking	Some changes in specifications	SV
SR12, 2	ICSR	Interrupt control status	Some changes in specifications	SV
SR13, 2	INTCFG	Interrupt function setting	No change	SV

### **3.3.3.1 ISPR — Priority of Interrupt being Serviced**

此寄存器保留被处理过的EI INT中断的优先级

This register holds the priority of the EIINTn interrupt being serviced. This priority value is then used to perform priority ceiling processing when multiple interrupts are generated.

此优先级的值被用于当多个中断发生的时候执行优先级上限处理（这翻译有问题，但是不知道怎么处理）

Register ISP7-0 bit map diagram showing bit positions 31 to 0. Bit 31 is labeled 'ISP7-0'.

Bit	Value after reset
31	0000 0000 <sub>H</sub>
8	
7	
0	

**Table 3.27 ISPR Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7 to 0	ISP7-0	<p>These bits indicate the acknowledgment status of an EIINTn interrupt with a priority*<sup>1</sup> that corresponds to the relevant bit position.</p> <p>0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged.</p> <p>1: An interrupt request for an interrupt whose priority corresponds to the relevant position is being serviced by the CPU core.</p>	R* <sup>3</sup>	0

When an interrupt request (EIINTn) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP7-0 bits that are set (0 is the highest priority) is cleared to 0<sup>\*2</sup>. While a bit in this register is set to 1, the same or lower priority interrupts (EIINTn) are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged. For details, see *Section 4.1.5, Interrupt exception priority and priority masking*, in the *RH850/G3K*.

When performing software-based priority control using the PMR register, be aware that it does not affect interrupt priority and priority masking, in the ARM® Cortex®-M3 User's Manual: Software.

- - Note 1. For details, see *Section 4.1.5, Interrupt exception priority and priority masking*, in the *RH850G3K User’s Manual: Software*.
  - Note 2. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.
  - Note 3. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register.

### 3.3.3.2 PMR — Interrupt Priority Masking

This register is used to mask the specified interrupt priority.  
用于屏蔽指定的中断优先级

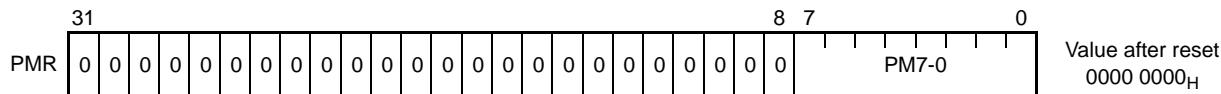


Table 3.28 PMR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7 to 0	PM7-0	<p>These bits mask an interrupt request with a priority level that corresponds to the relevant bit position.</p> <p>0: Servicing of an interrupt request with a priority that corresponds to the relevant bit position is enabled.</p> <p>1: Servicing of an interrupt request with a priority that corresponds to the relevant bit position is disabled.</p>	R/W	0

The bit positions correspond to the following priority levels:

Bit	Priority
0	Priority 0 (highest)
1	Priority 1
...	...
6	Priority 6
7	Priority 7 (lowest)

While a bit in this register is set to 1, interrupts (EIINTn) with the priority corresponding to that bit are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged\*1.

Note 1. Specify the masks by setting the bits to 1 in order from the lowest-priority bit. For example, F0H can be set, but 10H or 0FH cannot. 将此位设置为1要按照从lowest-priority bit顺序开始

### 3.3.3.3 ICSR — Interrupt Control Status

This register indicates the interrupt control status in the CPU.

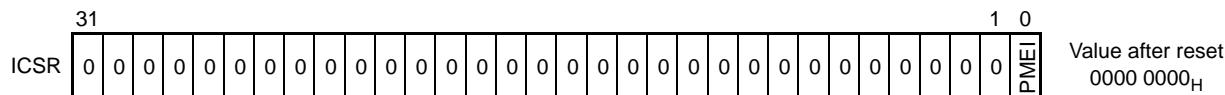


Table 3.29 ICSR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	PMEI	This bit indicates that an interrupt (EIINTn) with the priority level masked by the PMR register exists.	R	0

此位表明（带有被PMR寄存器屏蔽优先级的中断）存在，主语是中断

### 3.3.3.4 INTCFG — Interrupt Function Setting

This register is used to specify the interrupt function in the CPU.

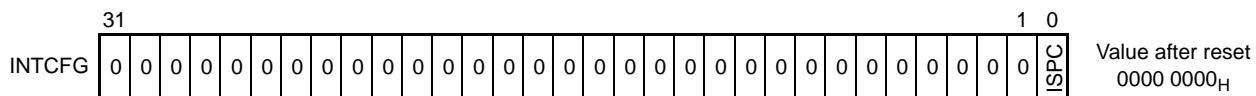


Table 3.30 INTCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	ISPC	<p>This bit specifies how the ISPR register is updated. <b>此位指定 ISPR 更新的方式</b></p> <p>0: ISPR is automatically updated. Updating by the program (LDSR, LDTC.SR) is ignored. <b>自动更新，由软件更新的方式将被忽略</b></p> <p>1: ISPR is not automatically updated. Updating by the program (LDSR, LDTC.SR) is performed. <b>由软件进行更新</b></p> <p>While this bit is cleared to 0, the respective bits in the ISPR register are automatically set to 1 by the acknowledgment of an interrupt (EIINTn) or automatically cleared to 0 by the execution of the EIRET instruction. In this state, updating by the program (LDSR, LDTC.SR) is ignored.</p> <p>While this bit is set to 1, the respective bits in the ISPR register are not updated by the acknowledgment of an interrupt (EIINTn) or execution of the EIRET instruction. In this state, updating by the program (LDSR, LDTC.SR) is enabled.</p> <p>Normally, use the ISPC bit being cleared to 0. If the interrupt priority is controlled by software, set this bit to 1 and control the priority by the PMR register. <b>一般来说，将 ISPC 清零。如果中断优先级被软件控制，设置其为1，由 PMR 控制优先级</b></p>	R/W	0

### 3.3.4 MPU Function Registers

Table 3.31 MPU Function System Registers (1/2)

Register No. (regID, sellID)	Symbol	Function	Changes from the RH850G3K User's Manual: Software	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	Some changes in specifications	SV
SR1, 5	MPRC	MPU region control	Some changes in specifications	SV
SR4, 5	MPBRGN	MPU base region number	Some changes in specifications	SV
SR5, 5	MPTRGN	MPU end region number	Some changes in specifications	SV
SR0, 6	MPLA0	Protection area minimum address	Some changes in specifications	SV
SR1, 6	MPUA0	Protection area maximum address	Some changes in specifications	SV
SR2, 6	MPAT0	Protection area attribute	Some changes in specifications	SV
SR4, 6	MPLA1	Protection area minimum address	Some changes in specifications	SV
SR5, 6	MPUA1	Protection area maximum address	Some changes in specifications	SV
SR6, 6	MPAT1	Protection area attribute	Some changes in specifications	SV
SR8, 6	MPLA2	Protection area minimum address	Some changes in specifications	SV
SR9, 6	MPUA2	Protection area maximum address	Some changes in specifications	SV
SR10, 6	MPAT2	Protection area attribute	Some changes in specifications	SV
SR12, 6	MPLA3	Protection area minimum address	Some changes in specifications	SV
SR13, 6	MPUA3	Protection area maximum address	Some changes in specifications	SV
SR14, 6	MPAT3	Protection area attribute	Some changes in specifications	SV

Table 3.31 MPU Function System Registers (2/2)

Register No. (regID, selID)	Symbol	Function	Changes from the RH850G3K User's Manual: Software	Access Permission
SR16, 6	MPLA4	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR17, 6	MPUA4			SV
SR18, 6	MPAT4			SV
SR20, 6	MPLA5			SV
SR21, 6	MPUA5			SV
SR22, 6	MPAT5			SV
SR24, 6	MPLA6			SV
SR25, 6	MPUA6			SV
SR26, 6	MPAT6			SV
SR28, 6	MPLA7			SV
SR29, 6	MPUA7			SV
SR30, 6	MPAT7			SV
SR0, 7	MPLA8			SV
SR1, 7	MPUA8			SV
SR2, 7	MPAT8			SV
SR4, 7	MPLA9			SV
SR5, 7	MPUA9			SV
SR6, 7	MPAT9			SV
SR8, 7	MPLA10			SV
SR9, 7	MPUA10			SV
SR10, 7	MPAT10			SV
SR12, 7	MPLA11			SV
SR13, 7	MPUA11			SV
SR14, 7	MPAT11			SV
SR16, 7	MPLA12			SV
SR17, 7	MPUA12			SV
SR18, 7	MPAT12			SV
SR20, 7	MPLA13			SV
SR21, 7	MPUA13			SV
SR22, 7	MPAT13			SV
SR24, 7	MPLA14			SV
SR25, 7	MPUA14			SV
SR26, 7	MPAT14			SV
SR28, 7	MPLA15			SV
SR29, 7	MPUA15			SV
SR30, 7	MPAT15			SV

### 3.3.4.1 MPM — Memory Protection Operation Mode

The memory protection mode register is used to define the basic operating state of the memory protection function.

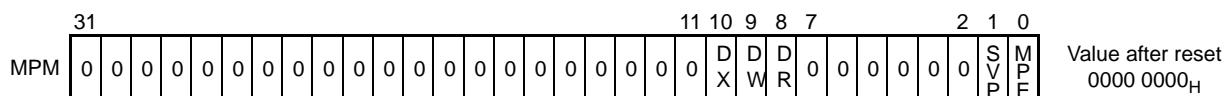


Table 3.32 MPM Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 11	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
10	DX	This bit specifies the default operation when an instruction is executed at an address that does not exist in a protection area.  0: Disable executing an instruction at an address that does not exist in a protection area. 1: Enable executing an instruction at an address that does not exist in a protection area.  The setting of this bit affects the access operation when the protection areas overlap.	R/W	0
9	DW	This bit specifies the default operation when writing to an address that does not exist in a protection area.  0: Disable writing to an address that does not exist in a protection area. 1: Enable writing to an address that does not exist in a protection area.  The setting of this bit affects the access operation when the protection areas overlap.	R/W	0
8	DR	This bit specifies the default operation when reading from an address that does not exist in a protection area.  0: Disable reading from an address that does not exist in a protection area. 1: Enable reading from an address that does not exist in a protection area.  The setting of this bit affects the access operation when the protection areas overlap.	R/W	0
7 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1	SVP	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
0	MPE	This bit is used to specify whether to enable or disable MPU function. 0: Disable 1: Enable	R/W	0

### **3.3.4.2 MPRC — MPU Region Control**

Bits used to perform special memory protection function operations are located in this register.

	31	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Value after reset 0000 0000H
MPRC	0	0	0	0	0	0	0	0	0	0	0	E	E	E	E	E	E	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			

**Table 3.33 MPRC Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 16	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
15 to 4	E15-4	These bits are not used in the RH850/F1L. Be sure to clear to 0.	R	0
3 to 0	E3-0	These are the enable bits for each protection area. Bit En is a copy of bit MPATn.E (where n = 3 to 0).	R/W	0

### **3.3.4.3 MPBRGN — MPU Base Region**

This register indicates the minimum usable MPU area number.

**Table 3.34 MPBRGN Register Contents**

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
4 to 0	MPBRGN	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0000 <sub>B</sub>

#### **3.3.4.4 MPTRGN — MPU End Region**

This register indicates the maximum usable MPU area number.

MPTRGN		Value after reset $0000\ 0003_H$
--------	--	-------------------------------------

**Table 3.35** MPTRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
4 to 0	MPTRGN	The RH850/F1L does not support these bits. Be sure to set to 0011 <sub>B</sub> .	R	0011 <sub>B</sub>

### 3.3.4.5 MPLAn — Protection Area Minimum Address

These registers indicate the minimum address of area n (where n = 0 to 3).

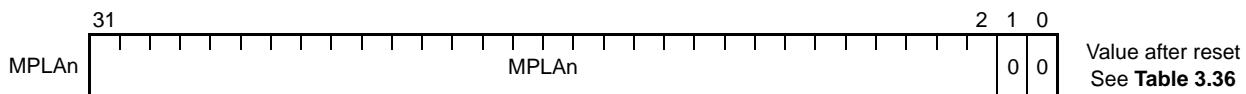


Table 3.36 MPLAn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	MPLA31-2	These bits indicate the minimum address of area n. The MPLA1-0 bits are implicitly set to 0.	R/W	Undefined
1, 0	—	Reserved for future expansion. Be sure to clear these bits to 0.	R	0

**CAUTION**

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of MPLAn.

### 3.3.4.6 MPUAn — Protection Area Maximum Address

These registers indicate the maximum address of area n (where n = 0 to 3).

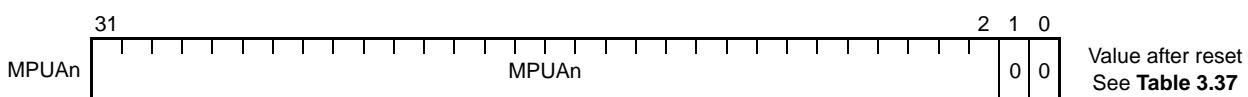


Table 3.37 MPUAn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	MPUA31-2	These bits indicate the maximum address of area n. The MPUA1-0 bits are implicitly set to 1.	R/W	Undefined
1, 0	—	Reserved for future expansion. Be sure to clear these bits to 0.	R	0

**CAUTION**

For a CPU whose instruction addressing range is 128 MB, the value resulting from sign-extending bit 26 is automatically set to bits 31 to 27 of MPUAn.

### 3.3.4.7 MPATn — Protection Area Attribute

These registers indicate the attributes of area n (where n = 0 to 3).

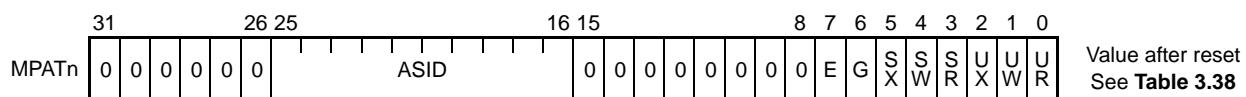


Table 3.38 MPATn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 26	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
25 to 16	ASID	Not supported by the RH850/F1L. Be sure to clear to 0.	R/W	Undefined
15 to 8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	E	This bit indicates whether area n is enabled or disabled. 0: The protection setting for area n is disabled. 1: The protection setting for area n is enabled.	R/W	0
6	G	If this bit is 1, the ASID bit is ignored under the area match condition. Be sure to set this bit to 1.	R/W	Undefined
5	SX	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	Undefined
4	SW	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	Undefined
3	SR	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	Undefined
2	UX	This bit indicates the execution privilege for the user mode. 0: The execution privilege is not granted to the user mode. 1: The execution privilege is granted to the user mode.	R/W	Undefined
1	UW	This bit indicates the write permission for the user mode. 0: The write permission is disabled for the user mode. 1: The write permission is enabled for the user mode.	R/W	Undefined
0	UR	This bit indicates the read permission for the user mode. 0: The read permission is disabled for the user mode. 1: The read permission is enabled for the user mode.	R/W	Undefined

### 3.3.5 Cache Operation Function Registers

The RH850/F1L does not include a cache operation function, so all the following registers return a value of 0 when read, and writing to these registers is ignored.

**Table 3.39 Cache Operation Function Registers**

Register No. (regID, selID)	Symbol	Function	Access Permission
SR12, 4	BWERRL	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR13, 4	BWERRH		SV
SR14, 4	BRERRL		SV
SR15, 4	BRERRH		SV
SR16, 4	ICTAGL		SV
SR17, 4	ICTAGH		SV
SR18, 4	ICDATL		SV
SR19, 4	ICDATH		SV
SR20, 4	DCTAGL		SV
SR21, 4	DCTAGH		SV
SR22, 4	DCDATL		SV
SR23, 4	DCDATH		SV
SR24, 4	ICCTRL		SV
SR25, 4	DCCTRL		SV
SR26, 4	ICCFG		SV
SR27, 4	DCCFG		SV
SR28, 4	ICERR		SV
SR29, 4	DCERR		SV

### 3.4 Instruction Reference

For the instructions used by the RH850/F1L, see the *RH850G3K User's Manual: Software (R01US0125E)*.

### 3.5 System Error Notification Setting Registers

SEG\_CONT and SEG\_FLAG registers described in this subsection can be written only in supervisor mode (PSW.UM=0).

### 3.5.1 SEG\_CONT — SYSERR Exception Notification Setting Registers

SEG\_CONT is an enable register to set whether to enable or disable error notification of each SYSERR source.

When bit 7, 6, 4, 2, or 1 is set to 1, the corresponding error results in a SYSERR exception. If the value of the corresponding enable bit is 0, the error flag is set when the error occurs but a SYSERR exception does not occur.

Be sure to set bits 15 to 8, 5, 3, and 0 to 0.

**Access:** SEG\_CONT can be read or written in 16-bit units.  
SEG\_CONTL can be read or written in 8-bit or 1-bit units.

**Address:** SEG\_CONT: FFFF 8C00<sub>H</sub>  
SEG\_CONTL: FFFF 8C00<sub>H</sub>

**Value after reset:** 0000<sub>H</sub> (Initialized by any reset source.)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEG_CONT DMAE	SEG_CONT RMWE	—	SEG_CONT RAME	—	SEG_CONT EXTE	SEG_CONT ROME	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R

Table 3.40 SEG\_CONT Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEG_CONT DMAE	DMA transfer error notification enable Sets whether to enable SYSERR notification at a DMA transfer error occurrence.
6	SEG_CONT RMWE	Read modify write error notification enable Sets whether to enable SYSERR notification for byte write or half-word write to the primary local RAM, secondary local RAM, or retention RAM.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SEG_CONT RAME	Primary local RAM/secondary local RAM/retention RAM area error notification enable Sets whether to enable SYSERR notification if an error occurs when the CPU accesses data in the primary local RAM, secondary local RAM, or retention RAM area.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	SEG_CONT EXTE	PBUS module access error notification enable Sets whether to enable SYSERR notification if an error occurs when the CPU accesses the PBUS module. <ul style="list-style-type: none"> <li>– Access to a register of a clock domain which has been disabled by the CKSC_xxxx_CTL register</li> <li>– Access to a register of a clock domain to which supply of the source clock selected by the CKSC_xxxx_CTL register has been stopped</li> </ul>
1	SEG_CONT ROME	Code flash error notification enable Sets whether to enable SYSERR notification if an error occurs when the CPU accesses data in the code flash area.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

### 3.5.2 SEG\_FLAG — System Error Source Storing Registers

SEG\_FLAG is the flag register to store occurrence state of each SYSERR source. When a SYSERR source occurs, the flag of the corresponding error source is set to 1. Each flag is cleared to 0 by reading the flag state of 1 and then writing 0.

Even if 0 is written to the flag (read as 0) in the subsequent write operation, the flag is set if an error occurs during the time between reading and writing.

Be sure to set bits 15 to 8, 5, 3, and 0 to 0.

**Access:** SEG\_FLAG can be read or written in 16-bit units.  
SEG\_FLAGL can be read or written in 8/1-bit units.

**Address:** SEG\_FLAG: FFFF 8C02<sub>H</sub>  
SEG\_FLAGL: FFFF 8C02<sub>H</sub>

**Value after reset:** 0000<sub>H</sub> (Initialized by any reset source.)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	SEG_FLAG DMAF	SEG_FLAG RMWF	—	SEG_FLAG RAMF	—	SEG_FLAG EXTF	SEG_FLAG ROMF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R

Table 3.41 SEG\_FLAG Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEG_FLAG DMAF	DMA transfer error flag DMAF is set to 1 when a DMA transfer error is detected during DMA access.
6	SEG_FLAG RMWF	Read modify write error flag SEG_FLAGRMWF is set to 1 if a read-modify-write error (ECC error in read phase) is detected during byte write or half-word write to the primary local RAM, secondary local RAM, or retention RAM.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SEG_FLAG RAMF	Primary local RAM/secondary local RAM/retention RAM area error flag SEG_FLAGRAMF is set to 1 if an error (ECC error in read phase) is detected when the CPU accesses data in the primary local RAM, secondary local RAM, or retention RAM area.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	SEG_FLAG EXTF	PBUS module access error flag EXTF is set to 1 if an error is detected when the CPU accesses the PBUS module.
1	SEG_FLAG ROMF	Code flash error flag ROMF is set to 1 if an error is detected when the CPU accesses data in the code flash area.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 3.6 Usage Notes

### 3.6.1 Synchronization of Store Instruction Completion and Subsequent Instruction Execution

When a control register is updated by a store instruction, there is a time gap between when the CPU executes the store instruction and when the control register is actually updated. Therefore, if the updated content of the control register is to be used by the instruction following the store instruction, the appropriate synchronization is required. The following shows the execution processes required to perform this synchronization.

#### (1) When the updated content of the control register is to be used by the subsequent instruction

Execute the procedure below if updating of a control register may make a difference to the order of program execution or if the correct operation of a subsequent instruction relies on the new value.

- (a) Store instructions for writing to peripheral function registers and instructions for access to DMA or INTc related registers

When an instruction for accessing a DMA- or INTc-related register (B) follows a store instruction for a peripheral function register (A), (B) may be executed before updating of the peripheral function register by (A) is actually completed. If the correct operation of a subsequent instruction relies on the new value of the control register, perform the following:

- Store instruction for the peripheral function register (A).
- Perform a dummy read of the above register.
- Execute the instruction for access to the DMA- or INTc-related register (B).

- (b) Store instructions for writing to DMA-related register or peripheral function register and EI/DI instructions, STSR instructions of ISPR/ICSR, or LDSR instructions to PMR

If an EI/DI instruction, STSR instruction of ISPR/ICSR, or LDSR instruction to PMR (B) follows a store instruction for a DMA-related register or peripheral function register (A), (B) may be executed before (A) is actually completed. If the correct operation of a subsequent instruction relies on the new value of the control register, perform the following:

- Store instruction for the DMA-related register or peripheral function register (A).
- Perform a dummy read of the above register.
- Execute the EI/DI instruction, STSR instruction of ISPR/ICSR, or LDSR instruction to PMR (B).

- (c) Store instructions for writing to INTc-related registers and EI/DI instructions, or STSR instructions of ISPR/ICSR, or LDSR instructions to PMR

If an EI/DI instruction, STSR instruction of ISPR/ICSR, or LDSR instruction to PMR (B) follows a store instruction for an INTc-related register (A), (B) may be executed before (A) is actually completed. If the correct operation of a subsequent instruction relies on the new value of the control register, perform the following:

- Store instruction for the INTc-related register (A).
- Perform a dummy read of the above register.
- Execute SYNCP.

- Execute the EI/DI instruction, STSR instruction of ISPR/ICSR, or LDSR instruction to PMR (B).

**(2) When the updated content of the control register or memory is to be used in the instruction fetch of the subsequent instruction**

For example, if an instruction is to be executed after it is written to memory. In this case, execute the following:

- Store instruction to update the memory (such as ST.W).
- Perform a dummy read of the above memory (such as LD.W).
- Execute SYNCI.
- Execute the subsequent instruction (such as a branch instruction).

### 3.6.2 System Register Hazards

To resolve potential hazards when updating the register values of some system registers, implement the following procedures.

- Instruction fetch:  
When fetching instructions after updating the following registers, start to do so only after executing the EIRET instruction, FERET instruction, or SYNCI instruction after executing the register update instruction.
  - PSW.UM, MCFG0.SPID, ASID, MPU: All related registers (Register number: sr\*, 5-7)
- SYSCALL instruction:  
When a SYSCALL instruction is to be executed after updating the register below, execute a SYNCOP instruction after the instruction to update the register and before the SYSCALL instruction.
  - SCCFG
- Load/Store:  
When executing instructions that involve load/store operations after updating the following registers, execute the load/store instruction only after executing the SYNCOP instruction after executing the register update instruction.
  - ASID, MPU protection area setting registers (Register number: sr\*, 6-7)

Do not execute instructions that involve load/store operations during the instruction preceding and the instruction following update of the following registers.

- MCTL.MA
- Interrupts  
Update the following registers in the interrupt disabled state (PSW.ID = 1).
  - PSW.EBV, EBASE, INTBP, ISPR, PMR, ICSR, INTCFG

### 3.6.3 Accessing Registers by Using Bit-Manipulation Instructions

Writing by using bit-manipulation instructions consists of read-modify-write processing in 8-bit units. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. If a register includes multiple flag bits, the read-modify-write operation may lead to the clearing of flags that were not actually targets for clearing.

### 3.6.4 Overwriting Context upon Acknowledgment of Multiple Exceptions

Acknowledgment of an exception depends on the type of exception source, regardless of the states of the ID and NP bits in the PSW register. When multiple exceptions are generated, the contents of the system register which hold the context information are overwritten. For the conditions for acknowledgment and whether correct return or recovery is possible for each exception source, see the List of Exception Sources in the *RH850G3K User's Manual: Software*.

### 3.6.5 Detection of ECC Errors during Prefetching

To maintain the throughput of instruction fetching, the CPU prefetches instructions (speculative instruction fetching) from an area to be executed beyond the current point of the running program. This prefetching may result in reading from memory areas to which no instruction codes have been assigned.

1. In the case of reading from a memory area to which no instruction codes have been assigned, the values read are not executed as instructions. Accordingly, a SYSERR is not generated in response to an ECC error detected in the case of reading for prefetching.
2. Since prefetching may lead to reading from memory areas to which no instruction codes have been assigned, leave at least 40 bytes of empty space after areas to which instruction codes are assigned when assigning instruction codes to memory.

Initialize the 40-byte space to prevent the generation of ECC errors.

## Section 4 Write-Protected Registers 写保护寄存器

This section contains a generic description of the write-protected registers.

The first part in this section describes the features specific to the write-protected registers, and the ensuing sections describe the various registers.

### 4.1 Overview

#### 4.1.1 Functional Overview

MCU需要一个特定的程序，使用写保护寄存器去设置会影响系统的重要寄存器设置

The RH850/F1L products require a special procedure using write-protected registers to set important registers that affect the system, such as clock, reset, and port-related registers. The settings of protected registers are protected against illegal writing by programs by requiring this special procedure. For details about the protected registers, see **Section 4.1.5, Write-Protection Target Registers**.

Write-protected registers are managed in units of protected registers called register protection clusters.

#### 4.1.2 Writing Procedure to Write-Protected Registers 往写保护寄存器中写入流程

Write access to a write-protected register is enabled by using the following protection unlock sequence:

1. Write the fixed value  $0000\ 00A5_H$  to the protection command register. **PROTCMD**
2. Write the desired value to the protected register. **将期望值写入保护寄存器**
3. Write the bit-wise inversion of the desired value to the protected register. **将期望值按位反转再次写入保护寄存器**
4. Write the desired value to the protected register. **重复2的步骤**
5. Verify that the desired value has been written to the protected register. **验证期望值已经被写入保护寄存器**  
Verify successful write of the desired value to the protected register by verifying that the error monitor bit in the protection status register is “0”. **保护状态寄存器中的error 监测bit为0，则说明期望值写入成功**  
In case the write was not successful, indicated by the error monitor bit set to “1”, the entire sequence has to be restarted at step 1. **如果没有成功（保护状态寄存器中的error监测bit为1），那么需要从步骤1开始重新操作**

If another register (second register) is accessed between step 1 and step 4 of the above sequence for writing to a write-protected register (first register), the protection mechanism operates as follows:  
**如果在上述的步骤1到步骤4之间操作另外一个保护寄存器，保护机制的工作如下所示**

- If the second register belongs to the same cluster, the write to the protected register fails (the error monitor bit is set to 1). The entire sequence has to be restarted at step 1.  
**如果第二个寄存器同样属于同一个cluster，那么写保护寄存器失败，整个顺序需要从步骤1重新开始**
- If the second register does not belong to the same cluster, the protection unlock sequence is not disrupted and the write to the first register completes successfully.  
**如果第二个寄存器不属于同一个cluster，保护解锁顺序不会被破坏，第一个寄存器可以顺利的完成**

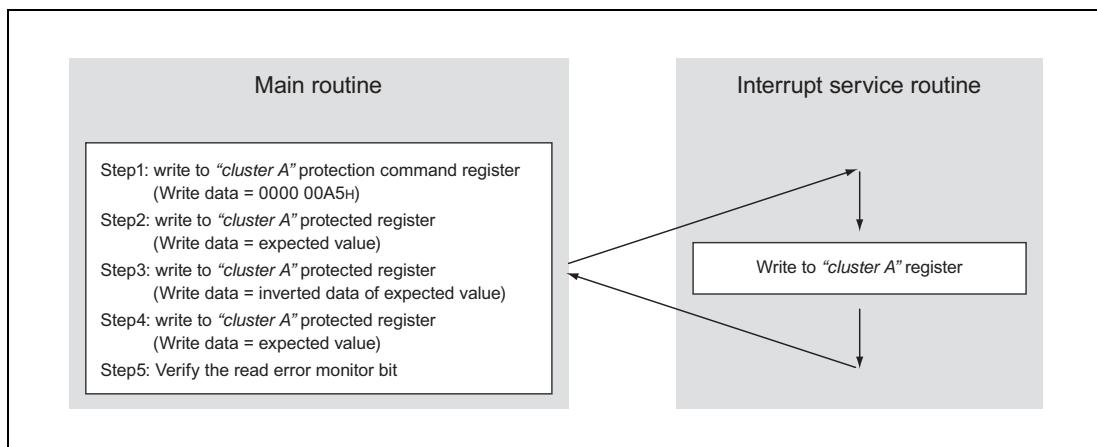
什么叫同一个  
cluster ? ? ? ? ? ? ?  
? 详见Table 4.1

### 4.1.3 Interrupt during Write Protection Unlock

If an interrupt occurs during the protection unlock sequence, the protection mechanism operates as follows: 在保护解锁序列期间，中断产生，那么保护机制的运行如下所示

- (1) If an interrupt request is accepted during the protection unlock sequence and write access to a register of the same cluster is performed 在保护解锁序列期间，中断产生，并且同一cluster下的寄存器写访问正在进行

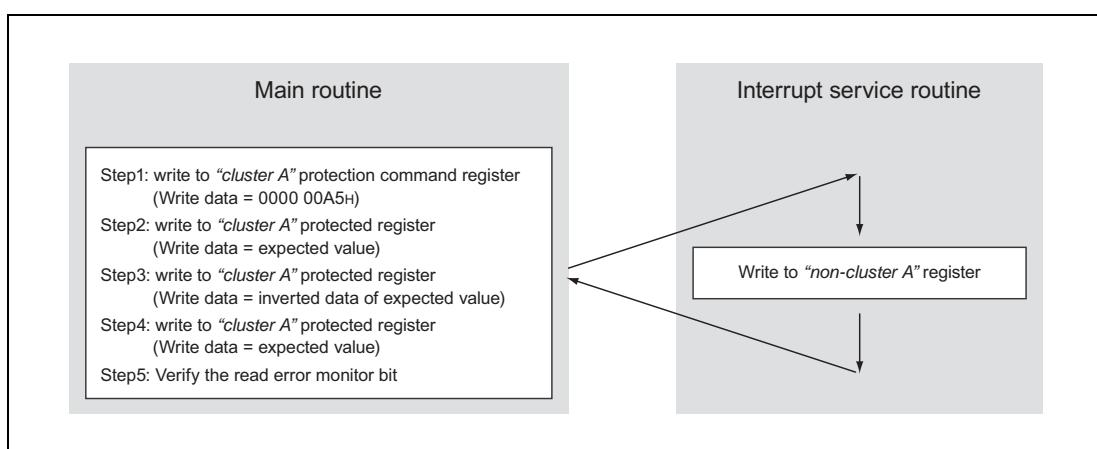
The protection unlock sequence is disrupted, so the write operation to the protected register cannot be completed upon returning from the interrupt service routine. **Figure 4.1** shows an execution example. 保护解锁序列被毁坏，因此向保护寄存器中的写操作在中断服务程序返回后 不能完成



**Figure 4.1 Example of Interruption of Register Protection Unlock Sequence**

- (2) If an interrupt request is accepted during the protection unlock sequence and write access to a register of a different cluster is performed 在保护解锁序列期间，中断产生，并且不同cluster下的寄存器写访问正在进行

The protection unlock sequence is not disrupted, so the write operation to the protected register is completed upon returning from the interrupt service routine. **Figure 4.2** shows an execution example. 保护解锁序列不会被毁坏，因此向保护寄存器中的写操作在中断服务程序返回后 能顺利完成



**Figure 4.2 Example of Successful Protection Unlock Sequence**

For more information on RH850/F1L register protection clusters, see **Section 4.1.5, Write-Protection Target Registers**.

#### 4.1.4 Emulation Break during Write Protection Unlock Sequence

If an emulation break occurs during the protection unlock sequence, e.g. because of a breakpoint hit, the register protection is suspended until normal operation is resumed.

Even if any register of the same cluster is accessed during the break, the protection unlock sequence is not disrupted and the error monitor bit is not set to 1.

#### 4.1.5 Write-Protection Target Registers

The registers that are protected through the write-protection control registers are listed below.

Table 4.1 Write-Protection Target Registers (1/2)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock controller	MOSCE	PROTCMD0	PROTS0	Control protection cluster 0
	ROSCE			
	SOSCE			
	CKSC_AWDTAD_CTL			
	CKSC_ATAUJS_CTL			
	CKSC_ATAUJD_CTL			
	CKSC_ARTCAS_CTL			
	CKSC_ARTCAD_CTL			
	CKSC_AADCAS_CTL			
	CKSC_AADCAD_CTL			
Stand-by function	STBC0PSC	PROTCMD1	PROTS1	Control protection cluster 1
	STBC0STPT			
	IOHOLD			
Reset function	LVICNT			
	SWRESA			
Clock Controller	PLLE			
	CKSC_CPUCLKS_CTL			
	CKSC_CPUCLKD_CTL			
	CKSC_IPERI1S_CTL			
	CKSC_IPERI2S_CTL			
	CKSC_ILINS_CTL			
	CKSC_IADCAS_CTL			
	CKSC_IADCAD_CTL			
	CKSC_ILIND_CTL			
	CKSC_ICANS_CTL			
	CKSC_ICANOSCD_CTL			
	CKSC_ICSIS_CTL			

Table 4.1 Write-Protection Target Registers (2/2)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock Monitor	CLMA0CTL0	CLMA0PCMD	CLMA0PS	Clock Monitor control protection cluster 0
	CLMA1CTL0	CLMA1PCMD	CLMA1PS	Clock Monitor control protection cluster 1
	CLMA2CTL0	CLMA2PCMD	CLMA2PS	Clock Monitor control protection cluster 2
	CLMATEST	PROTCMDCLMA	PROTSCLMA	Clock Monitor test protection cluster
Port* <sup>1</sup>	JPODC0	JPPCMD0	JPPRTS0	Port protection cluster 0
	PODC0	PPCMD0	PPRTS0	
	PODC1	PPCMD1	PPRTS1	
	PODC2	PPCMD2	PPRTS2	
	PODC8	PPCMD8	PPRTS8	
	PDSC0	PPCMD0	PPRTS0	
	PODC9	PPCMD9	PPRTS9	Port protection cluster 1
	PODC10	PPCMD10	PPRTS10	
	PODC11	PPCMD11	PPRTS11	
	PODC12	PPCMD12	PPRTS12	
	PODC18	PPCMD18	PPRTS18	
	PODC20	PPCMD20	PPRTS20	
Core Voltage Monitor	CVMF	PROTCMDCVM	PROTSCVM	Core Voltage Monitor protection cluster
	CVMDIAG			
Self-programming function	FLMDCNT	FLMDPCMD	FLMDPS	Self-programming protection cluster

Note 1. Each port group has its own protection command register and status register. For details, see **(1) Port protection clusters** on the next page.

### (1) Port protection clusters

The following port control registers have write protection function:

- Port open drain control registers (PODCn, JPODC0)
- Port drive strength control registers (PDSCn)

The write protected port registers are divided into two port protection clusters as shown in the following table:

**Table 4.2 Port Protection Clusters**

Port Protection Cluster	Port Group
0	JP0, P0, P1, P2, P8
1	P9, P10, P11, P12, P18, P20

#### NOTE

Each port group n has its own port protection command register PPCMDn and port protection status register PPROTSn.

However, any port protection command registers of the same port protection cluster can be used in the protection unlock sequence. For instance, PPCMD1 can be used to unlock the protection of PODC2.

## 4.2 Registers

### 4.2.1 List of Registers

The following table lists the write-protection control registers.

**Table 4.3 List of Write-Protection Control Registers (1/2)**

Register Name	Symbol	Address
<b>Control protection clusters</b>		
Protection command register 0	PROTCMD0	FFF8 0000 <sub>H</sub>
Protection command register 1	PROTCMD1	FFF8 8000 <sub>H</sub>
Protection status register 0	PROTS0	FFF8 0004 <sub>H</sub>
Protection status register 1	PROTS1	FFF8 8004 <sub>H</sub>
<b>Clock monitor control and test protection cluster</b>		
Protection command register 0	CLMA0PCMD	FFF8 C010 <sub>H</sub>
Protection command register 1	CLMA1PCMD	FFF8 D010 <sub>H</sub>
Protection command register 2	CLMA2PCMD	FFF8 E010 <sub>H</sub>
Protection status register 0	CLMA0PS	FFF8 C014 <sub>H</sub>
Protection status register 1	CLMA1PS	FFF8 D014 <sub>H</sub>
Protection status register 2	CLMA2PS	FFF8 E014 <sub>H</sub>
Protection command register	PROTCMDCLMA	FFF8 C200 <sub>H</sub>
Protection status register	PROTSCLMA	FFF8 C204 <sub>H</sub>
<b>Port protection cluster 0</b>		
Protection command registers	JPPCMD0	FFC2 04C0 <sub>H</sub>
	PPCMD0	FFC1 4C00 <sub>H</sub>
	PPCMD1	FFC1 4C04 <sub>H</sub>
	PPCMD2	FFC1 4C08 <sub>H</sub>
	PPCMD8	FFC1 4C20 <sub>H</sub>
Protection status registers	JPPROTS0	FFC2 04B0 <sub>H</sub>
	PPROTS0	FFC1 4B00 <sub>H</sub>
	PPROTS1	FFC1 4B04 <sub>H</sub>
	PPROTS2	FFC1 4B08 <sub>H</sub>
	PPROTS8	FFC1 4B20 <sub>H</sub>
<b>Port protection cluster 1</b>		
Protection command registers	PPCMD9	FFC1 4C24 <sub>H</sub>
	PPCMD10	FFC1 4C28 <sub>H</sub>
	PPCMD11	FFC1 4C2C <sub>H</sub>
	PPCMD12	FFC1 4C30 <sub>H</sub>
	PPCMD18	FFC1 4C48 <sub>H</sub>
	PPCMD20	FFC1 4C50 <sub>H</sub>

**Table 4.3 List of Write-Protection Control Registers (2/2)**

<b>Register Name</b>	<b>Symbol</b>	<b>Address</b>
Protection status registers	PPROTS9	FFC1 4B24 <sub>H</sub>
	PPROTS10	FFC1 4B28 <sub>H</sub>
	PPROTS11	FFC1 4B2C <sub>H</sub>
	PPROTS12	FFC1 4B30 <sub>H</sub>
	PPROTS18	FFC1 4B48 <sub>H</sub>
	PPROTS20	FFC1 4B50 <sub>H</sub>
<b>Core Voltage Monitor protection cluster</b>		
Protection command register	PROTCMDCVM	FFF5 0100 <sub>H</sub>
Protection status register	PROTSCVM	FFF5 0104 <sub>H</sub>
<b>Self-programming protection cluster</b>		
Protection command register	FLMDPCMD	FFA0 0004 <sub>H</sub>
Protection error status register	FLMDPS	FFA0 0008 <sub>H</sub>

## 4.2.2 Details of Control Protection Cluster Registers

### 4.2.2.1 PROTCMDn — Protection Command Register

This register is used to initiate the write protection unlock sequence for write-protected registers.

#### Index n

An index “n” denotes the number of protection command registers. For details, see **Table 4.1, Write-Protection Target Registers**.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** See **Table 4.3, List of Write-Protection Control Registers**.

**Value after reset:** XXXX XX00<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
PCMDn[7:0]																

**Table 4.4 PROTCMDn Register Contents**

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	PCMDn[7:0]	Protection command register bits to enable writing to protection target registers of control protection cluster

#### 4.2.2.2 PROTSn — Protection Status Register

This register indicates the status of the protection unlock sequence performed by PROTCMDn.

##### Index n

An index “n” denotes the number of protection command registers. For details, see **Table 4.1, Write-Protection Target Registers**.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** See **Table 4.3, List of Write-Protection Control Registers**.

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PROTSnERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 4.5 PROTSn Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PROTSnERR	Write sequence protection error monitor 0: No protection error occurred 1: Protection error occurred

## 4.2.3 Details of Clock Monitor Protection Cluster Registers

### 4.2.3.1 CLMAnPCMD — CLMAn Protection Command Register

This register is a protection command register for the CLMAnCTL0 register.

#### Index n

An index “n” denotes the number of protection command registers. For details, see **Table 4.1, Write-Protection Target Registers**.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** See **Table 4.3, List of Write-Protection Control Registers**.

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
CLMAnREG[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

**Table 4.6 CLMAnPCMD Register Contents**

Bit Position	Bit Name	Function
7 to 0	CLMAnREG[7:0]	Protection command register bits to enable writing to the CLMAnCTL0 register

#### 4.2.3.2 CLMAnPS — CLMAn Protection Status Register

This register is used to verify whether the write-protected register (CLMAnCTL0) has been successfully written or not.

##### Index n

An index “n” denotes the number of protection command registers. For details, see **Table 4.1, Write-Protection Target Registers**.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** See **Table 4.3, List of Write-Protection Control Registers**.

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMAnPRERR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 4.7 CLMAnPS Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	CLMAnPRERR	Write protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

#### 4.2.3.3 PROTCMDCLMA — Clock Monitor Test Protection Command Register

This register is a protection command register for the CLMATEST register.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** See [Table 4.3, List of Write-Protection Control Registers](#).

**Value after reset:** XXXX XX00<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMATREG[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

**Table 4.8 PROTCMDCLMA Register Contents**

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write “0”.
7 to 0	CLMATREG[7:0]	Protection command register bits to enable writing to the CLMATEST register

#### 4.2.3.4 PROT SCLMA — Clock Monitor Test Protection Status Register

This register is used to verify whether the write-protected register (CLMATEST) has been successfully written or not.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** See Table 4.3, List of Write-Protection Control Registers.

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMATPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 4.9** PROT SCLMA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CLMATPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

## 4.2.4 Details of Core Voltage Monitor Register

### 4.2.4.1 PROTCMDCVM — Core Voltage Monitor Protection Command Register

This register is a protection command register for the CVMF and CVMDIAG registers.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** See Table 4.3, List of Write-Protection Control Registers.

**Value after reset:** XXXX XX00<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
CVMFREG[7:0]																

Table 4.10 PROTCMDCVM Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	CVMFREG[7:0]	Protection command register bits to enable writing to the CVMF and CVMDIAG registers

#### 4.2.4.2 PROTSCVM — Core Voltage Monitor Protection Status Register

This register is used to verify whether the write-protected register (CVMF, CVMDIAG) has been successfully written or not.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** See Table 4.3, List of Write-Protection Control Registers.

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVMFP RERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.11 PROTSCVM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CVMFPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

## 4.2.5 Details of Port Protection Cluster Registers

### 4.2.5.1 PPCMDn — Port Protection Command Register

PPCMDn is a protection command register for port group n.

#### Index n

An index “n” denotes the number of protection command registers. For details, see **Table 4.1, Write-Protection Target Registers**.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** See **Table 4.3, List of Write-Protection Control Registers**.

**Value after reset:** XXXX XX00<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

#### NOTE

The protection command register for port group JP0 is JPPCMD0. Its bits are JPPCMD0[7:0].

**Table 4.12 PPCMDn Register Contents**

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	PPCMDn[7:0]	Protection command register bits that enable writing to port protection cluster registers

#### 4.2.5.2 PPROTS<sub>n</sub> — Port Protection Status Register

PPROTS<sub>n</sub> is a protection status register for write-protected registers of port group n. It indicates the status of the protection sequence operated by PPCMDn.

##### Index n

An index “n” denotes the number of protection command registers. For details, see **Table 4.1, Write-Protection Target Registers**.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** See **Table 4.3, List of Write-Protection Control Registers**.

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPROT SnPRE RR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

##### NOTE

The protection status register for port group JP0 is JPPROTS0. Its bit is JPPROTS0PRERR.

**Table 4.13 PPROTS<sub>n</sub> Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PPROTS <sub>n</sub> PRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

## 4.2.6 Details of Self-Programming Protection Cluster Registers

### 4.2.6.1 FLMDPCMD — FLMD Protection Command Register

FLMDPCMD is a protection command register for the FLMDCNT register.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** See Table 4.3, List of Write-Protection Control Registers.

**Value after reset:** XXXX XX00<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FLMDPC[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.14 FLMDPCMD Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	FLMDPC[7:0]	Protection command register bits that enable writing to FLMDCNT register

#### 4.2.6.2 FLMDPS — FLMD Protection Error Status Register

This register is used to verify whether the write-protected register (FLMDCNT) has been successfully written or not.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** See Table 4.3, List of Write-Protection Control Registers.

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDP RERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.15 FLMDPS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	FLMDPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

## Section 5 Operating Mode

This section describes the operating mode and mode selection of the RH850/F1L.

The RH850/F1L has the operating mode shown below.

- **Normal operating mode**

This mode is for execution of the user program. The on-chip debug functions also use this mode. If FLMD0 is pulled up high during operation in this mode, writing to the code flash memory through self-programming is enabled.

- **Serial programming mode**

The dedicated flash memory programmer enables erasing/writing to flash memory.

- **Boundary scan mode**

This mode allows boundary scan tests compliant with IEEE Standard 1149.1.

When an external reset or power-on-clear reset is generated, the state of the FLMD0, FLMD1, MODE0, and MODE1 pins are used to determine the operating mode after reset is released. The operating mode is fixed by the release of these reset factors. **Table 5.1** lists the relationship between the pin settings and the operating mode.

**Table 5.1 Selection of Operating Mode**

Pins				Operating Mode
FLMD0	FLMD1 (P10_8)	MODE0 (P10_1)	MODE1 (P10_2)	
0	x	x	x	Normal operating mode
1	0	x	x	Serial programming mode
1	1	0	1	Boundary scan mode
Other than the above				Setting prohibited

### CAUTION

To change operating mode, restart from power-on clear reset. (Remove the power supply once and apply it again.) In the case of only by the external reset, some functions are not initialized after the mode transitions. For details of functions not initialized by the external reset, see Section 8.1.1, Reset Sources.

## Section 6 Exceptions/Interrupts

### 6.1 Features

为响应某个事件，从当前运行的程序切换到另一个不同程序的过程叫做exception

The act of branching from a currently running program to a different program in response to an event is called an exception. This microcontroller supports the following types of exceptions.

The details on exceptions are described in the *RH850G3K User's Manual: Software*.

Table 6.1 List of Exception Sources

Name	Symbol	Source	Priority	Saved to
Reset	RESET	Reset input	High	—
FE level non-maskable interrupt <sup>*1</sup>	FENMI	FENMI input		FE
System error exception	SYSERR	SYSERR input		FE
FE level maskable interrupt <sup>*1</sup>	FEINT	FEINT input		FE
EI level maskable interrupt <sup>*1</sup>	EIINT	Interrupt controller		EI
Memory protection exception (execution right)	MIP	Memory protection violation		FE
System error exception	SYSERR	Error input at instruction fetch		FE
Reserved instruction exception	RIE	Execution of reserved instruction		FE
Coprocessor unusable exception	UCPOP	Execution of coprocessor instruction/ access right violation		FE
Privileged instruction exception	PIE	Execution of privileged instruction		FE
Misalign exception	MAE	Generation of misaligned access		FE
Memory protection exception (access right)	MDP	Memory protection violation		FE
System call	SYSCALL	Execution of SYSCALL instruction		EI
FE level trap	FETRAP	Execution of FETRAP instruction		FE
EI level trap 0	TRAP0	Execution of TRAP instruction		EI
EI level trap 1	TRAP1	Execution of TRAP instruction	Low	EI

Note 1. These interrupt exceptions are described in this section.

## (1) Interrupts

The following three exceptions in **Table 6.1** are called interrupts, and are described in this section.

- FE level non-maskable interrupt (FENMI)

An FENMI interrupt is acknowledged even if another FE level interrupt - FEINT - has been generated.

- FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
- Return from an FENMI interrupt is not possible and neither is recovery.

- FE level maskable interrupt (FEINT)

- FEINT can be acknowledged if the CPU system register PSW.NP = 0. It is masked if PSW.NP = 1.
- Return from an FEINT interrupt is possible and so is recovery.

- EI level maskable interrupt (EIINT)

An EIINT interrupt can be acknowledged if an FE level interrupt - FENMI or FEINT - has not been generated.

- EIINT can be acknowledged if the CPU system register PSW.NP = 0.
- It is masked if PSW.NP = 1, EIINT with a higher priority is being processed, or PSW.ID = 1.
- Return from an EIINT interrupt is possible and so is recovery.
- Interrupt masking can be specified for each interrupt channel.
- 8 interrupt priority levels can be specified for each interrupt channel.

EIINT not be acknowledged

- In this section, the EIINT that corresponds to interrupt channel n is indicated by “INTn”, whereas the EIINT that corresponds to interrupt source xxx is indicated by “INTxxx”.

For the PSW register, see **Section 3.3.2.5, PSW — Program Status Word** and the *RH850G3K User’s Manual: Software*.

### NOTE

**返回**：指示程序是否从中断处恢复运行

Return: Indicates whether or not the program can resume from where it was interrupted.

Recovery: Indicates whether or not the processor status (status of processor resources including general purpose registers and system registers) can be restored to the status they were in when the program was interrupted.

**恢复**：指示处理器的状态是否可以恢复到中断发生时的状态，此状态包含处理器资源（一般寄存器和系统寄存器）

These interrupt sources are described in the following pages.

## 6.2 RH850/F1L Interrupt Sources

### 6.2.1 Interrupt Sources

#### 6.2.1.1 FE Level Non-Maskable Interrupts

##### (1) Priority

See **Table 6.1, List of Exception Sources.**

##### (2) Return PC

(不可返回或者恢复的中断)

Return or recovery from an FE non-maskable interrupt is not possible.

##### (3) Status Register

See **Section 6.4.3, FNC — FE Level NMI Status Register.**

##### (4) Return Instruction

None

**Table 6.2 FE Level Non-Maskable Interrupt Requests**

Interrupt			Interrupt Request		Unit	Priority	Exception Source Code	Handler Address 00000...
Symbol	Control Register		Name	Source				
	Name	Address	Name	Source				
FENMI	FNC	FFFF 9078 <sub>H</sub>	TNMI	NMI pin	Port	*1	E0 <sub>H</sub>	0E0H
			WDTA0NMI	WDTA0 FENMI interrupt	WDTA0			
			WDTA1NMI	WDTA1 FENMI interrupt	WDTA1			

Note 1. See **Table 6.1, List of Exception Sources.**

The source of the FENMI interrupt can be evaluated by a dedicated flag register. See **Section 6.2.2, FE Level Non-Maskable Interrupt Sources** for details.

### 6.2.1.2 FE Level Maskable Interrupts

#### (1) Priority

See **Table 6.1, List of Exception Sources**.

#### (2) Return PC

The return PC returned from an interrupt handling routine by the FERET instruction is the PC from when the program was suspended (current PC).

#### (3) Status Register

See **Section 6.4.4, FIC — FE Level Maskable Interrupt Status Register**

#### (4) Return Instruction

FERET

**Table 6.3 FE Level Maskable Interrupt Requests**

Interrupt			Interrupt Request		Unit	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Priority	Exception Source Code	Handler Address 0000...	
Symbol	Control Register		Name	Source											
	Name	Address													
FEINT	FIC	FFFF 907A <sub>H</sub>	INTLVIL	LVI voltage detection (falling)	LVI	✓	✓	✓	✓	✓	✓	✓	*1	F0 <sub>H</sub>	0F0 <sub>H</sub>
			INTECCDEEP0	ECC 2-bit error interrupt of data flash	Data flash	✓	✓	✓	✓	✓	✓	✓			
			INTECCDCNRAM	ECC 2-bit error interrupt of RSCAN0	RSCAN0	✓	✓	✓	✓	✓	✓	✓			
			INTECCDCSIH0	ECC 2-bit error interrupt of CSIH0	CSIH0	✓	✓	✓	✓	✓	✓	✓			
			INTECCDCSIH1	ECC 2-bit error interrupt of CSIH1	CSIH1	—	—	✓	✓	✓	✓	✓			
			INTECCDCSIH2	ECC 2-bit error interrupt of CSIH2	CSIH2	—	—	✓	✓	✓	✓	✓			
			INTECCDCSIH3	ECC 2-bit error interrupt of CSIH3	CSIH3	—	—	—	✓	✓	✓	✓			
			INTECCSDFLI0	ECC 1-bit error interrupt of Code Flash	Code flash	✓	✓	✓	✓	✓	✓	✓			
			INTECCRAM <sup>*2</sup>	ECC 1-bit error interrupt of RAM	RAM	✓	✓	✓	✓	✓	✓	✓			
			INTOSTM0_FE <sup>*3</sup>	OSTM0 interrupt	OSTM0	✓	✓	✓	✓	✓	✓	✓			
			INTLVIH	LVI voltage detection (rising)	LVI	✓	✓	✓	✓	✓	✓	✓			

Note 1. See **Table 6.1, List of Exception Sources**.

Note 2. The primary local RAM is not supported by 256-Kbyte products. Only 2-Mbyte products have secondary local RAM.

Note 3. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it both ways at the same time is not possible.

When INTOSTM0 is used as FEINT, it is generated by the TSU (timing supervision unit) function of OSTM0.

\* Timing monitor (TSU)

This prevents the illicit use of CPU time by non-trusted programs, manages properties, and controls the intervals over which interrupts are disabled.

### 6.2.1.3 EI Level Maskable Interrupts

#### (1) Interrupt Naming Rules

The composition of the interrupt request signal names, their assigned interrupt control registers and the bits in these registers follow special rules.

In the following the name of the specific interrupt request is represented by  $\langle name \rangle$ .

For details of the names used in **IC** $\langle name \rangle$ , see **Table 6.4**.

- Interrupt request name: INT $\langle name \rangle$

The prefix “INT” is appended to  $\langle name \rangle$ .

- Interrupt control register: IC $\langle name \rangle$

The prefix “IC” is appended to  $\langle name \rangle$ .

The 16-bit register **IC** $\langle name \rangle$  can also be accessed in byte units.

**16bit寄存器** **IC** $\langle name \rangle$  可以以字节的方式访问

**低8位** – Low byte (bits [7:0]) of the **IC** $\langle name \rangle$  register: **IC** $\langle name \rangle$ **L**

The suffix “L” is appended to the register name **IC** $\langle name \rangle$ .

**高8位** – High byte (bits [15:8]) of the **IC** $\langle name \rangle$  register: **IC** $\langle name \rangle$ **H**

The suffix “H” is appended to the register name **IC** $\langle name \rangle$ .

- Interrupt control register bit names: CT $\langle name \rangle$ , RF $\langle name \rangle$ , MK $\langle name \rangle$ , TB $\langle name \rangle$ ,

P2 $\langle name \rangle$ , P1 $\langle name \rangle$ , P0 $\langle name \rangle$

The bit prefix “CT”, “RF”, “MK”, “TB”, “P2”, “P1”, or “P0” is appended to the interrupt  $\langle name \rangle$ .

#### Example

The interrupt request from channel 2 of TAUD0 channel ( $\langle name \rangle = TAUD0I2$ ) is named

**INTTAUD0I2**

The related interrupt control registers are

**ICTAUD0I2**, **ICTAUD0I2L**, **ICTAUD0I2H**

The bits in this register are

**CTTAUD0I2**, **RFTAUD0I2**, **MKTAUD0I2**, **TBTAUD0I2**, **P2TAUD0I2**,  
**P1TAUD0I2**, **P0TAUD0I2**

#### (2) Priority

See **Table 6.1, List of Exception Sources**.

#### (3) Return PC

The return PC returned from an interrupt handling routine by the EIRET instruction is the PC from when the program was suspended (current PC).

#### (4) Control Register

EI level maskable interrupt control register

See **Section 6.4.1, ICxxx — EI Level Interrupt Control Registers**.

#### (5) Return Instruction

EIRET instruction

## (6) Configuration

EI level maskable interrupts are supported on a total of 288 channels with a cascade connection of INTC1 and INTC2.

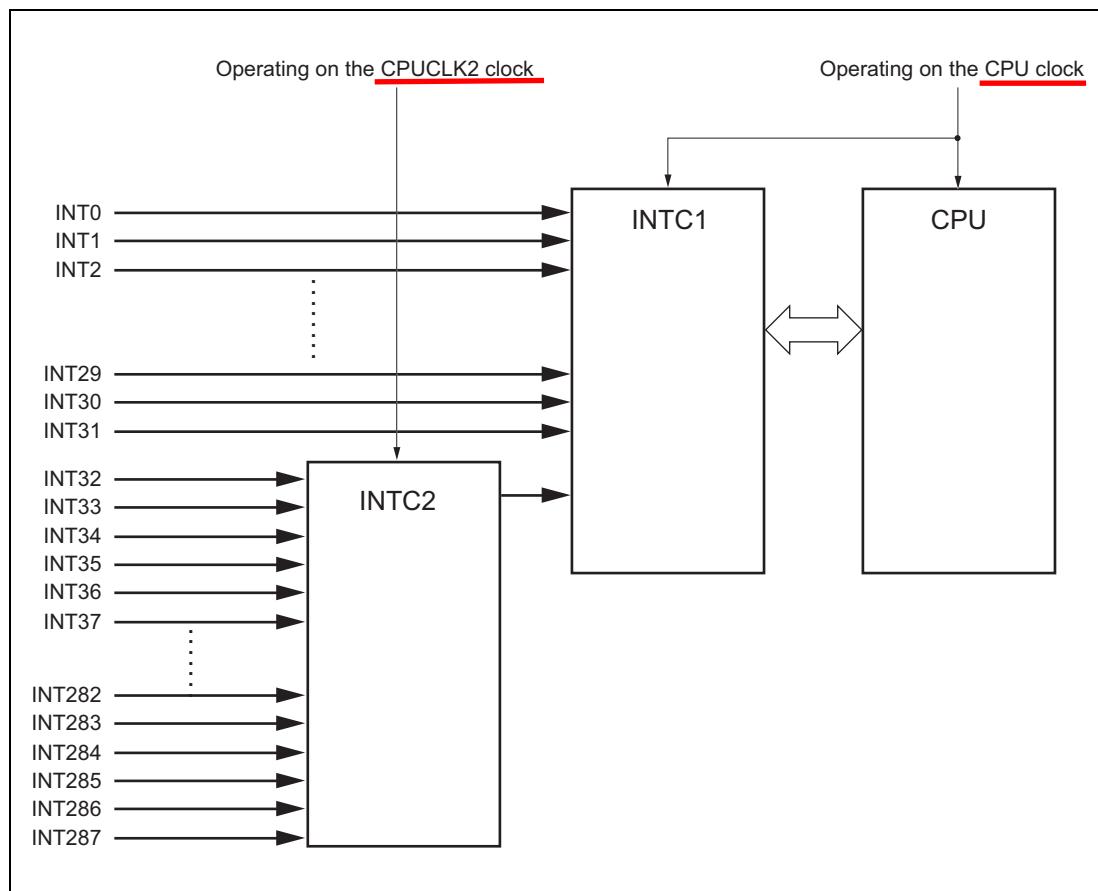


Figure 6.1 Configuration Diagram of EI Level Maskable Interrupt

### CAUTION

As CPUCLK2 is the operating clock for INTC2, the INT32 to INT287 interrupts, which are connected to INTC2, are delayed more than the interrupts directly connected to INTC1. 因为CPUCLK2连接到INTC，所以INT32 - INT287比直接连接到INTC1的中断会有一些延迟

**Table 6.4** lists EI level maskable interrupts.

中断源

**Table 6.4 EI Level Maskable Interrupt Sources (1/8)**

Interrupt No.* <sup>1</sup>	Interrupt		Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)* <sup>8</sup>								
	Control Register		Name	Source	Unit	Detection Type* <sup>2</sup>								Direct Jumping to an Address	Reference to a Table* <sup>5</sup>							
	Name	Address												RINT = 0	RINT = 1							
0	ICTAUD0I0	FFFF 9000 <sub>H</sub>	INTTAUD0I0* <sup>6</sup>	Interrupt for TAUD0 channel 0	TAUD0	Edge	1000 <sub>H</sub>	✓	✓	✓	✓	✓	✓	*3	*4	+000 <sub>H</sub>						
	ICCSIH2IC_1	FFFF 9000 <sub>H</sub>	INTCSIH2IC_1* <sup>6</sup>	CSIH2 communication status interrupt	CSIH2	Edge		—	—	—	—	—	—									
1	ICTAUD0I2	FFFF 9002 <sub>H</sub>	INTTAUD0I2* <sup>6</sup>	Interrupt for TAUD0 channel 2	TAUD0	Edge	1001 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+004 <sub>H</sub>								
	ICCSIH3IC_1	FFFF 9002 <sub>H</sub>	INTCSIH3IC_1* <sup>6</sup>	CSIH3 communication status interrupt	CSIH3	Edge		—	—	—	—	—	—									
2	ICTAUD0I4	FFFF 9004 <sub>H</sub>	INTTAUD0I4	Interrupt for TAUD0 channel 4	TAUD0	Edge	1002 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+008 <sub>H</sub>						
3	ICTAUD0I6	FFFF 9006 <sub>H</sub>	INTTAUD0I6	Interrupt for TAUD0 channel 6	TAUD0	Edge	1003 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+00C <sub>H</sub>						
4	ICTAUD0I8	FFFF 9008 <sub>H</sub>	INTTAUD0I8	Interrupt for TAUD0 channel 8	TAUD0	Edge	1004 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+010 <sub>H</sub>						
5	ICTAUD0I10	FFFF 900A <sub>H</sub>	INTTAUD0I10* <sup>6</sup>	Interrupt for TAUD0 channel 10	TAUD0	Edge	1005 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+014 <sub>H</sub>								
	ICCSIH3IR_1	FFFF 900A <sub>H</sub>	INTCSIH3IR_1* <sup>6</sup>	CSIH3 receive status interrupt	CSIH3	Edge		—	—	—	—	—	—									
6	ICTAUD0I12	FFFF 900C <sub>H</sub>	INTTAUD0I12* <sup>6</sup>	Interrupt for TAUD0 channel 12	TAUD0	Edge	1006 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+018 <sub>H</sub>								
	ICCSIH3IRE_1	FFFF 900C <sub>H</sub>	INTCSIH3IRE_1* <sup>6</sup>	CSIH3 communication error interrupt	CSIH3	Edge		—	—	—	—	—	—									
7	ICTAUD0I14	FFFF 900E <sub>H</sub>	INTTAUD0I14* <sup>6</sup>	Interrupt for TAUD0 channel 14	TAUD0	Edge	1007 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+01C <sub>H</sub>								
	ICCSIH3JJC_1	FFFF 900E <sub>H</sub>	INTCSIH3JJC_1* <sup>6</sup>	CSIH3 job completion interrupt	CSIH3	Edge		—	—	—	—	—	—									
8	ICTAPA0IPEK0	FFFF 9010 <sub>H</sub>	INTTAPA0IPEK0* <sup>6</sup>	TAPA0 peak interrupt 0	TAPA0	Edge	1008 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+020 <sub>H</sub>								
	ICCSIH1IC_1	FFFF 9010 <sub>H</sub>	INTCSIH1IC_1* <sup>6</sup>	CSIH1 communication status interrupt	CSIH1	Edge		—	—	—	—	—	—									
9	ICTAPA0IVLY0	FFFF 9012 <sub>H</sub>	INTTAPA0IVLY0* <sup>6</sup>	TAPA0 valley interrupt 0	TAPA0	Edge	1009 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+024 <sub>H</sub>								
	ICCSIH1IR_1	FFFF 9012 <sub>H</sub>	INTCSIH1IR_1* <sup>6</sup>	CSIH1 receive status interrupt	CSIH1	Edge		—	—	—	—	—	—									
10	ICADCA0I0	FFFF 9014 <sub>H</sub>	INTADCA0I0	ADCA0 scan group 1 (SG1) end interrupt	ADCA0	Edge	100A <sub>H</sub>	✓	✓	✓	✓	✓	✓			+028 <sub>H</sub>						
11	ICADCA0I1	FFFF 9016 <sub>H</sub>	INTADCA0I1	ADCA0 scan group 2 (SG2) end interrupt	ADCA0	Edge	100B <sub>H</sub>	✓	✓	✓	✓	✓	✓			+02C <sub>H</sub>						
12	ICADCA0I2	FFFF 9018 <sub>H</sub>	INTADCA0I2	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge	100C <sub>H</sub>	✓	✓	✓	✓	✓	✓			+030 <sub>H</sub>						
13	ICDCUTDI	FFFF 901A <sub>H</sub>	INTDCUTDI	Dedicated interrupt for on-chip debug function	Port	Edge	100D <sub>H</sub>	✓	✓	✓	✓	✓	✓			+034 <sub>H</sub>						
14	ICRCANGERR	FFFF 901C <sub>H</sub>	INTRCANGERR	CAN global error interrupt	RSCAN0	Level	100E <sub>H</sub>	✓	✓	✓	✓	✓	✓			+038 <sub>H</sub>						
15	ICRCANGRECC	FFFF 901E <sub>H</sub>	INTRCANGRECC	CAN receive FIFO interrupt	RSCAN0	Level	100F <sub>H</sub>	✓	✓	✓	✓	✓	✓			+03C <sub>H</sub>						
16	ICRCAN0ERR	FFFF 9020 <sub>H</sub>	INTRCAN0ERR	CAN0 error interrupt	RSCAN0	Level	1010 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+040 <sub>H</sub>						
17	ICRCAN0REC	FFFF 9022 <sub>H</sub>	INTRCAN0REC	CAN0 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	1011 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+044 <sub>H</sub>						
18	ICRCAN0TRX	FFFF 9024 <sub>H</sub>	INTRCAN0TRX	CAN0 transmit interrupt	RSCAN0	Level	1012 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+048 <sub>H</sub>						
19	ICCSIG0IC	FFFF 9026 <sub>H</sub>	INTCSIG0IC* <sup>6</sup>	CSIG0 communication status interrupt	CSIG0	Edge	1013 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+04C <sub>H</sub>								
	ICCSIH1IRE_1	FFFF 9026 <sub>H</sub>	INTCSIH1IRE_1* <sup>6</sup>	CSIH1 communication error interrupt	CSIH1	Edge		—	—	—	—	—	—									
20	ICCSIG0IR	FFFF 9028 <sub>H</sub>	INTCSIG0IR* <sup>6</sup>	CSIG0 receive status interrupt	CSIG0	Edge	1014 <sub>H</sub>	✓	✓	✓	✓	✓	✓	+050 <sub>H</sub>								
	ICCSIH1JJC_1	FFFF 9028 <sub>H</sub>	INTCSIH1JJC_1* <sup>6</sup>	CSIH1 job complete interrupt	CSIH1	Edge		—	—	—	—	—	—									
21	ICCSIH0IC	FFFF 902A <sub>H</sub>	INTCSIH0IC	CSIH0 communication status interrupt	CSIH0	Edge	1015 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+054 <sub>H</sub>						
22	ICCSIH0IR	FFFF 902C <sub>H</sub>	INTCSIH0IR	CSIH0 receive status interrupt	CSIH0	Edge	1016 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+058 <sub>H</sub>						
23	ICCSIH0IRE	FFFF 902E <sub>H</sub>	INTCSIH0IRE	CSIH0 communication error interrupt	CSIH0	Edge	1017 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+05C <sub>H</sub>						
24	ICCSIH0JJC	FFFF 9030 <sub>H</sub>	INTCSIH0JJC	CSIH0 job completion interrupt	CSIH0	Edge	1018 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+060 <sub>H</sub>						
25	ICRLIN30	FFFF 9032 <sub>H</sub>	INTRLIN30	RLIN30 interrupt	RLIN30	Edge	1019 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+064 <sub>H</sub>						
26	ICRLIN30UR0	FFFF 9034 <sub>H</sub>	INTRLIN30UR0	RLIN30 transmit interrupt	RLIN30	Edge	101A <sub>H</sub>	✓	✓	✓	✓	✓	✓			+068 <sub>H</sub>						
27	ICRLIN30UR1	FFFF 9036 <sub>H</sub>	INTRLIN30UR1	RLIN30 reception complete interrupt	RLIN30	Edge	101B <sub>H</sub>	✓	✓	✓	✓	✓	✓			+06C <sub>H</sub>						
28	ICRLIN30UR2	FFFF 9038 <sub>H</sub>	INTRLIN30UR2	RLIN30 status interrupt	RLIN30	Edge	101C <sub>H</sub>	✓	✓	✓	✓	✓	✓			+070 <sub>H</sub>						
29	ICP0	FFFF 903A <sub>H</sub>	INTP0* <sup>6</sup>	External interrupt	Port	Edge	101D <sub>H</sub>	✓	✓	✓	✓	✓	✓	+074 <sub>H</sub>								
	ICCSIH2IR_1	FFFF 903A <sub>H</sub>	INTCSIH2IR_1* <sup>6</sup>	CSIH2 reception status interrupt	CSIH2	Edge		—	—	—	—	—	—									

Table 6.4 EI Level Maskable Interrupt Sources (2/8)

Interrupt		Interrupt Request				Exception Source Code	Handler Address (Offset) <sup>*8</sup>	Direct Jumping to an Address		Reference to a Table <sup>*5</sup>			
Interrupt No. <sup>*1</sup>	Control Register		Name	Source	Unit	Detection Type <sup>*2</sup>		48 pins	64 pins	80 pins	100 pins	144 pins	
	Name	Address						RINT = 0	RINT = 1	*3	*4		
30	ICP1	FFFF 903C <sub>H</sub>	INTP1 <sup>*6</sup>	External interrupt	Port	Edge	101E <sub>H</sub>	✓	✓	✓	✓	✓	+078 <sub>H</sub>
	ICCSIH2IRE_1	FFFF 903C <sub>H</sub>	INTCSIH2IRE_1 <sup>*6</sup>	CSIH2 communication error interrupt	CSIH2	Edge		—	—				
31	ICP2	FFFF 903E <sub>H</sub>	INTP2 <sup>*6</sup>	External interrupt	Port	Edge	101F <sub>H</sub>	✓	✓	✓	✓	✓	+07C <sub>H</sub>
	ICCSIH2JC_1	FFFF 903E <sub>H</sub>	INTCSIH2JC_1 <sup>*6</sup>	CSIH2 job completion interrupt	CSIH2	Edge		—	—				
32	ICWDTA0	FFFF A040 <sub>H</sub>	INTWDTA0	WDTA0 75% interrupt	WDTA0	Edge	1020 <sub>H</sub>	✓	✓	✓	✓	✓	+080 <sub>H</sub>
33	ICWDTA1	FFFF A042 <sub>H</sub>	INTWDTA1	WDTA1 75% interrupt	WDTA1	Edge	1021 <sub>H</sub>	✓	✓	✓	✓	✓	+084 <sub>H</sub>
34	ICP3	FFFF A044 <sub>H</sub>	INTP3	External interrupt	Port	Edge	1022 <sub>H</sub>	✓	✓	✓	✓	✓	+088 <sub>H</sub>
35	ICP4	FFFF A046 <sub>H</sub>	INTP4	External interrupt	Port	Edge	1023 <sub>H</sub>	✓	✓	✓	✓	✓	+08C <sub>H</sub>
36	ICP5	FFFF A048 <sub>H</sub>	INTP5	External interrupt	Port	Edge	1024 <sub>H</sub>	✓	✓	✓	✓	✓	+090 <sub>H</sub>
37	ICP10	FFFF A04A <sub>H</sub>	INTP10	External interrupt	Port	Edge	1025 <sub>H</sub>	✓	✓	✓	✓	✓	+094 <sub>H</sub>
38	ICP11	FFFF A04C <sub>H</sub>	INTP11	External interrupt	Port	Edge	1026 <sub>H</sub>	✓	✓	✓	✓	✓	+098 <sub>H</sub>
39	ICTAUD0I1	FFFF A04E <sub>H</sub>	INTTAUD0I1	Interrupt for TAUD0 channel 1	TAUD0	Edge	1027 <sub>H</sub>	✓	✓	✓	✓	✓	+09C <sub>H</sub>
40	ICTAUD0I3	FFFF A050 <sub>H</sub>	INTTAUD0I3	Interrupt for TAUD0 channel 3	TAUD0	Edge	1028 <sub>H</sub>	✓	✓	✓	✓	✓	+0A0 <sub>H</sub>
41	ICTAUD0I5	FFFF A052 <sub>H</sub>	INTTAUD0I5	Interrupt for TAUD0 channel 5	TAUD0	Edge	1029 <sub>H</sub>	✓	✓	✓	✓	✓	+0A4 <sub>H</sub>
42	ICTAUD0I7	FFFF A054 <sub>H</sub>	INTTAUD0I7	Interrupt for TAUD0 channel 7	TAUD0	Edge	102A <sub>H</sub>	✓	✓	✓	✓	✓	+0A8 <sub>H</sub>
43	ICTAUD0I9	FFFF A056 <sub>H</sub>	INTTAUD0I9	Interrupt for TAUD0 channel 9	TAUD0	Edge	102B <sub>H</sub>	✓	✓	✓	✓	✓	+0AC <sub>H</sub>
44	ICTAUD0I11	FFFF A058 <sub>H</sub>	INTTAUD0I11	Interrupt for TAUD0 channel 11	TAUD0	Edge	102C <sub>H</sub>	✓	✓	✓	✓	✓	+0B0 <sub>H</sub>
45	ICTAUD0I13	FFFF A05A <sub>H</sub>	INTTAUD0I13	Interrupt for TAUD0 channel 13	TAUD0	Edge	102D <sub>H</sub>	✓	✓	✓	✓	✓	+0B4 <sub>H</sub>
46	ICTAUD0I15	FFFF A05C <sub>H</sub>	INTTAUD0I15	Interrupt for TAUD0 channel 15	TAUD0	Edge	102E <sub>H</sub>	✓	✓	✓	✓	✓	+0B8 <sub>H</sub>
47	ICADCA0ERR	FFFF A05E <sub>H</sub>	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Edge	102F <sub>H</sub>	✓	✓	✓	✓	✓	+0BC <sub>H</sub>
48	Reserved						1030 <sub>H</sub>	—	—	—	—	—	+0C0 <sub>H</sub>
49	ICCSIG0IRE	FFFF A062 <sub>H</sub>	INTCSIG0IRE	CSIG0 communication error interrupt	CSIG0	Edge	1031 <sub>H</sub>	✓	✓	✓	✓	✓	+0C4 <sub>H</sub>
50	ICRLIN20	FFFF A064 <sub>H</sub>	INTRLIN20	RLIN20 interrupt	RLIN240	Edge	1032 <sub>H</sub>	✓	✓	✓	✓	✓	+0C8 <sub>H</sub>
51	ICRLIN21	FFFF A066 <sub>H</sub>	INTRLIN21	RLIN21 interrupt	RLIN240	Edge	1033 <sub>H</sub>	✓	✓	✓	✓	✓	+0CC <sub>H</sub>
52	ICDMA0	FFFF A068 <sub>H</sub>	INTDMA0	DMA0 transfer completion	DMAC	Edge	1034 <sub>H</sub>	✓	✓	✓	✓	✓	+0D0 <sub>H</sub>
53	ICDMA1	FFFF A06A <sub>H</sub>	INTDMA1	DMA1 transfer completion	DMAC	Edge	1035 <sub>H</sub>	✓	✓	✓	✓	✓	+0D4 <sub>H</sub>
54	ICDMA2	FFFF A06C <sub>H</sub>	INTDMA2	DMA2 transfer completion	DMAC	Edge	1036 <sub>H</sub>	✓	✓	✓	✓	✓	+0D8 <sub>H</sub>
55	ICDMA3	FFFF A06E <sub>H</sub>	INTDMA3	DMA3 transfer completion	DMAC	Edge	1037 <sub>H</sub>	✓	✓	✓	✓	✓	+0DC <sub>H</sub>
56	ICDMA4	FFFF A070 <sub>H</sub>	INTDMA4	DMA4 transfer completion	DMAC	Edge	1038 <sub>H</sub>	✓	✓	✓	✓	✓	+0E0 <sub>H</sub>
57	ICDMA5	FFFF A072 <sub>H</sub>	INTDMA5	DMA5 transfer completion	DMAC	Edge	1039 <sub>H</sub>	✓	✓	✓	✓	✓	+0E4 <sub>H</sub>
58	ICDMA6	FFFF A074 <sub>H</sub>	INTDMA6	DMA6 transfer completion	DMAC	Edge	103A <sub>H</sub>	✓	✓	✓	✓	✓	+0E8 <sub>H</sub>
59	ICDMA7	FFFF A076 <sub>H</sub>	INTDMA7	DMA7 transfer completion	DMAC	Edge	103B <sub>H</sub>	✓	✓	✓	✓	✓	+0EC <sub>H</sub>
60	ICDMA8	FFFF A078 <sub>H</sub>	INTDMA8	DMA8 transfer completion	DMAC	Edge	103C <sub>H</sub>	✓	✓	✓	✓	✓	+0F0 <sub>H</sub>
61	ICDMA9	FFFF A07A <sub>H</sub>	INTDMA9	DMA9 transfer completion	DMAC	Edge	103D <sub>H</sub>	✓	✓	✓	✓	✓	+0F4 <sub>H</sub>
62	ICDMA10	FFFF A07C <sub>H</sub>	INTDMA10	DMA10 transfer completion	DMAC	Edge	103E <sub>H</sub>	✓	✓	✓	✓	✓	+0F8 <sub>H</sub>
63	ICDMA11	FFFF A07E <sub>H</sub>	INTDMA11	DMA11 transfer completion	DMAC	Edge	103F <sub>H</sub>	✓	✓	✓	✓	✓	+0FC <sub>H</sub>
64	ICDMA12	FFFF A080 <sub>H</sub>	INTDMA12	DMA12 transfer completion	DMAC	Edge	1040 <sub>H</sub>	✓	✓	✓	✓	✓	+100 <sub>H</sub>
65	ICDMA13	FFFF A082 <sub>H</sub>	INTDMA13	DMA13 transfer completion	DMAC	Edge	1041 <sub>H</sub>	✓	✓	✓	✓	✓	+104 <sub>H</sub>
66	ICDMA14	FFFF A084 <sub>H</sub>	INTDMA14	DMA14 transfer completion	DMAC	Edge	1042 <sub>H</sub>	✓	✓	✓	✓	✓	+108 <sub>H</sub>
67	ICDMA15	FFFF A086 <sub>H</sub>	INTDMA15	DMA15 transfer completion	DMAC	Edge	1043 <sub>H</sub>	✓	✓	✓	✓	✓	+10C <sub>H</sub>
68	ICRIIC0TI	FFFF A088 <sub>H</sub>	INTRIIC0TI	RIIC transmit data empty interrupt	RIIC0	Edge	1044 <sub>H</sub>	✓	✓	✓	✓	✓	+110 <sub>H</sub>
69	ICRIIC0TEI	FFFF A08A <sub>H</sub>	INTRIIC0TEI	RIIC transmit complete interrupt	RIIC0	Level	1045 <sub>H</sub>	✓	✓	✓	✓	✓	+114 <sub>H</sub>
70	ICRIIC0RI	FFFF A08C <sub>H</sub>	INTRIIC0RI	RIIC receive complete interrupt	RIIC0	Edge	1046 <sub>H</sub>	✓	✓	✓	✓	✓	+118 <sub>H</sub>

Table 6.4 EI Level Maskable Interrupt Sources (3/8)

Interrupt No. <sup>1</sup>	Interrupt		Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset) <sup>8</sup>								
	Control Register		Name	Source	Unit	Detection Type <sup>2</sup>								Direct Jumping to an Address								
	Name	Address												RINT = 0	RINT = 1	Reference to a Table <sup>5</sup>						
71	ICRIIC0EE	FFFF A08E <sub>H</sub>	INTRIIC0EE	RIIC communication error/event interrupt	RIIC0	Level	1047 <sub>H</sub>	✓	✓	✓	✓	✓	✓	*3	*4	+11C <sub>H</sub>						
72	ICTAUJ0I0	FFFF A090 <sub>H</sub>	INTTAUJ0I0	Interrupt for TAUJ0 channel 0	TAUJ0	Edge	1048 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+120 <sub>H</sub>						
73	ICTAUJ0I1	FFFF A092 <sub>H</sub>	INTTAUJ0I1	Interrupt for TAUJ0 channel 1	TAUJ0	Edge	1049 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+124 <sub>H</sub>						
74	ICTAUJ0I2	FFFF A094 <sub>H</sub>	INTTAUJ0I2	Interrupt for TAUJ0 channel 2	TAUJ0	Edge	104A <sub>H</sub>	✓	✓	✓	✓	✓	✓			+128 <sub>H</sub>						
75	ICTAUJ0I3	FFFF A096 <sub>H</sub>	INTTAUJ0I3	Interrupt for TAUJ0 channel 3	TAUJ0	Edge	104B <sub>H</sub>	✓	✓	✓	✓	✓	✓			+12C <sub>H</sub>						
76	ICOSTMO	FFFF A098 <sub>H</sub>	INTOSTMO <sup>11</sup>	OSTMO interrupt	OSTMO	Edge	104C <sub>H</sub>	✓	✓	✓	✓	✓	✓			+130 <sub>H</sub>						
77	ICENCA0IOV	FFFF A09A <sub>H</sub>	INTENCA0IOV <sup>7</sup>	ENCA0 overflow interrupt	ENCA0	Edge	104D <sub>H</sub>	✓	✓	✓	✓	✓	✓			+134 <sub>H</sub>						
	ICPWGA4	FFFF A09A <sub>H</sub>	INTPWGA4 <sup>7</sup>	PWGA4 interrupt	PWGA4	Edge																
78	ICENCA0IUD	FFFF A09C <sub>H</sub>	INTENCA0IUD <sup>7</sup>	ENCA0 underflow interrupt	ENCA0	Edge	104E <sub>H</sub>	✓	✓	✓	✓	✓	✓			+138 <sub>H</sub>						
	ICPWGA5	FFFF A09C <sub>H</sub>	INTPWGA5 <sup>7</sup>	PWGA5 interrupt	PWGA5	Edge																
79	ICENCA0I0	FFFF A09E <sub>H</sub>	INTENCA0I0 <sup>7</sup>	ENCA0 capture/compare match interrupt 0	ENCA0	Edge	104F <sub>H</sub>	✓	✓	✓	✓	✓	✓			+13C <sub>H</sub>						
	ICPWGA6	FFFF A09E <sub>H</sub>	INTPWGA6 <sup>7</sup>	PWGA6 interrupt	PWGA6	Edge																
80	ICENCA0I1	FFFF A0A0 <sub>H</sub>	INTENCA0I1 <sup>7</sup>	ENCA0 capture/compare match interrupt 1	ENCA0	Edge	1050 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+140 <sub>H</sub>						
	ICPWGA7	FFFF A0A0 <sub>H</sub>	INTPWGA7 <sup>7</sup>	PWGA7 interrupt	PWGA7	Edge																
81	ICENCA0IEC	FFFF A0A2 <sub>H</sub>	INTENCA0IEC	ENCA0 encoder clear interrupt	ENCA0	Edge	1051 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+144 <sub>H</sub>						
82	ICKR0	FFFF A0A4 <sub>H</sub>	INTKR0	KR0 key interrupt	KR0	Edge	1052 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+148 <sub>H</sub>						
83	ICQFULL	FFFF A0A6 <sub>H</sub>	INTQFULL	PWSA queue full interrupt	PWSA	Edge	1053 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+14C <sub>H</sub>						
84	ICPWGA0	FFFF A0A8 <sub>H</sub>	INTPWGA0	PWGA0 interrupt	PWGA0	Edge	1054 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+150 <sub>H</sub>						
85	ICPWGA1	FFFF A0AA <sub>H</sub>	INTPWGA1	PWGA1 interrupt	PWGA1	Edge	1055 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+154 <sub>H</sub>						
86	ICPWGA2	FFFF A0AC <sub>H</sub>	INTPWGA2	PWGA2 interrupt	PWGA2	Edge	1056 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+158 <sub>H</sub>						
87	ICPWGA3	FFFF A0AE <sub>H</sub>	INTPWGA3	PWGA3 interrupt	PWGA3	Edge	1057 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+15C <sub>H</sub>						
88	ICPWGA8	FFFF A0B0 <sub>H</sub>	INTPWGA8	PWGA8 interrupt	PWGA8	Edge	1058 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+160 <sub>H</sub>						
89	ICPWGA9	FFFF A0B2 <sub>H</sub>	INTPWGA9	PWGA9 interrupt	PWGA9	Edge	1059 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+164 <sub>H</sub>						
90	ICPWGA10	FFFF A0B4 <sub>H</sub>	INTPWGA10	PWGA10 interrupt	PWGA10	Edge	105A <sub>H</sub>	✓	✓	✓	✓	✓	✓			+168 <sub>H</sub>						
91	ICPWGA11	FFFF A0B6 <sub>H</sub>	INTPWGA11	PWGA11 interrupt	PWGA11	Edge	105B <sub>H</sub>	✓	✓	✓	✓	✓	✓			+16C <sub>H</sub>						
92	ICPWGA12	FFFF A0B8 <sub>H</sub>	INTPWGA12	PWGA12 interrupt	PWGA12	Edge	105C <sub>H</sub>	✓	✓	✓	✓	✓	✓			+170 <sub>H</sub>						
93	ICPWGA13	FFFF A0BA <sub>H</sub>	INTPWGA13	PWGA13 interrupt	PWGA13	Edge	105D <sub>H</sub>	—	✓	✓	✓	✓	✓			+174 <sub>H</sub>						
94	ICPWGA14	FFFF A0BC <sub>H</sub>	INTPWGA14	PWGA14 interrupt	PWGA14	Edge	105E <sub>H</sub>	—	✓	✓	✓	✓	✓			+178 <sub>H</sub>						
95	ICPWGA15	FFFF A0BE <sub>H</sub>	INTPWGA15	PWGA15 interrupt	PWGA15	Edge	105F <sub>H</sub>	—	✓	✓	✓	✓	✓			+17C <sub>H</sub>						
96	Reserved						1060 <sub>H</sub>	—	—	—	—	—	—			+180 <sub>H</sub>						
97	Reserved						1061 <sub>H</sub>	—	—	—	—	—	—			+184 <sub>H</sub>						
98	Reserved						1062 <sub>H</sub>	—	—	—	—	—	—			+188 <sub>H</sub>						
99	Reserved						1063 <sub>H</sub>	—	—	—	—	—	—			+18C <sub>H</sub>						
100	Reserved						1064 <sub>H</sub>	—	—	—	—	—	—			+190 <sub>H</sub>						
101	Reserved						1065 <sub>H</sub>	—	—	—	—	—	—			+194 <sub>H</sub>						
102	ICFLERR	FFFF A0CC <sub>H</sub>	INTFLERR	Flash sequencer access error interrupt <sup>12</sup>	Flash	Level	1066 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+198 <sub>H</sub>						
103	ICFLENDNM	FFFF A0CE <sub>H</sub>	INTFLENDNM	Flash sequencer end interrupt <sup>12</sup>	Flash	Edge	1067 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+19C <sub>H</sub>						
104	ICCWEND	FFFF A0D0 <sub>H</sub>	INTCWEND	LPS0 port polling end interrupt	LPS	Edge	1068 <sub>H</sub>	✓	✓	✓	✓	✓	✓			+1A0 <sub>H</sub>						
105	ICRCAN1ERR	FFFF A0D2 <sub>H</sub>	INTRCAN1ERR	CAN1 error interrupt	RSCAN0	Level	1069 <sub>H</sub>	—	✓	✓	✓	✓	✓			+1A4 <sub>H</sub>						
106	ICRCAN1REC	FFFF A0D4 <sub>H</sub>	INTRCAN1REC	CAN1 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	106A <sub>H</sub>	—	✓	✓	✓	✓	✓			+1A8 <sub>H</sub>						
107	ICRCAN1TRX	FFFF A0D6 <sub>H</sub>	INTRCAN1TRX	CAN1 transmit interrupt	RSCAN0	Level	106B <sub>H</sub>	—	✓	✓	✓	✓	✓			+1AC <sub>H</sub>						
108	ICCSIH1IC	FFFF A0D8 <sub>H</sub>	INTCSIH1IC <sup>6</sup>	CSIH1 communication status interrupt	CSIH1	Edge	106C <sub>H</sub>	—	—	✓	✓	✓	✓			+1B0 <sub>H</sub>						
	ICTAPA0IPEK	FFFF A0D8 <sub>H</sub>	INTTAPA0IPEK <sub>0_2</sub> <sup>6</sup>	TAPA0 peak interrupt 0	TAPA0	Edge																

Table 6.4 EI Level Maskable Interrupt Sources (4/8)

Interrupt		Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset) <sup>8</sup>								
Interrupt No. <sup>1</sup>	Control Register		Name	Source	Unit								Direct Jumping to an Address								
	Name	Address											RINT = 0	RINT = 1	Reference to a Table <sup>5</sup>						
109	ICCSIH1IR	FFFF A0DA <sub>H</sub>	INTCSIH1IR <sup>6</sup>	CSIH1 receive status interrupt	CSIH1	Edge	106D <sub>H</sub>	—	—	✓	✓	✓	✓	✓	+3	+4	+1B4 <sub>H</sub>				
	ICTAPA0IVLY_0_2	FFFF A0DA <sub>H</sub>	INTTAPA0IVLY_0_2 <sup>6</sup>	TAPA0 valley interrupt 0	TAPA0	Edge												+1B8 <sub>H</sub>			
110	ICCSIH1IRE	FFFF A0DC <sub>H</sub>	INTCSIH1IRE <sup>6</sup>	CSIH1 communication error interrupt	CSIH1	Edge	106E <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+1BC <sub>H</sub>			
	ICCSIG0IC_2	FFFF A0DC <sub>H</sub>	INTCSIG0IC_2 <sup>6</sup>	CSIG0 communication status interrupt	CSIG0	Edge												+1C8 <sub>H</sub>			
111	ICCSIH1JJC	FFFF A0DE <sub>H</sub>	INTCSIH1JJC <sup>6</sup>	CSIH1 job completion interrupt	CSIH1	Edge	106F <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+1CC <sub>H</sub>			
	ICCSIG0IR_2	FFFF A0DE <sub>H</sub>	INTCSIG0IR_2 <sup>6</sup>	CSIG0 receive status interrupt	CSIG0	Edge												+1D0 <sub>H</sub>			
112	ICRLIN31	FFFF A0E0 <sub>H</sub>	INTRLIN31	RLIN31 interrupt	RLIN31	Edge	1070 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1C0 <sub>H</sub>			
113	ICRLIN31UR0	FFFF A0E2 <sub>H</sub>	INTRLIN31UR0	RLIN31 transmit interrupt	RLIN31	Edge	1071 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1C4 <sub>H</sub>			
114	ICRLIN31UR1	FFFF A0E4 <sub>H</sub>	INTRLIN31UR1	RLIN31 reception complete interrupt	RLIN31	Edge	1072 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1D4 <sub>H</sub>			
115	ICRLIN31UR2	FFFF A0E6 <sub>H</sub>	INTRLIN31UR2	RLIN31 status interrupt	RLIN31	Edge	1073 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1D8 <sub>H</sub>			
116	ICPWGA20	FFFF A0E8 <sub>H</sub>	INTPWGA20	PWGA20 interrupt	PWGA20	Edge	1074 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1DC <sub>H</sub>			
117	ICPWGA21	FFFF A0EA <sub>H</sub>	INTPWGA21	PWGA21 interrupt	PWGA21	Edge	1075 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1E0 <sub>H</sub>			
118	ICPWGA22	FFFF A0EC <sub>H</sub>	INTPWGA22	PWGA22 interrupt	PWGA22	Edge	1076 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1E4 <sub>H</sub>			
119	ICPWGA23	FFFF A0EE <sub>H</sub>	INTPWGA23	PWGA23 interrupt	PWGA23	Edge	1077 <sub>H</sub>	—	✓	✓	✓	✓	✓	✓				+1F0 <sub>H</sub>			
120	ICP6	FFFF A0F0 <sub>H</sub>	INTP6	External interrupt	Port	Edge	1078 <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+1F4 <sub>H</sub>			
121	ICP7	FFFF A0F2 <sub>H</sub>	INTP7	External interrupt	Port	Edge	1079 <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+1F8 <sub>H</sub>			
122	ICP8	FFFF A0F4 <sub>H</sub>	INTP8	External interrupt	Port	Edge	107A <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+1EC <sub>H</sub>			
123	ICP12	FFFF A0F6 <sub>H</sub>	INTP12	External interrupt	Port	Edge	107B <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+200 <sub>H</sub>			
124	ICCSIH2IC	FFFF A0F8 <sub>H</sub>	INTCSIH2IC <sup>6</sup>	CSIH2 communication status interrupt	CSIH2	Edge	107C <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+204 <sub>H</sub>			
	ICTAUD010_2	FFFF A0F8 <sub>H</sub>	INTTAUD010_2 <sup>6</sup>	Interrupt for TAUD0 channel 0	TAUD0	Edge												+208 <sub>H</sub>			
125	ICCSIH2IR	FFFF A0FA <sub>H</sub>	INTCSIH2IR <sup>6</sup>	CSIH2 receive status interrupt	CSIH2	Edge	107D <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+212 <sub>H</sub>			
	ICP0_2	FFFF A0FA <sub>H</sub>	INTP0_2 <sup>6</sup>	External interrupt	Port	Edge												+216 <sub>H</sub>			
126	ICCSIH2IRE	FFFF A0FC <sub>H</sub>	INTCSIH2IRE <sup>6</sup>	CSIH2 communication error interrupt	CSIH2	Edge	107E <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+220 <sub>H</sub>			
	ICP1_2	FFFF A0FC <sub>H</sub>	INTP1_2 <sup>6</sup>	External interrupt	Port	Edge												+224 <sub>H</sub>			
127	ICCSIH2JJC	FFFF A0FE <sub>H</sub>	INTCSIH2JJC <sup>6</sup>	CSIH2 job completion interrupt	CSIH2	Edge	107F <sub>H</sub>	—	—	✓	✓	✓	✓	✓				+228 <sub>H</sub>			
	ICP2_2	FFFF A0FE <sub>H</sub>	INTP2_2 <sup>6</sup>	External interrupt	Port	Edge												+232 <sub>H</sub>			
128	Reserved						1080 <sub>H</sub>	—	—	—	—	—	—	—				+236 <sub>H</sub>			
129	Reserved						1081 <sub>H</sub>	—	—	—	—	—	—	—				+240 <sub>H</sub>			
130	Reserved						1082 <sub>H</sub>	—	—	—	—	—	—	—				+244 <sub>H</sub>			
131	Reserved						1083 <sub>H</sub>	—	—	—	—	—	—	—				+248 <sub>H</sub>			
132	Reserved						1084 <sub>H</sub>	—	—	—	—	—	—	—				+252 <sub>H</sub>			
133	Reserved						1085 <sub>H</sub>	—	—	—	—	—	—	—				+256 <sub>H</sub>			
134	ICTAUB010	FFFF A10C <sub>H</sub>	INTTAUB010	Interrupt for TAUB0 channel 0	TAUB0	Edge	1086 <sub>H</sub>	—	—	—	✓	✓	✓	✓				+260 <sub>H</sub>			
135	ICTAUB011	FFFF A10E <sub>H</sub>	INTTAUB011	Interrupt for TAUB0 channel 1	TAUB0	Edge	1087 <sub>H</sub>	—	—	—	✓	✓	✓	✓				+264 <sub>H</sub>			
136	ICTAUB012	FFFF A110 <sub>H</sub>	INTTAUB012	Interrupt for TAUB0 channel 2	TAUB0	Edge	1088 <sub>H</sub>	—	—	—	✓	✓	✓	✓				+268 <sub>H</sub>			
137	ICTAUB013	FFFF A112 <sub>H</sub>	INTTAUB013 <sup>7</sup>	Interrupt for TAUB0 channel 3	TAUB0	Edge	1089 <sub>H</sub>	—	—	—	✓	✓	✓	✓				+272 <sub>H</sub>			
	ICPWGA16	FFFF A112 <sub>H</sub>	INTPWGA16 <sup>7</sup>	PWGA16 interrupt	PWGA16	Edge		—	✓	✓								+276 <sub>H</sub>			
138	ICTAUB014	FFFF A114 <sub>H</sub>	INTTAUB014	Interrupt for TAUB0 channel 4	TAUB0	Edge	108A <sub>H</sub>	—	—	—	✓	✓	✓	✓				+280 <sub>H</sub>			
139	ICTAUB015	FFFF A116 <sub>H</sub>	INTTAUB015 <sup>7</sup>	Interrupt for TAUB0 channel 5	TAUB0	Edge	108B <sub>H</sub>	—	—	—	✓	✓	✓	✓				+284 <sub>H</sub>			
	ICPWGA17	FFFF A116 <sub>H</sub>	INTPWGA17 <sup>7</sup>	PWGA17 interrupt	PWGA17	Edge		—	✓	✓								+288 <sub>H</sub>			
140	ICTAUB016	FFFF A118 <sub>H</sub>	INTTAUB016	Interrupt for TAUB0 channel 6	TAUB0	Edge	108C <sub>H</sub>	—	—	—	✓	✓	✓	✓				+292 <sub>H</sub>			

Table 6.4 EI Level Maskable Interrupt Sources (5/8)

Interrupt No. <sup>1</sup>	Interrupt		Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset) <sup>8</sup>										
	Control Register		Name	Source	Unit	Detection Type <sup>2</sup>								Direct Jumping to an Address	RINT = 0	RINT = 1	Reference to a Table <sup>5</sup>							
	Name	Address																						
141	ICTAUB017	FFFF A11A <sub>H</sub>	INTTAUB017 <sup>7</sup>	Interrupt for TAUB0 channel 7	TAUB0	Edge	108D <sub>H</sub>	— — —	—	✓	✓	✓	✓	*3	*4	+234 <sub>H</sub>								
	ICPWGA18	FFFF A11A <sub>H</sub>	INTPWGA18 <sup>7</sup>	PWGA18 interrupt	PWGA18	Edge		— ✓	✓															
142	ICTAUB018	FFFF A11C <sub>H</sub>	INTTAUB018	Interrupt for TAUB0 channel 8	TAUB0	Edge	108E <sub>H</sub>	— — —	—	✓	✓	✓	✓			+238 <sub>H</sub>								
143	ICTAUB019	FFFF A11E <sub>H</sub>	INTTAUB019 <sup>7</sup>	Interrupt for TAUB0 channel 9	TAUB0	Edge	108F <sub>H</sub>	— — —	—	✓	✓	✓	✓			+23C <sub>H</sub>								
	ICPWGA19	FFFF A11E <sub>H</sub>	INTPWGA19 <sup>7</sup>	PWGA19 interrupt	PWGA19	Edge		— ✓	✓															
144	ICTAUB0110	FFFF A120 <sub>H</sub>	INTTAUB0110	Interrupt for TAUB0 channel 10	TAUB0	Edge	1090 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+240 <sub>H</sub>								
145	ICTAUB0111	FFFF A122 <sub>H</sub>	INTTAUB0111 <sup>7</sup>	Interrupt for TAUB0 channel 11	TAUB0	Edge	1091 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+244 <sub>H</sub>								
	ICPWGA26	FFFF A122 <sub>H</sub>	INTPWGA26 <sup>7</sup>	PWGA26 interrupt	PWGA26	Edge																		
146	ICTAUB0112	FFFF A124 <sub>H</sub>	INTTAUB0112	Interrupt for TAUB0 channel 12	TAUB0	Edge	1092 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+248 <sub>H</sub>								
147	ICTAUB0113	FFFF A126 <sub>H</sub>	INTTAUB0113 <sup>7</sup>	Interrupt for TAUB0 channel 13	TAUB0	Edge	1093 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+24C <sub>H</sub>								
	ICPWGA30	FFFF A126 <sub>H</sub>	INTPWGA30 <sup>7</sup>	PWGA30 interrupt	PWGA30	Edge																		
148	ICTAUB0114	FFFF A128 <sub>H</sub>	INTTAUB0114	Interrupt for TAUB0 channel 14	TAUB0	Edge	1094 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+250 <sub>H</sub>								
149	ICTAUB0115	FFFF A12A <sub>H</sub>	INTTAUB0115 <sup>7</sup>	Interrupt for TAUB0 channel 15	TAUB0	Edge	1095 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+254 <sub>H</sub>								
	ICPWGA31	FFFF A12A <sub>H</sub>	INTPWGA31 <sup>7</sup>	PWGA31 interrupt	PWGA31	Edge																		
150	ICCSIH3IC	FFFF A12C <sub>H</sub>	INTCSIH3IC <sup>6</sup>	CSIH3 communication status interrupt	CSIH3	Edge	1096 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+258 <sub>H</sub>								
	ICTAUD012_2	FFFF A12C <sub>H</sub>	INTTAUD012_2 <sup>6</sup>	Interrupt for TAUD0 channel 2	TAUD0	Edge																		
151	ICCSIH3IR	FFFF A12E <sub>H</sub>	INTCSIH3IR <sup>6</sup>	CSIH3 receive status interrupt	CSIH3	Edge	1097 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+25C <sub>H</sub>								
	ICTAUD0110_2	FFFF A12E <sub>H</sub>	INTTAUD0110_2 <sup>6</sup>	Interrupt for TAUD0 channel 10	TAUD0	Edge																		
152	ICCSIH3IRE	FFFF A130 <sub>H</sub>	INTCSIH3IRE <sup>6</sup>	CSIH3 communication error interrupt	CSIH3	Edge	1098 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+260 <sub>H</sub>								
	ICTAUD0112_2	FFFF A130 <sub>H</sub>	INTTAUD0112_2 <sup>6</sup>	Interrupt for TAUD0 channel 12	TAUD0	Edge																		
153	ICCSIH3JC	FFFF A132 <sub>H</sub>	INTCSIH3JC <sup>6</sup>	CSIH3 job completion interrupt	CSIH3	Edge	1099 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+264 <sub>H</sub>								
	ICTAUD0114_2	FFFF A132 <sub>H</sub>	INTTAUD0114_2 <sup>6</sup>	Interrupt for TAUD0 channel 14	TAUD0	Edge																		
154	ICRLIN22	FFFF A134 <sub>H</sub>	INTRLIN22	RLIN22 interrupt	RLIN240	Edge	109A <sub>H</sub>	— — —	—	✓	✓	✓	✓			+268 <sub>H</sub>								
155	ICRLIN23	FFFF A136 <sub>H</sub>	INTRLIN23	RLIN23 interrupt	RLIN240	Edge	109B <sub>H</sub>	— — —	—	—	✓	✓	✓			+26C <sub>H</sub>								
156	ICRLIN32	FFFF A138 <sub>H</sub>	INTRLIN32	RLIN32 interrupt	RLIN32	Edge	109C <sub>H</sub>	— — —	✓	✓	✓	✓	✓			+270 <sub>H</sub>								
157	ICRLIN32UR0	FFFF A13A <sub>H</sub>	INTRLIN32UR0	RLIN32 transmit interrupt	RLIN32	Edge	109D <sub>H</sub>	— — —	✓	✓	✓	✓	✓			+274 <sub>H</sub>								
158	ICRLIN32UR1	FFFF A13C <sub>H</sub>	INTRLIN32UR1	RLIN32 reception complete interrupt	RLIN32	Edge	109E <sub>H</sub>	— — —	✓	✓	✓	✓	✓			+278 <sub>H</sub>								
159	ICRLIN32UR2	FFFF A13E <sub>H</sub>	INTRLIN32UR2	RLIN32 status interrupt	RLIN32	Edge	109F <sub>H</sub>	— — —	✓	✓	✓	✓	✓			+27C <sub>H</sub>								
160	ICTAUJ110	FFFF A140 <sub>H</sub>	INTTAUJ110	Interrupt for TAUJ1 channel 0	TAUJ1	Edge	10A0 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+280 <sub>H</sub>								
161	ICTAUJ111	FFFF A142 <sub>H</sub>	INTTAUJ111	Interrupt for TAUJ1 channel 1	TAUJ1	Edge	10A1 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+284 <sub>H</sub>								
162	ICTAUJ112	FFFF A144 <sub>H</sub>	INTTAUJ112	Interrupt for TAUJ1 channel 2	TAUJ1	Edge	10A2 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+288 <sub>H</sub>								
163	ICTAUJ113	FFFF A146 <sub>H</sub>	INTTAUJ113	Interrupt for TAUJ1 channel 3	TAUJ1	Edge	10A3 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+28C <sub>H</sub>								
164	Reserved						10A4 <sub>H</sub>	— — —	—	—	—	—	—			+290 <sub>H</sub>								
165	Reserved						10A5 <sub>H</sub>	— — —	—	—	—	—	—			+294 <sub>H</sub>								
166	Reserved						10A6 <sub>H</sub>	— — —	—	—	—	—	—			+298 <sub>H</sub>								
167	Reserved						10A7 <sub>H</sub>	— — —	—	—	—	—	—			+29C <sub>H</sub>								
168	Reserved						10A8 <sub>H</sub>	— — —	—	—	—	—	—			+2A0 <sub>H</sub>								
169	Reserved						10A9 <sub>H</sub>	— — —	—	—	—	—	—			+2A4 <sub>H</sub>								
170	Reserved						10AA <sub>H</sub>	— — —	—	—	—	—	—			+2A8 <sub>H</sub>								
171	Reserved						10AB <sub>H</sub>	— — —	—	—	—	—	—			+2AC <sub>H</sub>								
172	Reserved						10AC <sub>H</sub>	— — —	—	—	—	—	—			+2B0 <sub>H</sub>								
173	Reserved						10AD <sub>H</sub>	— — —	—	—	—	—	—			+2B4 <sub>H</sub>								
174	Reserved						10AE <sub>H</sub>	— — —	—	—	—	—	—			+2B8 <sub>H</sub>								
175	Reserved						10AF <sub>H</sub>	— — —	—	—	—	—	—			+2BC <sub>H</sub>								
176	ICPWGA24	FFFF A160 <sub>H</sub>	INTPWGA24	PWGA24 interrupt	PWGA24	Edge	10B0 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+2C0 <sub>H</sub>								
177	ICPWGA25	FFFF A162 <sub>H</sub>	INTPWGA25	PWGA25 interrupt	PWGA25	Edge	10B1 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+2C4 <sub>H</sub>								
178	ICPWGA27	FFFF A164 <sub>H</sub>	INTPWGA27	PWGA27 interrupt	PWGA27	Edge	10B2 <sub>H</sub>	— — —	—	✓	✓	✓	✓			+2C8 <sub>H</sub>								

Table 6.4 EI Level Maskable Interrupt Sources (6/8)

Interrupt No. <sup>1</sup>	Interrupt		Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset) <sup>8</sup>								
	Control Register		Name	Source	Unit	Detection Type <sup>2</sup>								Direct Jumping to an Address								
	Name	Address												RINT = 0	RINT = 1	Reference to a Table <sup>5</sup>						
179	ICPWGA28	FFFF A166 <sub>H</sub>	INTPWGA28	PWGA28 interrupt	PWGA28	Edge	10B3 <sub>H</sub>	—	—	—	✓	✓	✓	*3	*4	+2CC <sub>H</sub>						
180	ICPWGA29	FFFF A168 <sub>H</sub>	INTPWGA29	PWGA29 interrupt	PWGA29	Edge	10B4 <sub>H</sub>	—	—	—	✓	✓	✓			+2D0 <sub>H</sub>						
181	ICPWGA32	FFFF A16A <sub>H</sub>	INTPWGA32	PWGA32 interrupt	PWGA32	Edge	10B5 <sub>H</sub>	—	—	—	✓	✓	✓			+2D4 <sub>H</sub>						
182	ICPWGA33	FFFF A16C <sub>H</sub>	INTPWGA33	PWGA33 interrupt	PWGA33	Edge	10B6 <sub>H</sub>	—	—	—	✓	✓	✓			+2D8 <sub>H</sub>						
183	ICPWGA34	FFFF A16E <sub>H</sub>	INTPWGA34	PWGA34 interrupt	PWGA34	Edge	10B7 <sub>H</sub>	—	—	—	✓	✓	✓			+2DC <sub>H</sub>						
184	ICPWGA35	FFFF A170 <sub>H</sub>	INTPWGA35	PWGA35 interrupt	PWGA35	Edge	10B8 <sub>H</sub>	—	—	—	✓	✓	✓			+2E0 <sub>H</sub>						
185	ICPWGA36	FFFF A172 <sub>H</sub>	INTPWGA36	PWGA36 interrupt	PWGA36	Edge	10B9 <sub>H</sub>	—	—	—	✓	✓	✓			+2E4 <sub>H</sub>						
186	ICPWGA37	FFFF A174 <sub>H</sub>	INTPWGA37	PWGA37 interrupt	PWGA37	Edge	10BA <sub>H</sub>	—	—	—	✓	✓	✓			+2E8 <sub>H</sub>						
187	ICPWGA38	FFFF A176 <sub>H</sub>	INTPWGA38	PWGA38 interrupt	PWGA38	Edge	10BB <sub>H</sub>	—	—	—	✓	✓	✓			+2EC <sub>H</sub>						
188	ICPWGA39	FFFF A178 <sub>H</sub>	INTPWGA39	PWGA39 interrupt	PWGA39	Edge	10BC <sub>H</sub>	—	—	—	✓	✓	✓			+2F0 <sub>H</sub>						
189	ICPWGA40	FFFF A17A <sub>H</sub>	INTPWGA40	PWGA40 interrupt	PWGA40	Edge	10BD <sub>H</sub>	—	—	—	✓	✓	✓			+2F4 <sub>H</sub>						
190	ICPWGA41	FFFF A17C <sub>H</sub>	INTPWGA41	PWGA41 interrupt	PWGA41	Edge	10BE <sub>H</sub>	—	—	—	✓	✓	✓			+2F8 <sub>H</sub>						
191	ICPWGA42	FFFF A17E <sub>H</sub>	INTPWGA42	PWGA42 interrupt	PWGA42	Edge	10BF <sub>H</sub>	—	—	—	✓	✓	✓			+2FC <sub>H</sub>						
192	ICPWGA43	FFFF A180 <sub>H</sub>	INTPWGA43	PWGA43 interrupt	PWGA43	Edge	10C0 <sub>H</sub>	—	—	—	✓	✓	✓			+300 <sub>H</sub>						
193	ICPWGA44	FFFF A182 <sub>H</sub>	INTPWGA44	PWGA44 interrupt	PWGA44	Edge	10C1 <sub>H</sub>	—	—	—	✓	✓	✓			+304 <sub>H</sub>						
194	ICPWGA45	FFFF A184 <sub>H</sub>	INTPWGA45	PWGA45 interrupt	PWGA45	Edge	10C2 <sub>H</sub>	—	—	—	✓	✓	✓			+308 <sub>H</sub>						
195	ICPWGA46	FFFF A186 <sub>H</sub>	INTPWGA46	PWGA46 interrupt	PWGA46	Edge	10C3 <sub>H</sub>	—	—	—	✓	✓	✓			+30C <sub>H</sub>						
196	ICPWGA47	FFFF A188 <sub>H</sub>	INTPWGA47	PWGA47 interrupt	PWGA47	Edge	10C4 <sub>H</sub>	—	—	—	✓	✓	✓			+310 <sub>H</sub>						
197	ICP9	FFFF A18A <sub>H</sub>	INTP9	External interrupt	Port	Edge	10C5 <sub>H</sub>	—	—	—	—	✓	✓			+314 <sub>H</sub>						
198	ICP13	FFFF A18C <sub>H</sub>	INTP13	External interrupt	Port	Edge	10C6 <sub>H</sub>	—	—	—	✓	✓	✓			+318 <sub>H</sub>						
199	ICP14	FFFF A18E <sub>H</sub>	INTP14	External interrupt	Port	Edge	10C7 <sub>H</sub>	—	—	—	—	✓	✓			+31C <sub>H</sub>						
200	ICP15	FFFF A190 <sub>H</sub>	INTP15	External interrupt	Port	Edge	10C8 <sub>H</sub>	—	—	—	—	✓	✓			+320 <sub>H</sub>						
201	ICRTCA01S	FFFF A192 <sub>H</sub>	INTRTCA01S	RTCA0 1 second interval interrupt	RTCA	Edge	10C9 <sub>H</sub>	—	—	—	—	✓	✓			+324 <sub>H</sub>						
202	ICRTCA0AL	FFFF A194 <sub>H</sub>	INTRTCA0AL	RTCA0 alarm interrupt	RTCA	Edge	10CA <sub>H</sub>	—	—	—	—	✓	✓			+328 <sub>H</sub>						
203	ICRTCA0R	FFFF A196 <sub>H</sub>	INTRTCA0R	RTCA0 periodic interrupt	RTCA	Edge	10CB <sub>H</sub>	—	—	—	—	✓	✓			+32C <sub>H</sub>						
204	ICADCA1ERR	FFFF A198 <sub>H</sub>	INTADCA1ERR	ADCA1 error interrupt	ADCA1	Edge	10CC <sub>H</sub>	—	—	—	—	✓	✓			+330 <sub>H</sub>						
205	ICADCA1I0	FFFF A19A <sub>H</sub>	INTADCA1I0	ADCA1 scan group 1 (SG1) end interrupt	ADCA1	Edge	10CD <sub>H</sub>	—	—	—	—	✓	✓			+334 <sub>H</sub>						
206	ICADCA1I1	FFFF A19C <sub>H</sub>	INTADCA1I1	ADCA1 scan group 2 (SG2) end interrupt	ADCA1	Edge	10CE <sub>H</sub>	—	—	—	—	✓	✓			+338 <sub>H</sub>						
207	ICADCA1I2	FFFF A19E <sub>H</sub>	INTADCA1I2	ADCA1 scan group 3 (SG3) end interrupt	ADCA1	Edge	10CF <sub>H</sub>	—	—	—	—	✓	✓			+33C <sub>H</sub>						
208	Reserved						10D0 <sub>H</sub>	—	—	—	—	—	—			+340 <sub>H</sub>						
209	ICRCAN2ERR	FFFF A1A2 <sub>H</sub>	INTRCAN2ERR	CAN2 error interrupt	RSCAN0	Level	10D1 <sub>H</sub>	—	✓	✓	✓	✓	✓			+344 <sub>H</sub>						
210	ICRCAN2REC	FFFF A1A4 <sub>H</sub>	INTRCAN2REC	CAN2 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	10D2 <sub>H</sub>	—	✓	✓	✓	✓	✓			+348 <sub>H</sub>						
211	ICRCAN2TRX	FFFF A1A6 <sub>H</sub>	INTRCAN2TRX	CAN2 transmit interrupt	RSCAN0	Level	10D3 <sub>H</sub>	—	✓	✓	✓	✓	✓			+34C <sub>H</sub>						
212	ICRCAN3ERR	FFFF A1A8 <sub>H</sub>	INTRCAN3ERR	CAN3 error interrupt	RSCAN0	Level	10D4 <sub>H</sub>	—	—	—	*9	✓	✓			+350 <sub>H</sub>						
213	ICRCAN3REC	FFFF A1AA <sub>H</sub>	INTRCAN3REC	CAN3 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	10D5 <sub>H</sub>	—	—	—	*9	✓	✓			+354 <sub>H</sub>						
214	ICRCAN3TRX	FFFF A1AC <sub>H</sub>	INTRCAN3TRX	CAN3 transmit interrupt	RSCAN0	Level	10D6 <sub>H</sub>	—	—	—	*9	✓	✓			+358 <sub>H</sub>						
215	ICCSIG1IC	FFFF A1AE <sub>H</sub>	INTCSIG1IC	CSIG1 communication status interrupt	CSIG1	Edge	10D7 <sub>H</sub>	—	—	—	—	✓	✓			+35C <sub>H</sub>						
216	ICCSIG1IR	FFFF A1B0 <sub>H</sub>	INTCSIG1IR	CSIG1 receive status interrupt	CSIG1	Edge	10D8 <sub>H</sub>	—	—	—	—	✓	✓			+360 <sub>H</sub>						
217	ICCSIG1IRE	FFFF A1B2 <sub>H</sub>	INTCSIG1IRE	CSIG1 communication error interrupt	CSIG1	Edge	10D9 <sub>H</sub>	—	—	—	—	✓	✓			+364 <sub>H</sub>						
218	ICRLIN24	FFFF A1B4 <sub>H</sub>	INTRLIN24	RLIN24 interrupt	RLIN241	Edge	10DA <sub>H</sub>	—	—	—	—	✓	✓			+368 <sub>H</sub>						
219	ICRLIN25	FFFF A1B6 <sub>H</sub>	INTRLIN25	RLIN25 interrupt	RLIN241	Edge	10DB <sub>H</sub>	—	—	—	—	✓	✓			+36C <sub>H</sub>						
220	ICRLIN33	FFFF A1B8 <sub>H</sub>	INTRLIN33	RLIN33 interrupt	RLIN33	Edge	10DC <sub>H</sub>	—	—	—	✓	✓	✓			+370 <sub>H</sub>						

Table 6.4 EI Level Maskable Interrupt Sources (7/8)

Interrupt			Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset) <sup>*8</sup>								
Interrupt No. <sup>*1</sup>	Control Register		Name	Source	Unit	Detection Type <sup>*2</sup>								Direct Jumping to an Address								
	Name	Address												RINT = 0	RINT = 1	Reference to a Table <sup>*5</sup>						
221	ICRLIN33UR0	FFFF A1BA <sub>H</sub>	INTRLIN33UR0	RLIN33 transmit interrupt	RLIN33	Edge	10DD <sub>H</sub>	—	—	✓	✓	✓	✓	+374 <sub>H</sub>	+3	+4						
222	ICRLIN33UR1	FFFF A1BC <sub>H</sub>	INTRLIN33UR1	RLIN33 reception complete interrupt	RLIN33	Edge	10DE <sub>H</sub>	—	—	✓	✓	✓	✓	+378 <sub>H</sub>								
223	ICRLIN33UR2	FFFF A1BE <sub>H</sub>	INTRLIN33UR2	RLIN33 status interrupt	RLIN33	Edge	10DF <sub>H</sub>	—	—	✓	✓	✓	✓	+37C <sub>H</sub>								
224	ICRLIN34	FFFF A1C0 <sub>H</sub>	INTRLIN34	RLIN34 interrupt	RLIN34	Edge	10E0 <sub>H</sub>	—	—	—	—	✓	✓	+380 <sub>H</sub>								
225	ICRLIN34UR0	FFFF A1C2 <sub>H</sub>	INTRLIN34UR0	RLIN34 transmit interrupt	RLIN34	Edge	10E1 <sub>H</sub>	—	—	—	—	✓	✓	+384 <sub>H</sub>								
226	ICRLIN34UR1	FFFF A1C4 <sub>H</sub>	INTRLIN34UR1	RLIN34 reception complete interrupt	RLIN34	Edge	10E2 <sub>H</sub>	—	—	—	—	✓	✓	+388 <sub>H</sub>								
227	ICRLIN34UR2	FFFF A1C6 <sub>H</sub>	INTRLIN34UR2	RLIN34 status interrupt	RLIN34	Edge	10E3 <sub>H</sub>	—	—	—	—	✓	✓	+38C <sub>H</sub>								
228	ICRLIN35	FFFF A1C8 <sub>H</sub>	INTRLIN35	RLIN35 interrupt	RLIN35	Edge	10E4 <sub>H</sub>	—	—	—	—	✓	✓	+390 <sub>H</sub>								
229	ICRLIN35UR0	FFFF A1CA <sub>H</sub>	INTRLIN35UR0	RLIN35 transmit interrupt	RLIN35	Edge	10E5 <sub>H</sub>	—	—	—	—	✓	✓	+394 <sub>H</sub>								
230	ICRLIN35UR1	FFFF A1CC <sub>H</sub>	INTRLIN35UR1	RLIN35 reception complete interrupt	RLIN35	Edge	10E6 <sub>H</sub>	—	—	—	—	✓	✓	+398 <sub>H</sub>								
231	ICRLIN35UR2	FFFF A1CE <sub>H</sub>	INTRLIN35UR2	RLIN35 status interrupt	RLIN35	Edge	10E7 <sub>H</sub>	—	—	—	—	✓	✓	+39C <sub>H</sub>								
232	ICPWGA48	FFFF A1D0 <sub>H</sub>	INTPWGA48	PWGA48 interrupt	PWGA48	Edge	10E8 <sub>H</sub>	—	—	—	—	✓	✓	+3A0 <sub>H</sub>								
233	ICPWGA49	FFFF A1D2 <sub>H</sub>	INTPWGA49	PWGA49 interrupt	PWGA49	Edge	10E9 <sub>H</sub>	—	—	—	—	✓	✓	+3A4 <sub>H</sub>								
234	ICPWGA50	FFFF A1D4 <sub>H</sub>	INTPWGA50	PWGA50 interrupt	PWGA50	Edge	10EA <sub>H</sub>	—	—	—	—	✓	✓	+3A8 <sub>H</sub>								
235	ICPWGA51	FFFF A1D6 <sub>H</sub>	INTPWGA51	PWGA51 interrupt	PWGA51	Edge	10EB <sub>H</sub>	—	—	—	—	✓	✓	+3AC <sub>H</sub>								
236	ICPWGA52	FFFF A1D8 <sub>H</sub>	INTPWGA52	PWGA52 interrupt	PWGA52	Edge	10EC <sub>H</sub>	—	—	—	—	✓	✓	+3B0 <sub>H</sub>								
237	ICPWGA53	FFFF A1DA <sub>H</sub>	INTPWGA53	PWGA53 interrupt	PWGA53	Edge	10ED <sub>H</sub>	—	—	—	—	✓	✓	+3B4 <sub>H</sub>								
238	ICPWGA54	FFFF A1DC <sub>H</sub>	INTPWGA54	PWGA54 interrupt	PWGA54	Edge	10EE <sub>H</sub>	—	—	—	—	✓	✓	+3B8 <sub>H</sub>								
239	ICPWGA55	FFFF A1DE <sub>H</sub>	INTPWGA55	PWGA55 interrupt	PWGA55	Edge	10EF <sub>H</sub>	—	—	—	—	✓	✓	+3BC <sub>H</sub>								
240	ICPWGA56	FFFF A1E0 <sub>H</sub>	INTPWGA56	PWGA56 interrupt	PWGA56	Edge	10FO <sub>H</sub>	—	—	—	—	✓	✓	+3C0 <sub>H</sub>								
241	ICPWGA57	FFFF A1E2 <sub>H</sub>	INTPWGA57	PWGA57 interrupt	PWGA57	Edge	10F1 <sub>H</sub>	—	—	—	—	✓	✓	+3C4 <sub>H</sub>								
242	ICPWGA58	FFFF A1E4 <sub>H</sub>	INTPWGA58	PWGA58 interrupt	PWGA58	Edge	10F2 <sub>H</sub>	—	—	—	—	✓	✓	+3C8 <sub>H</sub>								
243	ICPWGA59	FFFF A1E6 <sub>H</sub>	INTPWGA59	PWGA59 interrupt	PWGA59	Edge	10F3 <sub>H</sub>	—	—	—	—	✓	✓	+3CC <sub>H</sub>								
244	ICPWGA60	FFFF A1E8 <sub>H</sub>	INTPWGA60	PWGA60 interrupt	PWGA60	Edge	10F4 <sub>H</sub>	—	—	—	—	✓	✓	+3D0 <sub>H</sub>								
245	ICPWGA61	FFFF A1EA <sub>H</sub>	INTPWGA61	PWGA61 interrupt	PWGA61	Edge	10F5 <sub>H</sub>	—	—	—	—	✓	✓	+3D4 <sub>H</sub>								
246	ICPWGA62	FFFF A1EC <sub>H</sub>	INTPWGA62	PWGA62 interrupt	PWGA62	Edge	10F6 <sub>H</sub>	—	—	—	—	✓	✓	+3D8 <sub>H</sub>								
247	ICPWGA63	FFFF A1EE <sub>H</sub>	INTPWGA63	PWGA63 interrupt	PWGA63	Edge	10F7 <sub>H</sub>	—	—	—	—	✓	✓	+3DC <sub>H</sub>								
248	ICTAUB1I0	FFFF A1F0 <sub>H</sub>	INTTAUB1I0	Interrupt for TAUB1 channel 0	TAUB1	Edge	10F8 <sub>H</sub>	—	—	—	—	—	✓	+3E0 <sub>H</sub>								
249	ICTAUB1I1	FFFF A1F2 <sub>H</sub>	INTTAUB1I1	Interrupt for TAUB1 channel 1	TAUB1	Edge	10F9 <sub>H</sub>	—	—	—	—	—	✓	+3E4 <sub>H</sub>								
250	ICTAUB1I2	FFFF A1F4 <sub>H</sub>	INTTAUB1I2	Interrupt for TAUB1 channel 2	TAUB1	Edge	10FA <sub>H</sub>	—	—	—	—	—	✓	+3E8 <sub>H</sub>								
251	ICTAUB1I3	FFFF A1F6 <sub>H</sub>	INTTAUB1I3	Interrupt for TAUB1 channel 3	TAUB1	Edge	10FB <sub>H</sub>	—	—	—	—	—	✓	+3EC <sub>H</sub>								
252	ICTAUB1I4	FFFF A1F8 <sub>H</sub>	INTTAUB1I4	Interrupt for TAUB1 channel 4	TAUB1	Edge	10FC <sub>H</sub>	—	—	—	—	—	✓	+3F0 <sub>H</sub>								
253	ICTAUB1I5	FFFF A1FA <sub>H</sub>	INTTAUB1I5	Interrupt for TAUB1 channel 5	TAUB1	Edge	10FD <sub>H</sub>	—	—	—	—	—	✓	+3F4 <sub>H</sub>								
254	ICTAUB1I6	FFFF A1FC <sub>H</sub>	INTTAUB1I6	Interrupt for TAUB1 channel 6	TAUB1	Edge	10FE <sub>H</sub>	—	—	—	—	—	✓	+3F8 <sub>H</sub>								
255	ICTAUB1I7	FFFF A1FE <sub>H</sub>	INTTAUB1I7	Interrupt for TAUB1 channel 7	TAUB1	Edge	10FF <sub>H</sub>	—	—	—	—	—	✓	+3FC <sub>H</sub>								
256	ICTAUB1I8	FFFF A200 <sub>H</sub>	INTTAUB1I8	Interrupt for TAUB1 channel 8	TAUB1	Edge	1100 <sub>H</sub>	—	—	—	—	—	✓	+400 <sub>H</sub>								
257	ICTAUB1I9	FFFF A202 <sub>H</sub>	INTTAUB1I9	Interrupt for TAUB1 channel 9	TAUB1	Edge	1101 <sub>H</sub>	—	—	—	—	—	✓	+404 <sub>H</sub>								
258	ICTAUB1I10	FFFF A204 <sub>H</sub>	INTTAUB1I10	Interrupt for TAUB1 channel 10	TAUB1	Edge	1102 <sub>H</sub>	—	—	—	—	—	✓	+408 <sub>H</sub>								
259	ICTAUB1I11	FFFF A206 <sub>H</sub>	INTTAUB1I11	Interrupt for TAUB1 channel 11	TAUB1	Edge	1103 <sub>H</sub>	—	—	—	—	—	✓	+40C <sub>H</sub>								
260	ICTAUB1I12	FFFF A208 <sub>H</sub>	INTTAUB1I12	Interrupt for TAUB1 channel 12	TAUB1	Edge	1104 <sub>H</sub>	—	—	—	—	—	✓	+410 <sub>H</sub>								
261	ICTAUB1I13	FFFF A20A <sub>H</sub>	INTTAUB1I13	Interrupt for TAUB1 channel 13	TAUB1	Edge	1105 <sub>H</sub>	—	—	—	—	—	✓	+414 <sub>H</sub>								
262	ICTAUB1I14	FFFF A20C <sub>H</sub>	INTTAUB1I14	Interrupt for TAUB1 channel 14	TAUB1	Edge	1106 <sub>H</sub>	—	—	—	—	—	✓	+418 <sub>H</sub>								
263	ICTAUB1I15	FFFF A20E <sub>H</sub>	INTTAUB1I15	Interrupt for TAUB1 channel 15	TAUB1	Edge	1107 <sub>H</sub>	—	—	—	—	—	✓	+41C <sub>H</sub>								
264	ICRCAN4ERR	FFFF A210 <sub>H</sub>	INTRCAN4ERR	CAN4 error interrupt	RSCAN0	Level	1108 <sub>H</sub>	—	—	—	*9	*10	✓	+420 <sub>H</sub>								
265	ICRCAN4REC	FFFF A212 <sub>H</sub>	INTRCAN4REC	CAN4 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	1109 <sub>H</sub>	—	—	—	*9	*10	✓	+424 <sub>H</sub>								
266	ICRCAN4TRX	FFFF A214 <sub>H</sub>	INTRCAN4TRX	CAN4 transmit interrupt	RSCAN0	Level	110A <sub>H</sub>	—	—	—	*9	*10	✓	+428 <sub>H</sub>								

Table 6.4 EI Level Maskable Interrupt Sources (8/8)

Interrupt			Interrupt Request				Exception Source Code	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset) <sup>*8</sup>								
Interrupt No. <sup>*1</sup>	Control Register		Name	Source	Unit	Detection Type <sup>*2</sup>								Direct Jumping to an Address								
	Name	Address												RINT = 0	RINT = 1	Reference to a Table						
267	ICRLIN26	FFFF A216 <sub>H</sub>	INTRLIN26	RLIN26 interrupt	RLIN241	Edge	110B <sub>H</sub>	—	—	—	—	—	✓	*3	*4	+42C <sub>H</sub>						
268	ICRLIN27	FFFF A218 <sub>H</sub>	INTRLIN27	RLIN27 interrupt	RLIN241	Edge	110C <sub>H</sub>	—	—	—	—	—	✓			+430 <sub>H</sub>						
269	ICPWGA64	FFFF A21A <sub>H</sub>	INTPWGA64	PWGA64 interrupt	PWGA64	Edge	110D <sub>H</sub>	—	—	—	—	—	✓			+434 <sub>H</sub>						
270	ICPWGA65	FFFF A21C <sub>H</sub>	INTPWGA65	PWGA65 interrupt	PWGA65	Edge	110E <sub>H</sub>	—	—	—	—	—	✓			+438 <sub>H</sub>						
271	ICPWGA66	FFFF A21E <sub>H</sub>	INTPWGA66	PWGA66 interrupt	PWGA66	Edge	110F <sub>H</sub>	—	—	—	—	—	✓			+43C <sub>H</sub>						
272	ICPWGA67	FFFF A220 <sub>H</sub>	INTPWGA67	PWGA67 interrupt	PWGA67	Edge	1110 <sub>H</sub>	—	—	—	—	—	✓			+440 <sub>H</sub>						
273	ICPWGA68	FFFF A222 <sub>H</sub>	INTPWGA68	PWGA68 interrupt	PWGA68	Edge	1111 <sub>H</sub>	—	—	—	—	—	✓			+444 <sub>H</sub>						
274	ICPWGA69	FFFF A224 <sub>H</sub>	INTPWGA69	PWGA69 interrupt	PWGA69	Edge	1112 <sub>H</sub>	—	—	—	—	—	✓			+448 <sub>H</sub>						
275	ICPWGA70	FFFF A226 <sub>H</sub>	INTPWGA70	PWGA70 interrupt	PWGA70	Edge	1113 <sub>H</sub>	—	—	—	—	—	✓			+44C <sub>H</sub>						
276	ICPWGA71	FFFF A228 <sub>H</sub>	INTPWGA71	PWGA71 interrupt	PWGA71	Edge	1114 <sub>H</sub>	—	—	—	—	—	✓			+450 <sub>H</sub>						
277	ICRLIN28	FFFF A22A <sub>H</sub>	INTRLIN28	RLIN28 interrupt	RLIN210	Edge	1115 <sub>H</sub>	—	—	—	—	—	✓			+454 <sub>H</sub>						
278	ICRLIN29	FFFF A22C <sub>H</sub>	INTRLIN29	RLIN29 interrupt	RLIN211	Edge	1116 <sub>H</sub>	—	—	—	—	—	✓			+458 <sub>H</sub>						
279	ICRCAN5ERR	FFFF A22E <sub>H</sub>	INTRCAN5ERR	CAN5 error interrupt	RSCAN0	Level	1117 <sub>H</sub>	—	—	*9	*10	✓	+45C <sub>H</sub>									
280	ICRCAN5REC	FFFF A230 <sub>H</sub>	INTRCAN5REC	CAN5 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	1118 <sub>H</sub>	—	—	*9	*10	✓	+460 <sub>H</sub>									
281	ICRCAN5TRX	FFFF A232 <sub>H</sub>	INTRCAN5TRX	CAN5 transmit interrupt	RSCAN0	Level	1119 <sub>H</sub>	—	—	*9	*10	✓	+464 <sub>H</sub>									
282	Reserved						111A <sub>H</sub>	—	—	—	—	—	—			+468 <sub>H</sub>						
283	Reserved						111B <sub>H</sub>	—	—	—	—	—	—			+46C <sub>H</sub>						
284	Reserved						111C <sub>H</sub>	—	—	—	—	—	—			+470 <sub>H</sub>						
285	Reserved						111D <sub>H</sub>	—	—	—	—	—	—			+474 <sub>H</sub>						
286	Reserved						111E <sub>H</sub>	—	—	—	—	—	—			+478 <sub>H</sub>						
287	Reserved						111F <sub>H</sub>	—	—	—	—	—	—			+47C <sub>H</sub>						

- Note 1. Each interrupt is connected to INTC1 channel 0 to 31 and INTC2 channel 32 to 287.
- Note 2. This indicates whether an interrupt source is detected at the level or edge. This also affects the value after reset of an EI level interrupt control register. For details, see **Section 6.4.1, ICxxx — EI Level Interrupt Control Registers**. For detection at level, an interrupt source is cleared by accessing to the register that retains an interrupt source. Dummy-reading of the register and execution of the SYNCNP instruction are required to reflect the result of the register update to the subsequent instruction  
不管中断channel是 偏移地址根据优先级(0~7)确定从+100H到170H的范围
- Note 3. Irrespective of interrupt channels, an offset address is determined in the range from +100<sub>H</sub> to 170<sub>H</sub> according to the priority (0 to 7).
- Note 4. 不管优先级 偏移地址都是+100H
- Note 5. The table reference method uses a table for reading an exception handler address for each interrupt channel, and it extracts handler address by referencing that table. Table reference position is determined by the following formula.  
Exception handler address read position = INTBP register + channel number × 4 bytes
- Note 6. 同一个中断源分配给不同的中断channel  
The same interrupt source is assigned to different interrupt channels. For details, see **Section 6.5.1, SELB\_INTC1 — INTC1 Interrupt Select Register**.
- Note 7. 不同的中断源指定给同一个中断channel  
Two interrupt sources are assigned to the same interrupt channel. For details, see **Section 6.5.2, SELB\_INTC2 — INTC2 Interrupt Select Register**.
- Note 8. For details, see **Section 6.10, Exception Handler Address**.
- Note 9. Only supported in the F1L for Gateway.
- Note 10. Only supported in devices with 1.5- and 2-MB code flash memories.
- Note 11. INTOSTM0 can operate as an EIINT or FEINT interrupt, but using it both ways at the same time is not possible.  
It is used as FEINT when OSTM0 functions as the TSU (timing supervision unit). It is used as EIINT when OSTM0 functions as anything other than the TSU.
- Note 12. For details on the interrupt source, see the *RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface*.

## 6.2.2 FE Level Non-Maskable Interrupt Sources 中断源

### 6.2.2.1 WDTNMIF — FENMI Factor Register

This register contains information about which source has generated the FE level non-maskable interrupt (FENMI). This register is initialized by any reset.

<b>Access:</b>	This register is a read-only register that can be read in 32-bit units.															
<b>Address:</b>	$FFF8\ 0200_H$															
<b>Value after reset:</b>	$0000\ 0000_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA1 NMIF	WDTA0 NMIF	TNMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.5 WDTNMIF Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	WDTA1NMIF	WDTA1NMI flag 0: No WDTA1NMI occurred 1: WDTA1NMI has occurred
1	WDTA0NMIF	WDTA0NMI flag 0: No WDTA0NMI occurred 1: WDTA0NMI has occurred
0	TNMF	Input signal flag from the NMI pin 0: No TNMI occurred 1: TNMI has occurred

### 6.2.2.2 WDTNMIFC — WDTNMI Factor Clear Register

This register clears the FE level non-maskable interrupt flags of the WDTNMIF register.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** FFF8 0208<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA1 NMIFC	WDTA0 NMIFC	TNMIF C
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

**Table 6.6 WDTNMIFC Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing, write the value after reset.
2	WDTA1NMIFC	WDTA1NMIF flag clear 0: — 1: Clear
1	WDTA0NMIFC	WDTA0NMIF flag clear 0: — 1: Clear
0	TNMIFC	TNMIF flag clear 0: — 1: Clear

## 6.2.3 FE Level Maskable Interrupt Sources 中断源

### 6.2.3.1 FEINTF — FEINT Factor Register

This register contains information about which source has generated the FE level maskable interrupt (FEINT). This register is initialized by any reset.

<b>Access:</b>	This register is a read-only register that can be read in 32-bit units.															
<b>Address:</b>	FFF8 0300H															
<b>Value after reset:</b>	0000 0000H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIH FEIF	OSTM0 FEIF	ECCRA MFEIF	ECCSD FLI0 FEIF	—	ECCDC SIH3 FEIF	ECCDC SIH2 FEIF	ECCDC SIH1 FEIF	ECCDC SIH0 FEIF	ECCDC NRAM FEIF	—	ECCDE EPO FEIF	—	—	—	LVIL FEIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.7 FEINTF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	LVIHFEIF	INTLVIH interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
14	OSTM0 FEIF	INTOSTM0_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
13	ECCRAM FEIF	INTECCRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
12	ECCSDFLI0 FEIF	INTECCSDFLI0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
11	Reserved	When read, the value after reset is returned.
10	ECCDCSIH3 FEIF <sup>*1</sup>	INTECCDCSIH3 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
9	ECCDCSIH2 FEIF <sup>*1</sup>	INTECCDCSIH2 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
8	ECCDCSIH1 FEIF <sup>*1</sup>	INTECCDCSIH1 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
7	ECCDCSIH0 FEIF	INTECCDCSIH0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
6	ECCDCNRAM FEIF <sup>*1</sup>	INTECCDCNRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
5	Reserved	When read, the value after reset is returned.

**Table 6.7 FEINTF Register Contents (2/2)**

Bit Position	Bit Name	Function
4	ECCDEEP0 FEIF	INTECCDEEP0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
3 to 1	Reserved	When read, the value after reset is returned.
0	LVLFEIF	INTLVIL interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

Note 1. For the supported products, see **Table 6.3, FE Level Maskable Interrupt Requests**.

### 6.2.3.2 FEINTFMSK — FEINT Factor Mask Register

This register masks the FE level maskable interrupt (FEINT). This register is initialized by any reset factor.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 0304<sub>H</sub>

**Value after reset:** FFFF FFFF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	LVLIFEI FMSK
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R/W

Table 6.8 FEINTFMSK Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	LVIHFEIFMSK	INTLVIH interrupt mask 0: Not masked 1: Masked
14	OSTM0FEI FMSK	INTOSTM0_FE interrupt mask 0: Not masked 1: Masked
13	ECCRAMFEI FMSK	INTECCRAM interrupt mask 0: Not masked 1: Masked
12	ECCSDFLI0FEI FMSK	INTECCSDFLI0 interrupt mask 0: Not masked 1: Masked
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECCDCSIH3FEI FMSK <sup>*1</sup>	INTECCDCSIH3 interrupt mask 0: Not masked 1: Masked
9	ECCDCSIH2FEI FMSK <sup>*1</sup>	INTECCDCSIH2 interrupt mask 0: Not masked 1: Masked
8	ECCDCSIH1FEI FMSK <sup>*1</sup>	INTECCDCSIH1 interrupt mask 0: Not masked 1: Masked
7	ECCDCSIH0FEI FMSK	INTECCDCSIH0 interrupt mask 0: Not masked 1: Masked
6	ECCDCNRAMF EIFMSK <sup>*1</sup>	INTECCDCNRAM interrupt mask 0: Not masked 1: Masked
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

**Table 6.8 FEINTFMSK Register Contents (2/2)**

Bit Position	Bit Name	Function
4	ECCDEEP0FEI FMSK	INTECCDEEP0 interrupt mask 0: Not masked 1: Masked
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LVLFEIFMSK	INTLVIL interrupt mask 0: Not masked 1: Masked

Note 1. For the supported products, see **Table 6.3, FE Level Maskable Interrupt Requests**.

### 6.2.3.3 FEINTFC — FEINT Factor Clear Register

This register clears the bits of the FEINT factor register (FEINTF).

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** FFF8 0308<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIH FEIFC	OSTM0 FEIFC	ECC RAM FEIFC	ECCSD FLI0 FEIFC	—	ECCDC SIH3 FEIFC	ECCDC SIH2 FEIFC	ECCDC SIH1 FEIFC	ECCDC SIH0 FEIFC	ECCDC NRAM FEIFC	—	ECC DEEP0 FEIFC	—	—	—	LVIL FEIFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	R	W	W	W	W	W	R	W	R	R	R	W

**Table 6.9 FEINTFC Register Contents (1/2)**

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing, write the value after reset.
15	LVIHFEIFC	LVIHFEIF flag clear 0: — 1: Clear
14	OSTM0 FEIFC	OSTM0FEIF flag clear 0: — 1: Clear
13	ECCRAM FEIFC	ECCRAMFEIF flag clear 0: — 1: Clear
12	ECCSDFLI0 FEIFC	ECCSDFLI0FEIF flag clear 0: — 1: Clear
11	Reserved	When writing, write the value after reset.
10	ECCDCSIH3 FEIFC <sup>*1</sup>	ECCDCSIH3FEIF flag clear 0: — 1: Clear
9	ECCDCSIH2 FEIFC <sup>*1</sup>	ECCDCSIH2FEIF flag clear 0: — 1: Clear
8	ECCDCSIH1 FEIFC <sup>*1</sup>	ECCDCSIH1FEIF flag clear 0: — 1: Clear
7	ECCDCSIH0 FEIFC	ECCDCSIH0FEIF flag clear 0: — 1: Clear
6	ECCDCNRAM FEIFC <sup>*1</sup>	ECCDCNRAMFEIFC flag clear 0: — 1: Clear
5	Reserved	When writing, write the value after reset.
4	ECCDEEP0 FEIFC	ECCDEEP0FEIF flag clear 0: — 1: Clear
3 to 1	Reserved	When writing, write the value after reset.

**Table 6.9 FEINTFC Register Contents (2/2)**

Bit Position	Bit Name	Function
0	LVLFEIFC	LVLFEIF flag clear 0: — 1: Cleared

Note 1. For the supported products, see **Table 6.3, FE Level Maskable Interrupt Requests**.

## 6.3 Edge/Level Detection 边沿/电平触发

在一个外部的pin检测到上升沿，下降沿，上升/下降沿，高 低电平的时候就会触发外部中断

External interrupts (TNMI and INTPm) can be specified to be generated when a rising edge, falling edge, rising or falling edge, low level, or high level is detected at an external interrupt pin.

The following registers are used to specify the edge and level of each interrupt:

下面的寄存器用来指定每个中断的触发方式

**Table 6.10 External Interrupt Edge/Level Detection Registers**

Interrupt	Register
TNMI	FCLA0CTL0_NMI
INTP0	FCLA0CTL0_INTPL
INTP1	FCLA0CTL1_INTPL
INTP2	FCLA0CTL2_INTPL
INTP3	FCLA0CTL3_INTPL
INTP4	FCLA0CTL4_INTPL
INTP5	FCLA0CTL5_INTPL
INTP6 <sup>*1</sup>	FCLA0CTL6_INTPL
INTP7 <sup>*1</sup>	FCLA0CTL7_INTPL
INTP8 <sup>*1</sup>	FCLA0CTL0_INTPH
INTP9 <sup>*1</sup>	FCLA0CTL1_INTPH
INTP10	FCLA0CTL2_INTPH
INTP11	FCLA0CTL3_INTPH
INTP12 <sup>*1</sup>	FCLA0CTL4_INTPH
INTP13 <sup>*1</sup>	FCLA0CTL5_INTPH
INTP14 <sup>*1</sup>	FCLA0CTL6_INTPH
INTP15 <sup>*1</sup>	FCLA0CTL7_INTPH

Note 1. For the supported products, see **Table 6.4, EI Level Maskable Interrupt Sources**.

See **Section 2, Pin Function** for details of these registers.

## 6.4 Interrupt Controller Control Registers

Writing to the ICxxx, IMR<sub>m</sub> ( $m = 0$  to  $8$ ), FNC, and FIC registers is enabled only in supervisor mode (PSW.UM = 0). 只有在这种情况下，上面的寄存器才会写入

### 6.4.1 ICxxx — EI Level Interrupt Control Registers

One of these registers is assigned to each EI level maskable interrupt (EIINT) channel and is used to set the conditions for controlling that channel. This register is initialized by any reset.

**Access:** ICxxx can be read or written in 16-bit units.  
ICxxxH and ICxxxL can be read or written in 8- or 1-bit units.  
Access to the bit 14, 13, 11 to 8, 5 to 3 by using a SET1, CLR1, or NOT1 instruction is prohibited.

**Address:** See Table 6.4, EI Level Maskable Interrupt Sources.

**Value after reset:** 0087<sub>H</sub> (edge detection), 8087<sub>H</sub> (level detection)<sup>\*1</sup>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTxxx	—	—	RFxxx	—	—	—	—	MKxxx	TBxxx	—	—	—	P2xxx	P1xxx	P0xxx
Value after reset	0/1 <sup>*1</sup>	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W

Note 1. The value after reset differs depending on the detection type of a given interrupt (edge detection: 0, level detection: 1). For details, see Table 6.4, EI Level Maskable Interrupt Sources.

Table 6.11 ICxxx Register Contents (1/2)

Bit Position	Bit Name	Function
15	CTxxx <b>中断检测类型</b>	This bit indicates the type of interrupt detection. This bit is read only. 0: Edge detection 1: Level detection When writing in 8-bit or 16-bit units, write the value after reset.
14, 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	RFxxx <b>中断请求标志</b>	This is an interrupt request flag. 可以由程序写入 The RFxxx bit can be written from a program. Setting the RFxxx bit to 1 generates an EI level maskable interrupt n (EIINT <sub>n</sub> ), just as when an interrupt request is acknowledged. 此位被置位，产生一个EIINT <sub>n</sub> ，就像一个中断请求被探测一样 0: No interrupt request is made. 1: Interrupt request is made.
		<b>中断请求被CPU核探测，会被自动清除</b>
		Edge detection (CTxxx = 0) This bit is automatically cleared when an interrupt request is acknowledged by the CPU core. It can be set and cleared by software. 可以被软件置位和清除
		Level detection (CTxxx = 1) This bit cannot be set or cleared by software. It can only be read. 此位不可以被软件置位或者清除，只读 It is not cleared even if an interrupt request is acknowledged by the CPU core. 即便中断请求被CPU核探测，也不会被清除
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MKxxx <b>中断请求屏蔽位</b>  如果此位设置，从某个channel来的中断请求会被屏蔽，不会提交给CPU核。对于被屏蔽的channel，其中断状态不会反映在ICSR.PMEI bit上。对于屏蔽的channel，其RFxx仍然反映其的中断请求，并且可以被软件处理。  当MKXXX被清除，某个channel中断请求会被提交给CPU核进行下一步处理	This is the interrupt request mask bit. When the MKxxx bit is set, interrupt requests from the channel are masked and are not issued to the CPU core. The interrupt pending status is not reflected in the ICSR.PMEI bit for any channels that are masked. When the interrupt request from the channel is masked with MKxxx = 1, the RFxxx still reflects the interrupt request for the channel and can be polled in software. When the MKxxx bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the MKxxx bit is also reflected in the corresponding IMR <sub>m</sub> register. 0: Enables interrupt processing 1: Disables interrupt processing
		MKxxx的状态同样反映在对应的IMR寄存器

Table 6.11 ICxxx Register Contents (2/2)

Bit Position	Bit Name	Function
6	TBxxx	This bit is used to select the way to determine the interrupt vector. <b>决定中断向量的选择方式</b> 0: Direct jumping to an address determined from the level of priority 1: Table reference  For details on the way to determine the interrupt vector, see the <u>RH850G3K User's Manual: Software</u> .
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	P2xxx to P0xxx	These bits specify the interrupt priority as one of 8 levels, with 0 as the highest and 7 as the lowest. <b>指定8个中断优先级中的一个，0代表最高，7代表最低</b> <b>当多个EI级别中断请求同时发生具有最高优先级中断源的出来的中断被选择传送至CPU核进行处理</b> <b>当同优先级的中断发生，低channel number优先处理</b> <b>CAUTION</b> When multiple EI level-interrupt requests are made simultaneously, the interrupt from the source with the highest priority setting in these bits is selected and conveyed to the CPU core for servicing first. When P2xxx to P0xxx bits specify the same priority level for simultaneously occurring interrupt requests, the source with the lower channel number takes priority.
Do not access ICxxx registers of interrupt channels listed as "Reserved" in <b>Table 6.4, EI Level Maskable Interrupt Sources</b> and of the channels which are not incorporated in the product.		

### 6.4.2 IMRm — EI Level Interrupt Mask Registers ( $m = 0$ to 8)

这些寄存器是ICxxx寄存器中MKxxxbits的集合

IMRm的每一个bit都反映了对应MKxxxbit的设置

These registers are a collection of the MKxxx bits of the ICxxx registers. Each bit of IMRm reflects the setting of the corresponding MKxxx bit. The setting of IMRm is also reflected in the corresponding MKxxx bit. This register is initialized by any reset.

**Access:** IMRm can be read or written in 32-bit units.

IMRmH and IMRmL can be read or written in 16-bit units.

IMRmHH, IMRmHL, IMRmLH, and IMRmLL can be read or written in 8- or 1-bit units.

**Address:** IMR0: FFFF 90F0<sub>H</sub>

IMRm ( $m = 1$  to 8): FFFF A400<sub>H</sub> + (04<sub>H</sub> ×  $m$ )

**Value after reset:** FFFF FFFF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMRm EIMK ( $m \times 32 + 31$ )	IMRm EIMK ( $m \times 32 + 30$ )	IMRm EIMK ( $m \times 32 + 29$ )	IMRm EIMK ( $m \times 32 + 28$ )	IMRm EIMK ( $m \times 32 + 27$ )	IMRm EIMK ( $m \times 32 + 26$ )	IMRm EIMK ( $m \times 32 + 25$ )	IMRm EIMK ( $m \times 32 + 24$ )	IMRm EIMK ( $m \times 32 + 23$ )	IMRm EIMK ( $m \times 32 + 22$ )	IMRm EIMK ( $m \times 32 + 21$ )	IMRm EIMK ( $m \times 32 + 20$ )	IMRm EIMK ( $m \times 32 + 19$ )	IMRm EIMK ( $m \times 32 + 18$ )	IMRm EIMK ( $m \times 32 + 17$ )	IMRm EIMK ( $m \times 32 + 16$ )
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMRm EIMK ( $m \times 32 + 15$ )	IMRm EIMK ( $m \times 32 + 14$ )	IMRm EIMK ( $m \times 32 + 13$ )	IMRm EIMK ( $m \times 32 + 12$ )	IMRm EIMK ( $m \times 32 + 11$ )	IMRm EIMK ( $m \times 32 + 10$ )	IMRm EIMK ( $m \times 32 + 9$ )	IMRm EIMK ( $m \times 32 + 8$ )	IMRm EIMK ( $m \times 32 + 7$ )	IMRm EIMK ( $m \times 32 + 6$ )	IMRm EIMK ( $m \times 32 + 5$ )	IMRm EIMK ( $m \times 32 + 4$ )	IMRm EIMK ( $m \times 32 + 3$ )	IMRm EIMK ( $m \times 32 + 2$ )	IMRm EIMK ( $m \times 32 + 1$ )	IMRm EIMK ( $m \times 32 + 0$ )
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R/W															

Table 6.12 IMRm Register Contents

Bit Position	Bit Name	Function
31 to 0	IMRmEIMK ( $m \times 32 + 31$ ) to IMRmEIMK ( $m \times 32 + 0$ )	These are interrupt mask bits for EI level maskable interrupt (EIINT) channels 0 to 287. m的范围 0: Enables interrupt servicing 1: Disables interrupt servicing

#### CAUTION

MKxxx bits which correspond to channels listed as "Reserved" in **Table 6.4, EI Level Maskable Interrupt Sources**, and to channels which are not incorporated in the product must be set to "1".

### Non-maskable Interrupt

#### 6.4.3 FNC — FE Level NMI Status Register

This register indicates the status of an FE level non-maskable interrupt (FENMI).

**Access:** FNC is a read-only register that can be read in 16-bit units. **只读寄存器**  
FNCH is a read-only register that can be read in 8-bit or 1-bit units.

**Address:** FNC: FFFF 9078<sub>H</sub>  
FNCH: FFFF 9079<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FNRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.13 FNC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FNRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred  This bit is automatically cleared when an FE level NMI interrupt request is acknowledged by the CPU core. <b>FE NMI 中断请求一旦被CPU核所探知，此位会被自动清零</b>
11 to 0	Reserved	When read, the value after reset is returned.

#### 6.4.4 FIC — FE Level Maskable Interrupt Status Register

This register indicates the status of an FE level maskable interrupt (FEINT).

**Access:** FIC is a read-only register that can be read in 16-bit units. **只读寄存器**  
FICH is a read-only register that can be read in 8- or 1-bit units.

**Address:** FIC: FFFF 907A<sub>H</sub>  
FICH: FFFF 907B<sub>H</sub>

**Value after reset:** 8000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FIRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.14 FIC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FIRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred <b>此位不能由软件置位或者清除，只读</b> This bit cannot be set or cleared by software. It can only be read.
11 to 0	Reserved	When read, the value after reset is returned.

## 6.5 EI Level Maskable Interrupt Select Registers

以下寄存器选择一个EI level 可屏蔽中断

The following registers are used to select an EI level maskable interrupt.

### 6.5.1 SELB\_INTC1 — INTC1 Interrupt Select Register

When two interrupt sources are assigned to one interrupt channel, this register selects which interrupt sources is enabled. 当两个中断源被指定给一个中断channel 的时候。此寄存器选择哪个中断源被使能

#### NOTE

The channel described in each bit setting indicates the channel of an interrupt and the priority. For details on channels, see **Table 6.4, EI Level Maskable Interrupt Sources**.

<b>Access:</b>	This register can be read or written in 16-bit units.															
<b>Address:</b>	FFBC 0300 <sub>H</sub>															
<b>Value after reset:</b>	0000 <sub>H</sub>															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SELB_INTC1_11	SELB_INTC1_10	SELB_INTC1_9	SELB_INTC1_8	SELB_INTC1_7	SELB_INTC1_6	SELB_INTC1_5	SELB_INTC1_4	SELB_INTC1_3	SELB_INTC1_2	SELB_INTC1_1	SELB_INTC1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.15 SELB\_INTC1 Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	SELB_INTC1_11	Interrupt channel selection 0: INTTAUD0I14 (Channel 7) <span style="color:red">0-31</span> INTCSIH3IJC (Channel 153) <span style="color:red">32-287</span> 1: INTCSIH3IJC_1 (Channel 7) INTTAUD0I14_2 (Channel 153)
10	SELB_INTC1_10	Interrupt channel selection 0: INTTAUD0I12 (Channel 6) INTCSIH3IRE (Channel 152) 1: INTCSIH3IRE_1 (Channel 6) INTTAUD0I12_2 (Channel 152)
9	SELB_INTC1_9	Interrupt channel selection 0: INTTAUD0I10 (Channel 5) INTCSIH3IR (Channel 151) 1: INTCSIH3IR_1 (Channel 5) INTTAUD0I10_2 (Channel 151)
8	SELB_INTC1_8	Interrupt channel selection 0: INTTAUD0I2 (Channel 1) INTCSIH3IC (Channel 150) 1: INTCSIH3IC_1 (Channel 1) INTTAUD0I2_2 (Channel 150)
7	SELB_INTC1_7	Interrupt channel selection 0: INTP2 (Channel 31) INTCSIH2IJC (Channel 127) 1: INTCSIH2IJC_1 (Channel 31) INTP2_2 (Channel 127)
6	SELB_INTC1_6	Interrupt channel selection 0: INTP1 (Channel 30) INTCSIH2IRE (Channel 126) 1: INTCSIH2IRE_1 (Channel 30) INTP1_2 (Channel 126)

**Table 6.15 SELB\_INTC1 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
5	SELB_INTC1_5	Interrupt channel selection 0: INTPO (Channel 29) INTCSIH2IR (Channel 125) 1: INTCSIH2IR_1 (Channel 29) INTP0_2 (Channel 125)
4	SELB_INTC1_4	Interrupt channel selection 0: INTTAUD0I0 (Channel 0) INTCSIH2IC (Channel 124) 1: INTCSIH2IC_1 (Channel 0) INTTAUD0I0_2 (Channel 124)
3	SELB_INTC1_3	Interrupt channel selection 0: INTCSIG0IR (Channel 20) INTCSIH1JC (Channel 111) 1: INTCSIH1JC_1 (Channel 20) INTCSIG0IR_2 (Channel 111)
2	SELB_INTC1_2	Interrupt channel selection 0: INTCSIG0IC (Channel 19) INTCSIH1IRE (Channel 110) 1: INTCSIH1IRE_1 (Channel 19) INTCSIG0IC_2 (Channel 110)
1	SELB_INTC1_1	Interrupt channel selection 0: INTTAPA0IVLY0 (Channel 9) INTCSIH1IR (Channel 109) 1: INTCSIH1IR_1 (Channel 9) INTTAPA0IVLY0_2 (Channel 109)
0	SELB_INTC1_0	Interrupt channel selection 0: INTTAPA0IPEK0 (Channel 8) INTCSIH1IC (Channel 108) 1: INTCSIH1IC_1 (Channel 8) INTTAPA0IPEK0_2 (Channel 108)

**CAUTION**

 The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB\_INTC1.

## 6.5.2 SELB\_INTC2 — INTC2 Interrupt Select Register

When two interrupt sources are assigned to one interrupt channel, this register selects which interrupt sources is enabled.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFBC 0304<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SELB_INTC2_10	SELB_INTC2_9	SELB_INTC2_8	SELB_INTC2_7	SELB_INTC2_6	SELB_INTC2_5	SELB_INTC2_4	SELB_INTC2_3	SELB_INTC2_2	SELB_INTC2_1	SELB_INTC2_0
Value after reset	0	0	0	0	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.16 SELB\_INTC2 Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	SELB_INTC2_10	Interrupt channel 149 selection <sup>*1</sup> 0: INTTAUB0I15 1: INTPWGA31
9	SELB_INTC2_9	Interrupt channel 147 selection <sup>*1</sup> 0: INTTAUB0I13 1: INTPWGA30
8	SELB_INTC2_8	Interrupt channel 145 selection <sup>*1</sup> 0: INTTAUB0I11 1: INTPWGA26
7	SELB_INTC2_7	Interrupt channel 143 selection <sup>*1</sup> 0: INTTAUB0I9 1: INTPWGA19
6	SELB_INTC2_6	Interrupt channel 141 selection <sup>*1</sup> 0: INTTAUB0I7 1: INTPWGA18
5	SELB_INTC2_5	Interrupt channel 139 selection <sup>*1</sup> 0: INTTAUB0I5 1: INTPWGA17
4	SELB_INTC2_4	Interrupt channel 137 selection <sup>*1</sup> 0: INTTAUB0I3 1: INTPWGA16
3	SELB_INTC2_3	Interrupt channel 80 selection 0: INTENCA0I1 1: INTPWGA7
2	SELB_INTC2_2	Interrupt channel 79 selection 0: INTENCA0I0 1: INTPWGA6
1	SELB_INTC2_1	Interrupt channel 78 selection 0: INTENCA0IUD 1: INTPWGA5
0	SELB_INTC2_0	Interrupt channel 77 selection 0: INTENCA0IOV 1: INTPWGA4

Note 1. For the supported products, see **Table 6.4, EI Level Maskable Interrupt Sources**.

**CAUTION**

---

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB\_INTC2.

---

## 6.6 Interrupt Function System Registers

See [Section 3.3.3, Interrupt Function Registers](#).

### 6.6.1 ISPR — Priority of Interrupt being Serviced

See [Section 3.3.3.1, ISPR — Priority of Interrupt being Serviced](#).

### 6.6.2 PMR — Interrupt Priority Masking

See [Section 3.3.3.2, PMR — Interrupt Priority Masking](#).

### 6.6.3 ICSR — Interrupt Control Status

See [Section 3.3.3.3, ICSR — Interrupt Control Status](#).

### 6.6.4 INTCFG — Interrupt Function Setting

See [Section 3.3.3.4, INTCFG — Interrupt Function Setting](#).

## 6.7 Operation when Acknowledging an Interrupt 收到中断时的操作

Check whether each interrupt that is reported during instruction execution is acknowledged according to the priority. The procedure for acknowledging each interrupt is shown below.

**检查认定条件是否满足，根据其优先级检查中断是否被收到**

- (1) Check whether the acknowledgment conditions are satisfied and whether interrupts are acknowledged according to their priority.
- (2) Calculate the exception handler address according to the current PSW value.\*<sup>1</sup>  
**根据当前PSW的值计算异常处理地址  
对于FE-level non-maskable/maskable中断，以下步骤要被执行**
- (3) For FE-level non-maskable/maskable interrupts, the following processing is performed:
  - Save the PC to the FEPC.
  - Save the PSW to the FEPSW.
  - Store the exception source code in the FEIC.
  - Update the PSW and MCTL.\*<sup>2</sup>
  - Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.
- (4) For EI level exceptions, the following processing is performed:
  - Save the PC to the EIPC.
  - Save the PSW to the EIPSW.
  - Store the exception source code in the EIIC.
  - Update the PSW and MCTL.\*<sup>2</sup>
  - Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

**Note 1.** For details, see **Section 6.10, Exception Handler Address**.

**Note 2.** For the values to be updated, see Table 4-1 Exception Cause List in the *RH850G3K User's Manual: Software*.

The following figure shows steps (1) to (4).

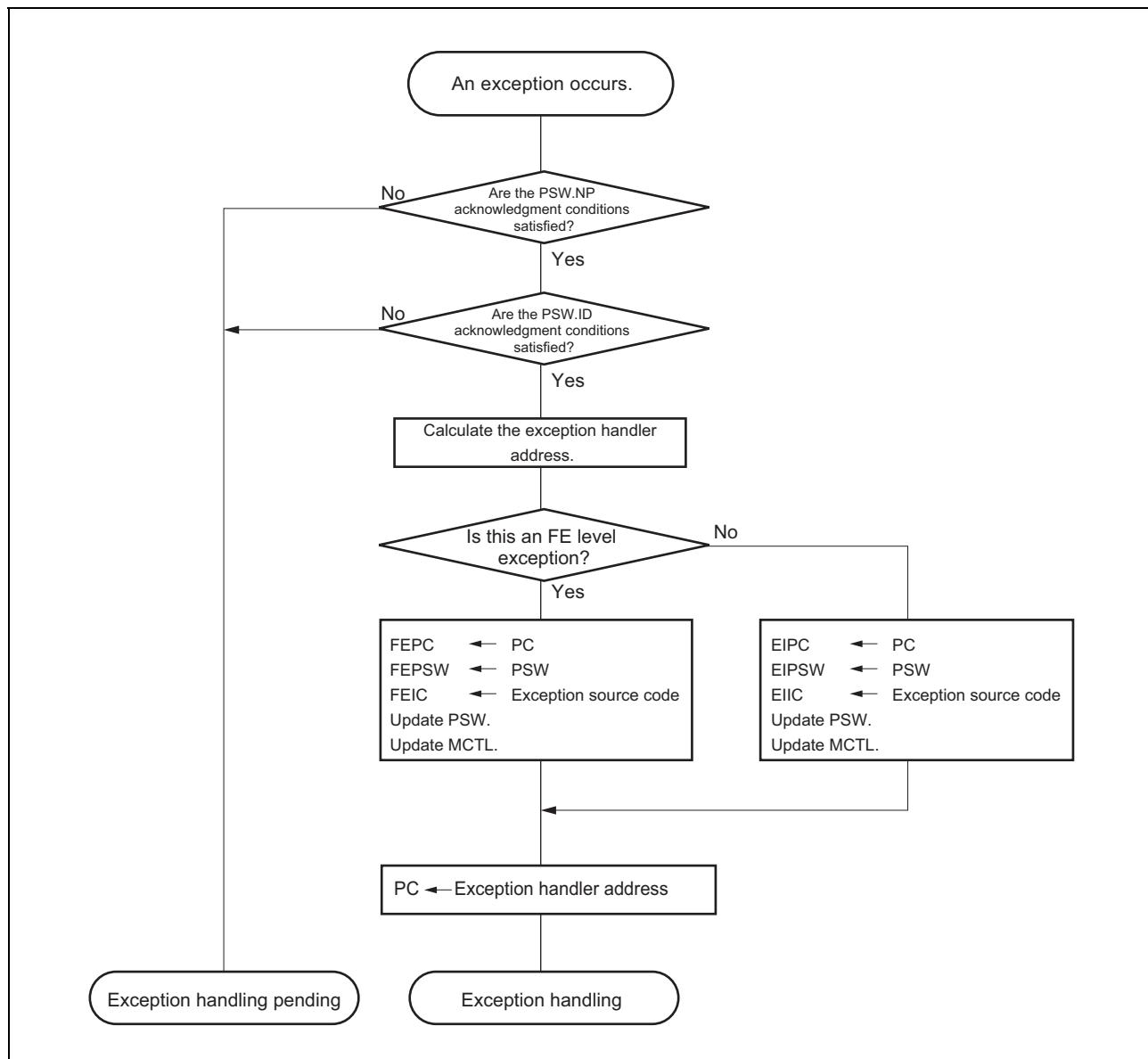


Figure 6.2 Operation when Acknowledging an Interrupt

### 6.7.1 Exception Source Codes When an SYSERR Exception Occurs

The following table lists exception source codes when an SYSERR exception occurs.

**Table 6.17    Exception Source Codes when an SYSERR Exception Occurs**

Exception Source Code	Source of SYSERR Generation
11 <sub>H</sub>	<ul style="list-style-type: none"> <li>Detection of an error during the fetching of an instruction from the code flash memory, or from the primary local, secondary local, or retention RAM areas</li> </ul>
14 <sub>H</sub> <sup>*1</sup> 具体是右边所列的哪一种，需要读取 SEG_FLAG 寄存器	<ul style="list-style-type: none"> <li>Detection of an error during access to data in the code flash area</li> <li>Detection of an error during access to data in the primary local, secondary local, or retention RAM areas</li> <li>Detection of a read-modify-write error during byte write or half-word write to the primary local, secondary local, or retention RAM</li> <li>Detection of an error during access to the PBUS module</li> </ul>
1F <sub>H</sub>	<ul style="list-style-type: none"> <li>Detection of a DMA transfer error</li> </ul>

Note 1. The actual source of a SYSERR for which the exception source code is 14H is checked by reading the SEG\_FLAG register.

## 6.8 Return from Interrupts

从中断处理中返回，执行对应与相关中断等级的返回指令（EIRET或者FERET）

To return from interrupt handling, execute the return instruction (EIRET or FERET) corresponding to each relevant interrupt level.

**当上下文被保存在stack，执行返回指令之前，上下文必须被恢复**

When a context has been saved in a stack and the like, the context must be restored before executing the return instruction.

The EIRET instruction is used to return from the EI level maskable interrupt handling and the FERET instruction is used to return from FE-level maskable interrupt handling.

**当EIRET或者FERET指令被执行，CPU执行如下的步骤，并且把控制权交给返回的PC地址**

When the EIRET or FERET instruction is executed, the CPU performs the following processing and then passes its control to the return PC address:

- (1) When returning from the service routine for an EI-level exception, the PC and PSW values on return are loaded from the EIPC and EIPSW registers.
- When returning from the service routine for an FE-level exception, the PC and PSW values on return are loaded from the FEPC and FEPFW registers.
- (2) Control is passed to the addresses indicated by the return PC and PSW that were loaded.
- (3) When EP = 0 and INTCFG.ISPC = 0, the CPU updates the ISPR register.  
**PSW.EP**

The flows for returning from exception handling using the EIRET and FERET instructions are shown below.

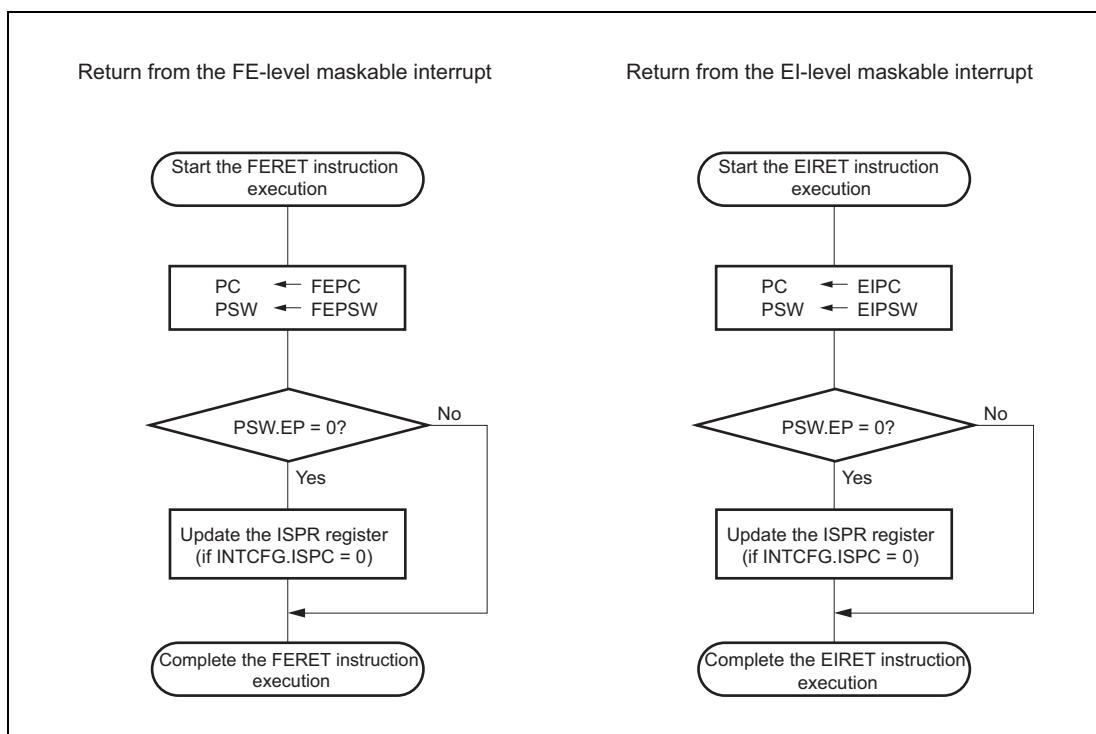


Figure 6.3 Flow of Return from Interrupts

## 6.9 Interrupt Operation

### 6.9.1 Interrupt Mask Function of EI Level Maskable Interrupt (EIINT)

Interrupt masking can be specified for each respective interrupt channel of EIINT. Interrupt masking is performed by the following register settings.

Table 6.18 Operation of the MKxxx Bit

ICxxx.MKxxx	Operation
1	Masks interrupt
0	Enables interrupt

The ICxxx.MKxxx bits can also be read and written via the corresponding IMRmEIMKn bits of the IMRm registers. The interrupt mask state is reflected in both the ICxxx registers and the IMRm registers.

#### [ Operation example ]

- (1) When a 1 is written to an IMRm.IMRmEIMKn bit, interrupts are prohibited for the corresponding channel.
- (2) When the corresponding ICxxx.MKxxx bit is read, 1 is returned.

#### CAUTION

中断请求进行时 (RFxxx = 1) If the MKxxx bit is set to 0 while an interrupt request is pending (RFxxx = 1), the MKxxx处于清零状态, 中断服务会被准时执行 (根据优先级) , interrupt service routine will be executed at that time (subject to the rules of interrupt prioritization). Even if an interrupt request is issued in software by setting the RFxxx 如果MKxxx被置位, 即使使用软件以RFxxx=1的形式产生中断请求bit to 1, the interrupt will not occur as long as the interrupt is masked with MKxxx = 1. 中断也不会发生 To cancel an interrupt request that is pending, clear the corresponding RFxxx bit in 如果取消待处理的中断请求, software. 使用软件将RFxxxbit置为 0

### 6.9.2 Interrupt Priority Level Judgment 中断优先级的判定

When FE level non-maskable interrupts (FENMI), FE level maskable interrupts (FEINT), and EI level maskable interrupts (EIINT) are input, priorities including other exceptions are determined, and the exception with the highest priority (including interrupts) is processed. Exceptions occurred at the same time (including interrupts) are processed in a pre-allocated priority order (the default priority order). The priority orders of FENMI, FEINT, and EIINT interrupts are as follows.

默认的优先级顺序如下所示

FENMI > FEINT > EIINT

See the *RH850G3K User's Manual: Software* for other exceptions.

对于EI INT (INTn) 中断, 每个中断源可以单独设置中断  
For EIINT (INTn) interrupts, the interrupt priority can be set independently for each interrupt source.  
在bit P2xxx到P0xxx上指定中断优先级  
Specify the interrupt priority with the bits P2xxx to P0xxx. The interrupt priority levels can be set from 0 to 7: 0 is the highest and 7 is the lowest. Among multiple EIINT (INTn) interrupts with the same priority level, the interrupt with the lowest interrupt channel number has priority.

在多个具有相同优先级的中断中, 低中断channel number 的中断最优先

Table 6.19 Example of EIINT (INTn) Interrupt Priority Level Settings and Priority Levels During Operation

channel number	寄存器设定的优先级	实际的优先级排序
EIINT (INTn)	ICxxx.P[2:0]xxx Setting	Priority Level During Operation
INT0	3	10
INT1	4	11
INT2	0	1
INT3	0	2
INT4	1	3
INT5	2	6
INT6	2	7
INT7	1	4
INT8	1	5
INT9	2	8
INT10	2	9

The interrupt controller executes multiple interrupt handling when another interrupt request is acknowledged while interrupt processing is being executed. When multiple EIINT (INTn) interrupts occur at the same time, the interrupt to be acknowledged is determined by the following procedure.

当很多的EIINT中断同一时间产生，中断的感知按照下面的程序来进行

### 与当前正在处理的中断优先级进行比较

#### 6.9.2.1 Comparison with the Priority Level of the Interrupt Currently being Handled

Interrupts with the same or lower priority level as the interrupt currently being handled are held pending.

当前正在处理的中断优先级相同或者比其低优先级的中断设置为待执行  
The priority level of the interrupt currently being handled is stored in the ISPR register.

比当前正在处理的中断优先级高的中断被放在下一次的优先级诊断阶段中进行  
Interrupts with a higher priority level than the interrupt currently being handled proceed to the next priority judgment stage.

#### 6.9.2.2 Masking through Priority Mask Register (PMR)

在PMR中被使能的中断才会在下一次优先级诊断阶段中进行  
Only interrupts enabled by the PMR register proceed to the next priority judgment stage.

For the PMR register, see **Section 3.3.3.2, PMR — Interrupt Priority Masking**, or the *RH850G3K User's Manual: Software*.

#### 6.9.2.3 The Requested Interrupt Source with the Highest Priority Level is Selected

由多个中断源产生的中断请求，优先级高者优先  
When interrupts are requested simultaneously from multiple sources, the interrupt set the highest priority by ICxxx.P[2:0]xxx bits takes priority. When there are multiple highest priority interrupts(ICxxx.P[2:0]xxx bits = 0), the lowest interrupt channel number is selected.

#### 6.9.2.4 Interrupt Hold by CPU

中断优先级的感知与否要根据PSW中的NP和ID的状态  
Interrupt acknowledgment is held according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among EIINT interrupts, and priority judgment among EIINT, FEINT and FENMI interrupts are performed even while interrupt acknowledgment is pending, and the interrupt with the highest priority is selected when the acknowledgment condition is satisfied.  
当感知的条件被满足，具有高优先级的中断会被选择

### Example 例子

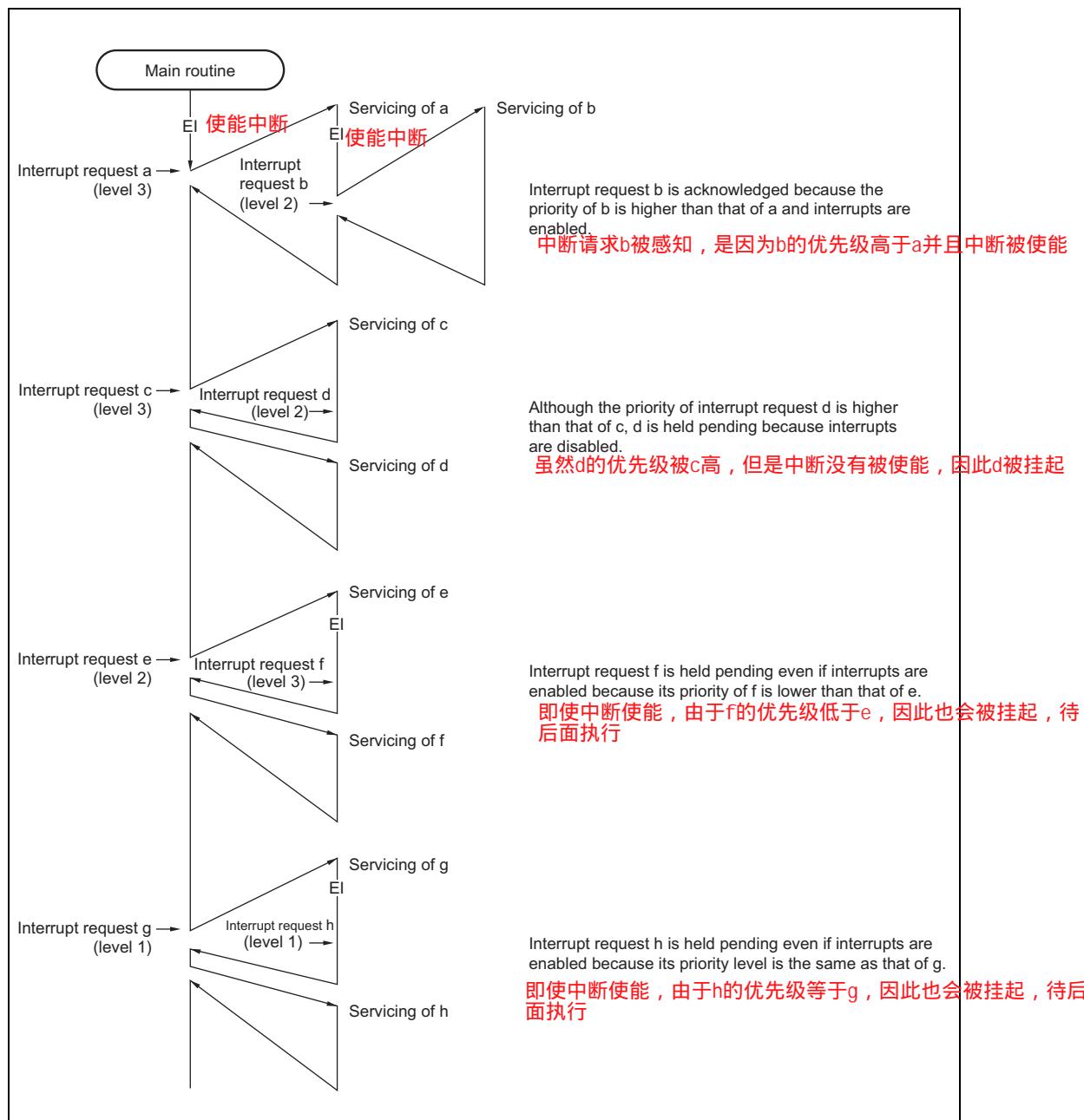
An EIINT interrupt with the priority level 5 has already been requested and interrupt generation is pending because the value of the PSW.ID bit is 1. If a subsequent EIINT interrupt with the priority 优先级为5的EIINT中断已经被请求，由于PSW.ID为1，其中断的产生将被挂起

如果随之而来既有优先级为3的请求，并且PSW.ID被清零，则优先级为3的中断将会产生 level 3 is requested and the PSW.ID bit is cleared to 0, the latter EIINT interrupt (with the priority level 3) will be generated.

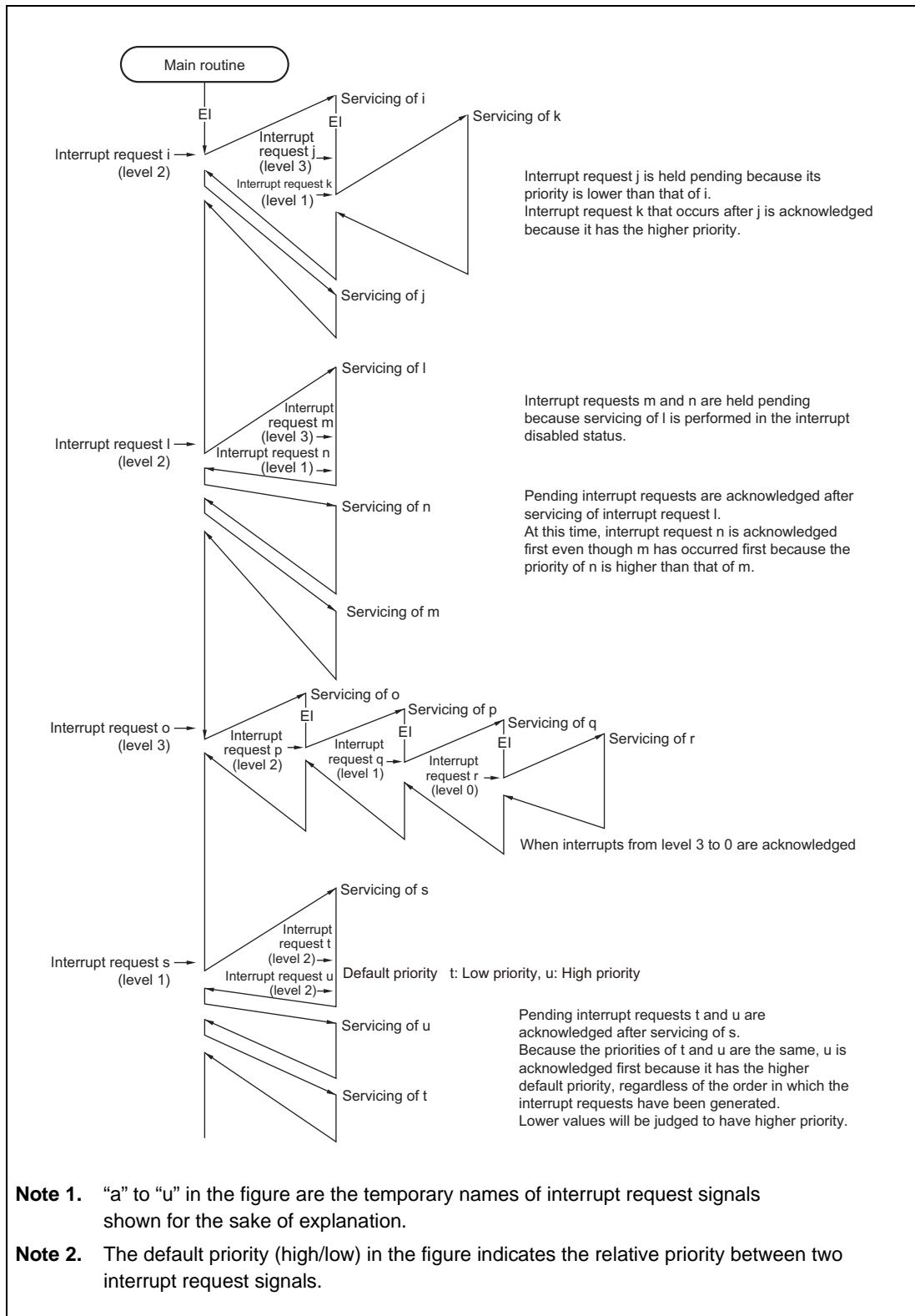
**Figure 6.4** shows an example of multiple interrupt handling when another interrupt request is acknowledged while interrupt processing is being executed.

当中断请求信号被感知，PSW.ID会被置位，因此ID标志应该被清零，这样多中断处理才能进行 When an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, the ID flag should be cleared to 0 to execute multiple interrupt handling. Specifically, execute the EI instruction and the like in an interrupt handling program to enable the interrupt.

特别的，在中断处理程序中执行EI指令可以使能中断



**Figure 6.4 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1)**



**Figure 6.5 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (2)**

**CAUTION** 为了实现多中断，EIPC核EIPSW寄存器在执行EI指令之前必须被保存

To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction. 从多中断服务中返回时，在执行DI指令以后，要恢复EIPC核EIPSW的值

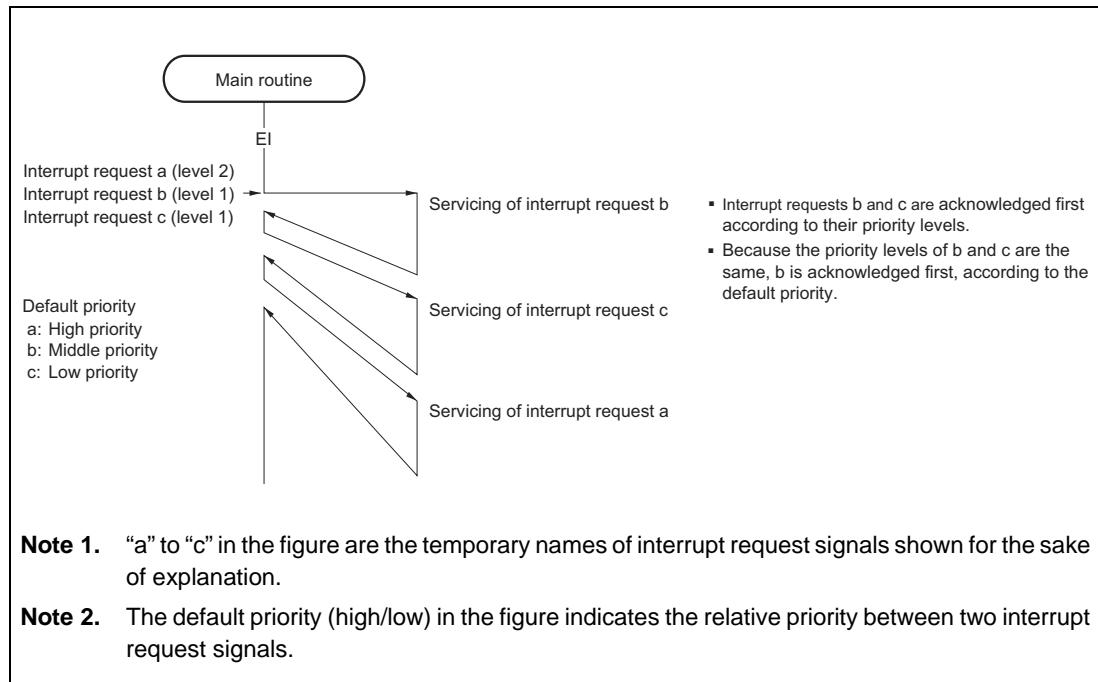


Figure 6.6 Example of Servicing Simultaneously Generated Interrupt Requests

### 6.9.3 Interrupt Request Acknowledgement Conditions and the Priority

See the *RH850G3K User's Manual: Software*.

### 6.9.4 Exception Priority of Interrupts and the Priority Mask

See the *RH850G3K User's Manual: Software*.

### 6.9.5 Interrupt Priority Mask

See the *RH850G3K User's Manual: Software*.

## 6.9.6 Priority Mask Function

**优先级屏蔽函数用来禁止具有指定优先级的所有EIINT中断**

The priority mask function prohibits all EIINT interrupts of the specified interrupt priority level.

**中断优先级的屏蔽在PMR寄存器中进行**

The interrupt priority levels to be masked are specified in the PMR register. Masking and acknowledgment can be set for each priority level. **屏蔽或者感知可以针对每个优先级来设置**

The following operations are possible using this function:**使用这些函数可以实现下面的操作**

- Temporary prohibition of interrupts that have a priority level that is lower than a given priority level
- Temporary prohibition of interrupts that have a given priority level

**Table 6.20 Operation of PMR.PMm Bit**

PMR.PMm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

**Note:** m = 0 to 7

The presence of EIINT interrupts held pending with this function can be checked with **Section 6.9.7, Exception Management**.

For details on the PMR register, see **Section 3.3.3.2, PMR — Interrupt Priority Masking**, or the *RH850G3K User's Manual: Software*.

## 6.9.7 Exception Management

**挂起的中断可以被检测**

Pending interrupts can be checked in the RH850/F1L. For details, see the *RH850G3K User's Manual: Software*.

## 6.10 Exception Handler Address

In the RH850/F1L, the exception handler address from which the handler is executed after a reset is input or when an exception or interrupt is acknowledged can be changed according to a setting.

异常处理地址（用于reset或者异常（含中断）），由直接向量方法决定

The exception handler address for resets and exceptions (including interrupts) is determined with the direct vector method, in which the reference point of the exception handler address can be changed by using the PSW.EBV bit, the RBASE register, and the EBASE register. For interrupts, the direct vector method and table reference method can be selected for each channel. If the table reference method is selected, execution can branch to the address indicated by the exception handler table allocated in the memory. 对于中断，每个channel可以选择直接向量方法和表参考方法。如果表参考方法被选择，执行程序可以切换至某个地址。这个地址是由异常处理表（内存中）所指示的

### CAUTION

The exception handler address of EIINT (INTn) selected using the direct vector method differs from that of the V850E2 core products. In the V850E2 core products, a different exception handler address is individually assigned to each interrupt channel (EIINT (INTn)). In the RH850/F1L, one exception handler address is assigned to each interrupt priority. Consequently, interrupts that have the same priority level branch to the same exception handler.

在RH850/F1L，一个异常处理地址被分配给每个中断优先级，结果，具有相同优先级的中断切换至相同的异常处理。

### 6.10.1 Direct Vector Method 直接向量方法

The CPU uses the result of adding the offset shown in **Table 6.21, Selection of Base Register/Offset Address** to the base address indicated by the RBASE or EBASE register as the exception handler address.

**Offset Address** CPU使用偏移地址+基地址来计算作为异常处理地址，基地址由EBASE核RBASE来指示

Select whether the RBASE or EBASE register is used as the base address by using the PSW.EBV bit<sup>\*1</sup>.  
当PSW.EBV == 1, 选择EBASE寄存器的值作为基地址  
When the PSW.EBV bit is set to 1, the value of the EBASE register is used as the base address. When  
当PSW.EBV == 0, 选择RBASE寄存器的值作为基地址  
the PSW.EBV bit is cleared to 0, the value of the RBASE register is used as the base address.

对于复位输入和一些异常，RBASE寄存器总是用作参考  
For reset input and some exceptions<sup>\*2</sup>, however, the RBASE register is always used for reference.

除此之外，用户中断查看对应基地址寄存器RINTbit，并且会根据其bit状态减小偏移地址  
In addition, user interrupts see the RINT bit of the corresponding base register, and reduce the offset  
address according to the bit status. If the RBASE.RINT bit or EBASE.RINT bit is set to 1, 所有的用户中断  
使用固定的偏移地址100H, 如果此bit被清零, 偏移地址根据表6.21中的来计算  
interrupts are handled using an offset of 100H. If the bit is cleared to 0, the offset address is determined  
according to **Table 6.21, Selection of Base Register/Offset Address**.

是不是就是INTn ( priority level 0 --- 7 ) ??

**Note 1.** Exception acknowledgment itself may sometimes update the status of the PSW.EBV bit. In this case, the base register is selected based on the updated value.

**Note 2.** The exceptions that always see the RBASE register are determined according to the hardware specifications.

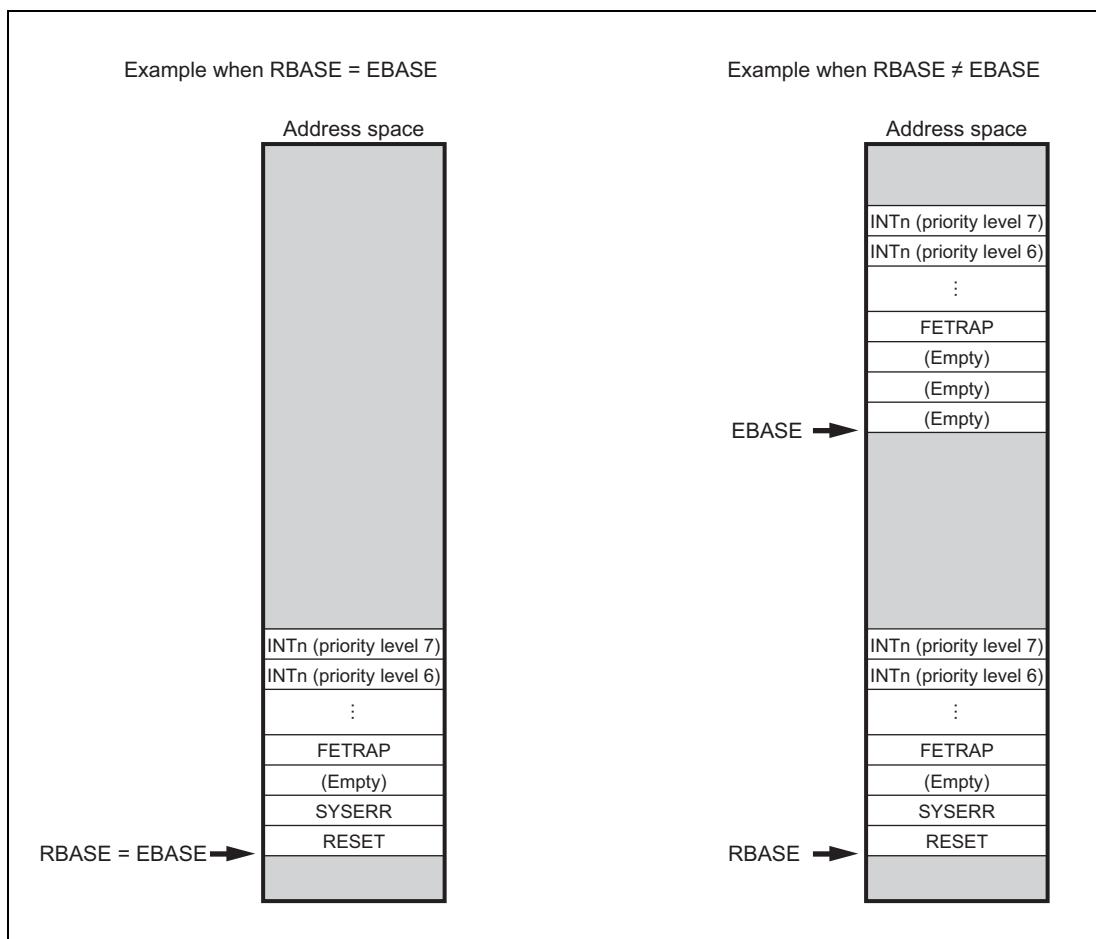


Figure 6.7 Direct Vector Method

The table below shows how base register selection and offset address reduction function for each exception to determine the exception handler address. The value of the PSW bit determines the exception handler address on the basis of the value updated by the acknowledgment of an exception.

**Table 6.21 Selection of Base Register/Offset Address**

Exception/Interrupt	PSW.EBV = 0	PSW.EBV = 1	RINT = 0	RINT = 1
	Base Register		Offset Address	
RESET	RBASE	N.A.	000 <sub>H</sub>	000 <sub>H</sub>
SYSERR			010 <sub>H</sub>	010 <sub>H</sub>
Reserved		EBASE	020 <sub>H</sub>	020 <sub>H</sub>
FETRAP			030 <sub>H</sub>	030 <sub>H</sub>
TRAP0			040 <sub>H</sub>	040 <sub>H</sub>
TRAP1			050 <sub>H</sub>	050 <sub>H</sub>
RIE			060 <sub>H</sub>	060 <sub>H</sub>
FPP/FPI			070 <sub>H</sub>	070 <sub>H</sub>
UCPOP			080 <sub>H</sub>	080 <sub>H</sub>
MIP/MDP			090 <sub>H</sub>	090 <sub>H</sub>
PIE			0A0 <sub>H</sub>	0A0 <sub>H</sub>
Debug			0B0 <sub>H</sub>	0B0 <sub>H</sub>
MAE			0C0 <sub>H</sub>	0C0 <sub>H</sub>
Reserved			0D0 <sub>H</sub>	0D0 <sub>H</sub>
FENMI			0E0 <sub>H</sub>	0E0 <sub>H</sub>
FEINT			0F0 <sub>H</sub>	0F0 <sub>H</sub>
INTn (Priority level 0)		100 <sub>H</sub>	100 <sub>H</sub>	100 <sub>H</sub>
INTn (Priority level 1)			110 <sub>H</sub>	
INTn (Priority level 2)			120 <sub>H</sub>	
INTn (Priority level 3)			130 <sub>H</sub>	
INTn (Priority level 4)			140 <sub>H</sub>	
INTn (Priority level 5)			150 <sub>H</sub>	
INTn (Priority level 6)			160 <sub>H</sub>	
INTn (Priority level 7)			170 <sub>H</sub>	

Base register selection is used to execute exception handling for resets and some hardware errors by using the programs in a relatively reliable area such as ROM instead of the areas that are easily affected by software errors such as RAM and cache area. The user interrupt offset address reduction function is used to reduce the memory occupation size required by the exception handler for specific system-internal operating modes. The main purpose of this is to minimize the amount of memory consumed in operating modes that use only the minimum functionality, for example, during system maintenance and diagnosis.

### 6.10.2 Table Reference Method 表向量方法

对直接向量方法，对一个中断优先级对应只有一个用

With the direct vector method, there is one user-interrupt exception handler for each interrupt priority level, and interrupt channels that indicate multiple interrupts with the same priority branch to the same interrupt handler, but some users might want to use different code areas for each interrupt handler from the beginning.

The RH850/F1L uses the table reference method for interrupts that assume the above usage.

如果表参考方法被指定作为中断channel 向量选择方法，

If the table reference method is specified as the interrupt channel vector selection method in the interrupt controller and the like, the method for determining the exception handler address when an interrupt request corresponding to that interrupt channel is acknowledged differs as follows.

<1> In any of the following cases, the exception handler address is determined by using the direct vector method:  
在下列情况下，使用直接向量方法来决定异常处理地址

- When PSW.EBV = 0 and RBASE.RINT = 1
- When PSW.EBV = 1 and EBASE.RINT = 1
- When the interrupt channel setting is not the table reference method

<2> In cases other than <1>, calculate the table reference position. 除了<1>之外，计算表参考位置

Exception handler address read position = INTBP register + channel number × 4 bytes

<3> Read word data starting at the interrupt handler address read position calculated in <2>.

<4> Use the word data read in <3> as the exception handler address.

**Table 6.22** shows the exception handler address read positions corresponding to each interrupt channel and **Figure 6.8** shows an overview of the allocation in memory.

**Table 6.22 Exception Handler Address Expansion**

Type of Interrupt	Exception Handler Address Read Position
EI level maskable interrupt channel 0	INTBP register value + 0 × 4
EI level maskable interrupt channel 1	INTBP register value + 1 × 4
EI level maskable interrupt channel 2	INTBP register value + 2 × 4
:	:
EI level maskable interrupt channel 286	INTBP register value + 286 × 4
EI level maskable interrupt channel 287	INTBP register value + 287 × 4

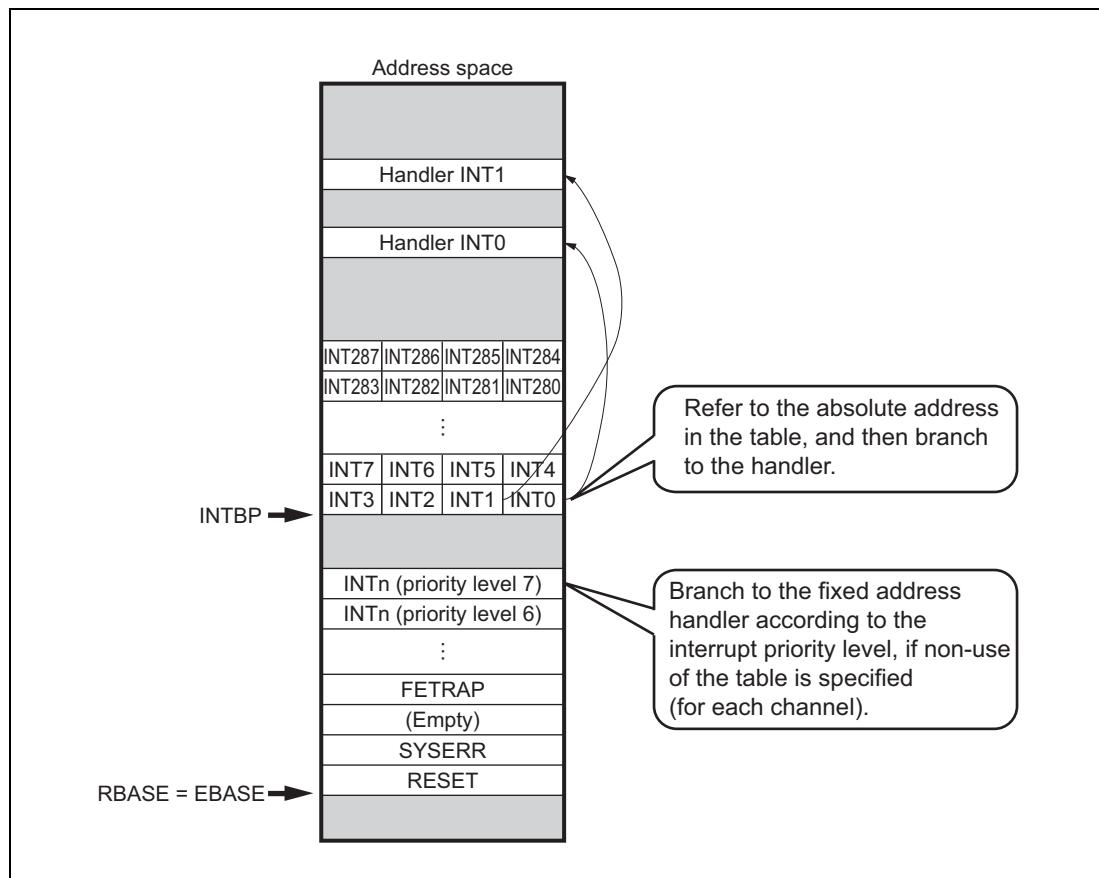


Figure 6.8 Table Reference Method

## Section 7 DMA

This section contains a generic description of the DMA controller (DMA).

The first part in this section describes the features specific to the RH850/F1L, including the number of channels and register base addresses. The ensuing sections describe the DMA functions and registers.

### 7.1 Features of RH850/F1L DMA

#### 7.1.1 Number of Channels

The products of the RH850/F1L series incorporate a DMA with the number of channels indicated below.

**Table 7.1 Number of Channels**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of channels	16 ch (8 ch + 8 ch)					

**Table 7.2 Index**

Index	Description
m	In this section, the DMA channels are identified with an index "m". For example, the DMA source address register is described as DSAm.

#### 7.1.2 Register Base Address

The DMA base address is indicated in the following table.

The DMA register addresses are expressed as an offset of the base address.

**Table 7.3 Register Base Address**

Base Address Name	Base Address
<DMA_base>	FFFF 8300 <sub>H</sub>

### 7.1.3 Interrupt Requests

The DMA interrupt requests are listed in the following table.

**Table 7.4 Interrupt Requests**

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
INTDMA0	DMA0 transfer completion	52	—
INTDMA1	DMA1 transfer completion	53	—
INTDMA2	DMA2 transfer completion	54	—
INTDMA3	DMA3 transfer completion	55	—
INTDMA4	DMA4 transfer completion	56	—
INTDMA5	DMA5 transfer completion	57	—
INTDMA6	DMA6 transfer completion	58	—
INTDMA7	DMA7 transfer completion	59	—
INTDMA8	DMA8 transfer completion	60	—
INTDMA9	DMA9 transfer completion	61	—
INTDMA10	DMA10 transfer completion	62	—
INTDMA11	DMA11 transfer completion	63	—
INTDMA12	DMA12 transfer completion	64	—
INTDMA13	DMA13 transfer completion	65	—
INTDMA14	DMA14 transfer completion	66	—
INTDMA15	DMA15 transfer completion	67	—

**Table 7.5 Interrupt Requests (System Error Interrupts)**

Unit Interrupt Name	Description	Interrupt Name	DMA Trigger Number
—	DMA transfer error occurrence	SYSERR	—

### 7.1.4 DMA Trigger Factors

DMA trigger factors can be selected by setting the DTFRm.DTFRmIFCm[5:0] bits.

The following table lists all DMA trigger factors which can be selected by the DTFRm register.

**Table 7.6 DMA Trigger Factors (m = 0 to 7) (1/2)**

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	DMA Trigger Factor	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
0	INTTAUD0I0	✓	✓	✓	✓	✓	✓
1	INTTAUD0I4	✓	✓	✓	✓	✓	✓
2	INTTAUD0I8	✓	✓	✓	✓	✓	✓
3	INTTAUD0I12	✓	✓	✓	✓	✓	✓
4	INTADCA0I0	✓	✓	✓	✓	✓	✓
5	INTADCA0I1	✓	✓	✓	✓	✓	✓
6	INTADCA0I2	✓	✓	✓	✓	✓	✓
7	ADC_CONV_END0	✓	✓	✓	✓	✓	✓
8	INTCSIG0IC	✓	✓	✓	✓	✓	✓
9	INTCSIG0IR	✓	✓	✓	✓	✓	✓
10	INTRLIN30UR0	✓	✓	✓	✓	✓	✓
11	INTRLIN30UR1	✓	✓	✓	✓	✓	✓
12	INTP0	✓	✓	✓	✓	✓	✓
13	INTP2	✓	✓	✓	✓	✓	✓
14	INTP4	✓	✓	✓	✓	✓	✓
15	INTTAUD0I1	✓	✓	✓	✓	✓	✓
16	INTTAUD0I5	✓	✓	✓	✓	✓	✓
17	INTTAUD0I9	✓	✓	✓	✓	✓	✓
18	INTTAUD0I13	✓	✓	✓	✓	✓	✓
19	INTRIIC0TI	✓	✓	✓	✓	✓	✓
20	INTRIIC0RI	✓	✓	✓	✓	✓	✓
21	INTTAUJ0I0	✓	✓	✓	✓	✓	✓
22	INTTAUJ0I3	✓	✓	✓	✓	✓	✓
23	Setting prohibited	—	—	—	—	—	—
24	Setting prohibited	—	—	—	—	—	—
25	Setting prohibited	—	—	—	—	—	—
26	Setting prohibited	—	—	—	—	—	—
27	Setting prohibited	—	—	—	—	—	—
28	INTCSIH1IC	—	—	✓	✓	✓	✓
29	INTCSIH1IR	—	—	✓	✓	✓	✓
30	INTCSIH1JC	—	—	✓	✓	✓	✓
31	INTP6	—	—	✓	✓	✓	✓
32	INTP8	—	—	✓	✓	✓	✓
33	INTTAUB0I0	—	—	—	✓	✓	✓
34	INTTAUB0I2	—	—	—	✓	✓	✓
35	INTTAUB0I4	—	—	—	✓	✓	✓
36	INTTAUB0I6	—	—	—	✓	✓	✓
37	INTTAUB0I9	—	—	—	✓	✓	✓
38	INTTAUB0I11	—	—	—	✓	✓	✓

Table 7.6 DMA Trigger Factors ( $m = 0$  to 7) (2/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	DMA Trigger Factor	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
39	INTTAUB0I13	—	—	—	✓	✓	✓
40	INTTAUB0I15	—	—	—	✓	✓	✓
41	INTCSIH3IC	—	—	—	✓	✓	✓
42	INTCSIH3IR	—	—	—	✓	✓	✓
43	INTCSIH3IJC	—	—	—	✓	✓	✓
44	INTRLIN32UR0	—	—	✓	✓	✓	✓
45	INTRLIN32UR1	—	—	✓	✓	✓	✓
46	INTTAUJ1I0	—	—	—	✓	✓	✓
47	INTTAUJ1I2	—	—	—	✓	✓	✓
48	Setting prohibited	—	—	—	—	—	—
49	Setting prohibited	—	—	—	—	—	—
50	INTRLIN34UR0	—	—	—	—	✓	✓
51	INTRLIN34UR1	—	—	—	—	✓	✓
52	INTTAUB1I0	—	—	—	—	—	✓
53	INTTAUB1I2	—	—	—	—	—	✓
54	INTTAUB1I4	—	—	—	—	—	✓
55	INTTAUB1I6	—	—	—	—	—	✓
56	INTTAUB1I9	—	—	—	—	—	✓
57	INTTAUB1I11	—	—	—	—	—	✓
58	INTTAUB1I13	—	—	—	—	—	✓
59	INTTAUB1I15	—	—	—	—	—	✓
60	Setting prohibited	—	—	—	—	—	—
61	Setting prohibited	—	—	—	—	—	—
62	Setting prohibited	—	—	—	—	—	—
63	Setting prohibited	—	—	—	—	—	—

Table 7.7 DMA Trigger Factors (m = 8 to 15) (1/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	DMA Trigger Factor	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
0	INTTAUD0I2	✓	✓	✓	✓	✓	✓
1	INTTAUD0I6	✓	✓	✓	✓	✓	✓
2	INTTAUD0I10	✓	✓	✓	✓	✓	✓
3	INTTAUD0I14	✓	✓	✓	✓	✓	✓
4	Setting prohibited	—	—	—	—	—	—
5	Setting prohibited	—	—	—	—	—	—
6	INTCSIH0IC	✓	✓	✓	✓	✓	✓
7	INTCSIH0IR	✓	✓	✓	✓	✓	✓
8	INTCSIH0IJC	✓	✓	✓	✓	✓	✓
9	INTP1	✓	✓	✓	✓	✓	✓
10	INTP3	✓	✓	✓	✓	✓	✓
11	INTP5	✓	✓	✓	✓	✓	✓
12	INTTAUD0I3	✓	✓	✓	✓	✓	✓
13	INTTAUD0I7	✓	✓	✓	✓	✓	✓
14	INTTAUD0I11	✓	✓	✓	✓	✓	✓
15	INTTAUD0I15	✓	✓	✓	✓	✓	✓
16	INTTAUJ0I1	✓	✓	✓	✓	✓	✓
17	INTTAUJ0I2	✓	✓	✓	✓	✓	✓
18	Setting prohibited	—	—	—	—	—	—
19	Setting prohibited	—	—	—	—	—	—
20	Setting prohibited	—	—	—	—	—	—
21	INTDMAFL <sup>*1</sup>	✓	✓	✓	✓	✓	✓
22	INTRLIN31UR0	—	✓	✓	✓	✓	✓
23	INTRLIN31UR1	—	✓	✓	✓	✓	✓
24	INTP7	—	—	✓	✓	✓	✓
25	INTCSIH2IC	—	—	✓	✓	✓	✓
26	INTCSIH2IR	—	—	✓	✓	✓	✓
27	INTCSIH2IJC	—	—	✓	✓	✓	✓
28	INTTAUB0I1	—	—	—	✓	✓	✓
29	INTTAUB0I3	—	—	—	✓	✓	✓
30	INTTAUB0I5	—	—	—	✓	✓	✓
31	INTTAUB0I7	—	—	—	✓	✓	✓
32	INTTAUB0I8	—	—	—	✓	✓	✓
33	INTTAUB0I10	—	—	—	✓	✓	✓
34	INTTAUB0I12	—	—	—	✓	✓	✓
35	INTTAUB0I14	—	—	—	✓	✓	✓
36	INTTAUJ1I1	—	—	—	✓	✓	✓
37	INTTAUJ1I3	—	—	—	✓	✓	✓
38	INTP9	—	—	—	—	✓	✓
39	INTADCA1I0	—	—	—	—	✓	✓
40	INTADCA1I1	—	—	—	—	✓	✓
41	INTADCA1I2	—	—	—	—	✓	✓
42	ADC_CONV_END1	—	—	—	—	✓	✓

Table 7.7 DMA Trigger Factors ( $m = 8$  to  $15$ ) (2/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	DMA Trigger Factor	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
43	Setting prohibited	—	—	—	—	—	—
44	Setting prohibited	—	—	—	—	—	—
45	INTCSIG1IC	—	—	—	—	✓	✓
46	INTCSIG1IR	—	—	—	—	✓	✓
47	INTRLIN33UR0	—	—	—	✓	✓	✓
48	INTRLIN33UR1	—	—	—	✓	✓	✓
49	INTRLIN35UR0	—	—	—	—	✓	✓
50	INTRLIN35UR1	—	—	—	—	✓	✓
51	INTTAUB1I1	—	—	—	—	—	✓
52	INTTAUB1I3	—	—	—	—	—	✓
53	INTTAUB1I5	—	—	—	—	—	✓
54	INTTAUB1I7	—	—	—	—	—	✓
55	INTTAUB1I8	—	—	—	—	—	✓
56	INTTAUB1I10	—	—	—	—	—	✓
57	INTTAUB1I12	—	—	—	—	—	✓
58	INTTAUB1I14	—	—	—	—	—	✓
59	Setting prohibited	—	—	—	—	—	—
60	Setting prohibited	—	—	—	—	—	—
61	Setting prohibited	—	—	—	—	—	—
62	Setting prohibited	—	—	—	—	—	—
63	Setting prohibited	—	—	—	—	—	—

Note 1. For details, see the *RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface*.

## 7.2 Overview

### 7.2.1 Functional Overview

#### 7.2.1.1 DMA Transfer Function

##### **DMA controller (DMAC) function**

Registers to store transfer information (transfer address, transfer size, etc.) and registers to control DMAC are included.

##### **Number of channels**

2 units × 8 channels (Total: 16 channels)

##### **Transfer data size**

8 bits, 16 bits, 32 bits

##### **Transfer data**

- Little endian
- Misaligned data not supported

##### **Maximum transfer count**

32768 ( $2^{15}$ ) times

##### **Channel priority control**

Fixed priority (highest priority (CH0) → lowest priority (CH15))

##### **Transfer target**

The relationship between the source and destination of transfer is shown below.

**Table 7.8 Relationship between the Source and Destination of Transfer**

		Destination of Transfer				
		Peripheral I/O	Primary Local RAM, Secondary Local RAM, Retention RAM*2,*3	Code Flash	Data Flash	External Memory*1
Source of Transfer	Peripheral I/O	√	√	—	—	√
	Primary Local RAM, Secondary Local RAM, Retention RAM	√	√	—	—	√
	Code flash	√	√	—	—	√
	Data flash	√	√	—	—	√
	External memory	√	√	—	—	√

Note 1. The external memory is supported only in 176 pin devices.

Note 2. The primary local RAM is not supported in 256-Kbyte products.

Note 3. Only 2-Mbyte products have secondary local RAM.

### Transfer type

2-cycle transfer (dual address transfer)

The address of both the transfer source and destination is accessed. Two bus cycles are required to execute one transfer (read cycle + write cycle). Because the bus is not locked between the read cycle and write cycle, a CPU cycle may occur between the read and write access.

### Transfer mode

- Single transfer mode (when hardware DMA transfer request is generated)

When a hardware DMA transfer request is generated, the bus mastership is acquired, and the bus is always released after transfer has been executed once. If another hardware DMA transfer request is generated after that, transfer is executed once again. This operation is repeated until transfer has been executed the number of times specified by the transfer count register (DTCm).

- Single-step transfer mode (when software DMA transfer request is generated)

When a software DMA transfer request is generated, the bus mastership is acquired, and the bus is released each time transfer has been executed once. Once the software DMA transfer request has been acknowledged, this operation is repeated until transfer has been executed the number of times specified by the transfer count register (DTCm).

### Transfer address control

The address of the transfer source and destination can be incremented or fixed per channel.

### Transfer error support

If any of the errors below occur during access for DMA transfer, DMA transfer is suspended and a SYSERR exception is output to the CPU.

- An 2-bit ECC error is generated in the primary local RAM, secondary local RAM, or retention RAM.
- An 2-bit ECC error is generated in the code flash.
- A peripheral function register not being supplied with a clock signal is accessed.

### DMA transfer request

Any hardware or software transfer request can be selected for each channel (**Figure 7.5, Configuration Diagram of Transfer Requests of Channels**).

The trigger factor of hardware DMA transfer is selected by the DTFRm register. A status bit (the DTSm.DTSmDR bit) is also available to indicate whether a hardware transfer request has been issued.

Software DMA transfer requests can be selected by setting the DTSm.DTSmSR bit to 1.

### Transfer completion interrupt output function

This function outputs a transfer completion interrupt request signal (INTDMA15 to INTDMA0) when DMA transfer of each channel has been completed the number of times specified by the transfer count register (DTCm).

### Stand-by support

DMA transfer is suspended momentarily by shifting to STOP mode.

### DMA transfer suspend function

This function supports suspending DMA transfer by software.

### Protection function

This function enables protection against writing to the DMA control register by setting an access master.

### Loop function

At the completion of DMA transfer, this function controls the acceptance of the request for the next DMA transfer.

#### 7.2.1.2 DTFR Function

The DMA trigger factor register (DTFR) selects DMA trigger factors from among interrupt request signals, and enables DMA transfer requests to be issued to DMAC. The signals to be used for DMA transfer requests can be selected from among the 128 ( $64 \times 2$ ) interrupt request input signals by using the DTFRm ( $m = 15$  to 0) registers.

##### Number of trigger factors

DMA transfer requests (for 16 channels) are selected from among 128 ( $64 \times 2$ ) interrupt request signals.

##### DMAC interface

The DMA transfer request signal m ( $m = 15$  to 0) is output.

The DMA transfer request signal m is cleared by an acknowledge signal from DMA.

##### CPU interface

The last transfer signal from DMA is output as a CPU interrupt request signal.

##### Clearing of transfer request

Transfer request signal m sent to DMA can be cleared by register access.

##### Confirmation of transfer request

The status of transfer request signal m sent to DMA can be checked by register access.

#### 7.2.1.3 DMA Access Memory Map

See [Section 3.2.4, RH850/F1L Memory Map](#) for details.

#### 7.2.1.4 Prioritization of Channels

When multiple DMA triggers occur simultaneously in DMAC, bus access is determined by the following priority.

CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15 (CH0 has the highest priority)

## 7.2.2 Terms

The terms used in this section are defined as follows.

**Table 7.9 Definition of Terms**

Term	Function
DMA transfer	Period from the start of the first DMA cycle to assertion of INTDMAm
DMA cycle	Period for transferring one unit of data (2-cycle transfer: a read cycle and write cycle are generated on the DMA bus.)
Hardware DMA transfer request	DMA transfer request by the DMA trigger factor selected in the DTFRm register (It is possible to check whether there is any request by reading the DTSm.DTSmDR bit.)
Software DMA transfer request	DMA transfer request by internal register (setting 1 to the DTSm.DTSmSR bit)
DMA transfer request	Hardware DMA transfer request and software DMA transfer request
Read cycle	DMA cycle for reading data
Write cycle	DMA cycle for writing data
Single transfer	The DMAC executes one DMA cycle per transfer request. INTDMAm is generated on completion of the number of transfers specified in the DMA transfer count register m (DTCm). (A single transfer can be executed only by a hardware DMA transfer request.)
Single step transfer	The number of transfers set in the transfer count setting register (DTCm) is executed per software DMA transfer request. Since the bus is released after each DMA cycle (after a read cycle or write cycle), a CPU access may occur between DMA cycles. If a higher-priority transfer request occurs during execution of a single step transfer, the single step transfer is suspended while the higher-priority transfer request is executed. (A single-step transfer can be executed only by a software DMA transfer request.)

### 7.2.3 Block Diagram

The following figure shows the main components of the DMA.

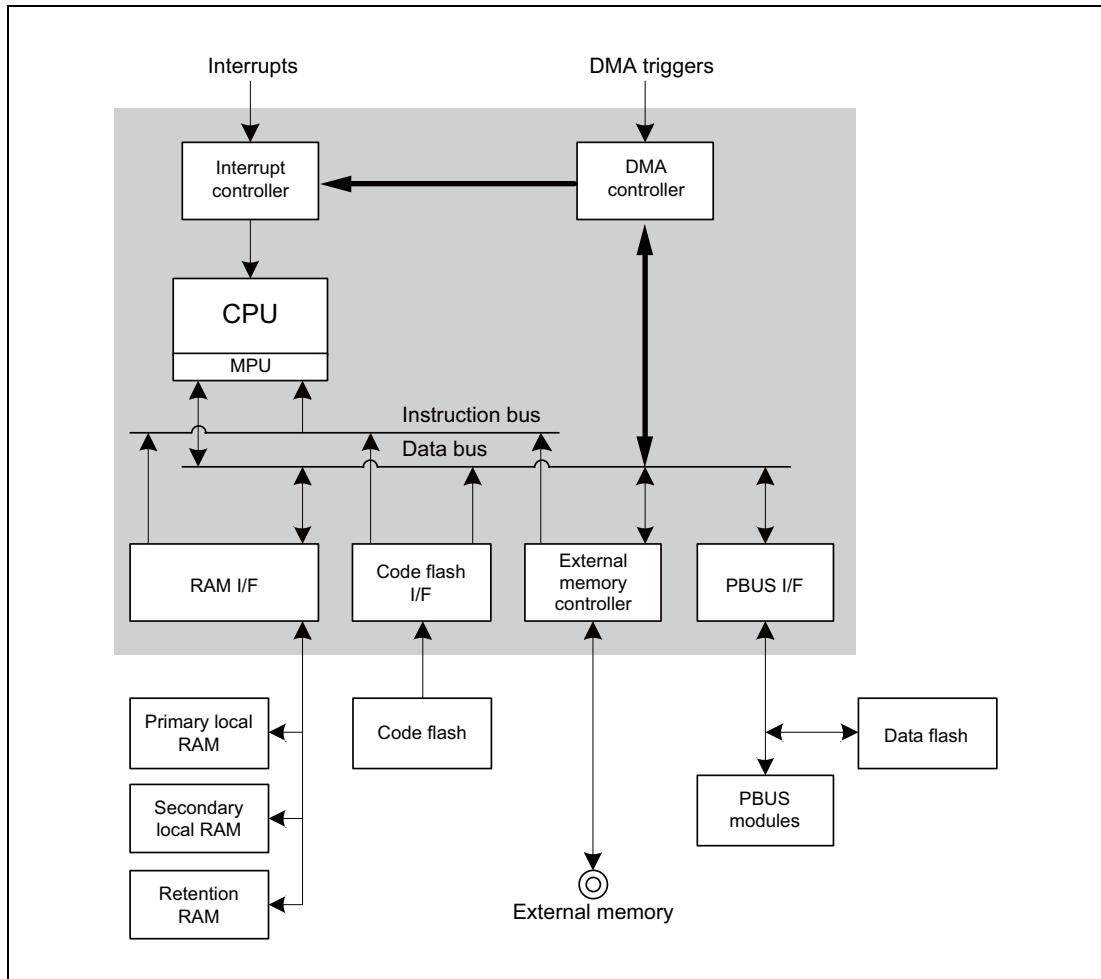


Figure 7.1 Block Diagram of the DMA

## 7.3 Registers

### 7.3.1 List of Registers

The following table lists the DMA registers.

For details about <DMA\_base>, see **Section 7.1.2, Register Base Address**.

Table 7.10 List of Registers (1/2)

Module Name	Register Name	Symbol	Address
DMA	<b>Channel common registers</b>		
	DMA transfer request control register 0	DTRC0	<DMA_base> + 00H
	DMA channel master setting register 0	DMCM0	<DMA_base> + 04H
	<b>Channel registers (m = 0 to 7)</b>		
	DMA source address register CHm	DSAm	<DMA_base> + (m × 30H) + 14H
	DMA source address register L CHm	DSAmL	<DMA_base> + (m × 30H) + 14H
	DMA source address register H CHm	DSAmH	<DMA_base> + (m × 30H) + 16H
	DMA destination address register CHm	DDAm	<DMA_base> + (m × 30H) + 24H
	DMA destination address register L CHm	DDAmL	<DMA_base> + (m × 30H) + 24H
	DMA destination address register H CHm	DDAmH	<DMA_base> + (m × 30H) + 26H
	DMA transfer count register CHm	DTCm	<DMA_base> + (m × 30H) + 32H
	DMA transfer control register CHm	DTCTm	<DMA_base> + (m × 30H) + 38H
	DMA transfer status register CHm	DTSm	<DMA_base> + (m × 30H) + 3AH
	<b>Channel registers (m = 8 to 15)</b>		
	DMA source address register CHm	DSAm	<DMA_base> + (m × 30H) + 94H
	DMA source address register L CHm	DSAmL	<DMA_base> + (m × 30H) + 94H
	DMA source address register H CHm	DSAmH	<DMA_base> + (m × 30H) + 96H
	DMA destination address register CHm	DDAm	<DMA_base> + (m × 30H) + A4H
	DMA destination address register L CHm	DDAmL	<DMA_base> + (m × 30H) + A4H
	DMA destination address register H CHm	DDAmH	<DMA_base> + (m × 30H) + A6H
	DMA transfer count register CHm	DTCm	<DMA_base> + (m × 30H) + B2H
	DMA transfer control register CHm	DTCTm	<DMA_base> + (m × 30H) + B8H
	DMA transfer status register CHm	DTSm	<DMA_base> + (m × 30H) + BAH
	<b>Trigger factor registers</b>		
	DMA trigger factor register 0	DTFR0	<DMA_base> + 800H
	DMA trigger factor register 1	DTFR1	<DMA_base> + 802H
	DMA trigger factor register 2	DTFR2	<DMA_base> + 804H
	DMA trigger factor register 3	DTFR3	<DMA_base> + 806H
	DMA trigger factor register 4	DTFR4	<DMA_base> + 808H
	DMA trigger factor register 5	DTFR5	<DMA_base> + 80AH
	DMA trigger factor register 6	DTFR6	<DMA_base> + 80CH
	DMA trigger factor register 7	DTFR7	<DMA_base> + 80EH
	DMA trigger factor register 8	DTFR8	<DMA_base> + 810H
	DMA trigger factor register 9	DTFR9	<DMA_base> + 812H
	DMA trigger factor register 10	DTFR10	<DMA_base> + 814H
	DMA trigger factor register 11	DTFR11	<DMA_base> + 816H
	DMA trigger factor register 12	DTFR12	<DMA_base> + 818H

**Table 7.10 List of Registers (2/2)**

<b>Module Name</b>	<b>Register Name</b>	<b>Symbol</b>	<b>Address</b>
DMA	DMA trigger factor register 13	DTFR13	<DMA_base> + 81A <sub>H</sub>
	DMA trigger factor register 14	DTFR14	<DMA_base> + 81C <sub>H</sub>
	DMA trigger factor register 15	DTFR15	<DMA_base> + 81E <sub>H</sub>
	DMA request clear register	DRQCLR	<DMA_base> + 840 <sub>H</sub>
	DMA request check register	DRQSTR	<DMA_base> + 844 <sub>H</sub>

### 7.3.2 Enabling or Disabling Writing Control Registers

The following control registers cannot be written while DMA transfer is enabled (DTSm.DTSmDTE = 1). All these registers can always be read, however.

**Table 7.11 Enabling/Disabling Writing Control Registers**

Always writable	DTRC0, DTSm, DRQCLR, DTCTm.DTCTmLE
Writing prohibited while DMA transfer is enabled (DTSm.DTSmDTE = 1) (Operation is not guaranteed if these registers are written.)	DSAm, DSAmL, DSAmH, DDAm, DDAmL, DDAmH, DTClm, DTCTm (excluding the DTCTmLE bit), DTFRm

**Note:** m = 0 to 15

### 7.3.3 DMAC Control Registers

#### 7.3.3.1 DTRC0 — DMA Transfer Request Control Register 0

**Access:** This register can be read or written in 8- or 1-bit units.

**Address:** <DMA\_base> + 00H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	DTRC0ERR	—	—	—	—	—	—	DTRC0ADS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W

Table 7.12 DTRC0 Registers Contents

Bit Position	Bit Name	Function
7	DTRC0ERR	DMA transfer error status This bit indicates that an error response has been received from the transfer target during DMA transfer. If an error response is received, the DTRC0ERR and DTRC0ADS bits are set and a SYSERR exception is generated. To clear this bit, write "0" to it. 0: No DMA transfer error 1: DMA transfer error
6 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DTRC0ADS	DMA transfer suspended This bit indicates that DMA transfer has been suspended by the generation of DMA transfer error. In addition, the current DMA transfer can be suspended by writing "1" to this bit. 0: DMA transfer not suspended 1: DMA transfer suspended/DMA transfer suspended request

#### 7.3.3.2 DMCM0 — DMA Channel Master Setting Register 0

**Access:** This register can be read or written in 16-bit units.

**Address:** <DMA\_base> + 04H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMCM0UM15	DMCM0UM14	DMCM0UM13	DMCM0UM12	DMCM0UM11	DMCM0UM10	DMCM0UM9	DMCM0UM8	DMCM0UM7	DMCM0UM6	DMCM0UM5	DMCM0UM4	DMCM0UM3	DMCM0UM2	DMCM0UM1	DMCM0UM0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.13 DMCM0 Registers Contents

Bit Position	Bit Name	Function
15 to 0	DMCM0UM15 to DMCM0UM0	These bits set access masters to be assigned to each channel. 0: Supervisor mode is assigned as the access master 1: User mode is assigned as the access master
<b>CAUTION</b>		
The register of a channel for which the access master is set to user mode is accessible in supervisor mode, but if the register of a channel for which the access master is set to supervisor mode is accessed in user mode, an access error will occur.		

### 7.3.3.3 DSAm — DMA Source Address Register

**Access:** DSAm can be read or written in 32-bit units.  
DSAmL and DSAmH can be read or written in 16-bit units.

**Address:**

- When m is 0 to 7:  
DSAm: <DMA\_base> + (m × 30<sub>H</sub>) + 14<sub>H</sub>  
DSAmL: <DMA\_base> + (m × 30<sub>H</sub>) + 14<sub>H</sub>  
DSAmH: <DMA\_base> + (m × 30<sub>H</sub>) + 16<sub>H</sub>
- When m is 8 to 15:  
DSAm: <DMA\_base> + (m × 30<sub>H</sub>) + 94<sub>H</sub>  
DSAmL: <DMA\_base> + (m × 30<sub>H</sub>) + 94<sub>H</sub>  
DSAmH: <DMA\_base> + (m × 30<sub>H</sub>) + 96<sub>H</sub>

**Value after reset:** 0XXX XXXX<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSAmH SA26	DSAmH SA25	DSAmH SA24	DSAmH SA23	DSAmH SA22	DSAmH SA21	DSAmH SA20	DSAmH SA19	DSAmH SA18	DSAmH SA17	DSAmH SA16
Value after reset	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R/W										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSAmL SA15	DSAmL SA14	DSAmL SA13	DSAmL SA12	DSAmL SA11	DSAmL SA10	DSAmL SA9	DSAmL SA8	DSAmL SA7	DSAmL SA6	DSAmL SA5	DSAmL SA4	DSAmL SA3	DSAmL SA2	DSAmL SA1	DSAmL SA0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.14 DSAm Registers Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	DSAmHSA26 to DSAmHSA16	DMA source address These bits set bits 26 to 16 of the transfer source address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DSAmL and DSAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.
15 to 0	DSAmLSA15 to DSAmLSA0	DMA source address These bits set bits 15 to 0 of the transfer source address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DSAmL and DSAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

#### CAUTIONS

- Writing this register is prohibited while DMA transfer is enabled (DTSm.DTSMDE = 1). If it is written, the operation is not guaranteed.
- Set an address by accessing in 32-bit units while the DTSMDE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
- If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the source address is updated.
- DMA transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (— indicates any bit).  
The operation is not guaranteed if a setting other than the following is made.

Data Size	DSAm LSA1	DSAm LSA0
8 bits	—	—
16 bits	—	0
32 bits	0	0

5. When the DSAm register is read in 32-bit units while DMA transfer is enabled (DTSm.DTSmDTE = 1), the read value may not be correct. Whether the read value is correct or not can be judged using the following method.  
(Method of judging)
  - Read DSAm register in 32-bit units twice in succession  
If the upper 16 bits of the first and the second read values are the same, the second read value is correct.  
If the upper 16 bits of the first and the second read values are different, the first read value is correct.
6. The DMA access source set by DSAm is an address where bit 26 is sign-extended for the higher-order 5 bits of DSAmH. However, when reading the DSAmH register, "0" is read from the higher-order 5 bits.

### 7.3.3.4 DDAm — DMA Destination Address Register

**Access:** DDAm can be read or written in 32-bit units.  
DDAmL and DDAmH can be read or written in 16-bit units.

**Address:**

- When m is 0 to 7:  
DDAm: <DMA\_base> + (m × 30<sub>H</sub>) + 24<sub>H</sub>  
DDAmL: <DMA\_base> + (m × 30<sub>H</sub>) + 24<sub>H</sub>  
DDAmH: <DMA\_base> + (m × 30<sub>H</sub>) + 26<sub>H</sub>
- When m is 8 to 15:  
DDAm: <DMA\_base> + (m × 30<sub>H</sub>) + A4<sub>H</sub>  
DDAmL: <DMA\_base> + (m × 30<sub>H</sub>) + A4<sub>H</sub>  
DDAmH: <DMA\_base> + (m × 30<sub>H</sub>) + A6<sub>H</sub>

**Value after reset:** 0XXX XXXX<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DDAmH DA26	DDAmH DA25	DDAmH DA24	DDAmH DA23	DDAmH DA22	DDAmH DA21	DDAmH DA20	DDAmH DA19	DDAmH DA18	DDAmH DA17	DDAmH DA16
Value after reset	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R/W										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDAmL DA15	DDAmL DA14	DDAmL DA13	DDAmL DA12	DDAmL DA11	DDAmL DA10	DDAmL DA9	DDAmL DA8	DDAmL DA7	DDAmL DA6	DDAmL DA5	DDAmL DA4	DDAmL DA3	DDAmL DA2	DDAmL DA1	DDAmL DA0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.15 DDAm Registers Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	DDAmHDA26 to DDAmHDA16	DMA destination address These bits set bits 26 to 16 of the transfer destination address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DDAmL and DDAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.
15 to 0	DDAmLDA15 to DDAmLDA0	DMA destination address These bits set bits 15 to 0 bits of the transfer destination address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DDAmL and DDAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

#### CAUTIONS

- Writing these bits is prohibited while DMA transfer is enabled (DTSm.DTSmDTE = 1). If they are written, the operation is not guaranteed.
- Set an address by accessing in 32-bit units while the DTsmDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
- If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.
- DMA transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (— indicates any bit).  
The operation is not guaranteed if a setting other than the following is made.

Data Size	DDAm LDA1	DDAm LDA0
8 bits	—	—
16 bits	—	0
32 bits	0	0

5. When the DDAm register is read in 32-bit units while DMA transfer is enabled (DTSm.DTSmDTE = 1), the read value may not be correct. Whether the read value is correct or not can be judged using the following method.  
(Method of judging)
  - Read DDAm register in 32-bit units twice in succession  
If the upper 16 bits of the first and the second read values are the same, the second read value is correct.  
If the upper 16 bits of the first and the second read values are different, the first read value is correct.
6. The DMA access destination set by DDAm is an address where bit 26 is sign-extended for the higher-order 5 bits of DDAmH. However, when reading the DDAmH register, "0" is read from the higher-order 5 bits.

### 7.3.3.5 DTCm — DMA Transfer Count Register (m = 0 to 15)

**Access:** This register can be read or written in 16-bit units.

**Address:**

- When m is 0 to 7:  
DTCm: <DMA\_base> + (m × 30<sub>H</sub>) + 32<sub>H</sub>
- When m is 8 to 15:  
DTCm: <DMA\_base> + (m × 30<sub>H</sub>) + B2<sub>H</sub>

**Value after reset:** Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DTCm DTC14	DTCm DTC13	DTCm DTC12	DTCm DTC11	DTCm DTC10	DTCm DTC9	DTCm DTC8	DTCm DTC7	DTCm DTC6	DTCm DTC5	DTCm DTC4	DTCm DTC3	DTCm DTC2	DTCm DTC1	DTCm DTC0
Value after reset	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.16 DTCm Registers Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 0	DTCmDTC14 to DTCmDTC0	DMA transfer count These bits specify the number of DMA transfers (DMA transfer count) for channel m. When this register is referenced during DMA transfer, the remaining number of DMA transfers to be executed can be read. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.
DTCmDTC[14:0]		Operation
0000 <sub>H</sub>		Transfer executed 32,768 times
0001 <sub>H</sub>		Transfer executed once or transfer to be executed once
:		:
7FFF <sub>H</sub>		Transfer executed 32,767 times or transfer to be executed 32,767 times

#### CAUTIONS

1. Writing these bits is prohibited while DMA transfer is enabled (DTSm.DTSmDTE bit = 1). If they are written, the operation is not guaranteed.
2. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the transfer count is updated.

### 7.3.3.6 DTCTm — DMA Transfer Control Register (m = 0 to 15)

**Access:** This register can be read or written in 16-bit units.

- Address:**
- When m is 0 to 7:  
DTCTm: <DMA\_base> + (m × 30<sub>H</sub>) + 38<sub>H</sub>
  - When m is 8 to 15:  
DTCTm: <DMA\_base> + (m × 30<sub>H</sub>) + B8<sub>H</sub>

**Value after reset:** x000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	DTCTm DS1	DTCTm DS0	DTCTm LE	—	—	—	—	DTCTm SACM1	—	DTCTm DPCM1	—	—	—	—	—
Value after reset	0	—	—	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W	R	R	R	R	R

Table 7.17 DTCTm Registers Contents

Bit Position	Bit Name	Function															
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
14, 13	DTCTmDS1 DTCTmDS0	DMA transfer data size These bits specify the DMA transfer data size of channel m.															
		<table border="1"> <thead> <tr> <th>DTCTmDS1</th> <th>DTCTmDS0</th> <th>DMA Transfer Data Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>8 bits</td> </tr> <tr> <td>0</td> <td>1</td> <td>16 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>32 bits</td> </tr> <tr> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> </tbody> </table>	DTCTmDS1	DTCTmDS0	DMA Transfer Data Size	0	0	8 bits	0	1	16 bits	1	0	32 bits	1	1	Setting prohibited
DTCTmDS1	DTCTmDS0	DMA Transfer Data Size															
0	0	8 bits															
0	1	16 bits															
1	0	32 bits															
1	1	Setting prohibited															
12	DTCTmLE	Loop enable This bit enables the repeat function (loop function) of DMA transfer. This bit specifies whether to acknowledge the next DMA transfer request, even if the DTSm.DTSmTC bit is not cleared (to "0") after DMA transfer has been completed in single transfer mode. Do not use this bit in single-step transfer mode. If this bit is set (to "1"), the DTSm.DTSmDTE bit is not cleared upon completion of DMA transfer. Even if the DTSm.DTSmTC bit is not cleared, DMA transfer is executed if a DMA transfer request is issued. This bit can be written while the DTSm.DTSmDTE bit is set (to "1"). 0: Clears the DTSm.DTSmDTE bit upon completion of DMA transfer. 1: Does not clear the DTSm.DTSmDTE bit upon completion of DMA transfer.															
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
7	DTCTmSACM1	DMA transfer source address counting direction This bit specifies the direction in which the transfer source address of channel m is to be counted. 0: Increment 1: Fixed															
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
5	DTCTmDPCM1	DMA transfer destination address counting direction This bit specifies the direction in which the transfer destination address of channel m is to be counted. 0: Increment 1: Fixed															
4 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															

**CAUTION**

Writing to this register while DMA transfer is enabled (DTSm.DTSmDTE bit = 1) only affects the DTCTmLE bit (only the DTCTmLE bit is updated). The other bits are not changed.

### 7.3.3.7 DTSm — DMA Transfer Status Register (m = 0 to 15)

**Access:** This register can be read or written in 8- or 1-bit units.

- Address:**
- When m is 0 to 7:  
DTSm: <DMA\_base> + (m × 30<sub>H</sub>) + 3A<sub>H</sub>
  - When m is 8 to 15:  
DTSm: <DMA\_base> + (m × 30<sub>H</sub>) + BA<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	DTSmTC	DTSmDT	—	—	DTSmER	DTSmDR	DTSmSR	DTSmDTE
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

**Table 7.18 DTSm Registers Contents (1/2)**

Bit Position	Bit Name	Function
7	DTSmTC	DMA transfer end status This bit indicates that DMA transfer has been completed. This bit should be cleared to acknowledge the next DMA transfer request. Write “0” to this bit to clear it after reading “1” from it. It is recommended to write this bit using bit manipulation such as CLR1. Note that, when bit manipulation (SET1, CLR1, NOT1, or TST1) is performed on the other bits of the DTS register while the DTSmTC bit is set, it has the same effect as reading “1” from this bit. 0: DMA transfer not completed 1: DMA transfer completed
6	DTSmDT	DMA transfer status This bit indicates that a DMA transfer request has been acknowledged and that DMA transfer is in progress. It is not set (to “1”) when a DMA transfer request is issued, but rather when it is acknowledged. This bit is cleared (to “0”) when DMA transfer has been completed. If the DTSmDTE bit is “0”, this bit can be cleared by the user. (It can also be written at the same time as the DTSmDTE bit.) After the clearance operation, when the ongoing DMA cycle has been completed, this bit is updated. 0: DMA transfer is not executed 1: DMA transfer in progress
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	DTSmER	DMA transfer error flag This bit indicates that a DMA transfer error has occurred in channel m. It is cleared (to “0”) when the DTRC0ERR bit is cleared. Note that this bit is read only. 0: No DMA transfer error 1: DMA transfer error
2	DTSmDR	Hardware DMA transfer request flag This bit indicates that channel m has a hardware DMA transfer request. It is cleared (to “0”) when the hardware DMA transfer request is deasserted. This bit operates regardless of the status of the DTSmDTE bit. It is not set (to “1”) by a software DMA transfer request. Note that this bit is read only. 0: No hardware DMA transfer request 1: Hardware DMA transfer request

**Table 7.18 DTSm Registers Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
1	DTSmSR	<p>Software DMA transfer request This bit sets a software DMA transfer request. If this bit is set to “1” while the DTSmDTE bit is set to “1”, DMA transfer is executed. This bit is automatically cleared (to “0”) when DMA transfer has been completed. Writing “0” to this bit during single-step transfer issued by this bit suspends DMA transfer.</p> <p>0: No software DMA transfer request 1: Software DMA transfer request</p>
0	DTSmDTE	<p>DMA transfer enable This bit enables or disables DMA transfer. DMA transfer is executed if “1” is written to this bit and a DMA transfer request is issued. This bit is automatically cleared (to “0”) if the DTCTmLE bit is “0” when DMA transfer has been completed. DMA transfer is suspended if “0” is written to this bit during DMA transfer.</p> <p>0: Disables DMA transfer 1: Enables DMA transfer</p>

## 7.3.4 DTFR Control Registers

### 7.3.4.1 DTFRm — DMA Trigger Factor Register m (m = 0 to 15)

**Access:** This register can be read or written in 16-bit units.

**Address:** DTFRm: <DMA\_base> + 800<sub>H</sub> + (m × 2)

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTFRm REQEN	—	—	—	—	—	—	—	—	—	—	DTFRm IFCm5	DTFRm IFCm4	DTFRm IFCm3	DTFRm IFCm2	DTFRm IFCm1	DTFRm IFCm0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.19 DTFRm Registers Contents

Bit Position	Bit Name	Function
15	DTFRmREQEN	This bit enables or disables the generation of the DMA trigger factor. 1: Enables the generation of the DMA trigger factor. 0: Disables the generation of the DMA trigger factor.
14 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	DTFRmIFCm5 to DTFRmIFCm0	These bits select the trigger factor. The set values are shown in <b>Table 7.6</b> and <b>Table 7.7</b> .

#### NOTE

The selected trigger factor is always sampled. When a trigger factor is generated, the DRQSTRRQST<sub>m</sub> bit in the DRQSTR register is always set.

#### CAUTION

Writing these bits is prohibited while DMA transfer is enabled (DTSm.DTSmDTE bit = 1). Otherwise the operation is not guaranteed.

### 7.3.4.2 DRQCLR — DMA Request Clear Register

**Access:** This register can be read or written in 16-bit units.

**Address:** <DMA\_base> + 840H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRQCL RRQCR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R/W															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.20 DRQCLR Register Contents

Bit Position	Bit Name	Function
15 to 0	DRQCLRRQCR	Set DRQCLRRQCRm to 1 to clear the corresponding transfer request held in channel m.
15 to 0	DRQCLRRQCR	
0		

#### NOTE

Writing "0" to bits 15 to 0 is ignored. When read, these bits are always read as 0.

### 7.3.4.3 DRQSTR — DMA Request Check Register

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <DMA\_base> + 844H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRQST RRQST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 7.21 DRQSTR Register Contents

Bit Position	Bit Name	Function
15 to 0	DRQSTRRQST	DMA transfer request status flag
15 to 0	DRQSTRRQST	1: Request issued (DMA transfer request signal m is "1")
0	DRQSTRRQST	0: No request (DMA transfer request signal m is "0")

#### NOTE

- The selected trigger factor is always sampled. When a trigger factor is generated, the DRQSTRRQSTM bit in the DRQSTR register is always set.
- The DMA transfer request status flag (DRQSTRRQSTM) is cleared in the following cases.
  - Acknowledgement of DMA transfer
  - Setting the DRQCLRRQCRm bit in the DMA request clear register (DRQCLR) to 1

## 7.4 Functions

### 7.4.1 DMAC Transfer Modes

Single transfer mode and single-step transfer mode are supported as transfer modes.

In either mode, transfer is executed in two cycles (dual address transfer) and therefore a read cycle and a write cycle are generated each time a DMA cycle is executed.

Because the bus is not locked, a CPU cycle may occur between the read and write cycles.

#### 7.4.1.1 Single Transfer Mode (when hardware DMA transfer request is generated)

When a hardware DMA transfer request is acknowledged, data of the selected transfer size (8, 16, or 32 bits) is transferred. Each time transfer has been executed, the bus is released and the DMA controller waits for a DMA transfer request.

Each time a hardware DMA transfer request has been acknowledged, transfer is executed once and the DMA transfer request is cleared. This operation is repeated the number of times specified by the DMA transfer count register m (DTCm ( $m = 15$  to  $0$ )). INTDMAm is generated when the transfer is completed, and the hardware DMA transfer request is cleared.

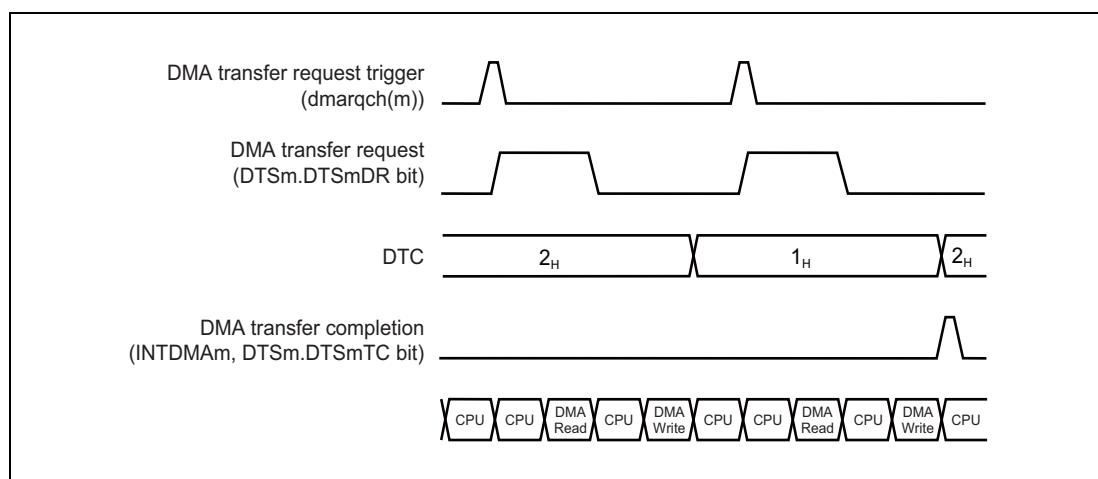


Figure 7.2 Example of Single Transfer (Transfer count is two times)

### 7.4.1.2 Single-Step Transfer Mode (when software DMA transfer request is generated)

When a software DMA transfer request is acknowledged, data of the selected transfer size (8, 16, or 32 bits) is transferred. Each time transfer has been executed, the bus is released.

Once a software DMA transfer request has been acknowledged, this operation is repeated the number of times specified by the DMA transfer count register m (DTCm ( $m = 15$  to  $0$ )). When the transfer is completed, INTDMAm is generated and software DMA transfer request is cleared. Because the priority is judged each time transfer is executed, the DMA cycle of a channel with a higher priority may occur between transfers.

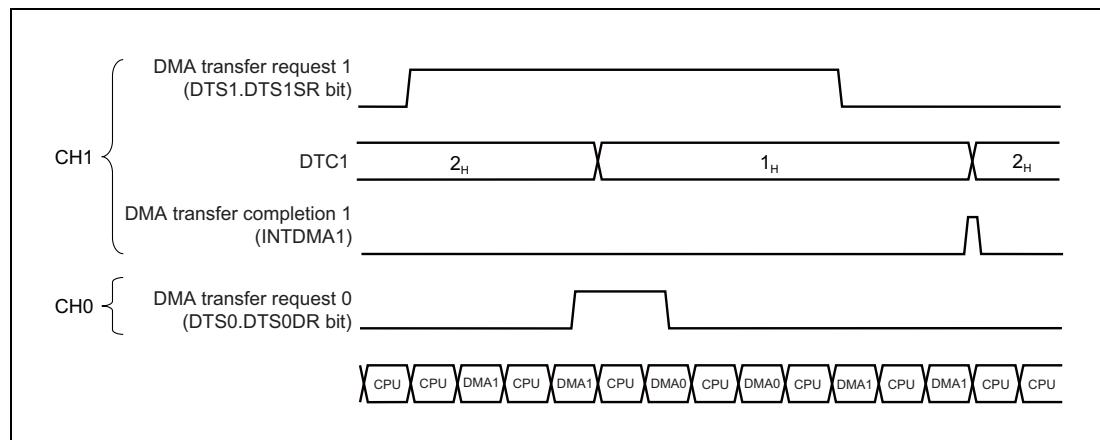


Figure 7.3 Example of Single-Step Transfer

### 7.4.2 DMAC Channel Priority Control

The priority of each channel is fixed and is as follows.

CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15

In single-step transfer, because the priority is judged each time transfer is executed, the DMA cycle of a channel with a higher priority may occur between transfers. However, DMA transfer with higher priority does not occur between the read and write cycles.

An example where DMA transfer request with a high priority is generated while DMA transfer is being executed is shown below.

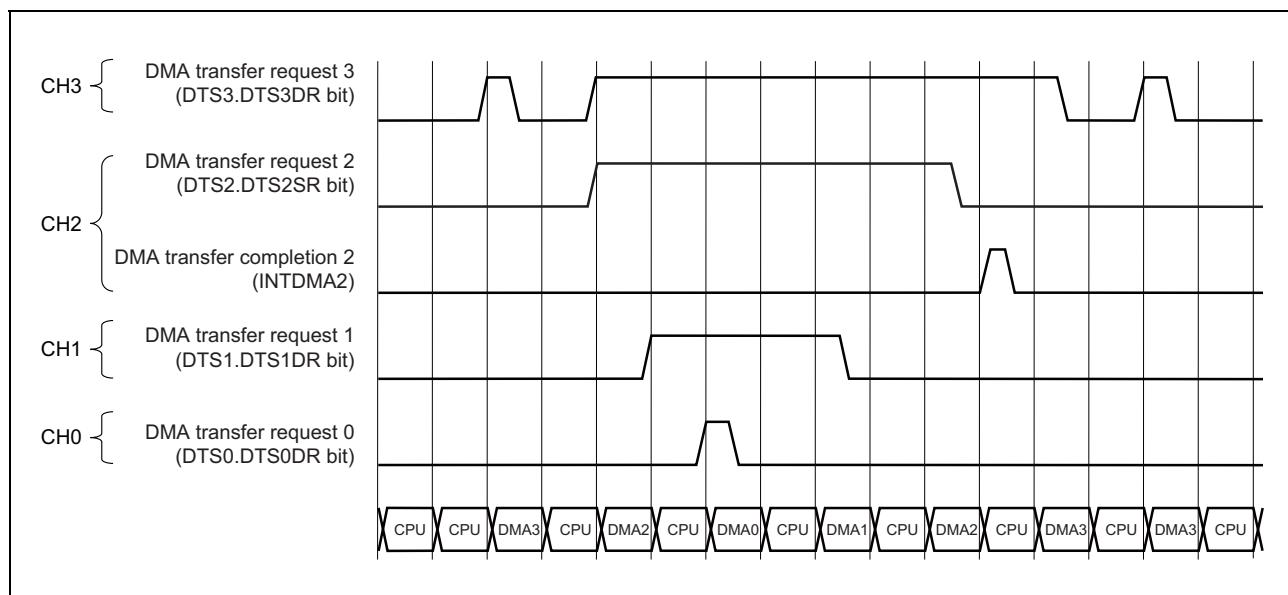


Figure 7.4 Example of Priority Control of DMA Transfer Channels

### 7.4.3 Valid DMAC Transfer Request Conditions

Whether a DMA transfer request of channel m is acknowledged depends on the setting of the DTRC0ERR and DTRC0ADS bits of the DMA transfer request control register 0 (DTRC0), the DTCTmLE bit of the DMA transfer control register (DTCTm), the DTSmTC and DTSmDTE bits of the DMA transfer status register (DTSm), and the DTFRmREQEN bit of the DMA trigger factor register m (DTFRm). **Table 7.22** shows the relationship between the setting of each of the above bits and whether a DMA transfer request is acknowledged.

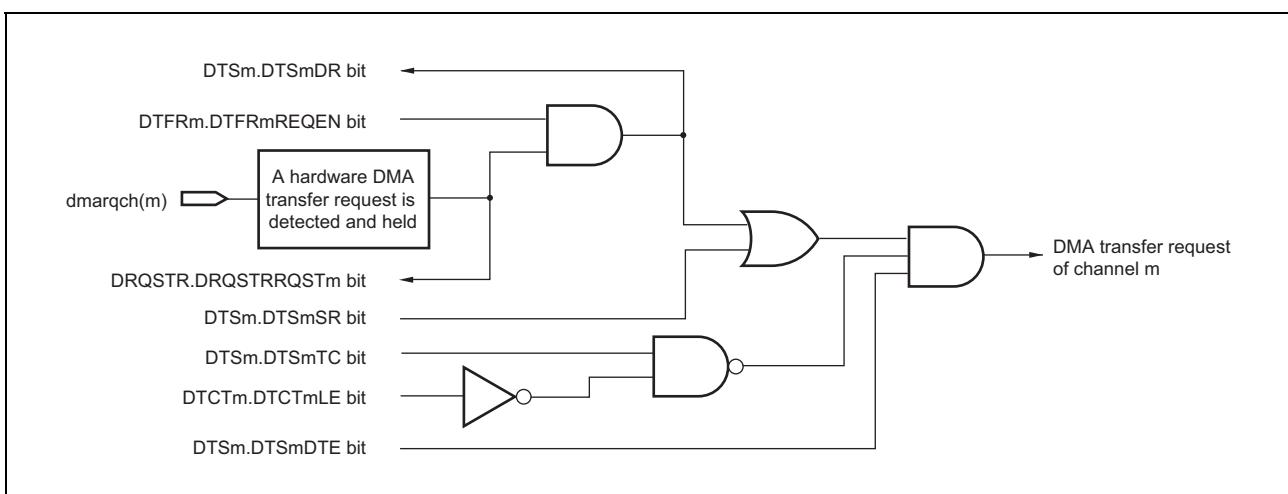
**Table 7.22** Valid DMA Transfer Request Conditions of Channel m

DTFRmREQEN	DTCTmLE	DTSmTC	DTSmDTE	DTRC0ERR	DTRC0ADS	Hardware DMA Transfer Request	Software DMA Transfer Request	Status
—	—	—	—	—	1	Disabled	Disabled	DMA transfer is suspended
—	—	—	—	1	—	Disabled	Disabled	DMA transfer error occurs
—	—	—	0	—	—	Disabled	Disabled	DMA transfer is disabled
0	—	0	1	0	0	Disabled	Enable	Hardware transfer request masked by DTFR
1	—	0	1	0	0	Enable <sup>*1</sup>	Enable <sup>*1</sup>	DMA transfer request can be acknowledged
—	0	1	0	0	0	Disabled	Disabled	DMA transfer is completed (LE disable)
0	1	1	1	0	0	Disabled	Enable	DMA transfer is completed (LE enable)
1	1	1	1	0	0	Enable <sup>*1</sup>	Enable <sup>*1</sup>	DMA transfer is completed (LE enable)

Note 1. The configuration diagram of DMA transfer requests is shown below. A hardware DMA transfer request and software DMA transfer request are logically ORed. Therefore, a hardware DMA transfer request and software DMA transfer request cannot be set at the same time. When DMA transfer is executed by a software DMA transfer request, the DTFR.RQEN bit can be set to 0 to mask hardware DMA transfer requests.

Note that, when a hardware DMA trigger factor (dmarqch (m)) is entered while the DTFRm.DTFRmREQEN bit is set to "0", the entered trigger factor is held in the DRQSTR register. In this case, even if a software DMA transfer request is acknowledged, the held hardware DMA trigger factor is not cleared.

**Note:** m = 15 to 0 (16-channel DMA)



**Figure 7.5** Configuration Diagram of Transfer Requests of Channels

## 7.4.4 Suspending/Resuming/Forcibly Terminating DMA Transfer

### 7.4.4.1 Suspending or Resuming DMA Transfer for All Channels through Software

By setting the DMA transfer suspend bit (DTRC0ADS) of the DMA transfer request control register 0 (DTRC0), the DMA transfer can be suspended. If the DMA transfer suspend bit is set during a DMA cycle, the next DMA transfer is suspended after the ongoing DMA cycle has been completed. Note that the DMA transfer enable bit (DTSmDTE) and the software DMA transfer request bit (DTSmSR) of the DMA transfer status register (DTSm) are not cleared.

To resume the suspended DMA transfer, clear the DTRC0ADS bit. If a DMA transfer is requested at that point, the transfer is executed starting from the channel having the highest priority at that time. To end DMA transfer, clear the DMA transfer request with the DTSmDTE bit cleared.

### 7.4.4.2 Suspending or Resuming DMA Transfer for Each Channel using DMA Transfer Enable Bit (DTSmDTE)

By clearing the DMA transfer enable bit (DTSmDTE) of the DMA transfer status register (DTSm), the DMA transfer can be suspended. If the DMA transfer enable bit is cleared during a DMA cycle, the next DMA transfer is suspended after the ongoing DMA cycle is completed. Note that the software DMA transfer request bit (DTSmSR) of DTSm is not cleared.

To resume the suspended DMA transfer, set the DTSmDTE bit. If another channel is not executing DMA transfer at that point, the priority is judged as usual. If another channel is executing DMA transfer, the priority is judged after that the transfer cycle has been completed. To end DMA transfer, clear the DMA transfer request with the DTSmDTE bit cleared.

### 7.4.4.3 Suspending or Resuming DMA Transfer by using Software DMA Transfer Request Bit (DTSmSR)

By clearing the software DMA transfer request bit (DTSmSR) of the DMA transfer status register (DTSm), the next DMA transfer and those that follow can be suspended. If the DMA transfer request bit is cleared during a DMA cycle, the next DMA transfer is suspended after the ongoing DMA cycle has been completed. Note that DMA transfer enable bit (DTSmDTE) of DMA transfer status register (DTSm) is not cleared.

To resume the suspended DMA transfer, set the DTSmSR bit. If another channel is not executing DMA transfer at that point, the priority is judged as usual. If another channel is executing DMA transfer, the priority is judged after that transfer has been completed.

### 7.4.4.4 Forcibly Terminating Suspended DMA Transfer

To end the suspended DMA transfer, perform the following procedure for each channel.

- Clear the DTSmDTE bit (to “0”) while the DMA transfer is suspended.
- Clear the DMA transfer status bit (DTSmDT) of the DMA transfer status register (DTSm).
- Clear the DMA transfer request (the DRQSTRRQSTm bit of the DRQSTR register for hardware DMA transfer requests, or the DTSmSR bit of the DTSm register for software DMA transfer requests) (to “0”).

Note that execution of the above procedure may not end DMA transfer immediately. When re-setting the DMA setting register after executing the procedure to end transfer, do so after confirming that the read value of the DTSmDT bit is 0.

Note also that DMA transfer cannot be resumed once the above procedure is executed.

## 7.4.5 Error Response

### 7.4.5.1 Suspending DMA Transfer by Error Response

When an error occurs at the DMA transfer source or transfer destination, the DMA transfer suspend bit (DTRC0ADS) of the DMA transfer request control register 0 (DTRC0) is set to suspend all of the subsequent DMA transfers. At the same time, the DMA transfer error status bit (DTRC0ERR) is set and a SYSERR exception is generated to the CPU. The user can evaluate the channel in which the error has occurred by using the DMA transfer error flag (DTSmER) of the DMA transfer status register (DTSm), when the user has confirmed that DTRC0ERR bit has been set.

In this case, note that, if an error response is acknowledged in the read cycle, the write cycle is not executed, but the transfer address and the transfer count are updated.

### 7.4.5.2 Canceling Transfer Suspend by Error Response

DMA transfer suspend can be canceled by clearing the DMA transfer suspend bit (DTRC0ADS) and DMA transfer error status bit (DTRC0ERR) of the DMA transfer request control register 0 (DTRC0).

Clear the DMA transfer enable bit (DTSmDTE) and the DMA transfer status bit (DTSmDT) in the DMA transfer status register (DTSm) in advance, so that DMA transfer is not resumed after its suspension has been canceled. In the case of a hardware DMA transfer request, the DRQSTRRQST<sub>m</sub> bit of the DRQSTR register must be cleared. In the case of a software DMA transfer request, the DTSmSR bit of the DTSm register must be cleared.

## 7.4.6 Stand-By Support

If a request to transition to STOP mode is issued, DMA transfer stops after completion of the DMA cycle currently being executed. Unlike DMA transfer suspend caused by software, this does not require any operation of the DMA control register. DMA transfer resumes upon cancellation of the STOP mode, and if a DMA request is retained, that DMA transfer starts in order of priority of each channel.

## 7.4.7 Loop function

When DMA transfer is finished with loop enabled (DTCTmLE is set), DTSmDTE bit is not cleared and DTSmTC bit is set. In that case, successive DMA transfer request can be acknowledged under the condition that DTSmTC bit is set.

When DMA transfer has been completed, the values of source address (DSALm and DSAHm registers), destination address (DDALm and DDAHm registers), transfer count (DTcm register) return to the values when DMA transfer was started.

Loop function can be used to execute DMA transfer continuously under the same settings without operations of clearing DTSmTC bit and setting DTSmDTE bit after DMA transfer completion.

In case that loop function is enabled, DMA transfer is started automatically with the same setting of previous DMA transfer when DMA transfer request is inputted after completion of previous DMA transfer.

DTCTmLE bit is configurable during DMA transfer and loop function is valid at the final transfer (DTcmDTc=0001<sub>H</sub>). It is recommended to change setting of DTCTmLE bit before starting the DMA cycle of final transfer.

To suspend DMA transfer while using loop function, clear DTCTmLE bit and then clear DTSmDTE bit.

## 7.5 Protection Function

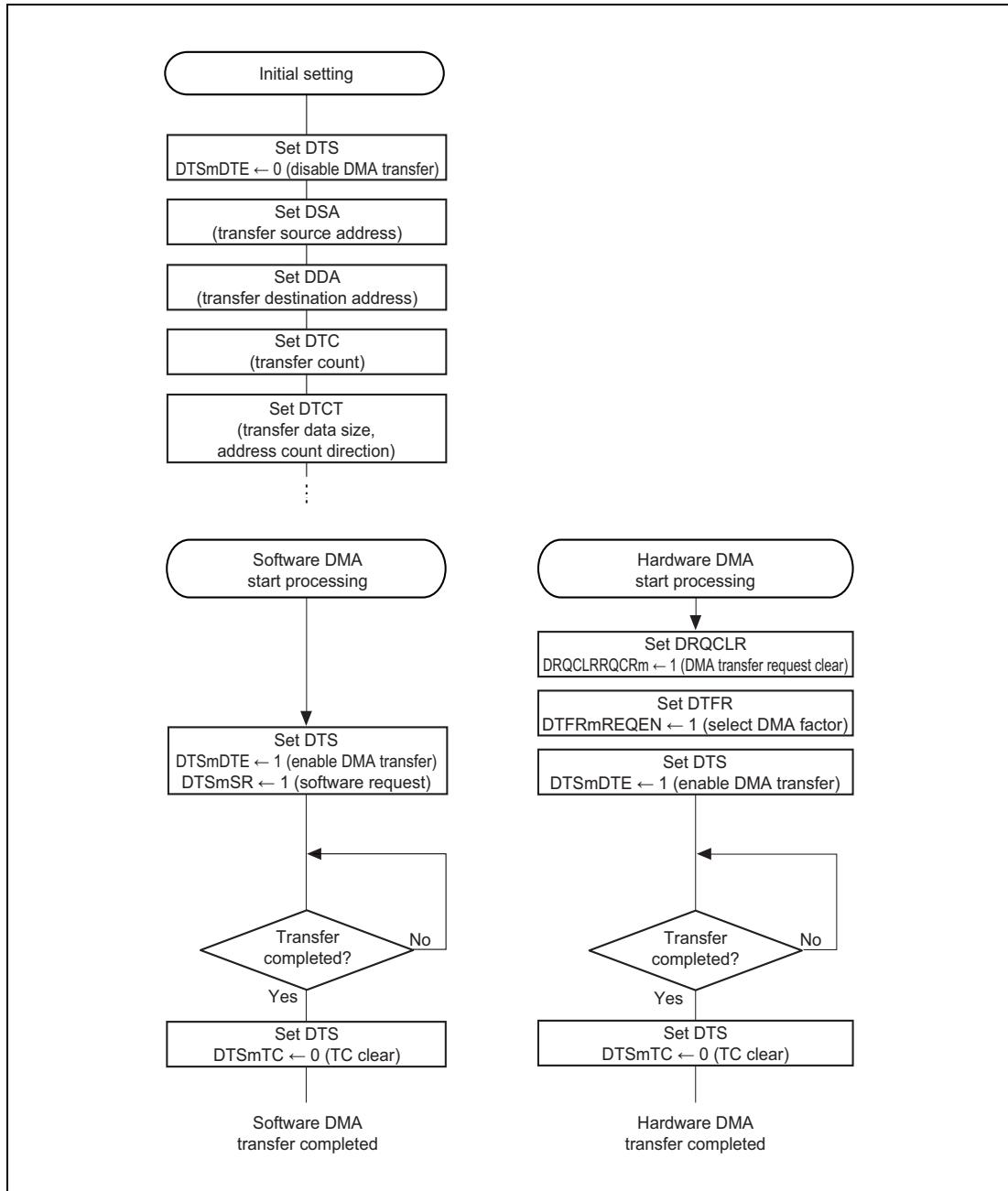
The accessibility of DMA control registers in supervisor mode or user mode can be set by using the DMCM0 register. Even if the user mode is set as the access master by the DMCM0 register, each register can be accessed in supervisor mode. Note that when the following registers are accessed in user mode, an access error occurs:

- DMA transfer request control register 0 (DTRC0)
- DMA channel master setting register 0 (DMCM0)
- DMA setting registers for which supervisor mode is assigned as the access master.

When an access error occurs, writing the DMA setting register is ignored, but reading it can be performed as usual.

## 7.6 DMAC Transfer Setting Flow

**Figure 7.6** shows the flow for setting DMA transfer.



**Figure 7.6** DMA Transfer Setting Flow

### CAUTION

Because DMA constantly samples hardware DMA trigger factors, a trigger factor may be held in the DRQSTR register during the initial settings of DMA transfer. The held trigger factor can be cleared by setting the DRQCLR register. If necessary, clear unnecessary trigger factors before setting the DTSmDTE bit to 1.

## Section 8 Reset Controller

### 8.1 Overview

Several system reset functions are provided in order to initialize CPU core and peripheral functions as well as their associated registers.

A reset can be caused by the following events:

- External reset (RESET)
- Power-on-clear (POCRES)
- Watchdog timer reset (WDTA0RES, WDTA1RES)
- Clock monitor reset (CLMA0RES, CLMA1RES, CLMA2RES)
- Low-voltage indicator reset (LVIRES)
- Software reset (SWRES)
- Debugger reset (DBRES)
- Core voltage monitor reset (CVMRES)
- Transition to DeepSTOP mode

#### 8.1.1 Reset Sources

Reset levels and reset sources are shown below.

Various reset sources are assigned to the different levels of the reset.

**Table 8.1 Reset Sources and Reset Targets**

Reset Level	Reset Source	Clock Generator (except PLL)/ Real-Time Clock/CVM/ LVI	AWO Modules <sup>*1</sup>	ISO Modules <sup>*3</sup>
1	Power-on-clear (POCRES) Debugger reset ( <u>DBRES</u> )	Reset <sup>*2</sup>	Reset	Reset
2	External reset (RESET) Watchdog timer reset (WDTA0RES, WDTA1RES) Clock monitor reset ( <u>CLMA0RES</u> , <u>CLMA1RES</u> , <u>CLMA2RES</u> ) Core voltage monitor reset ( <u>CVMRES</u> ) Low voltage indicator reset ( <u>LVIRES</u> ) Software reset (SWRES)	Not reset target <sup>*4</sup>	Reset	Reset
3	Reset by DeepSTOP mode	Not reset target	Not reset target	Reset

Note 1. Clock generator, real-time clock, CVM, and LVI are excluded.

Note 2. For a debugger reset, the CVM related registers (CVMF, CVMDE, and CVMDIAG) are not reset.

Note 3. PLL is included.

Note 4. In clock monitor reset, oscillator-related registers for clock monitoring are initialized.

Reset level 1: Initializes the entire microcontroller.

Reset level 2: For a quick return to normal operating mode by eliminating the oscillator stabilization time<sup>\*1</sup>, initializes the entire microcontroller except for the clock generator and the real-time clock.

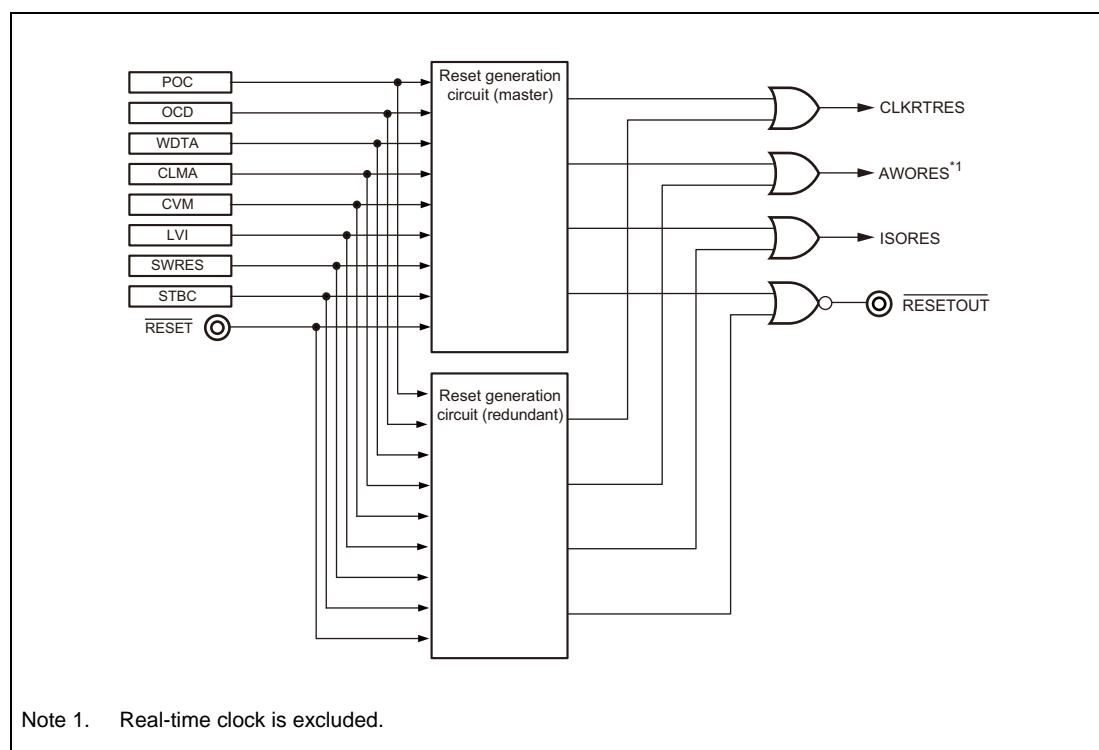
Reset level 3: At the transition to DeepSTOP mode, initializes all the isolated areas (ISO).

**Note 1.** MOSCSTPMASK=1

### 8.1.2 Reset Controller Redundancy

The reset controller of the microcontroller have redundant configuration, and includes duplicated reset generation circuits. Such configuration enables initialization of the reset targeted area without failure even if one of the two reset generation circuits fails.

The configuration of the reset controller is shown in the figure below.



**Figure 8.1** Reset Controller Redundancy

At the generation of a reset, the same reset source signal is input to two reset generation circuits.

According to the reset source, the two reset generation circuits output the Always-ON area reset signal (AWORES), Isolated area reset signal (ISORES), clock generator/real-time clock reset signal (CLKRTRES), and RESETOUT signal.

The AWORES, ISORES, CLKRTRES, and RESETOUT signals are generated by executing the logical OR of the signals output from the two reset generation circuits. Thus, a reset signal is generated normally even if one of the two reset generation circuits fails.

Whether a reset generation circuit operates normally can be checked by reading and comparing the reset factor registers of the respective reset generation circuits.

### 8.1.3 Reset Output (RESETOUT)

When a reset source of reset level 1 or 2 is generated, a reset output signal (RESETOUT) is output to the outside. Reset output is used to reset external devices at the same time as a reset is generated inside the microcontroller.

For details, refer to **Section 2.11.1.1, P0\_0: RESETOUT.**

### 8.1.4 Reset Flag

To identify a reset source, two registers with a flag for each reset source are provided. The main elements of the reset controller are shown in **Figure 8.2, Block Diagram of the Reset Controller.**

### 8.1.5 Clock Supply

The clock supply to the reset controller is shown in the following table.

Table 8.2     Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Reset	Register access clock	EMCLK

## 8.2 Configuration

### 8.2.1 Block Diagram

A block diagram of the reset controller is shown below.

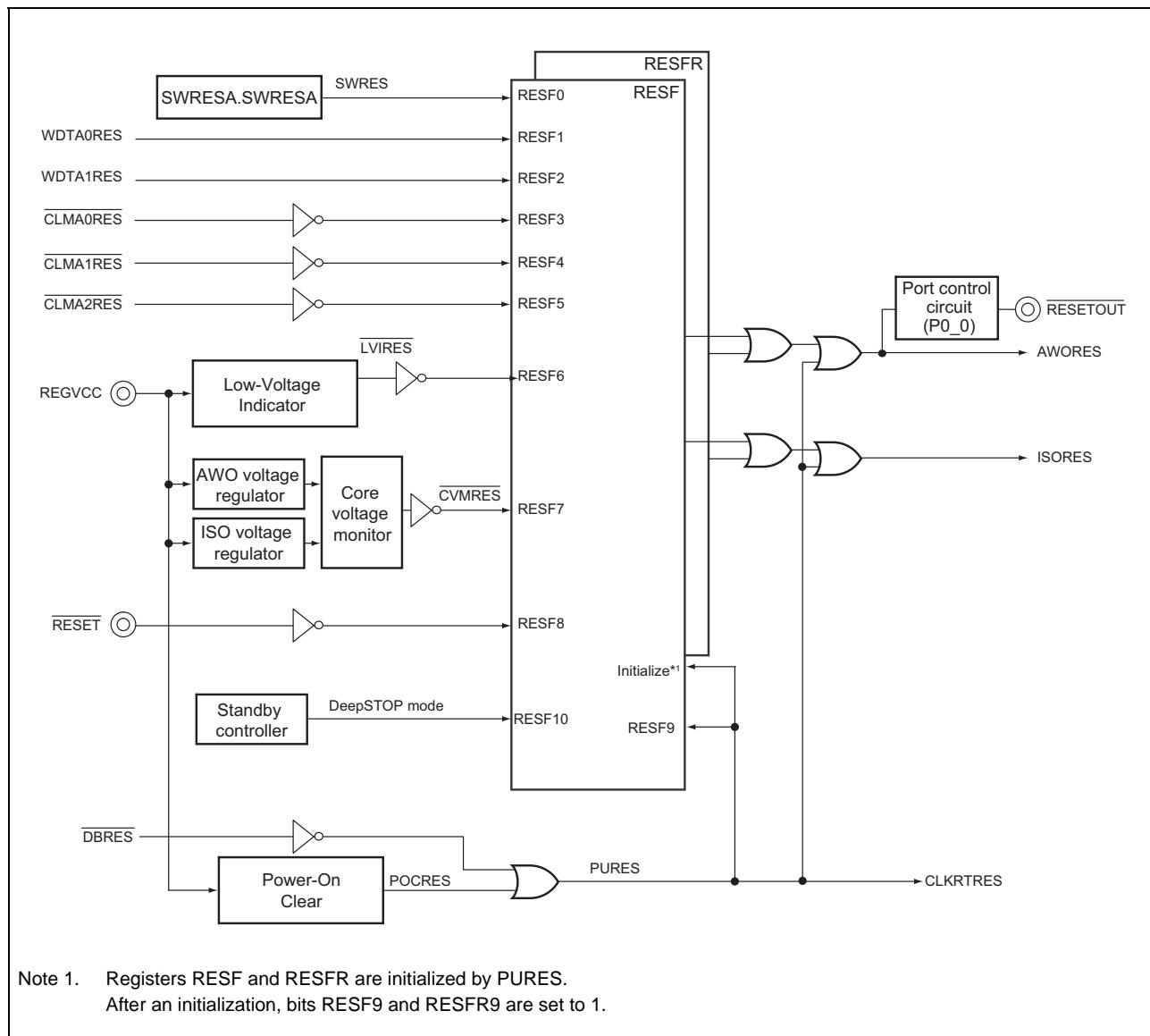


Figure 8.2 Block Diagram of the Reset Controller

## (1) Reset Signals

The reset controller manages the generation of three reset signals upon occurrence of reset signals from various reset sources:

- AWO area reset (AWORES)

AWORES is generated by all reset sources except the transition to DeepSTOP mode.

AWORES resets all modules in the AWO area except clock generation circuit, real-time clock, core voltage monitor, and low-voltage detection circuit.

- ISO area reset (ISORES)

ISORES is generated by all reset sources.

ISORES resets all modules (including PLL) in the ISO area.

- CLKRTRES

CLKRTRES is generated by the power-on clear or debugger reset sources.

CLKRTRES resets the clock generation circuit (excluding PLL) and real-time clock.

The power-up reset (PURES) is generated by the power-on clear and debugger reset sources.

Following the generation of an AWORES reset, with the exception of the PLL, all clock-generation circuits that were operating at the time (LS IntOSC, HS IntOSC, MainOSC, SubOSC) continue to operate. However, when MOSCSTPMISK is 0, MainOSC stops during any reset and resumes operation after release from the reset. On the generation of a  $\overline{\text{CLMA0RES}}$  reset, the HS IntOSC that was the target for CLMA0 monitoring is reset. On the generation of a  $\overline{\text{CLMA1RES}}$  reset, the MainOSC that was the target for CLMA1 monitoring is reset.

The PURES initializes all of the clock generation circuits. It is necessary to restart the clock generation circuit after recovery from the PURES.

The CPU reset is the ISO area reset (ISORES) to the CPU sub system.

## (2) Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) hold a flag for each reset source, and this flag is set when the corresponding reset is asserted.

All reset flags except RESF9 and RESFR9 are initialized by a power-up reset (PURES). (Bits RESF9 and RESFR9 are set to 1 after initialization.) In addition, all the bits can be cleared by software.

For details, see [Section 8.1.4, Reset Flag](#).

## (3) On-Chip Module Resets

### (a) Watchdog Timer Reset

The watchdog timers 0, 1 can generate two types of resets: WDTA0RES and WDTA1RES.

For details, see [Section 8.4.6, Watchdog Timer \(WDTA\) Reset](#).

### (b) Clock Monitor Resets

The clock monitors can generate three resets:  $\overline{\text{CLMA0RES}}$ ,  $\overline{\text{CLMA1RES}}$ , and  $\overline{\text{CLMA2RES}}$ .

For details, see [Section 8.4.8, Clock Monitor \(CLMA\) Reset](#).

### (c) Debugger Reset

A reset is generated by a command from a debugger. This leads to a generation of power-up reset PURES. For details, see [Section 8.4.9, Debugger Reset](#).

#### (4) Software Controlled Resets (SWRES)

A software reset SWRES can be generated by use of the software reset register SWRESA.

For details, see **Section 8.4.7, Software Reset**.

#### (5) Reset Output Signal

During reset and after release from the reset, port P0\_0 outputs low level as RESETOUT function. For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

#### (6) Power Supply Monitoring

The following power supply detection circuits observe the level of the external power supply REGVCC and core voltage.

##### (a) Low-Voltage Indicator

The low-voltage indicator (LVI) generates the LVIRES reset if the voltage level of REGVCC drops below a certain level. The level can be adjusted and the LVIRES can be masked.

For details, see **Section 8.4.3, Low-Voltage Indicator (LVI) Reset**.

##### (b) Power-On-Clear

The power-on-clear circuit (POC) continuously compares the power supply voltage REGVCC with an internal reference voltage. Thus, a reset is generated when the power supply voltage goes below a certain level.

For details, see **Section 8.4.2, Power-On-Clear (POC) Reset**.

##### (c) Core Voltage Monitor

A reset can be generated when the core voltage monitor (CVM) detects over- or undervoltage in core voltage. (Output/not output can be set by option byte.)

For details, see **Section 8.4.4, Core Voltage Monitor (CVM) Reset**.

#### (7) Masking of Reset Sources in Debugging Mode

The following reset sources can be masked during debugging:

**Table 8.3 Reset Sources to be Masked during Debugging**

Reset Source	Maskable/Non-maskable
Power-on clear (POCRES)	—
Debugger reset ( <u>DBRES</u> )	—
External reset ( <u>RESET</u> )	√
Low-voltage indicator reset ( <u>LVIRES</u> )	√
Clock monitor reset ( <u>CLMA0RES</u> , <u>CLMA1RES</u> , <u>CLMA2RES</u> )	√
Watchdog timer reset (WDTA0RES, WDTA1RES)	√
Core voltage monitor reset ( <u>CVMRES</u> )	√
Software reset (SWRES)	√
Reset by DeepSTOP mode	—

## 8.3 Registers

This section contains a description of all registers of the reset controller.

### 8.3.1 Reset Controller Registers Overview

The reset controller is controlled and operated by the following registers:

**Table 8.4 Reset Controller Registers Overview**

Register Name	Symbol	Address
<b>Reset flag registers</b>		
Reset factor register	RESF	FFF8 0760 <sub>H</sub>
Reset factor clear register	RESFC	FFF8 0768 <sub>H</sub>
Redundant reset factor register	RESFR	FFF8 0860 <sub>H</sub>
Redundant reset factor clear register	RESFCR	FFF8 0868 <sub>H</sub>
<b>Software reset control register</b>		
Software reset register	SWRESA	FFF8 0A04 <sub>H</sub>

#### NOTES

1. For the LVI related, RAM store related, and CVM related registers, see **Section 9, Supply Voltage Monitor**.
2. As for the protection registers, see **Section 4, Write-Protected Registers**.

## 8.3.2 Details of Reset Flag Registers

### 8.3.2.1 RESF — Reset Factor Register

This register contains information about which type of resets occurred after the last power-on clear reset. This register is initialized by a power-up reset PURES.

Each reset condition sets the corresponding flag in the register.

For example, if a clock monitor reset  $\overline{\text{CLMA0RES}}$  occurs after a watchdog timer reset WDTA0RES, RESF reads  $0000\ 000A_H$ .

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:**  $FFF8\ 0760_H$

**Value after reset:**  $0000\ 0200_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESF10	RESF9	RESF8	RESF7	RESF6	RESF5	RESF4	RESF3	RESF2	RESF1	RESF0
Value after reset	0	0	0	0	0	0	1	1/0 <sup>*1</sup>	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see [Figure 8.4, When RESET is Released before Execution of Flash Sequence](#).

Table 8.5 RESF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10	RESF10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESF9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESF8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESF7	CVM reset flag 0: No reset occurred 1: Reset has occurred
6	RESF6	LVI reset flag 0: No reset occurred 1: Reset has occurred
5	RESF5	CLMA2 reset flag 0: No reset occurred 1: Reset has occurred
4	RESF4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESF3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred

**Table 8.5 RESF Register Contents (2/2)**

Bit Position	Bit Name	Function
2	RESF2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESF1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred
0	RESF0	Software reset flag 0: No reset occurred 1: Reset has occurred

### 8.3.2.2 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** FFF8 0768<sub>H</sub>

**Value after reset:** Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESFC10	RESFC9	RESFC8	RESFC7	RESFC6	RESFC5	RESFC4	RESFC3	RESFC2	RESFC1	RESFC0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W

**Table 8.6** RESFC Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When writing, write “0”.
10	RESFC10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFC9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFC8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFC7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFC6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFC5	CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag
4	RESFC4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFC3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFC2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFC1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFC0	Software reset flag clear 0: Do not clear flag 1: Clear flag

### 8.3.2.3 RESFR — Redundant Reset Factor Register

This register is a duplication of the reset factor register. This register is initialized by a power-up reset PURSES.

In accordance with the setting conditions for each bit in the reset factor register, the same bits are set in this register.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 0860<sub>H</sub>

**Value after reset:** 0000 0200<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESFR10	RESFR9	RESFR8	RESFR7	RESFR6	RESFR5	RESFR4	RESFR3	RESFR2	RESFR1	RESFR0
Value after reset	0	0	0	0	0	0	1	1/0 <sup>*1</sup>	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 8.4, When RESET is Released before Execution of Flash Sequence.**

**Table 8.7 RESFR Register Contents (1/2)**

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10	RESFR10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESFR9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESFR8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESFR7	CVM reset flag 0: No reset occurred 1: Reset has occurred
6	RESFR6	LVI reset flag 0: No reset occurred 1: Reset has occurred
5	RESFR5	CLMA2 reset flag 0: No reset occurred 1: Reset has occurred
4	RESFR4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESFR3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred
2	RESFR2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESFR1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred

**Table 8.7 RESFR Register Contents (2/2)**

Bit Position	Bit Name	Function
0	RESFR0	Software reset flag 0: No reset occurred 1: Reset has occurred

### 8.3.2.4 RESFCR — Redundant Reset Factor Clear Register

This register clears the reset flags of the RESFR register.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** FFF8 0868<sub>H</sub>

**Value after reset:** Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	—	—	—	—	—	RESFCR10	RESFCR9	RESFCR8	RESFCR7	RESFCR6	RESFCR5	RESFCR4	RESFCR3	RESFCR2	RESFCR1	RESFCR0
R/W	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W

**Table 8.8** RESFCR Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When writing, write “0”.
10	RESFCR10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFCR9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFCR8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFCR7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFCR6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFCR5	CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag
4	RESFCR4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFCR3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFCR2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFCR1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFCR0	Software reset flag clear 0: Do not clear flag 1: Clear flag

### 8.3.3 Details of Software Reset Control Registers

#### 8.3.3.1 SWRESA — Software Reset Register

This register is used to generate a software reset SWRES. The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** FFF8 0A04<sub>H</sub>

**Value after reset:** Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRESA
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

**Table 8.9 SWRESA Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write “0”.
0	SWRESA	Software reset trigger 0: No Software reset trigger is generated. 1: Software reset trigger is generated.

## 8.4 Functional Description

### 8.4.1 Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) provide reset flags for each reset source.

If a reset has occurred, the corresponding flag is set. According to this, the source of the reset is evaluated.

RESF and RESFR are initialized by a power-up reset PURES (POCRES or DBRES) (though bits RESF9 and RESFR9 are set to 1 after initialization). In addition, flags in RESF and RESFR can be cleared by the reset factor clear register (RESFC) and the redundant reset factor clear register (RESFRC).

Each reset source can set the corresponding flag independently from other reset sources.

### 8.4.2 Power-On-Clear (POC) Reset

The power-on-clear circuit (POC) constantly compares the power supply voltage REGVCC with the internal reference voltage  $V_{POC}$ . It ensures that the microcontroller only operates as long as the power supply exceeds a certain level.

If REGVCC falls below the internal reference voltage ( $REGVCC < V_{POC}$ ), the internal reset signal POCRES and a power-up reset PURES are asserted.

For details on the specifications of the internal voltage reference level  $V_{POC}$ , see the Data Sheet.

The reset factor register (RESF) and the redundant reset factor register (RESFR) are cleared by the power-on clear reset. RESF9 and RESFR9 are set to 1 after initialization.

The power-on-clear function holds the microcontroller in reset state as long as the power supply voltage does not exceed the threshold level  $V_{POC}$ .

The following figure illustrates the timing of a POCRES.

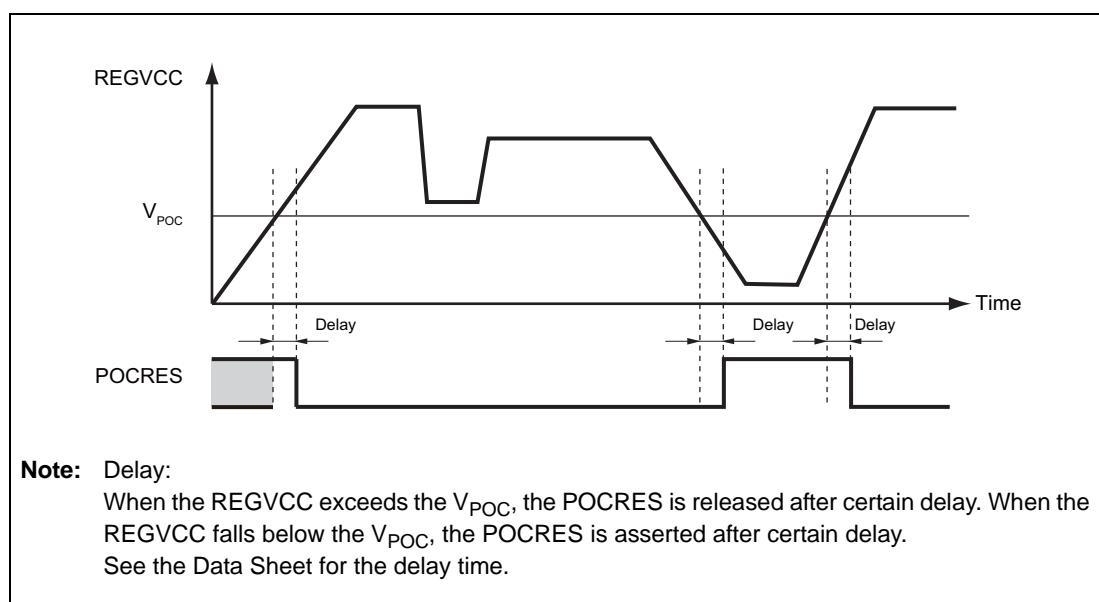


Figure 8.3 POC Reset Timing

### (1) Overview of CPU System Startup after Power-On-Clear

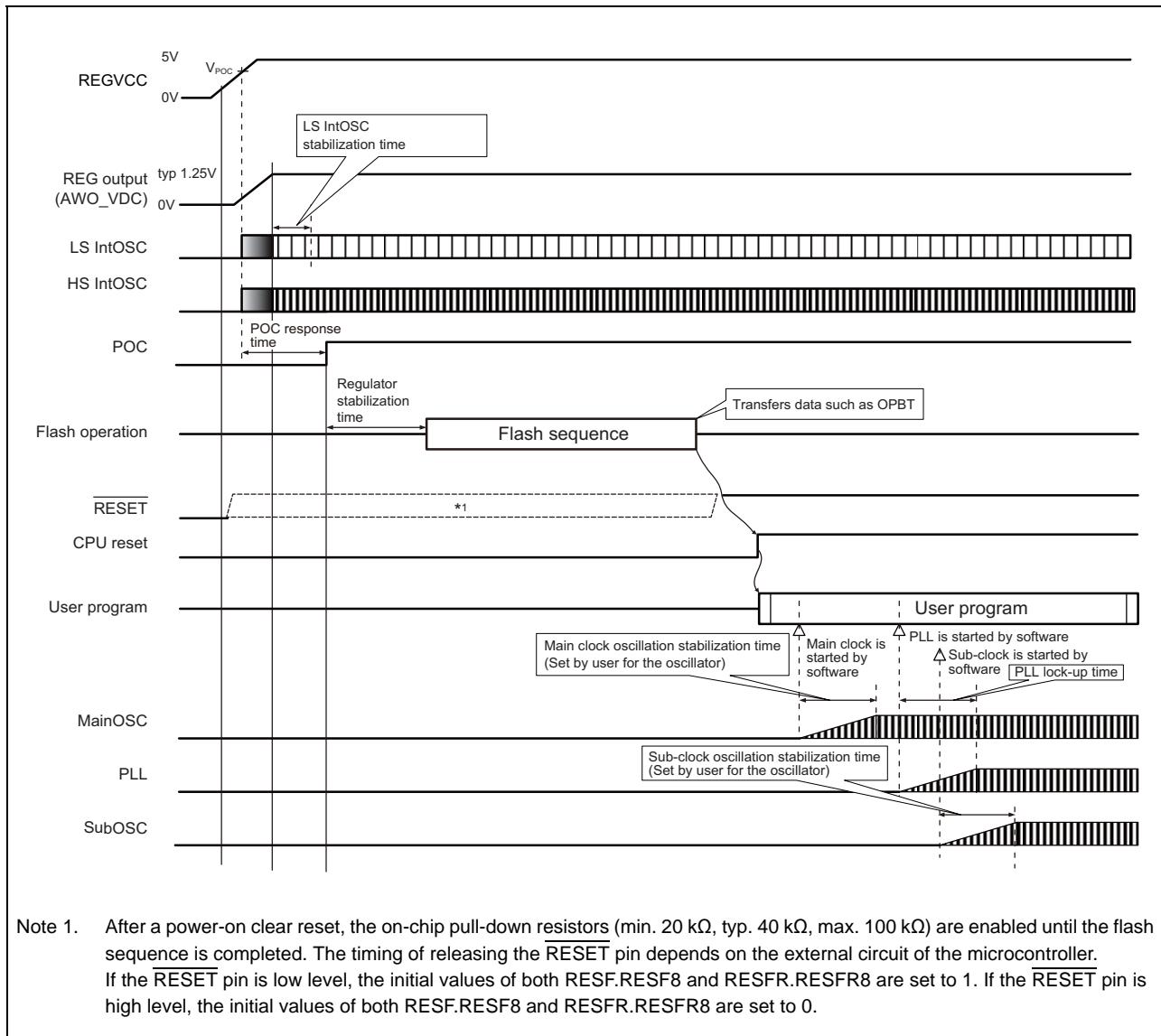


Figure 8.4 When **RESET** is Released before the Flash Sequence is Completed

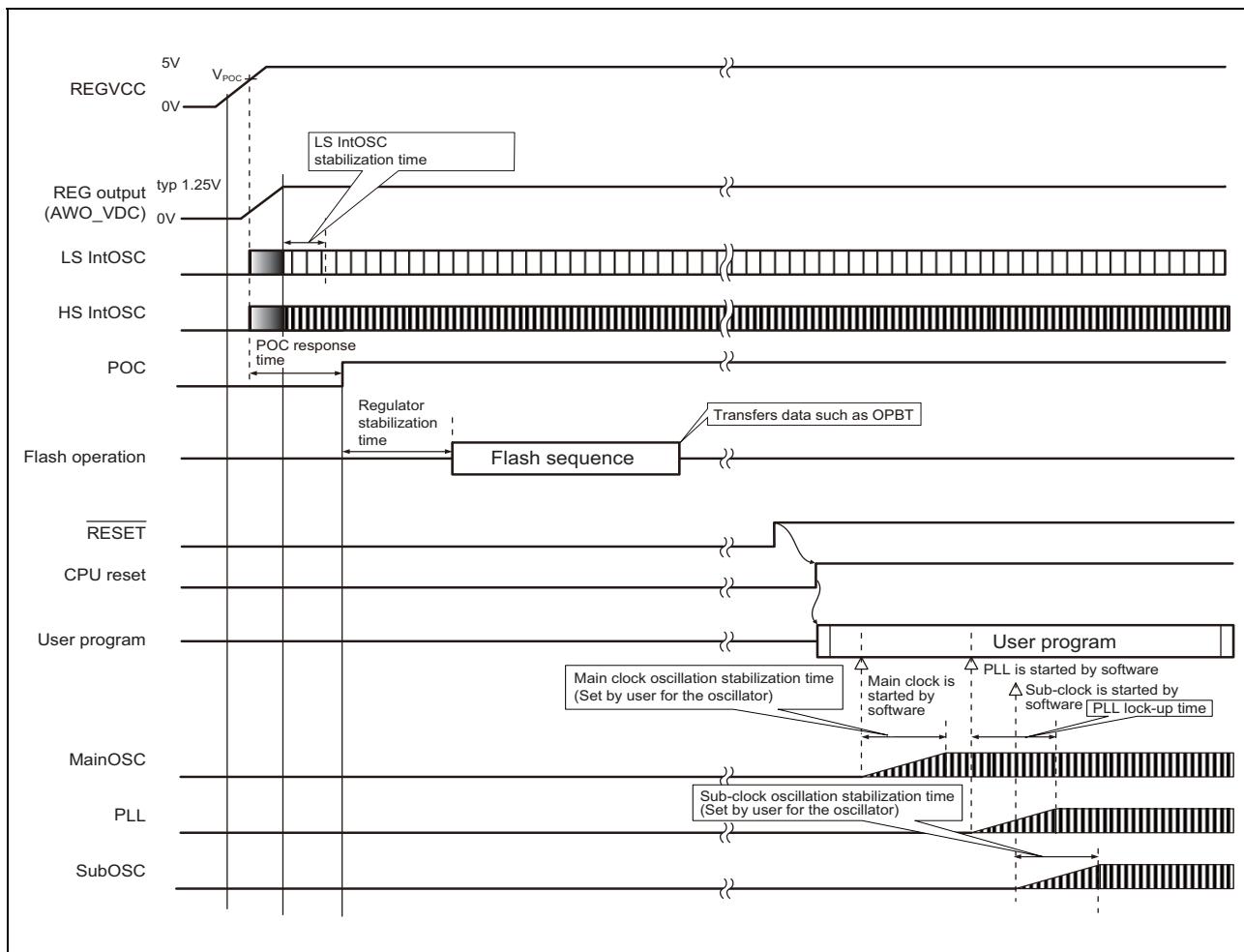


Figure 8.5 When RESET is Released after the Flash Sequence is Completed

### 8.4.3 Low-Voltage Indicator (LVI) Reset

The low-voltage indicator (LVI) permanently compares the power supply voltage REGVCC with the LVI internal reference voltage  $V_{LVI}$ .

When setting the LVI detection voltage and releasing the LVIRESMK, if REGVCC falls below the internal reference voltage ( $REGVCC < V_{LVI}$ ), the internal reset signal  $\overline{LVIRES}$  is generated.

Additionally, the  $\overline{LVIRES}$  flags (bits RESF.RESF6 and RESFR.RESFR6) are set.

After that, even if REGVCC exceeds  $V_{LVI}$ , bits RESF.RESF6 and RESFR.RESFR6 are not cleared automatically. They are cleared as described below.

- Setting the RESFC.RESFC6 bit to 1 clears the RESF.RESF6 bit.  
Setting the RESFCR.RESFCR6 bit to 1 clears the RESFR.RESFR6 bit.
- Power-up reset PURES (POCRES or  $\overline{DBRES}$ )

For details on the LVI functions, see **Section 9, Supply Voltage Monitor**.

The following figure illustrates the timing of a  $\overline{LVIRES}$  and bits RESF.RESF6 and RESFR.RESFR6.

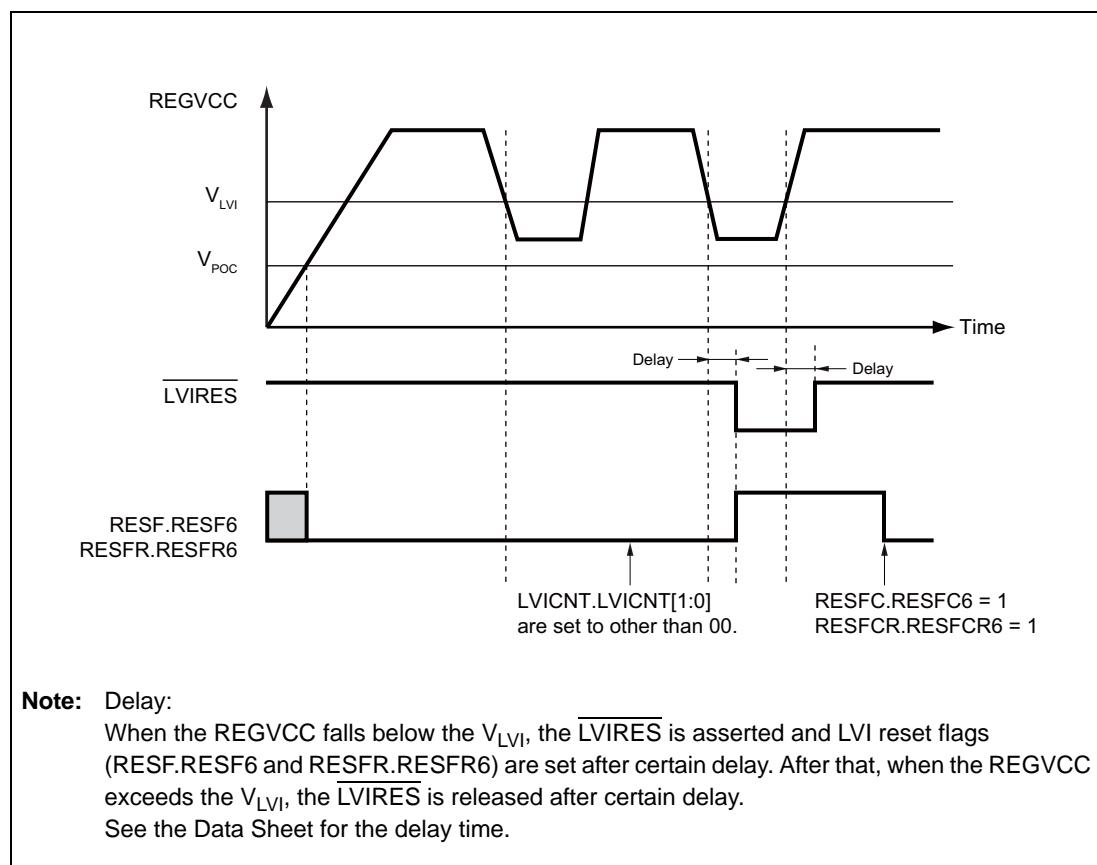


Figure 8.6 LVI Reset Timing

#### 8.4.4 Core Voltage Monitor (CVM) Reset

Core voltage monitor is used to monitor the core voltage inside the microcontroller.

The reset  $\overline{\text{CVMRES}}$  is generated if the core voltage is not in the specified voltage range while CVM is enabled. Moreover, the  $\overline{\text{CVMRES}}$  flags (RESF.RESF7 and RESFR.RESFR7) are set.

After that, the RESF.RESF7 and RESFR.RESFR7 bits are not automatically cleared even if the core voltage returns to the specified voltage range. The RESF.RESF7 and RESFR.RESFR7 bits are cleared as described below.

- Setting the RESFC.RESFC7 bit to 1 clears the RESF.RESF7 bit.  
Setting the RESFCR.RESFCR7 bit to 1 clears the RESFR.RESFR7 bit.
- Power-up reset PURES (POCRES or  $\overline{\text{DBRES}}$ )

If the CVM detects an abnormal high voltage, the power supply to the ISO area is switched off.

Once the  $\overline{\text{CVMRES}}$  by the high voltage detection is generated, the microcontroller stays in the reset state.

To cancel this state, it is mandatory to use the external reset ( $\overline{\text{RESET}}$ ) input.

Release the external reset ( $\overline{\text{RESET}}$ ) after the voltage level becomes lower than the high detection voltage.

For details on the CVM function, see **Section 9, Supply Voltage Monitor**.

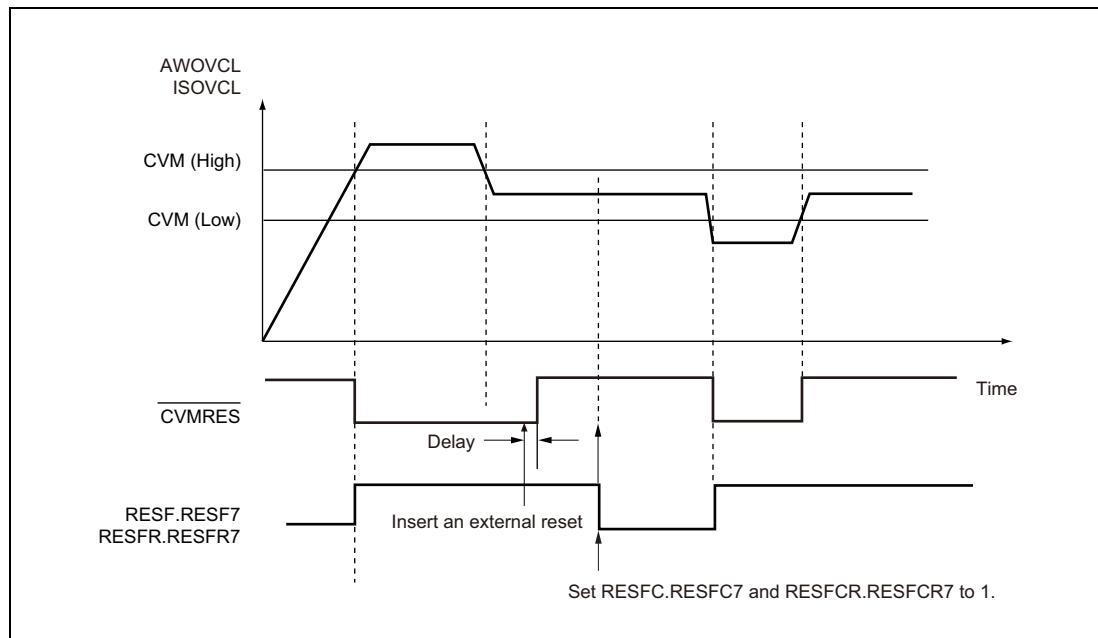


Figure 8.7 CVM Reset Timing

### 8.4.5 External Reset (RESET)

When a low level input is applied to the RESET pin, a reset is asserted and the RESF.RESF8 and RESFR.RESFR8 bits are set.

After that, bits RESF.RESF8 and RESFR.RESFR8 are not cleared automatically, even if the low-level input to the RESET pin is released. Bits RESF.RESF8 and RESFR.RESFR8 are cleared as described below.

- Setting the RESFC.RESFC8 bit to 1 clears the RESF.RESF8 bit.
- Setting the RESFCR.RESFCR8 bit to 1 clears the RESFR.RESFR8 bit.
- Power-up reset PURES (POCRES or DBRES)

The RESET pin includes an analog noise filter to prevent erroneous resets due to noise.

The following figure shows the timing when AWORES and ISORES are generated by an external reset. This figure also shows the effect of the noise filter.

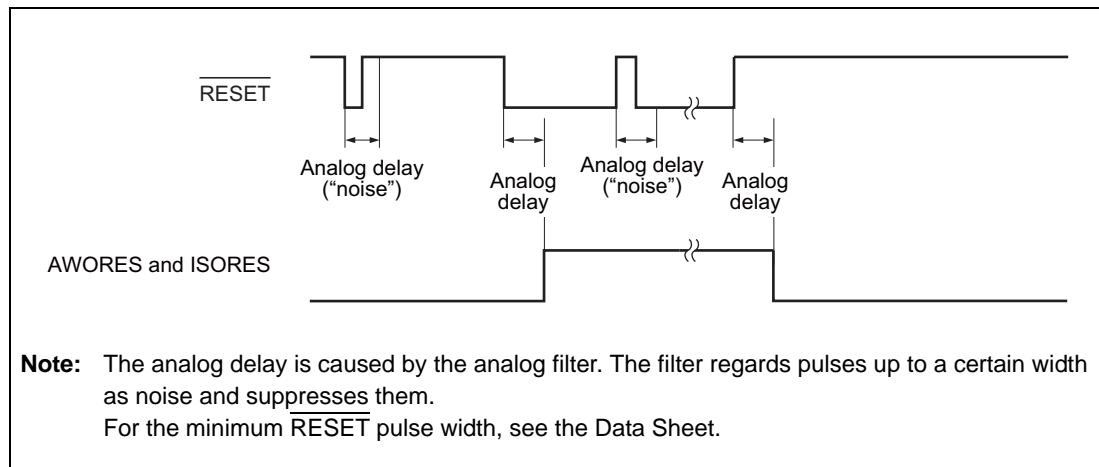


Figure 8.8 External Reset (RESET)

#### 8.4.6 Watchdog Timer (WDTA) Reset

The watchdog timers can be configured to generate a reset if the overflow time is exceeded. After a watchdog timer reset is asserted, the corresponding watchdog timer reset flags (the RESF.RESF1 and RESFR.RESFR1 bits for WDTA0RES, and the RESF.RESF2 and RESFR.RESFR2 bits for WDTA1RES) are set.

After that, bits RESF.RESF1 and RESFR.RESFR1 (or bits RESF.RESF2 and RESFR.RESFR2) are not cleared automatically, even if WDTA0RES (or WDTA1RES) is released.

Bits RESF.RESF1 and RESFR.RESFR1, and bits RESF.RESF2 and RESFR.RESFR2 are cleared as described below.

- WDTA0RES:
  - Setting the RESFC.RESFC1 bit to 1 clears the RESF.RESF1 bit.
  - Setting the RESFCR.RESFCR1 bit to 1 clears the RESFR.RESFR1 bit.
- WDTA1RES:
  - Setting the RESFC.RESFC2 bit to 1 clears the RESF.RESF2 bit.
  - Setting the RESFCR.RESFCR2 bit to 1 clears the RESFR.RESFR2 bit.
- Power-up reset PURES (POCRES or  $\overline{DBRES}$ )

#### 8.4.7 Software Reset

The software reset SWRES can be asserted by setting SWRESA.SWRESA to 1.

SWRES sets the reset flag RESF.RESF0 and the RESFR.RESFR0 bit.

RESF.RESF0 and RESFR.RESFR0 are not cleared automatically. RESF.RESF0 and RESFR.RESFR0 are cleared as described below.

- Setting the RESFC.RESFC0 bit to 1 clears the RESF.RESF0 bit.
  - Setting the RESFCR.RESFCR0 bit to 1 clears the RESFR.RESFR0 bit.
- Power-up reset PURES (POCRES or  $\overline{DBRES}$ )

### 8.4.8 Clock Monitor (CLMA) Reset

The clock monitors can generate the following resets:

- $\overline{\text{CLMA0RES}}$ , if a frequency abnormality in HS IntOSC is detected
- $\overline{\text{CLMA1RES}}$ , if a frequency abnormality in MainOSC is detected
- $\overline{\text{CLMA2RES}}$ , if a frequency abnormality in PLL is detected

When the Clock Monitor detects frequency abnormality of the respective clocks, resets  $\overline{\text{CLMA0RES}}$ ,  $\overline{\text{CLMA1RES}}$ , and  $\overline{\text{CLMA2RES}}$  are generated.

In addition, flags  $\overline{\text{CLMA0RES}}$ ,  $\overline{\text{CLMA1RES}}$ , and  $\overline{\text{CLMA2RES}}$  (RESF.RESF3, RESFR.RESFR3, RESF.RESF4, RESFR.RESFR4, RESF.RESF5, and RESFR.RESFR5) are set.

These flags are not cleared automatically. They are cleared as described below.

- $\overline{\text{CLMA0RES}}$ :  
Setting the RESFC.RESFC3 bit to 1 clears the RESF.RESF3 bit.  
Setting the RESFCR.RESFCR3 bit to 1 clears the RESFR.RESFR3 bit.
- $\overline{\text{CLMA1RES}}$ :  
Setting the RESFC.RESFC4 bit to 1 clears the RESF.RESF4 bit.  
Setting the RESFCR.RESFCR4 bit to 1 clears the RESFR.RESFR4 bit.
- $\overline{\text{CLMA2RES}}$ :  
Setting the RESFC.RESFC5 bit to 1 clears the RESF.RESF5 bit.  
Setting the RESFCR.RESFCR5 bit to 1 clears the RESFR.RESFR5 bit.
- Power-up reset PURES (POCRES or  $\overline{\text{DBRES}}$ )

### 8.4.9 Debugger Reset

Debugger reset ( $\overline{\text{DBRES}}$ ) is generated via a debugger command.  $\overline{\text{DBRES}}$  activates PURES, and therefore operates in the same way as the power-on-clear reset POCRES:

- The clock generators are reset and stops operating. The clock generators should be restarted after release from the reset state.
- The reset factor register RESF and the redundant reset factor register RESFR are cleared (Bits RESF9 and RESFR9 are set to 1 after initialization.).

## Section 9 Supply Voltage Monitor

This section explains in general about the supply voltage monitor.

The first part in this section describes the supply voltage monitor function, and the ensuing sections describe the registers.

### 9.1 Overview

#### 9.1.1 Functional Overview

The supply voltage monitor continuously monitors multiple external and internal supply voltages in order to ensure that the device operates with a supply voltage within the specified range. If the voltage drops below the reference voltage or comparison voltage, an interrupt request signal or internal reset signal is generated. The following table lists the supply voltage monitor functions.

**Table 9.1 Supply Voltage Monitor Functions**

Function Name	Monitor Voltage	Signal Generated when Voltage Drops below Certain Level
Power-On Clear (POC)	REGVCC	Internal reset signal
Low-Voltage Indicator Circuit (LVI)	REGVCC	Internal reset signal, interrupt request signal
Core Voltage Monitor (CVM)	AWO, ISO area voltage	Internal reset signal
RAM Retention Voltage Indicator (VLVI)	REGVCC	—

**Note:** The RAM Retention Voltage Indicator sets the very-low voltage detection flag (VLVF) when the voltage drops below the RAM retention voltage.

#### 9.1.2 Power-On Clear (POC)

The POC continuously monitors the external power supply voltage REGVCC. This ensures that the microcontroller only operates at or above power-on-clear detection voltage ( $V_{POC}$ ).

If REGVCC falls below the POC detection voltage ( $REGVCC < V_{POC}$ ), the internal reset signal (POCRES) is generated.

For details, see **Section 8.4.2, Power-On-Clear (POC) Reset**.

#### 9.1.3 Low Voltage Indicator Circuit (LVI)

The LVI continuously compares the external power supply voltage REGVCC with the LVI reference voltage  $V_{LVI}$ .

If REGVCC falls below the reference voltage ( $REGVCC < V_{LVI}$ ), an internal reset signal or interrupt request signal is generated.

##### 9.1.3.1 LVI Reference Voltage

The LVI reference voltage  $V_{LVI}$  can be selected from three different levels by LVICNT.LVICNT[1:0].

If LVICNT.LVICNT[1:0] is set to 00<sub>B</sub>, the LVI is disabled.

For the specifications of the reference voltage level  $V_{LVI}$ , see **Section 9.2.2.1, LVICNT — LVI Control Register**.

### 9.1.3.2 LVI Reset (LVIRES)

When the LVI detection voltage is set and LVIRESMK is cleared, if REGVCC falls below the reference voltage ( $\text{REGVCC} < V_{\text{LVI}}$ ), the internal reset signal  $\overline{\text{LVIRES}}$  is generated.

For the specifications of  $\overline{\text{LVIRES}}$  generation, see **Section 8.4.3, Low-Voltage Indicator (LVI) Reset**.

### 9.1.3.3 LVI Interrupt (INTLVIL / INTLVIH)

After the LVI detection voltage is set to the LVICNT.LVICNT[1:0] and the LVICNT.LVIRESMK is set to 1, if REGVCC falls below the reference voltage ( $\text{REGVCC} (\text{MIN}) < V_{\text{LVI}}$ ), the LVI interrupt INTLVIL is generated.

To use the LVI as an interrupt source, the INTLVIL interrupt must be unmasked.

INTLVIL interrupt can be used as wake-up source from all standby modes. For details, see **Section 11, Stand-By Controller (STBC)**.

After the LVI detection voltage is set to the LVICNT.LVICNT[1:0] and the LVICNT.LVIRESMK is set to 1, if REGVCC exceeds the reference voltage ( $\text{REGVCC} (\text{MIN}) > V_{\text{LVI}}$ ), LVI interrupt INTLVIH is generated.

When LVI is used as an interrupt source, INTLVIH interrupt must be unmasked.

The following figure illustrates the timing of INTLVIL / INTLVIH.

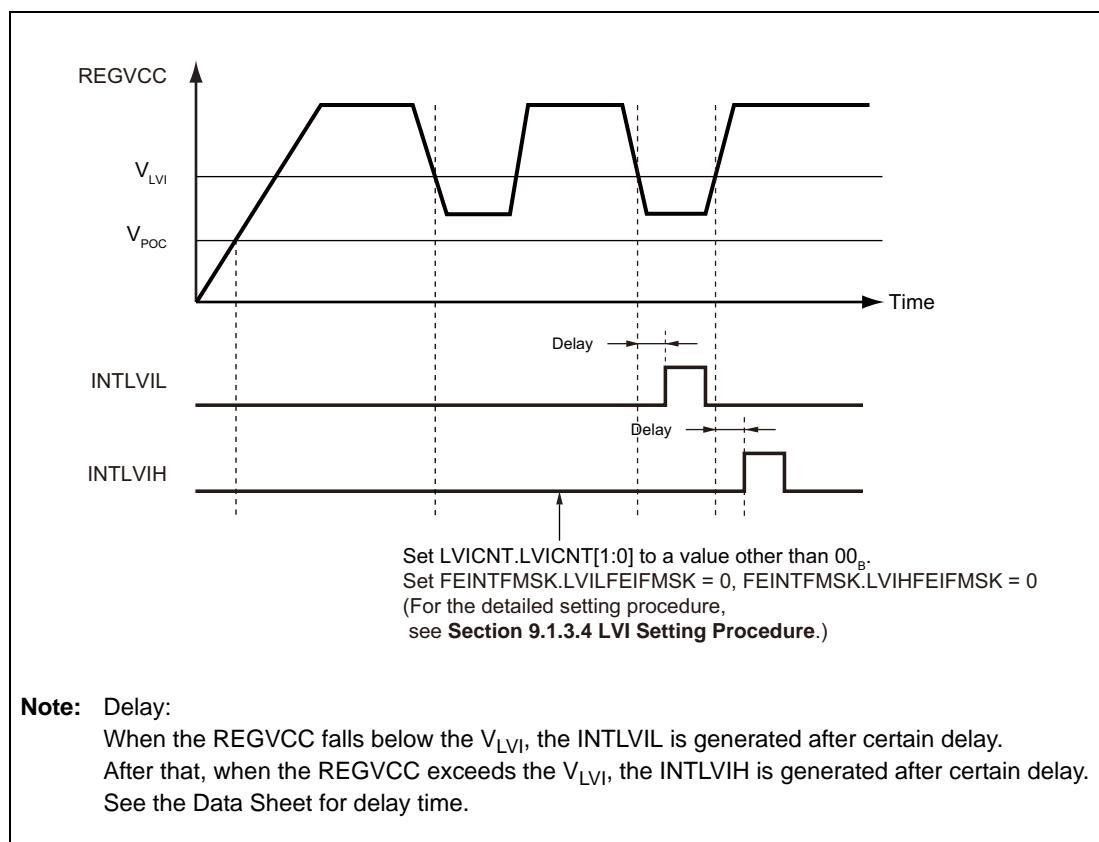


Figure 9.1 INTLVIL / INTLVIH Generation Timing

### 9.1.3.4 LVI Setting Procedure

The setting procedures for LVI are shown below.

#### (1) Using LVI as the reset source

- (1) Mask LVI reset. (LVICNT.LVIRESMK = 1)<sup>\*1</sup>
- (2) Mask LVI interrupt. (FEINTFMSK.LVLFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)
- (3) Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])<sup>\*1</sup>
- (4) Insert ample wait time by software (see Data Sheet).
- (5) Unmask LVI reset. (LVICNT.LVIRESMK = 0)<sup>\*1</sup>

#### (2) Using LVI as the interrupt source (FEINT)

- (1) Mask LVI reset. (LVICNT.LVIRESMK = 1)<sup>\*1</sup>
- (2) Mask LVI interrupt. (FEINTFMSK.LVLFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)
- (3) Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])<sup>\*1</sup>
- (4) Insert ample wait time by software (see Data Sheet).
- (5) Unmask LVI interrupt. (FEINTFMSK.LVLFEIFMSK = 0, FEINTFMSK.LVIHFEIFMSK = 0)

**Note 1.** Follow the register protection sequence to set LVICNT register because it is a write-protected register. For details on the write-protected registers, see **Section 4, Write-Protected Registers**.

#### CAUTION

If REGVCC is not stable around the LVI detection level ( $V_{LVI}$ ), correct judgment of whether INTLVIH or INTLVIL interrupt processing should proceed may not be possible. For example, if multiple interrupts consisting of both INTLVIH and INTLVIL occur during INTLVIL interrupt processing due to REGVCC being unstable, the software cannot detect which type of interrupt was generated last.

Consequently, if the last interrupt generated was an INTLVIL interrupt, regardless of REGVCC (min.) being greater than  $V_{LVI}$ , the software erroneously judges that REGVCC (min.) <  $V_{LVI}$ .

Accordingly, take measures such as programming the software so that LVI detection interrupt processing is completed before the next LVI interrupt. Also, consider control of REGVCC.

### 9.1.3.5 Clock Supply to the LVI

The clock supply to the LVI is shown in the following table.

Table 9.2 Clock Supply to the LVI

Unit Name	Unit Clock Name	Supply Clock Name
LVI	Register access clock	EMCLK

### 9.1.4 Core Voltage Monitor (CVM)

The core voltage monitor (CVM) monitors the AWO area and ISO area voltages (referred to as “core voltage” below) in the microcontroller.

If the regulator output voltage is outside the specified range, the internal reset signal (CVMRES) is generated.

If the CVM detects an abnormal high voltage, the power supply to the ISO area is switched off in addition to a reset being generated.

When operation shifts to diagnostic mode (DIAG mode), the CVM enters the abnormal core voltage detection state. An abnormal core voltage detected state can be intentionally created by using the DIAG mode so that the CVM abnormal voltage detected flag can be checked for failures.

#### **CAUTION**

---

**The CVM cannot detect drifts in the voltage being supplied to the on-chip voltage regulator, AWO area, and ISO area, or the increase or decrease of voltage.**

---

#### 9.1.4.1 CVM Reset (CVMRES)

If the core voltage exceeds the specified level while high-voltage monitor is enabled (CVMDE.H\_D\_E = 1), then CVMRES is generated and the power supply to the isolated area is stopped.

If the core voltage falls below the specified level while low voltage monitor is enabled (CVMDE.L\_D\_E = 1), CVMRES is generated.

For the specifications of CVMRES generation, see **Section 8.4.4, Core Voltage Monitor (CVM) Reset**.

#### 9.1.4.2 CVM Setting

Use the option byte to enable the high-voltage monitor and the low-voltage monitor. For details, see **Section 35.11, Option Bytes**.

#### 9.1.4.3 Diagnostic (DIAG) Mode

This product supports diagnostic mode.

In diagnostic (DIAG) mode, whether the CVM abnormal voltage detection flag is set to 1 can be checked.

In diagnostic mode, CVMRES is not output.

The setting procedure for diagnostic mode is described below.

Set the registers according to this procedure. Otherwise the operation is not guaranteed.

- (1) Set CVMDIAG.CVM\_DIAG\_MASK.\*<sup>1</sup>
- (2) Set CVMDIAG.CVM\_DIAG.\*<sup>1</sup>
- (3) Wait for 12 µs.\*<sup>2</sup>
- (4) Read the CVMF register to confirm that the H\_V\_F and L\_V\_F bits are set to 1 (if these bits are 0, the CVM does not operate normally, requiring error handling).

- (5) Clear CVMDIAG.CVM\_DIAG.\*<sup>1</sup>
- (6) Clear the CVMF register.\*<sup>1</sup>
- (7) Read the CVMF register to confirm that the H\_V\_F and L\_V\_F bits are set to 0 (if these bits are 1, go back to step 5 again).
- (8) Clear CVMDIAG.CVM\_DIAG\_MASK.\*<sup>1</sup>

**Note 1.** Follow the register protection sequence to set CVMF and CVMDIAG registers because these are write-protected registers. For details, see **Section 4, Write-Protected Registers**.

**Note 2.** At least 50 µs must elapse after the following conditions are fulfilled before step (4) is started.

- Release from HALT mode
- Release from STOP mode
- Release from the reset state, when a reset other than a reset due to the CVM is generated in RUN/HALT mode
- The CPU clock is switched
- Operation of the MainOSC is started or stopped
- Operation of the PLL is started or stopped

#### 9.1.4.4 Clock Supply to the CVM

The clock supply to the CVM is shown in the following table.

Table 9.3     Clock Supply to the CVM

Unit Name	Unit Clock Name	Supply Clock Name
CVM	Register access clock	CPUCLK2

### 9.1.5 RAM Retention Voltage Indicator (Very-Low-Voltage Detection Circuit:VLVI)

The very-low-voltage detection circuit (VLVI) is used to detect the RAM retention voltage, and continuously compares the power supply voltage REGVCC with the RAM retention voltage  $V_{VLVI}$ .

See the Data Sheet for the specifications of the RAM retention voltage level  $V_{VLVI}$ .

#### 9.1.5.1 Clock Supply to the VLVI

The clock supply to the VLVI is shown in the following table.

**Table 9.4 Clock Supply to the VLVI**

Unit Name	Unit Clock Name	Supply Clock Name
VLVI	Register access clock	EMCLK

#### 9.1.5.2 Retention RAM Content Retention

If the power supply voltage REGVCC does not fall below  $V_{VLVI}$ , the content of the Retention RAM (RRAM) is retained.

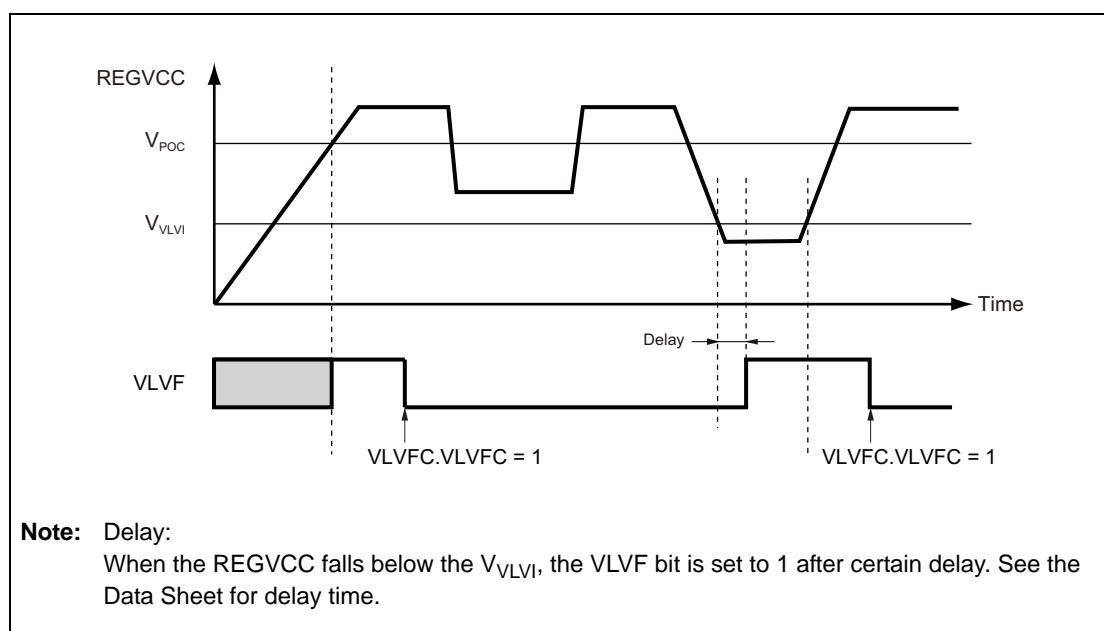
If REGVCC falls below  $V_{VLVI}$ , the RRAM content cannot be guaranteed. Thus the entire RRAM must be restored before continuing operation.

If REGVCC falls below the RAM retention voltage ( $REGVCC < V_{VLVI}$ ), the VLVF.VLVF bit is set.

After that, even if REGVCC exceeds  $V_{VLVI}$ , the VLVF.VLVF bit is not cleared automatically. It is cleared by

- setting VLVFC.VLVFC bit to 1.

The following figure illustrates the timing of VLVF.



**Figure 9.2 VLVF Operation Timing**

### 9.1.6 Block Diagram

The block diagram of the supply voltage monitor is shown below.

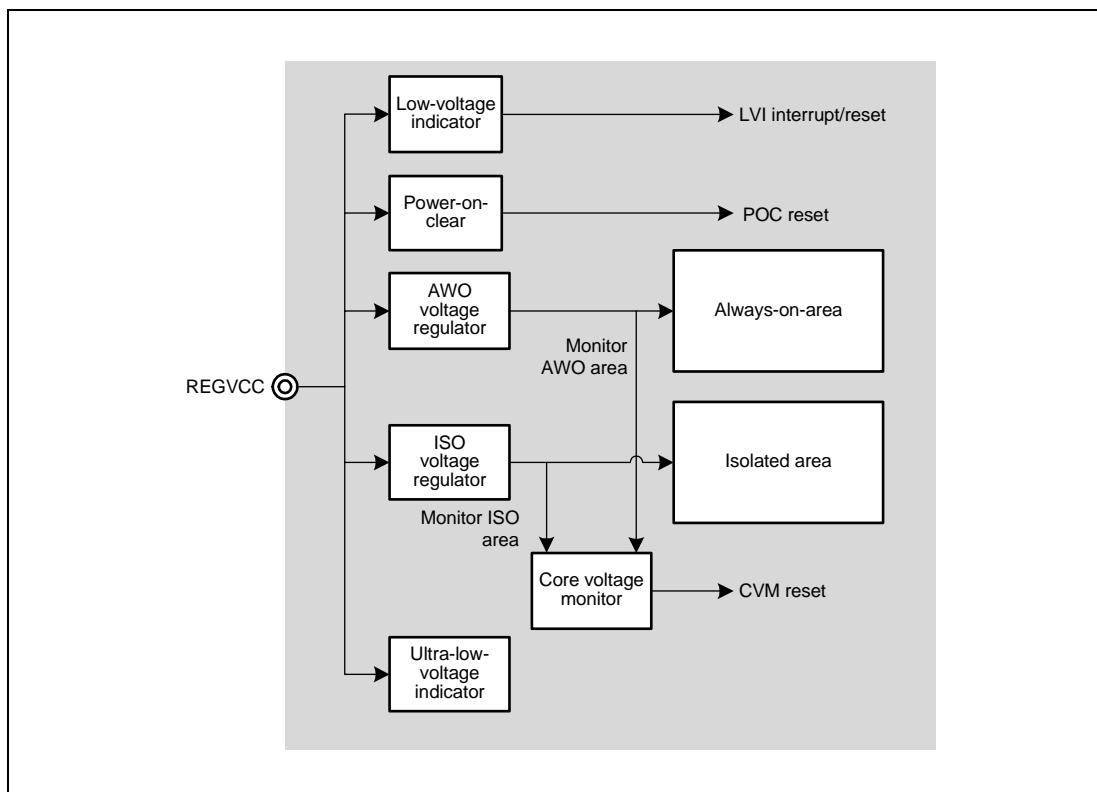


Figure 9.3 Supply Voltage Monitor

## 9.2 Registers

### 9.2.1 List of Registers

The following table lists the supply voltage monitor registers.

Table 9.5 Power Supply Voltage Monitor Registers

Register Name	Symbol	Address
<b>Low-voltage indicator reset control register</b>		
LVI control register	LVICNT	FFF8 0A00 <sub>H</sub>
<b>Core voltage monitor control register</b>		
CVM factor register	CVMF	FFF5 0000 <sub>H</sub>
CVM detection enable register	CVMDE	FFF5 0004 <sub>H</sub>
CVM diagnostic mode setting register	CVMDIAG	FFF5 0014 <sub>H</sub>
<b>Very-low-voltage detection control register</b>		
Very-low-voltage detection register	VLVF	FFF8 0980 <sub>H</sub>
Very-low-voltage detection clear register	VLVFC	FFF8 0988 <sub>H</sub>

## 9.2.2 Low-Voltage Indicator Reset Control Registers

### 9.2.2.1 LVICNT — LVI Control Register

This register is used to control the Low-Voltage Indicator and to select the LVI detection level.

This register is initialized by a power-up reset PURES.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 4, Write-Protected Registers**.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 0A00<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

**Table 9.6 LVICNT Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LVIRESMK	Mask LVI Reset 0: LVI reset is not masked 1: LVI reset is masked
1, 0	LVICNT[1:0]	Detection Level 0 0: LVI is disabled 0 1: 4.0+/-0.1 V (drop), 4.0+/-0.13 V (rise) 1 0: 3.7+/-0.1 V (drop), 3.7+/-0.13 V (rise) 1 1: 3.5+/-0.1 V (drop), 3.5+/-0.13 V (rise)

#### NOTE

To use an LVI interrupt, LVI reset must be masked (LVIRESMK = 1) by LVIRESMK.

## 9.2.3 Core Voltage Monitor Control Registers

### 9.2.3.1 CVMF — CVM Factor Register

This register records the core voltage failure state generated after the last POC reset.

If the L\_V\_F or H\_V\_F bit of this register is set to 1, that bit is not updated until it is initialized by a power-on-clear or by writing 0 to the CVMFL\_V\_F or CVMFH\_V\_F bit. However, it continuously monitors an error signal from the core voltage monitoring circuit in diagnostic mode.

The correct write sequence using the PROTCMDCVM register is required in order to update this register. For details, see **Section 4, Write-Protected Registers**.

	<b>Access:</b> This register can be read or written in 32-bit units.															
	<b>Address:</b> FFFF 0000 <sub>H</sub>															
	<b>Value after reset:</b> 0000 0000 <sub>H</sub>															
<b>Bit</b>																
31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
30	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
29	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
28	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
27	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
26	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
25	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
24	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
23	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
22	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
21	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
19	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Bit</b>																
15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_V_F	L_V_F
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 9.7 CVMF Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	H_V_F	High-Voltage Failure Detection of the Core Voltage by the CVM 0: No high-voltage failure state is detected 1: High-voltage failure state is detected
0	L_V_F	Low-Voltage Failure Detection of the Core Voltage by the CVM 0: No low-voltage failure state is detected 1: Low-voltage failure state is detected

### 9.2.3.2 CVMDE — CVM Detection Enable Register

This register is used to indicate the voltage detection enabled or disabled state. This register is initialized only by the Power-On-Clear.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF5 0004<sub>H</sub>

**Value after reset:** The value after reset depends on the option byte setting.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	H_D_E	L_D_E	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1 <sup>*1</sup>	0/1 <sup>*2</sup>
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The setting of the option byte OPBT0.CVM\_HD\_EN is reflected.

Note 2. The setting of the option byte OPBT0.CVM\_LD\_EN is reflected.

For details on the option byte, see [Section 35.11, Option Bytes](#).

**Table 9.8 CVMDE Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	H_D_E	High-Voltage Monitor Enable 0: High-voltage detection is disabled. 1: High-voltage detection is enabled.
0	L_D_E	Low-Voltage Monitor Enable 0: Low-voltage detection is disabled. 1: Low-voltage detection is enabled.

### 9.2.3.3 CVMDIAG — CVM Diagnostic Mode Setting Register

This register sets the CVM diagnostic mode.

This register is initialized only by the Power-On-Clear.

For details on the register settings in diagnostic mode, see **Section 9.1.4.3, Diagnostic (DIAG) Mode.**

The correct write sequence using the PROTCMDCVM register is required in order to update this register. For details, see **Section 4, Write-Protected Registers.**

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF5 0014<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVM DIAG MASK	CVM DIAG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 9.9 CVMDIAG Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CVM_DIAG_MASK	CVMRES Mask Control 0: CVMRES is not masked. 1: CVMRES output is masked.
0	CVM_DIAG	CVM Diagnostic Mode Setting 0: Normal mode 1: Diagnostic mode

## 9.2.4 Very-Low-Voltage Detection Control Registers

### 9.2.4.1 VLVF — Very-Low-Voltage Detection Register

The very-low-voltage detection register (VLVF) shows the state of the RAM retention voltage detection.

This register is set upon detection of a voltage below the RAM retention voltage ( $V_{VLF}$ ).

If VLVF is set, the Retention RAM content cannot be guaranteed.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 0980<sub>H</sub>

**Value after power-on:** 0000 0001<sub>H</sub>

This register is not initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.10 VLVF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	VLVF	Very-Low-Voltage Detection Flag 0: Very-low-voltage is not detected. 1: Very-low-voltage is detected.
<b>Note:</b> Very-low-voltage is the voltage status of REGVCC < RAM retention voltage ( $V_{VLF}$ ). For details, see 9.1.5.2, Retention RAM Content Retention		

### 9.2.4.2 VLVFC — Very-Low-Voltage Detection Clear Register

This register clears the VLVF.VLVF bit.

**Access:** This register is a write-only register that can be written in 32-bit units.

**Address:** FFF8 0988<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

**Table 9.11 VLVFC Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	VLVFC	Clear VLVF.VLVF bit. 0: Do not clear 1: Clear

## Section 10 Clock Controller

This section explains in general about the clock controller.

The first part in this section describes the specific features of the clock controller of the RH850/F1L microcontrollers. The ensuing sections describe the clock oscillators, clock output function, and clock monitor function that make up the clock controller.

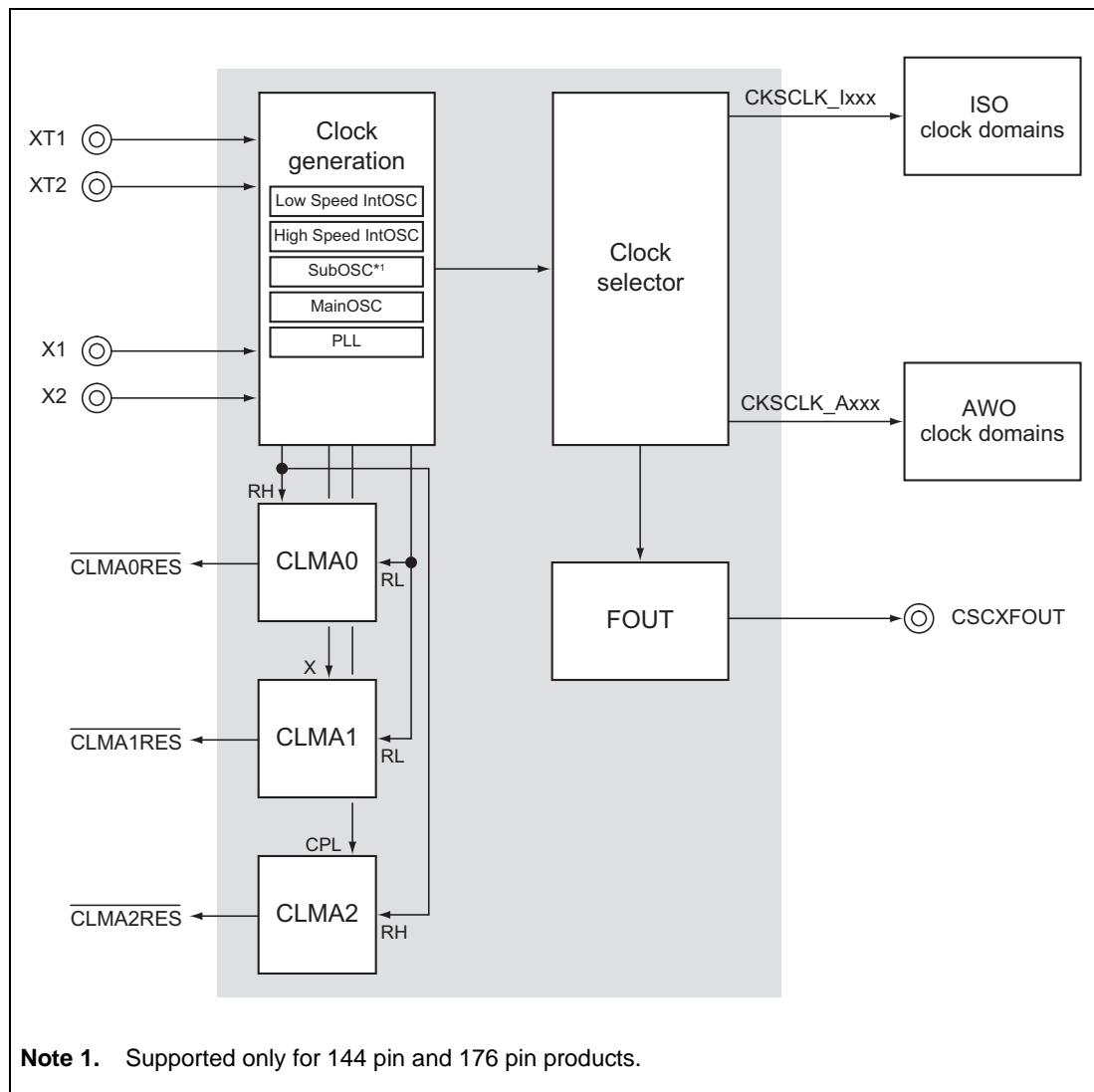
### 10.1 Features of Clock Controller of RH850/F1L

The clock controller of the RH850/F1L microcontrollers has the following features.

- Five on-chip clock oscillators
  - Main Oscillator (MainOSC) with an oscillation frequency of 8 to 24 MHz Source clock for PLL
  - Sub Oscillator (SubOSC) with an oscillation frequency of 32.768 kHz<sup>\*1</sup> Source clock for RTC
  - High Speed Internal Oscillator (HS IntOSC) with a nominal frequency of 8 MHz (Typ.) Clock source of CPU after reset
  - Low Speed Internal Oscillator (LS IntOSC) with a nominal frequency of 240 kHz (Typ.)
  - PLL Clock source for CPU and Peripheral
- Fine management of clock supply to peripheral modules through clock domains
- On-chip clock monitor that detects CPU clock anomalies when the Main Oscillator, High Speed Internal Oscillator, or PLL are in use
- CPU internal clock output (FOUT)

**Note 1.** The Sub Oscillator is supported only for 144 pin and 176 pin products.

**Figure 10.1** shows the schematic diagram of the clock controller.



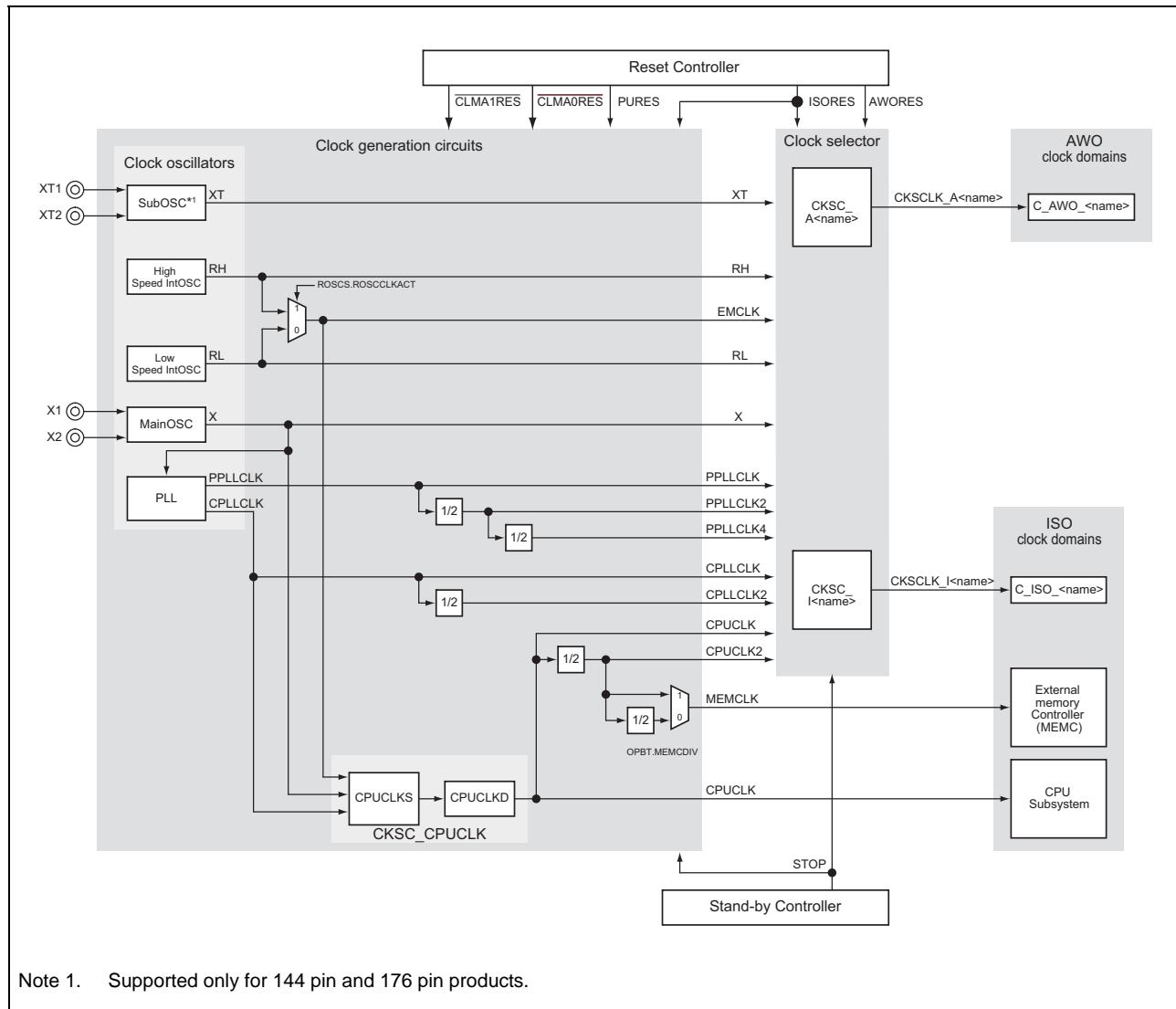
**Figure 10.1** Clock Controller Overview

## 10.2 Configuration of Clock Controller

This section describes the configuration of the clock controller.

The clock controller is composed of clock oscillators and clock generation circuits that generate the clocks for the CPU and the peripheral modules, a clock selector for selecting the optimum clock, and clock domains for the CPU and the peripheral modules.  
时钟选择器用来选择最佳的时钟

**Figure 10.2** shows the configuration of the clock controller.



Note 1. Supported only for 144 pin and 176 pin products.

**Figure 10.2 Clock Controller Configuration**

**NOTE****Clock domain and clock control register naming conventions**

The clock signals and their control registers, etc., described in this section are named according to the following naming conventions to reflect the power domain or clock domain to which they belong. The placeholder “<name>” is used to identify the target module in the clock domain:

- Clock domain names:
  - C\_AWO\_<name>: Always-On-Area<sup>\*</sup> clock domain
  - C\_ISO\_<name>: Isolated-Area<sup>\*</sup> clock domain
- Domain clock names:
  - CKSCLK\_A<name>: Always-On-Area domain clock
  - CKSCLK\_I<name>: Isolated-Area domain clock
- Clock selector names:
  - CKSC\_A<name>: Always-On-Area clock selector
  - CKSC\_I<name>: Isolated-Area clock selector
- Clock selector register names:
  - CKSC\_A<name>S\_CTL: Always-On-Area source clock selector register
  - CKSC\_A<name>D\_CTL: Always-On-Area source clock divider register
  - CKSC\_I<name>S\_CTL: Isolated-Area source clock selector register
  - CKSC\_I<name>D\_CTL: Isolated-Area source clock divider register

**Example 例子**

时钟信号 CKSCLK\_AADCA 所属时钟域 C\_AWO\_AADCA 选择寄存器 CKSC\_AADCAS\_CTL

The clock signal CKSCLK\_AADCA (placeholder <name> = ADCA) is the clock supplied to the clock domain C\_AWO\_AADCA in the Always-On-Area. This clock is selected by the clock selector register CKSC\_AADCAS\_CTL.

**AWO和ISO是指的Power Supply 域，前者持续供电，后者开关供电**

- Note 1.** Always-On-Area and Isolated-Area refer to the power supply domains. Always-On-Area (AWO) is an always-on power supply, and the Isolated\_Area (ISO) is a power supply that is switched on or off according to the operation mode. For details, see **Section 38, Power Supply and Power Domains.**

### 10.2.1 Clock Generation Circuits

Five clock oscillators are provided:

Four clock oscillators are located on the Always-On-Area (AWO) and PLL is located on the Isolated-Area (ISO).

#### Main Oscillator (MainOSC)

The MainOSC generates main clock X.

Generation of main clock X requires the connection of an external resonator to X1 and X2.

The clock X is used as the reference clock for the PLL.

作为PLL的时钟参考源

#### Sub Oscillator (SubOSC)

The SubOSC generates the sub-clock XT, which runs at a frequency of 32.768 kHz (Typ.). Generation of the sub clock XT requires the connection of an external resonator to XT1 and XT2.

This clock is mainly used for real-time clock applications.

用于RTC

#### High Speed Internal Oscillator (HS IntOSC)

The HS IntOSC generates a clock RH, which runs at a frequency of 8 MHz (Typ.).

#### Low Speed Internal Oscillator (LS IntOSC)

The LS IntOSC generates the clock RL, which runs at a frequency of 240 kHz (Typ.). It starts operation at power up and cannot be stopped, hence it is always operating.

随上电便开始工作并且不会停止，因此它总是在工作的

#### PLL (用于倍频和分频)

The PLL circuit generates high speed operation clocks CPLLCLK and PPLLCLK required for normal operation of the microcontroller.

The clocks supplied by the clock oscillators (X, XT, RH, RL, CPLLCLK, PPLLCLK) and their divided clocks (CPLLCLK2, PPLLCLK2, PPLLCLK4) are all generated in the clock generation circuits.

## 10.2.2 Clock Selection

The clocks generated by the clock oscillators are input to the clock selectors CKSC\_A<name>/CKSC\_I<name>.

Domain clocks CKSCLK\_A<name>/CKSCLK\_I<name> are selected by dedicated clock selectors from clocks directly input from the oscillators, or in some cases from clocks that have been divided by clock dividers.

- CKSC\_I<name>S\_CTL/CKSC\_I<name>D\_CTL registers: determine the clocks for the Isolated-Area clock domains.
- CKSC\_A<name>S\_CTL/CKSC\_A<name>D\_CTL registers: determine the clocks for the Always-On-Area clock domains.

Note that not all available clocks generated by the clock oscillators are input to each clock selector.  
并不是所有的有时钟晶振产生的可获得时钟都会被输入各自的Clock selector

The following clocks are supplied to the CPU and related modules from the clock generation circuits.

### Emergency Clock (EMCLK)

The emergency clock EMCLK is supplied by the

- HS IntOSC, if it is active
- LS IntOSC, if HS IntOSC is inactive

The selection is done automatically, so if the HS IntOSC is stopped for any reason, vital modules of the microcontroller are still in operation, since the LS IntOSC does not stop.

### CPU Subsystem Clock (CPUCLK)

CPU Subsystem clock CPUCLK 可以从CPLLCLK , MainOSC和EMCLK获取  
The CPU Subsystem clock CPUCLK is derived from PLL clock CPLLCLK, MainOSC, and EMCLK. The CPU clock selector CKSC\_CPUCLK incorporates the selector CPUCLKS, followed by the clock divider CPUCLKD. CKSC\_CPUCLK 包含CPUCLKS和CPUCLKD

The CPUCLK clock divider provides the frequency-divided CPUCLK2 clock signal derived from CPUCLK.

### External Memory Controller Clock (MEMCLK)

The operating clock for the external memory controller (MEMCLK) is provided by the CPUCLK clock divider.

The division ratio is selected by the option byte OPBT.MEMCDIV.

- MEMCDIV = 0:  $f_{MEMCLK} = f_{CPUCLK} / 2$
- MEMCDIV = 1:  $f_{MEMCLK} = f_{CPUCLK} / 4$

这个跟上面的框图是不由冲突？

The maximum frequency of MEMCLK is 40 MHz.

### 10.2.3 Clock Domains

The clock controller allows selection of the respective clocks for the CPU and peripheral modules. The clock control scope is called the clock domain. For the correspondence between the CPU and peripheral modules and clock domains, see **Section 10.5.3, Clock Domain Settings**.

### 10.2.4 Resetting Clock Oscillators

The clock oscillators on the Always-On-Area are reset by the PURES signal.

The HS IntOSC is reset when  $\overline{\text{CLMA0RES}}$  is generated and the MainOSC is reset when  $\overline{\text{CLMA1RES}}$  is generated.

The clock oscillator on the Isolated-Area is reset by the ISORES signal.

For further details on the clock oscillators, see **Section 10.3, Clock Oscillators**.

---

**CAUTION**

---

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, see the Data Sheet.

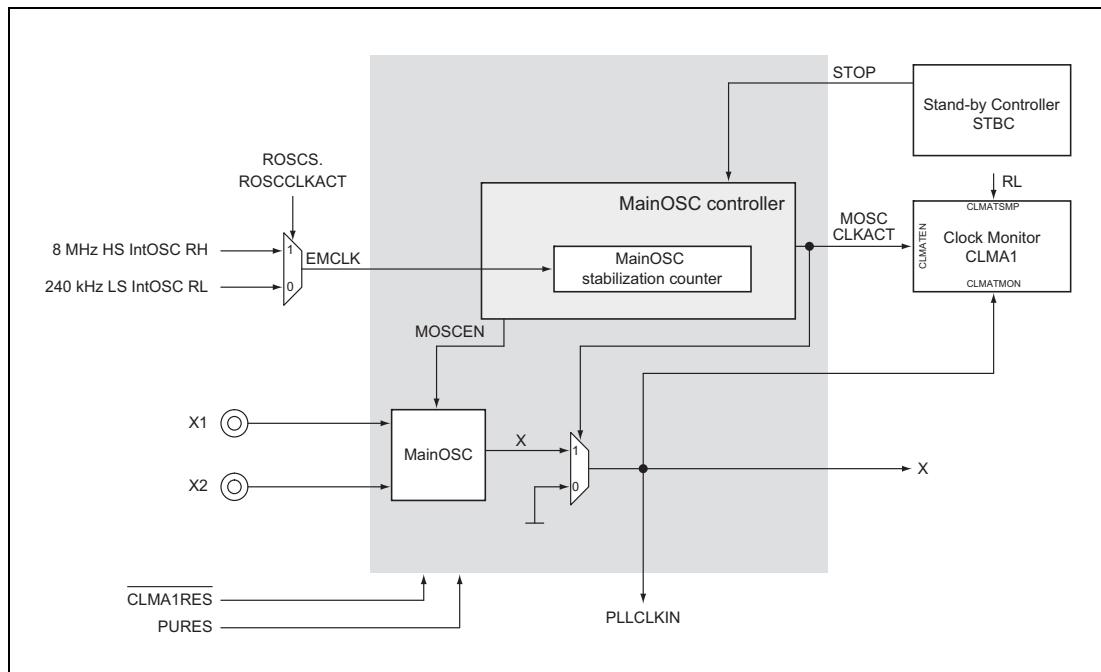
---

## 10.3 Clock Oscillators

### 10.3.1 Main Oscillator (MainOSC)

The Main Oscillator generates the clock X. X is also used as the PLL input clock PLLCLKIN.

**Figure 10.3** shows the basic configuration and signals of the MainOSC.



**Figure 10.3 Main Oscillator (MainOSC)**

#### MainOSC

The MainOSC stops operating after reset is released. To use the MainOSC, set the MainOSC enable trigger bit (MOSCE.MOSCENTRG) to 1 to start the MainOSC.

#### MainOSC stabilization 此位是设置MainOSC 晶振稳定下来的时间

The MOSCST.MOSCCLKST[16:0] bits set the MainOSC oscillation stabilization time.

The MainOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting. The oscillation stabilization time can be set to up to 最大可以达到  $2^{16} - 1$  EMCLK cycles.

As long as the MainOSC is not stable, the MOSCCLKACT signal disables the X output.

When the MainOSC stabilization counter reaches the value specified in MOSCST.MOSCCLKST[16:0], X is assumed to be stable and MOSCCLKACT switches from 0 to 1 to enable output of X when a waveform is output from MainOSC.

Stable and active X clock is indicated by MOSCS.MOSCCLKACT = 1.

时钟稳定和激活X时钟是由MOSCS. MOSCCLKACT=1来指示的

#### MainOSC amplification gain

By using MOSCC.MOSCAMPSEL[1:0], the MainOSC's input frequency, determined by the external resonator, can be selected in the range of 8 MHz to 24 MHz.

外部晶振可以使用的范围在8M-24MHz

### MainOSC STOP requests in stand-by mode

The STOP signal from the Stand-by Controller requests the MainOSC Controller to switch off the X clock in stand-by mode. **STOP信号可以关闭X clock (在待机模式下)**

The stop request mask bit **MOSCSTPM.MOSCSTPMSK** controls whether the MainOSC is stopped during stand-by or continues operation:

- MOSCSTPM.MOSCSTPMSK = 0: **MainOSC在待机模式下停止运行**  
The STOP request signal is not masked, so the MainOSC is stopped during stand-by.  
**如果在待机模式之前MainOSC是正常运行的，那么在Wakeup以后会自动重启，然后**  
If the MainOSC was in operation before stand-by, it is automatically re-started after wake-up from MainOSC 稳定时钟开始计算稳定的时间 and the MainOSC stabilization timer counts the oscillation stabilization time.
- MOSCSTPM.MOSCSTPMSK = 1: **MainOSC在待机模式下可以继续运行**  
The STOP request signal is masked, so the MainOSC continues to operate during stand-by.

控制MainOSC在待机模式下  
是停止工作还是继续运行

### Clock monitor control

The MainOSC activity signal MOSCCLKACT enables or disables supervision by the Clock Monitor CLMA1. In case the MainOSC is inactive (MOSCCLKACT = 0), supervision of its output clock X by CLMA1 is also disabled.

### MainOSC enable/disable trigger

The MainOSC can be enabled and disabled by the **MOSCE.MOSCENTRG** and **MOSCE.MOSCDISTRG** bits:

开启MainOSC，这种设置只有在  
MainOSC inactive的情况下才有效，  
比如MOSCS.MOSCCLKACT=0

- Enable trigger **MOSCE.MOSCENTRG** = 1 starts the MainOSC.  
Note that setting the enable trigger is only effective if the MainOSC is inactive, i.e. if MOSCS.MOSCCLKACT = 0.
- Disable trigger **MOSCE.MOSCDISTRG** = 1 stops the MainOSC.  
Note that setting the disable trigger is only effective if the MainOSC is active (MOSCS.MOSCCLKACT = 1).

关闭MainOSC，这种设置  
是由在MainOSC active  
的情况下才有效，比如  
MOSCS.MOSCCLKACT=1

### 10.3.2 Sub Oscillator (SubOSC)

The Sub Oscillator generates the sub clock XT. XT typically has a frequency of 32.768 kHz and is used for the Real-time Clock.

**用于RTC**

Figure 10.4 shows the basic configuration and signals of the SubOSC.

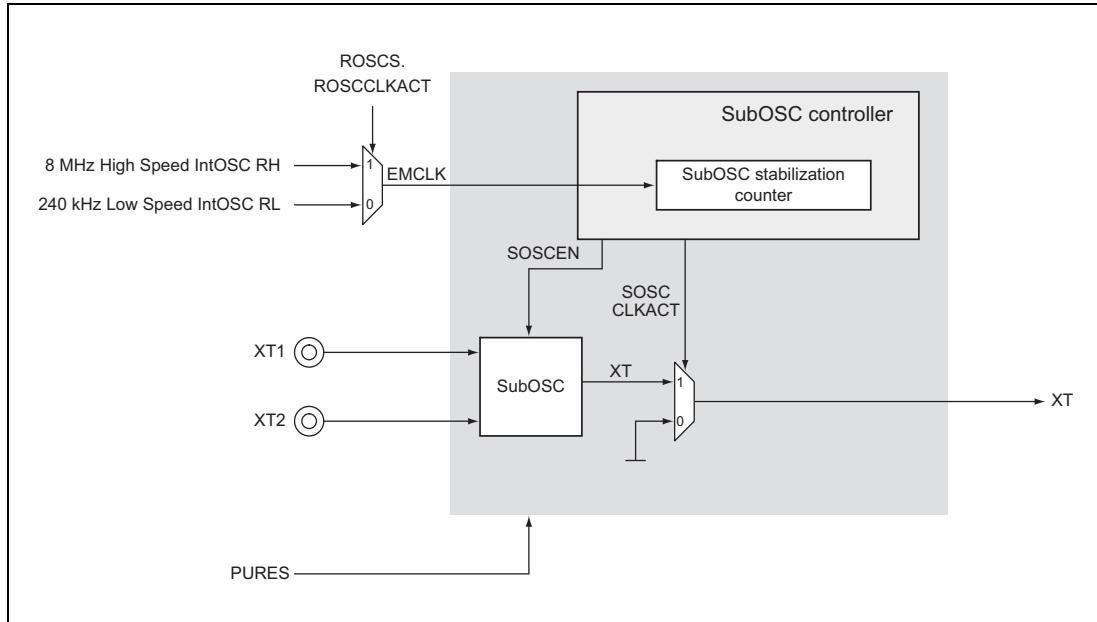


Figure 10.4 Sub Oscillator (SubOSC)

#### SubOSC enable

在reset被复位后SubOSC就停止了

如果使用SubOSC，设置SubOSC的使能位（SOCCE.SOSGENTRG）为1

The SubOSC stops operating after reset is released. To use the SubOSC, set SubOSC enable trigger bit (SOSCE.SOSCENTRG) to 1 to start the SubOSC.

#### SubOSC stabilization

设置SubOSC的稳定时间

The SOSCST.SOSCCLKST[29:0] bits set the SubOSC oscillation stabilization time.

The SubOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting. 在SubOSC没有稳定之前，SOSCCLKACT信号禁止XT的输出

As long as the SubOSC is not stable, the SOSCCLKACT signal disables the XT output.

When the SubOSC stabilization counter reaches the value specified in SOSCST.SOSCCLKST[29:0],

XT is assumed to be stable and SOSCCLKACT switches from 0 to 1 to enable output of XT.

Secure the stabilization time longer than 2 seconds. 安全的稳定时间要长于2s

Stable and active XT clock is indicated by SOSCS.SOSCCLKACT = 1. SOSCS.SOSCCLKACT=1证明XT clock已经激活

#### SubOSC input frequency

The SubOSC input frequency is 32.768 kHz (Typ.).

#### SubOSC enable trigger/disable trigger

开启SubOSC，这种设置只有在SubOSC inactive的情况下才有效，比如SOSCS.SOSCCLKACT = 0

SubOSC can be enabled or disabled by using enable/disable trigger control bit.

- Enable trigger SOSCE.SOSCENTRG = 1 starts the SubOSC.

Note that setting the enable trigger is only effective if the SubOSC is inactive, i.e. if SOSCS.SOSCCLKACT = 0.

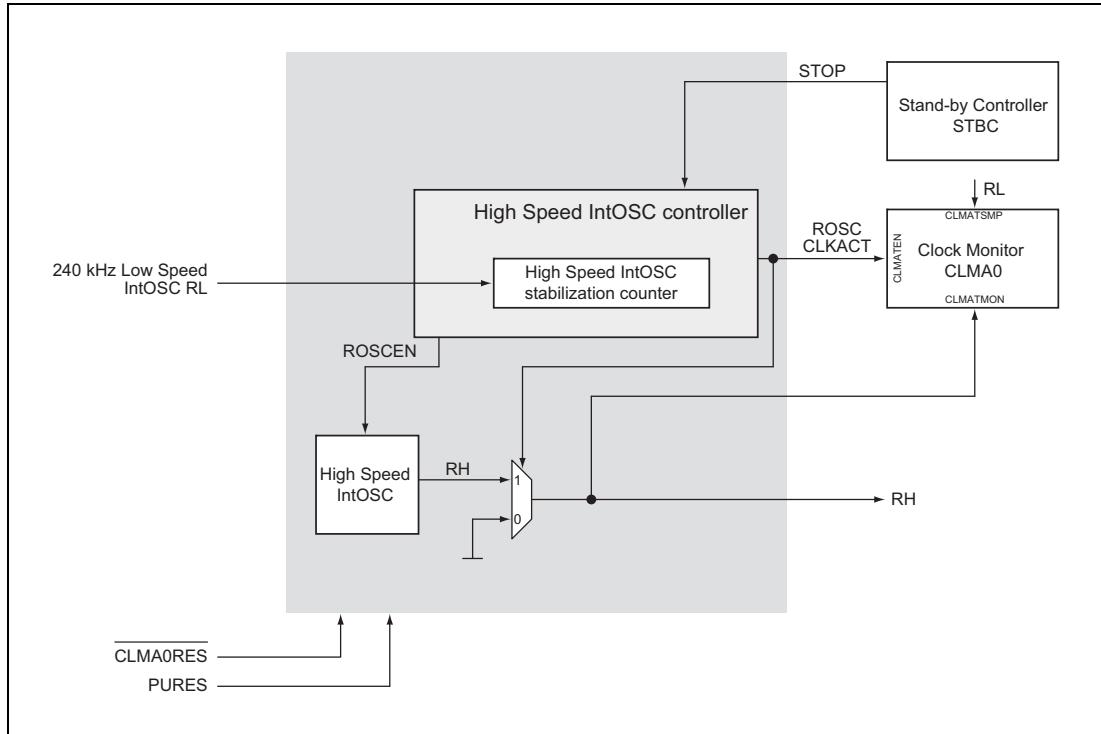
停止SubOSC，这种设置只有在  
SubOSC active的情况下才会有效，  
比如SOSCS.SOSCCLKACT = 1

- Disable trigger SOSCE.SOSCDISTRG = 1 stops the SubOSC.  
Note that setting the disable trigger is only effective if the SubOSC is active; that is, if SOSCS.SOSCCLKACT = 1.

### 10.3.3 High Speed Internal Oscillator (HS IntOSC)

The High Speed Internal Oscillator generates the clock RH. RH has a nominal frequency of 8 MHz.

**Figure 10.5** shows the basic configuration and signals of the HS IntOSC.



**Figure 10.5 High Speed Internal Oscillator (HS IntOSC)**

复位之后HS IntOSC开始工作

After reset release the HS IntOSC starts operation.

**NOTE**

HS IntOSC不能被软件操作启动或者停止

HS IntOSC在待机模式下可以被停掉

The HS IntOSC can neither be stopped nor started by software. It can only be stopped in stand-by mode. On the other hand, when CLMA0 is reset, the HS IntOSC can be stopped by software. 另外一方面，如果CLMA0被复位，HS IntOSC可以被软件停掉

#### HS IntOSC stabilization

HS IntOSC outputs RH when it is stabilized.

Stable and active RH clock is indicated by ROSCS.ROSCCLKACT = 1.  
ROSCE.ROSCCLKACT = 1表明HS IntOSC已经稳定并且RH时钟已经激活

#### HS IntOSC STOP requests in stand-by mode

STOP信号在待机模式下可以将RH关闭

The STOP signal from the stand-by controller requests the HS IntOSC Controller to switch off the RH clock in stand-by mode.

The stop request mask bit ROSCSTPM.ROSCSTPMASK controls whether the HS IntOSC is stopped during stand-by or continues operation:

可是控制HS IntOSC在待机模式下是继续运行还是停止工作

- **ROSCSTPM.ROSCSTPMSK = 0:** STOP信号不会被屏蔽，在待机模式下HS IntOSC被停止，并且在wakeup之后会自动重启

The STOP request signal is not masked, so the HS IntOSC is stopped during stand-by and automatically restarted after wake-up from stand-by.  
然而在下面的一些情况下，STOP 请求会被屏蔽掉，即使ROSCS, ROSCCLKACT = 0  
However, the STOP request is masked under the following conditions, even if  
ROSCSTPM.ROSCSTPMSK = 0. Therefore, the HS IntOSC will continue to operate even in standby mode.

  - If the stop mask is set (CKSC\_xxxx\_STPM = 0000\_0003<sub>H</sub>) for a clock domain for which the HS IntOSC is selected 对于HS IntOSC所在的时钟域，STOP mask被设置 (CKSC\_XXXX\_STPM=0x00000003)
  - If the low power sampler (LPS) is operating 在LPS状态下
- **ROSCSTPM.ROSCSTPMSK = 1:** STOP信号被屏蔽，在待机模式下HS IntOSC继续运行

The STOP request signal is masked, so the HS IntOSC continues to operate during stand-by.

### Clock Monitor control

The HS IntOSC activity signal ROSCCLKACT enables or disables supervision by the Clock Monitor CLMA0. In case the HS IntOSC is inactive (ROSCCLKACT = 0), supervision of its output clock by CLMA0 is also disabled.

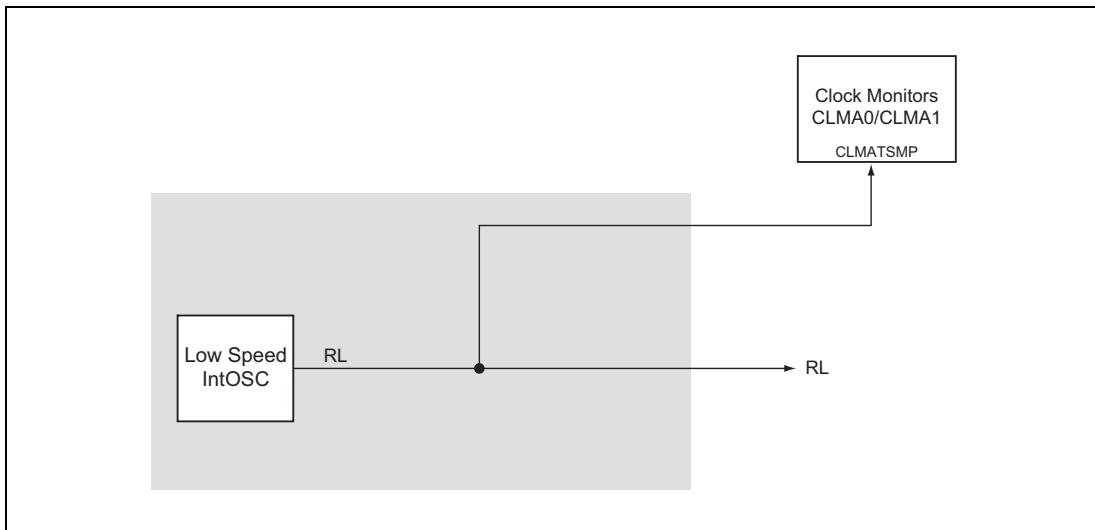
The HS IntOSC clock RH is used as the sampling clock for Clock Monitor CLMA2.  
RH用作CLMA2的采样时钟

**HS IntOSC disable trigger** 禁止掉HS IntOSC  
The disable trigger, ROSCE.ROSCDISTRG = 1, stops the HS IntOSC.

#### 10.3.4 Low Speed Internal Oscillator (LS IntOSC)

The Low Speed Internal Oscillator generates the clock RL. RL has a nominal frequency of 240 kHz.

**Figure 10.6** shows the basic configuration and signals of the LS IntOSC.



**Figure 10.6 Low Speed Internal Oscillator (LS IntOSC)**

当复位后LS IntOSC开始运行，并且不会停止（当然得在有点的情况下）  
After reset release the LS IntOSC starts operation. It cannot be stopped.

The LS IntOSC clock RL is used as the sampling clock for the Clock Monitors CLMA0 and CLMA1.  
RL作为CLMA0 和CLMA1的采样时钟

### 10.3.5 PLL

由Main OSC输出的X被输入到锁相环 (PLL), PLL输出的CPLLCLK和PPLLCLK作为MCU的主时钟使用  
The Main Oscillator clock X is input to the Phase-Locked Loops (PLL) clock oscillator. The PLL output clocks CPLLCLK and PPLLCLK serve as the main operation clocks for the microcontroller.

Figure 10.7 shows the basic configuration and signals of the PLL.

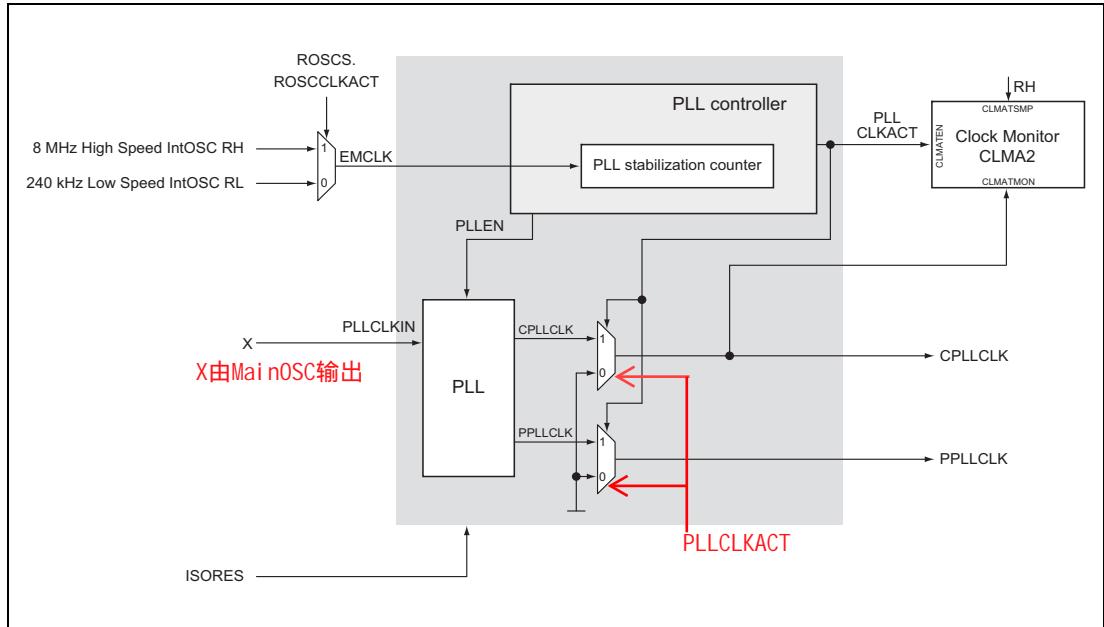


Figure 10.7 PLL

#### PLL enable

The PLL stops operating after reset is released. To use the PLL, set the PLL enable trigger bit (PLLE.PLLENTRG) to 1 to start the PLL. **如果使用PLL，设置 PLLE.PLLENTRG = 1 来开启PLL**

#### PLL stabilization

The PLL stabilization counter starts counting the stabilization time.

As long as the PLL is not stable, the PLLCLKACT signal disables the PPLLCLK and CPLLCLK outputs. **在PLL稳定之前，PLLCLKACT信号会禁止 PPLLCLK和CPLLCLK的输出**

When the PLL stabilization counter reaches the predefined value, PPLLCLK and CPLLCLK are assumed to be stable and PLLCLKACT switches from 0 to 1 to enable output of PPLLCLK and CPLLCLK. **当PLL稳定计数器达到预定值，PPLLCLK和CPLLCLK被认为稳定，PLLCLKACT从0切换为1来使能 PPLLCLK和CPLLCLK的输出**

The stable and active state of the PPLLCLK and CPLLCLK clocks is indicated by

PLLS.PLLCLKACT = 1.

**表明PPLLCLK和CPLLCLK被激活**

#### PLL in stand-by modes

**在STOP模式下，PLL会被自动禁止，从STOPmode wake\_up以后，PLL恢复运行（在进入STOPmode之前PLL是正常运行的）**  
In STOP mode, the PLL is automatically disabled and resumes operation after wake-up from STOP mode, if it was operating before entering STOP mode.

**在进入DeepSTOPmode后PLL会被自动禁止，注意：从DeepSTOP模式恢复的时候PLL必须重新配置**

The PLL is also automatically disabled when transitioning to DeepSTOP mode. However, after restoring from DeepSTOP mode, the PLL needs to be reconfigured.

In Cyclic RUN and Cyclic STOP mode, the PLL is not available. Do not enable the PLL by PLL Enable register in Cyclic RUN mode.

**在Cyclic Run 和Cyclic STOP模式下，PLL是不可用的。在Cyclic RUN模式下不要使用PLL enable 寄存器使能PLL**

### Clock Monitor control

The PLL activity signal PLLCLKACT enables or disables supervision by the Clock Monitor CLMA2. In case the PLL is inactive (PLLCLKACT = 0), supervision of the output clock CPLLCLK by CLMA2 is also disabled.

### PLL enable/disable trigger

The PLL can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger PLLE.PLLENTRG = 1 starts the PLL. 
- Note that setting the enable trigger is only effective if the PLL is inactive, i.e. if PLLS.PLLCLKACT = 0.
- Disable trigger PLLE.PLLDISTRG = 0 stops the PLL. 
- Note that setting the disable trigger is only effective if the PLL is active, i.e. if PLLS.PLLCLKACT = 1.

#### 10.3.5.1 PLL Parameters

The PLL is configured by a set of parameters, loaded from the control register PLLC.

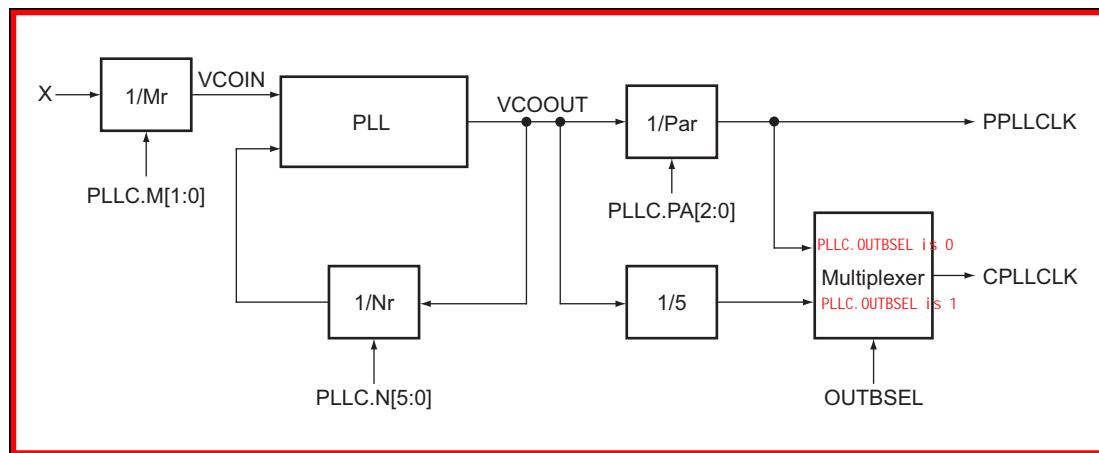


Figure 10.8 PLL Circuit Configuration

### CPLLCLK and PPLLCLK

**PPLLCLK** is the source of the clock supply for several functional modules and its frequency must not exceed 80 MHz. **PPLLCLK最大为80MHz**

The **CPLLCLK** is also the source of the CPU Subsystem clock CPUCLK.

The clock frequencies  $f_{PPLLCLK}$  and  $f_{CPLLCLK}$  are integer fractions of the VCO output frequency  $f_{VCOOUT}$ :

$$f_{VCOOUT} = f_X \times (Nr / Mr)$$

When  $f_{CPLLCLK}$  is to be set at 96 MHz, the  $f_{CPLLCLK}$  divider must be set with a dedicated divider (1/5) by setting PLLC.OUTBSEL to 1.  $f_{PPLLCLK}$  and  $f_{CPLLCLK}$  are calculated as follows:

When **PLLC.OUTBSEL** is 0:

$$f_{PPLLCLK}, f_{CPLLCLK} = f_X \times (Nr / Mr) \times 1/\text{Par} = f_{VCOOUT} \times 1/\text{Par}$$

When PLLC.OUTBSEL is 1:

$$f_{PPLLCLK} = f_X \times (Nr / Mr) \times 1/\text{Par} = f_{VCOOUT} \times 1/\text{Par}$$

$$f_{CPLLCLK} = f_X \times (Nr / Mr) \times 1/5 = f_{VCOOUT} \times 1/5$$

The values Nr, Mr, and Par are derived from PLLC register bits:

$$Nr = \text{PLLC.N}[5:0] + 1$$

$$Mr = \text{PLLC.M}[1:0] + 1$$

The setting range of Mr:  $1 \leq Mr \leq 3$

Par is determined by PLLC.PA[2:0] according to the following table:

PLLC.PA[2:0]	Par
$010_B$	4
$011_B$	6
$100_B$	8
$101_B$	16

## 10.4 Registers

### 10.4.1 List of Registers

The registers of the clock controller are listed below.

Table 10.1 List of Clock Controller Registers (1/2)

Register Name	Symbol	Address
<b>Clock oscillator registers:</b>		
MainOSC enable register	MOSCE	FFF8 1100 <sub>H</sub>
MainOSC status register	MOSCS	FFF8 1104 <sub>H</sub>
MainOSC control register	MOSCC	FFF8 1108 <sub>H</sub>
MainOSC stabilization time register	MOSCST	FFF8 110C <sub>H</sub>
MainOSC stop mask register	MOSCSTPM	FFF8 1118 <sub>H</sub>
SubOSC enable register	SOSCE <sup>*1</sup>	FFF8 1200 <sub>H</sub>
SubOSC status register	SOSCS <sup>*1</sup>	FFF8 1204 <sub>H</sub>
SubOSC stabilization time register	SOSCST <sup>*1</sup>	FFF8 120C <sub>H</sub>
HS IntOSC enable register	ROSCE	FFF8 1000 <sub>H</sub>
HS IntOSC status register	ROSCS	FFF8 1004 <sub>H</sub>
HS IntOSC stop mask register	ROSCSTPM	FFF8 1018 <sub>H</sub>
PLL enable register	PLLE	FFF8 9000 <sub>H</sub>
PLL status register	PLLS	FFF8 9004 <sub>H</sub>
PLL control register	PLLC	FFF8 9008 <sub>H</sub>
<b>Clock selector registers:</b>		
C_AWO_WDTA clock divider register	CKSC_AWDTAD_CTL	FFF8 2000 <sub>H</sub>
C_AWO_WDTA clock divider active register	CKSC_AWDTAD_ACT	FFF8 2008 <sub>H</sub>
C_AWO_WDTA stop mask register	CKSC_AWDTAD_STPM	FFF8 2018 <sub>H</sub>
C_AWO_TAUJ source clock selection register	CKSC_ATAUJS_CTL	FFF8 2100 <sub>H</sub>
C_AWO_TAUJ source clock active register	CKSC_ATAUJS_ACT	FFF8 2108 <sub>H</sub>
C_AWO_TAUJ clock divider register	CKSC_ATAUJD_CTL	FFF8 2200 <sub>H</sub>
C_AWO_TAUJ clock divider active register	CKSC_ATAUJD_ACT	FFF8 2208 <sub>H</sub>
C_AWO_TAUJ stop mask register	CKSC_ATAUJD_STPM	FFF8 2218 <sub>H</sub>
C_AWO_RTCA source clock selection register	CKSC_ARTCAS_CTL <sup>*1</sup>	FFF8 2300 <sub>H</sub>
C_AWO_RTCA source clock active register	CKSC_ARTCAS_ACT <sup>*1</sup>	FFF8 2308 <sub>H</sub>
C_AWO_RTCA clock divider register	CKSC_ARTCAD_CTL <sup>*1</sup>	FFF8 2400 <sub>H</sub>
C_AWO_RTCA clock divider active register	CKSC_ARTCAD_ACT <sup>*1</sup>	FFF8 2408 <sub>H</sub>
C_AWO_RTCA stop mask register	CKSC_ARTCAD_STPM <sup>*1</sup>	FFF8 2418 <sub>H</sub>
C_AWO_ADCA source clock selection register	CKSC_AADCAS_CTL	FFF8 2500 <sub>H</sub>
C_AWO_ADCA source clock active register	CKSC_AADCAS_ACT	FFF8 2508 <sub>H</sub>
C_AWO_ADCA clock divider register	CKSC_AADCAD_CTL	FFF8 2600 <sub>H</sub>
C_AWO_ADCA clock divider active register	CKSC_AADCAD_ACT	FFF8 2608 <sub>H</sub>
C_AWO_ADCA stop mask register	CKSC_AADCAD_STPM	FFF8 2618 <sub>H</sub>
C_AWO_FOUT source clock selection register	CKSC_AFOUTS_CTL	FFF8 2700 <sub>H</sub>
C_AWO_FOUT source clock active register	CKSC_AFOUTS_ACT	FFF8 2708 <sub>H</sub>
C_AWO_FOUT stop mask register	CKSC_AFOUTS_STPM	FFF8 2718 <sub>H</sub>
C_ISO_CPUCLK source clock selection register	CKSC_CPUCLKS_CTL	FFF8 A000 <sub>H</sub>
C_ISO_CPUCLK source clock active register	CKSC_CPUCLKS_ACT	FFF8 A008 <sub>H</sub>

**Table 10.1 List of Clock Controller Registers (2/2)**

<b>Register Name</b>	<b>Symbol</b>	<b>Address</b>
C_ISO_CPUCLK clock divider register	CKSC_CPUCLKD_CTL	FFF8 A100 <sub>H</sub>
C_ISO_CPUCLK clock divider active register	CKSC_CPUCLKD_ACT	FFF8 A108 <sub>H</sub>
C_ISO_PERI1 source clock selection register	CKSC_IPERI1S_CTL	FFF8 A200 <sub>H</sub>
C_ISO_PERI1 source clock active register	CKSC_IPERI1S_ACT	FFF8 A208 <sub>H</sub>
C_ISO_PERI2 source clock selection register	CKSC_IPERI2S_CTL	FFF8 A300 <sub>H</sub>
C_ISO_PERI2 source clock active register	CKSC_IPERI2S_ACT	FFF8 A308 <sub>H</sub>
C_ISO_LIN source clock selection register	CKSC_ILINS_CTL	FFF8 A400 <sub>H</sub>
C_ISO_LIN source clock active register	CKSC_ILINS_ACT	FFF8 A408 <sub>H</sub>
C_ISO_ADCA source clock selection register	CKSC_IADCAS_CTL <sup>*1</sup>	FFF8 A500 <sub>H</sub>
C_ISO_ADCA source clock active register	CKSC_IADCAS_ACT <sup>*1</sup>	FFF8 A508 <sub>H</sub>
C_ISO_ADCA clock divider register	CKSC_IADCAD_CTL <sup>*1</sup>	FFF8 A600 <sub>H</sub>
C_ISO_ADCA clock divider active register	CKSC_IADCAD_ACT <sup>*1</sup>	FFF8 A608 <sub>H</sub>
C_ISO_LIN clock divider register	CKSC_ILIND_CTL	FFF8 A800 <sub>H</sub>
C_ISO_LIN clock divider active register	CKSC_ILIND_ACT	FFF8 A808 <sub>H</sub>
C_ISO_LIN stop mask register	CKSC_ILIND_STPM	FFF8 A818 <sub>H</sub>
C_ISO_CAN source clock selection register	CKSC_ICANS_CTL	FFF8 A900 <sub>H</sub>
C_ISO_CAN source clock active register	CKSC_ICANS_ACT	FFF8 A908 <sub>H</sub>
C_ISO_CAN stop mask register	CKSC_ICANS_STPM	FFF8 A918 <sub>H</sub>
C_ISO_CANOSC clock divider register	CKSC_ICANOSCD_CTL	FFF8 AA00 <sub>H</sub>
C_ISO_CANOSC clock divider active register	CKSC_ICANOSCD_ACT	FFF8 AA08 <sub>H</sub>
C_ISO_CANOSC stop mask register	CKSC_ICANOSCD_STPM	FFF8 AA18 <sub>H</sub>
C_ISO_CSI source clock selection register	CKSC_ICSIS_CTL	FFF8 AB00 <sub>H</sub>
C_ISO_CSI source clock active register	CKSC_ICSIS_ACT	FFF8 AB08 <sub>H</sub>

Note 1. Supported only for 144 pin and 176 pin products.

## 10.4.2 Clock Oscillator Registers

### 10.4.2.1 MOSCE — MainOSC Enable Register

This register is used to start and stop the MainOSC.

**使用PROTCMD0寄存器，正确的写入顺序是必须的，这样才能更新此寄存器**  
The correct write sequence using the PROTCMD0 register is required in order to update this register.  
For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the power-up reset signals PURES and CLMA1RES.  
**此寄存器由power-up 复位信号PURES和...来初始化**

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 1100<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCDISTRG	MOSCENTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

**Table 10.2 MOSCE Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MOSCDISTRG	MainOSC Disable Trigger <sup>*1</sup> 0: No function 1: Stops MainOSC
0	MOSCENTRG	MainOSC Enable Trigger 0: No function 1: Starts MainOSC

Note 1. <Recommended procedure for stopping MainOSC using MOSCDISTRG>

1. Check that there is no clock domain for which MainOSC is selected.  
If MainOSC is selected for a clock domain, disable the setting or select a clock source other than MainOSC.
2. Stop the MainOSC (MOSCE.MOSCDISTRG = 1).
3. Confirm that the MainOSC has been stopped (MOSCS.MOSCCLKACT = 0).

**使用MOSCDISTRG来  
stop MainOSC的过程**

### 10.4.2.2 MOSCS — MainOSC Status Register

用来提供MainOSC的激活信息

This register provides active status information about the MainOSC.

This register is initialized by the power-up reset signals PURES and CLMA1RES.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 1104<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 <sup>*1</sup>	0 <sup>*1</sup>
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.

After masking bit 1 and 0, check only bit 2 to verify the status.

Table 10.3 MOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	MOSCCLKACT	MainOSC Active Status 0: MainOSC is inactive 1: MainOSC is active
1, 0	Reserved	When read, an undefined value is returned.

### 10.4.2.3 MOSCC — MainOSC Control Register

用来指定MainOSC的放大增益

This register is used to specify amplification gain of the MainOSC.

This register is initialized by the power-up reset signals PURES and CLMA1RES.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 1108<sub>H</sub>

**Value after reset:** 0000 0004<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCAMPSEL [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.4 MOSCC Register Contents

Bit Position	Bit Name	Function															
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
1, 0	MOSC AMPSEL[1:0]	MainOSC Frequency Selection															
		<table border="1"> <thead> <tr> <th>MOSCAMPSEL[1:0]</th> <th>Amplification Gain</th> <th>Typical Frequency Range for External Resonator 根据外振荡器部 (Depends on Oscillator Type)</th> </tr> </thead> <tbody> <tr> <td>00<sub>B</sub></td> <td>High</td> <td>20 MHz &lt; f<sub>X</sub> ≤ 24 MHz</td> </tr> <tr> <td>01<sub>B</sub></td> <td>Mid-high</td> <td>16 MHz &lt; f<sub>X</sub> ≤ 20 MHz</td> </tr> <tr> <td>10<sub>B</sub></td> <td>Mid-low</td> <td>8 MHz &lt; f<sub>X</sub> ≤ 16 MHz</td> </tr> <tr> <td>11<sub>B</sub></td> <td>Low</td> <td>8 MHz</td> </tr> </tbody> </table>	MOSCAMPSEL[1:0]	Amplification Gain	Typical Frequency Range for External Resonator 根据外振荡器部 (Depends on Oscillator Type)	00 <sub>B</sub>	High	20 MHz < f <sub>X</sub> ≤ 24 MHz	01 <sub>B</sub>	Mid-high	16 MHz < f <sub>X</sub> ≤ 20 MHz	10 <sub>B</sub>	Mid-low	8 MHz < f <sub>X</sub> ≤ 16 MHz	11 <sub>B</sub>	Low	8 MHz
MOSCAMPSEL[1:0]	Amplification Gain	Typical Frequency Range for External Resonator 根据外振荡器部 (Depends on Oscillator Type)															
00 <sub>B</sub>	High	20 MHz < f <sub>X</sub> ≤ 24 MHz															
01 <sub>B</sub>	Mid-high	16 MHz < f <sub>X</sub> ≤ 20 MHz															
10 <sub>B</sub>	Mid-low	8 MHz < f <sub>X</sub> ≤ 16 MHz															
11 <sub>B</sub>	Low	8 MHz															

#### CAUTION

Set this register when MainOSC is stopped. 当MainOSC停止后设置此寄存器

#### 10.4.2.4 MOSCST — MainOSC Stabilization Time Register

**此寄存器决定MainOSC的稳定时间**  
This register determines the MainOSC stabilization time.

This register is initialized by the power-up reset signals PURES and CLMA1RES.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 110CH

**Value after reset:** 0000 44C0H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	MOSC CLKST[16]
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSC CLKST[16]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MOSCCLKST[15:0]																	
Value after reset	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 10.5 MOSCST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16 to 0	MOSC CLKST[16:0]	The MOSCCLKST[16:0] bits specify the count value for the MainOSC stabilization counter. <ul style="list-style-type: none"> <li>If the HS IntOSC is <u>active</u> (ROSCS.ROSCCLKACT = 1): Stabilization time = MOSCCLKST[16:0] / f<sub>RH</sub></li> <li>If the HS IntOSC is <u>inactive</u> (ROSCS.ROSCCLKACT = 0): Stabilization time = MOSCCLKST[16:0] / f<sub>RL</sub></li> </ul>

#### NOTE

See the [Data Sheet](#) for information about the MainOSC stabilization time.

#### CAUTION

Set this register when MainOSC is stopped. **当MainOSC停止后设置此寄存器**

### 10.4.2.5 MOSCSTPM — MainOSC Stop Mask Register

This register is initialized by the power-up reset signals PURES and CLMA1RES.

<b>Access:</b>	This register can be read or written in 32-bit units.															
<b>Address:</b>	<u>FFF8 1118H</u>															
<b>Value after reset:</b>	0000 0002H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCS TPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 10.6 MOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCSTPMASK	MainOSC Stop Request Mask in Stand-by Mode 0: MainOSC stops operation in stand-by mode. 1: MainOSC continues operation in stand-by mode. Note that the MainOSC can be stopped in the case the MainOSC disable trigger MOSCE.MOSCDISTRG is set to 1 regardless of the setting of this bit.

注意如果MOSCE.MOSCDISTRG = 1, 这这一位的设置是无效的

### 10.4.2.6 SOSCE — SubOSC Enable Register

用于启动或者停止SubOSC

This register is used to start and stop the SubOSC.

**使用PROTCMD0寄存器 正确的写入顺序是必须的，这样才能更新此寄存器**

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 1200<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCDISTRG	SOSCE NTRG	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	

Table 10.7 SOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SOSCDISTRG	SubOSC Disable Trigger <sup>*1</sup> 0: No function 1: Stops SubOSC
0	SOSCENTRG	SubOSC Enable Trigger 0: No function 1: Starts SubOSC

Note 1. <Recommended procedure for stopping SubOSC using SOSCDISTRG>

1. Check that there is no clock domain for which SubOSC is selected.  
If SubOSC is selected for a clock domain, disable the setting or select a clock source other than SubOSC.
2. Stop the SubOSC (SOSCE.SOSCDISTRG = 1).
3. Confirm that the SubOSC has been stopped (SOSCS.SOSCCLKACT = 0).

使用SOSCDISTRG来  
stop SubOSC的过程

### 10.4.2.7 SOSCS — SubOSC Status Register

表明SubOSC的激活状态

This register provides active status information about the SubOSC.

This register is initialized by the power-up reset signal PURES.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 1204<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1*1	0*1
	—	—	—	—	—	—	—	—	—	—	—	—	SOSCC LKACT	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.

After masking bit 1 and 0, check only bit 2 to verify the status.

Table 10.8 SOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	SOSCCLKACT	SubOSC Active Status 0: SubOSC is inactive 1: SubOSC is active
1, 0	Reserved	When read, an undefined value is returned.

### 10.4.2.8 SOSCST — SubOSC Stabilization Time Register

决定SubOSC的稳定时间

This register determines the SubOSC stabilization time.

This register is initialized by the power-up reset signal PURES.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 120C<sub>H</sub>

**Value after reset:** 010C 8E00<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SOSCCLKST[29:16]													
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOSCCLKST[15:0]															
Value after reset	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.9 SOSCST Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 0	SOSCCLKST [29:0]	The SOSCCLKST[29:0] bits specify the count value for the SubOSC stabilization time counter. <ul style="list-style-type: none"> <li>If the HS IntOSC is active (ROSCS.ROSCCLKACT = 1): Stabilization time = SOSCCLKST[29:0] / f<sub>RH</sub></li> <li>If the HS IntOSC is inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = SOSCCLKST[29:0] / f<sub>RL</sub></li> </ul>

#### NOTE

See the Data Sheet for information about the SubOSC stabilization time.

#### CAUTION

Set this register when SubOSC is stopped. 在SubOSC停止后设置此寄存器

### 10.4.2.9 ROSCE — HS IntOSC Enable Register

用来停止HS IntOSC

This register is used to stop the HS IntOSC operation.

使用PROTCMD0寄存器 正确的写入顺序是必须的，这样才能更新此寄存器

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the power-up reset signals PURES and CLMA0RES.

#### CAUTION

Set the ROSCE.ROSCDISTRG bit only when the CLMA0RES has occurred. In other cases, setting this bit is prohibited. 注意：在CLMA0RES发生的时候，设置ROSCDISTRG bit才是有效的，其他情况都是不允许的

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 1000<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 10.10 ROSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ROSCDISTRG	HS IntOSC Disable Trigger 0: No function 1: Stops HS IntOSC
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

### 10.4.2.10 ROSCS — HS IntOSC Status Register

**提供表征HS IntOSC是否激活的状态信息**

This register provides active status information about the HS IntOSC.

This register is initialized by the power-up reset signals PURES and CLMA0RES.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 1004<sub>H</sub>

**Value after reset:** 0000 0007<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 <sup>*1</sup>	0 <sup>*1</sup>
	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCC LKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.

After masking bit 1 and 0, check only bit 2 to verify the status.

**Table 10.11 ROSCS Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	ROSCCLKACT	HS IntOSC Active Status 0: HS IntOSC is inactive 1: HS IntOSC is active
1, 0	Reserved	When read, an undefined value is returned.

### 10.4.2.11 ROSCSTPM — HS IntOSC Stop Mask Register

This register is initialized by the power-up reset signals PURES and CLMA0RES.

<b>Access:</b>	This register can be read or written in 32-bit units.															
<b>Address:</b>	<u>FFF8 1018H</u>															
<b>Value after reset:</b>	0000 0002H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCSTPM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 10.12 ROSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ROSCSTPM	HS IntOSC Stop Request Mask in Stand-by Mode 0: HS IntOSC stops operation in stand-by mode 1: HS IntOSC continues operation in stand-by mode Note that the HS IntOSC can be stopped in the case the HS IntOSC disable trigger ROSCE.ROSCDISTRG is set to 1 regardless of the setting of this bit.

注意如果ROSCE.ROSCDISTRG = 1, 这一位的设置是无效的

### 10.4.2.12 PLLE — PLL Enable Register

**使用PROTCMD0寄存器 正确的写入顺序是必须的，这样才能更新此寄存器**

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the ISORES signal.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 9000<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLDIS TRG	PLLLEN TRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 10.13 PLLE Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PLLDISTRG	PLL Disable Trigger* <sup>1</sup> 0: No function 1: Stops PLL
0	PLLLENTRG	PLL Enable Trigger* <sup>2</sup> 0: No function 1: Starts PLL

Note 1. <Recommended procedure for stopping PLL using PLLE.PLLDISTRG>

1. Check that there is no clock domain for which PLL is selected.  
If PLL is selected for a clock domain, disable the setting or select a clock source other than PLL.
2. Stop the PLL (PLLE.PLLDISTRG = 1).
3. Confirm that the PLL has been stopped (PLLS.PLLCLKACT = 0).

Note 2. Before starting PLL using PLLLENTRG, confirm that MainOSC is operating.

**在使用PLLLENTRG激活PLL之前，请确保MainOSC是工作的**

### 10.4.2.13 PLLS — PLL Status Register

表征PLL激活的状态

This register provides active status information about the PLL.

This register is initialized by the ISORES signal.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 9004<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 <sup>*1</sup>	0 <sup>*1</sup>
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.

After masking bit 1 and 0, check only bit 2 to verify the status.

Table 10.14 PLLS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLLCLKACT	PLL Active Status 0: PLL is inactive 1: PLL is active
1, 0	Reserved	When read, an undefined value is returned.

### 10.4.2.14 PLLC — PLL Control Register

**重要**

通常用于设置PLL输出的时钟频率 $f_{PLLCLK}$ 和 $f_{CPLLCLK}$   
This register is used to set the PLL output clock frequencies  $f_{PLLCLK}$  and  $f_{CPLLCLK}$ , shown in  
**Section 10.3.5.1, PLL Parameters.**

This register can only be written, if the PLL is disabled. 只有在PLL被disabled的情况下此寄存器才能被写入  
This register is initialized by the ISORES signal.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 9008<sub>H</sub>

**Value after reset:** 0001 133B<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUTBSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	M[1:0]		PA[2:0]		—	—			N[5:0]				
Value after reset	0	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.15 PLLC Register Contents (1/2)

Bit Position	Bit Name	Function																				
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																				
16	OUTBSEL	<ul style="list-style-type: none"> <li>Other than ADVANCE, PREMIUM, and Gateway-1MB: Be sure to write 0 to this bit (the value after reset is 1). 除非是ADVANCE, PREMIUM和Gateway-1MB以外，此位写为0，复位状态下为1。</li> <li>ADVANCE, PREMIUM, and Gateway-1MB: <math>f_{CPLLCLK}</math> selection bit 这是上述的选择位</li> </ul> 0: Same clock as $f_{PLLCLK}$ 1: Clock dividing $f_{VCOOUT}$ by 5																				
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																				
12, 11	M[1:0]	<table border="1"> <thead> <tr> <th>M1</th> <th>M0</th> <th>Mr-Value</th> <th>MainOSC frequency <math>f_X</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> <td>8 MHz <math>\leq f_X \leq</math> 24 MHz</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>16 MHz <math>\leq f_X \leq</math> 24 MHz</td> </tr> <tr> <td>1</td> <td>0</td> <td>3</td> <td><math>f_X = 24</math> MHz</td> </tr> <tr> <td>1</td> <td>1</td> <td></td> <td>Setting prohibited</td> </tr> </tbody> </table>	M1	M0	Mr-Value	MainOSC frequency $f_X$	0	0	1	8 MHz $\leq f_X \leq$ 24 MHz	0	1	2	16 MHz $\leq f_X \leq$ 24 MHz	1	0	3	$f_X = 24$ MHz	1	1		Setting prohibited
M1	M0	Mr-Value	MainOSC frequency $f_X$																			
0	0	1	8 MHz $\leq f_X \leq$ 24 MHz																			
0	1	2	16 MHz $\leq f_X \leq$ 24 MHz																			
1	0	3	$f_X = 24$ MHz																			
1	1		Setting prohibited																			

Table 10.15 PLLC Register Contents (2/2) 图10-8

Bit Position	Bit Name	Function																											
10 to 8	PA[2:0]	P Divider Selection When OUTBSEL = 0, these bits indicate the output range of $f_{PPLLCLK}$ and $f_{CPLLCLK}$ . When OUTBSEL = 1, these bits indicate the output range of $f_{PPLLCLK}$ .																											
		<table border="1"> <thead> <tr> <th>PA[2:0]</th> <th>Par-Value</th> <th>PLL output frequency range</th> </tr> </thead> <tbody> <tr> <td>000<sub>B</sub></td> <td>Setting prohibited</td> <td>—</td> </tr> <tr> <td>001<sub>B</sub></td> <td>Setting prohibited</td> <td>—</td> </tr> <tr> <td>010<sub>B</sub></td> <td>4</td> <td>60 MHz to 80 MHz <span style="color: red;">PPLLCLK最大为80MHz</span></td> </tr> <tr> <td>011<sub>B</sub></td> <td>6</td> <td>40 MHz to 80 MHz</td> </tr> <tr> <td>100<sub>B</sub></td> <td>8</td> <td>30 MHz to 60 MHz</td> </tr> <tr> <td>101<sub>B</sub></td> <td>16</td> <td>25 MHz to 30 MHz</td> </tr> <tr> <td>110<sub>B</sub></td> <td>Setting prohibited</td> <td>—</td> </tr> <tr> <td>111<sub>B</sub></td> <td>Setting prohibited</td> <td>—</td> </tr> </tbody> </table>	PA[2:0]	Par-Value	PLL output frequency range	000 <sub>B</sub>	Setting prohibited	—	001 <sub>B</sub>	Setting prohibited	—	010 <sub>B</sub>	4	60 MHz to 80 MHz <span style="color: red;">PPLLCLK最大为80MHz</span>	011 <sub>B</sub>	6	40 MHz to 80 MHz	100 <sub>B</sub>	8	30 MHz to 60 MHz	101 <sub>B</sub>	16	25 MHz to 30 MHz	110 <sub>B</sub>	Setting prohibited	—	111 <sub>B</sub>	Setting prohibited	—
PA[2:0]	Par-Value	PLL output frequency range																											
000 <sub>B</sub>	Setting prohibited	—																											
001 <sub>B</sub>	Setting prohibited	—																											
010 <sub>B</sub>	4	60 MHz to 80 MHz <span style="color: red;">PPLLCLK最大为80MHz</span>																											
011 <sub>B</sub>	6	40 MHz to 80 MHz																											
100 <sub>B</sub>	8	30 MHz to 60 MHz																											
101 <sub>B</sub>	16	25 MHz to 30 MHz																											
110 <sub>B</sub>	Setting prohibited	—																											
111 <sub>B</sub>	Setting prohibited	—																											
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																											
5 to 0	N[5:0]	Division ratio Nr is set. For N[5:0] example settings, see <b>Table 10.16, PLL Output Table</b> .																											
<b>CAUTION</b>																													
Set this register when PLL is stopped. Setting Par = 4 is prohibited when OUTBSEL = 1.																													

Table 10.16 PLL Output Table

MainOSC = 8 MHz (1/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
0	1	16	1	1	0	0	0	1	50	25.00	←
0	1	16	1	1	0	0	1	0	51	25.50	←
0	1	16	1	1	0	0	1	1	52	26.00	←
0	1	16	1	1	0	1	0	0	53	26.50	←
0	1	16	1	1	0	1	0	1	54	27.00	←
0	1	16	1	1	0	1	1	0	55	27.50	←
0	1	16	1	1	0	1	1	1	56	28.00	←
0	1	16	1	1	1	0	0	0	57	28.50	←
0	1	16	1	1	1	0	0	1	58	29.00	←
0	1	16	1	1	1	0	1	0	59	29.50	←
0	1	16	1	1	1	0	1	1	60	30.00	←
0	1	8	0	1	1	1	0	1	30	30.00	←
0	1	8	0	1	1	1	1	0	31	31.00	←
0	1	8	0	1	1	1	1	1	32	32.00	←
0	1	8	1	0	0	0	0	0	33	33.00	←
0	1	8	1	0	0	0	0	1	34	34.00	←
0	1	8	1	0	0	0	1	0	35	35.00	←

MainOSC = 8 MHz (2/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	f <sub>PPLLCLK</sub> Frequency (MHz)	f <sub>CPLLCLK</sub> Frequency (MHz)
0	1	8	1	0	0	0	1	1	36	36.00	←
0	1	8	1	0	0	1	0	0	37	37.00	←
0	1	8	1	0	0	1	0	1	38	38.00	←
0	1	8	1	0	0	1	1	0	39	39.00	←
0	1	8	1	0	0	1	1	1	40	40.00	←
0	1	8	1	0	1	0	0	0	41	41.00	←
0	1	8	1	0	1	0	0	1	42	42.00	←
0	1	8	1	0	1	0	1	0	43	43.00	←
0	1	8	1	0	1	0	1	1	44	44.00	←
0	1	8	1	0	1	1	0	0	45	45.00	←
0	1	8	1	0	1	1	0	1	46	46.00	←
0	1	8	1	0	1	1	1	0	47	47.00	←
0	1	8	1	0	1	1	1	1	48	48.00	←
0	1	8	1	1	0	0	0	0	49	49.00	←
0	1	8	1	1	0	0	0	1	50	50.00	←
0	1	8	1	1	0	0	1	0	51	51.00	←
0	1	8	1	1	0	0	1	1	52	52.00	←
0	1	8	1	1	0	1	0	0	53	53.00	←
0	1	8	1	1	0	1	0	1	54	54.00	←
0	1	8	1	1	0	1	1	0	55	55.00	←
0	1	8	1	1	0	1	1	1	56	56.00	←
0	1	8	1	1	1	0	0	0	57	57.00	←
0	1	8	1	1	1	0	0	1	58	58.00	←
0	1	8	1	1	1	0	1	0	59	59.00	←
0	1	8	1	1	1	0	1	1	60	60.00	←
0	1	4	0	1	1	1	0	1	30	60.00	←
0	1	4	0	1	1	1	1	0	31	62.00	←
0	1	4	0	1	1	1	1	1	32	64.00	←
0	1	4	1	0	0	0	0	0	33	66.00	←
0	1	4	1	0	0	0	0	1	34	68.00	←
0	1	4	1	0	0	0	1	0	35	70.00	←
0	1	4	1	0	0	0	1	1	36	72.00	←
0	1	4	1	0	0	1	0	0	37	74.00	←
0	1	4	1	0	0	1	0	1	38	76.00	←
0	1	4	1	0	0	1	1	0	39	78.00	←
0	1	4	1	0	0	1	1	1	40	80.00	←
1	1	6	1	1	1	0	1	1	60	80.00	96.00 <sup>*1</sup>

Note 1. 96 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

MainOSC = 12 MHz

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	f <sub>PPLLCLK</sub> Frequency (MHz)	f <sub>CPLLCLK</sub> Frequency (MHz)
0	1	16	1	0	0	0	0	1	34	25.50	←
0	1	16	1	0	0	0	1	0	35	26.25	←
0	1	16	1	0	0	0	1	1	36	27.00	←
0	1	16	1	0	0	1	0	0	37	27.75	←
0	1	16	1	0	0	1	0	1	38	28.50	←
0	1	16	1	0	0	1	1	0	39	29.25	←
0	1	16	1	0	0	1	1	1	40	30.00	←
0	1	8	0	1	0	0	1	1	20	30.00	←
0	1	8	0	1	0	1	0	0	21	31.50	←
0	1	8	0	1	0	1	0	1	22	33.00	←
0	1	8	0	1	0	1	1	0	23	34.50	←
0	1	8	0	1	0	1	1	1	24	36.00	←
0	1	8	0	1	1	0	0	0	25	37.50	←
0	1	8	0	1	1	0	0	1	26	39.00	←
0	1	8	0	1	1	0	1	0	27	40.50	←
0	1	8	0	1	1	0	1	1	28	42.00	←
0	1	8	0	1	1	1	0	0	29	43.50	←
0	1	8	0	1	1	1	0	1	30	45.00	←
0	1	8	0	1	1	1	1	0	31	46.50	←
0	1	8	0	1	1	1	1	1	32	48.00	←
0	1	8	1	0	0	0	0	0	33	49.50	←
0	1	8	1	0	0	0	0	1	34	51.00	←
0	1	8	1	0	0	0	1	0	35	52.50	←
0	1	8	1	0	0	0	1	1	36	54.00	←
0	1	8	1	0	0	1	0	0	37	55.50	←
0	1	8	1	0	0	1	0	1	38	57.00	←
0	1	8	1	0	0	1	1	0	39	58.50	←
0	1	8	1	0	0	1	1	1	40	60.00	←
0	1	4	0	1	0	0	1	1	20	60.00	←
0	1	4	0	1	0	1	0	0	21	63.00	←
0	1	4	0	1	0	1	0	1	22	66.00	←
0	1	4	0	1	0	1	1	0	23	69.00	←
0	1	4	0	1	0	1	1	1	24	72.00	←
0	1	4	0	1	1	0	0	0	25	75.00	←
0	1	4	0	1	1	0	0	1	26	78.00	←
1	1	6	1	0	0	1	1	1	40	80.00	96.00*1

Note 1. 96 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

MainOSC = 16 MHz (1/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	f <sub>PPLLCLK</sub> Frequency (MHz)	f <sub>CPLLCLK</sub> Frequency (MHz)
0	1	16	0	1	1	0	0	0	25	25.00	←
0	1	16	0	1	1	0	0	1	26	26.00	←
0	1	16	0	1	1	0	1	0	27	27.00	←
0	1	16	0	1	1	0	1	1	28	28.00	←
0	1	16	0	1	1	1	0	0	29	29.00	←
0	1	16	0	1	1	1	0	1	30	30.00	←
0	1	8	0	1	0	0	1	1	20	40.00	←
0	1	8	0	1	0	1	0	0	21	42.00	←
0	1	8	0	1	0	1	0	1	22	44.00	←
0	1	8	0	1	0	1	1	0	23	46.00	←
0	1	8	0	1	0	1	1	1	24	48.00	←
0	1	8	0	1	1	0	0	0	25	50.00	←
0	1	8	0	1	1	0	0	1	26	52.00	←
0	1	8	0	1	1	0	1	0	27	54.00	←
0	1	8	0	1	1	0	1	1	28	56.00	←
0	1	8	0	1	1	1	0	0	29	58.00	←
0	1	8	0	1	1	1	0	1	30	60.00	←
0	2	16	1	1	0	0	0	1	50	25.00	←
0	2	16	1	1	0	0	1	0	51	25.50	←
0	2	16	1	1	0	0	1	1	52	26.00	←
0	2	16	1	1	0	1	0	0	53	26.50	←
0	2	16	1	1	0	1	0	1	54	27.00	←
0	2	16	1	1	0	1	1	0	55	27.50	←
0	2	16	1	1	0	1	1	1	56	28.00	←
0	2	16	1	1	1	0	0	0	57	28.50	←
0	2	16	1	1	1	0	0	1	58	29.00	←
0	2	16	1	1	1	0	1	0	59	29.50	←
0	2	16	1	1	1	0	1	1	60	30.00	←
0	2	8	0	1	1	1	0	1	30	30.00	←
0	2	8	0	1	1	1	1	0	31	31.00	←
0	2	8	0	1	1	1	1	1	32	32.00	←
0	2	8	1	0	0	0	0	0	33	33.00	←
0	2	8	1	0	0	0	0	1	34	34.00	←
0	2	8	1	0	0	0	1	0	35	35.00	←
0	2	8	1	0	0	0	1	1	36	36.00	←
0	2	8	1	0	0	1	0	0	37	37.00	←
0	2	8	1	0	0	1	0	1	38	38.00	←
0	2	8	1	0	0	1	1	0	39	39.00	←
0	2	8	1	0	0	1	1	1	40	40.00	←
0	2	8	1	0	1	0	0	0	41	41.00	←
0	2	8	1	0	1	0	0	1	42	42.00	←
0	2	8	1	0	1	0	1	0	43	43.00	←

**MainOSC = 16 MHz (2/2)**

<b>OUTBSEL</b>	<b>Mr</b>	<b>Par</b>	<b>N5</b>	<b>N4</b>	<b>N3</b>	<b>N2</b>	<b>N1</b>	<b>N0</b>	<b>Nr</b>	<b>f<sub>PPLLCLK</sub> Frequency (MHz)</b>	<b>f<sub>CPLLCLK</sub> Frequency (MHz)</b>
0	2	8	1	0	1	0	1	1	44	44.00	←
0	2	8	1	0	1	1	0	0	45	45.00	←
0	2	8	1	0	1	1	0	1	46	46.00	←
0	2	8	1	0	1	1	1	0	47	47.00	←
0	2	8	1	0	1	1	1	1	48	48.00	←
0	2	8	1	1	0	0	0	0	49	49.00	←
0	2	8	1	1	0	0	0	0	50	50.00	←
0	2	8	1	1	0	0	1	0	51	51.00	←
0	2	8	1	1	0	0	1	1	52	52.00	←
0	2	8	1	1	0	1	0	0	53	53.00	←
0	2	8	1	1	0	1	0	1	54	54.00	←
0	2	8	1	1	0	1	1	0	55	55.00	←
0	2	8	1	1	0	1	1	1	56	56.00	←
0	2	8	1	1	1	0	0	0	57	57.00	←
0	2	8	1	1	1	0	0	0	58	58.00	←
0	2	8	1	1	1	0	1	0	59	59.00	←
0	2	8	1	1	1	0	1	1	60	60.00	←
0	2	4	0	1	1	1	0	1	30	60.00	←
0	2	4	0	1	1	1	1	0	31	62.00	←
0	2	4	0	1	1	1	1	1	32	64.00	←
0	2	4	1	0	0	0	0	0	33	66.00	←
0	2	4	1	0	0	0	0	0	34	68.00	←
0	2	4	1	0	0	0	1	0	35	70.00	←
0	2	4	1	0	0	0	0	1	36	72.00	←
0	2	4	1	0	0	1	0	0	37	74.00	←
0	2	4	1	0	0	1	0	1	38	76.00	←
0	2	4	1	0	0	1	1	0	39	78.00	←
0	2	4	1	0	0	1	1	1	40	80.00	←
1	2	6	1	1	1	0	1	1	60	80.00	96.00*1

Note 1. 96 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

**MainOSC = 20 MHz (1/3)**

<b>OUTBSEL</b>	<b>Mr</b>	<b>Par</b>	<b>N5</b>	<b>N4</b>	<b>N3</b>	<b>N2</b>	<b>N1</b>	<b>N0</b>	<b>Nr</b>	<b>f<sub>PPLLCLK</sub> Frequency (MHz)</b>	<b>f<sub>CPLLCLK</sub> Frequency (MHz)</b>
0	1	16	0	1	0	0	1	1	20	25.00	←
0	1	16	0	1	0	1	0	0	21	26.25	←
0	1	16	0	1	0	1	0	1	22	27.50	←
0	1	16	0	1	0	1	1	0	23	28.75	←
0	1	16	0	1	0	1	1	1	24	30.00	←
0	1	8	0	1	0	0	1	1	20	50.00	←
0	1	8	0	1	0	1	0	0	21	52.50	←
0	1	8	0	1	0	1	0	1	22	55.00	←
0	1	8	0	1	0	1	1	0	23	57.50	←

MainOSC = 20 MHz (2/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	f <sub>PPLLCLK</sub> Frequency (MHz)	f <sub>CPLLCLK</sub> Frequency (MHz)
0	1	8	0	1	0	1	1	1	24	60.00	←
0	2	16	1	0	0	1	1	1	40	25.00	←
0	2	16	1	0	1	0	0	0	41	25.63	←
0	2	16	1	0	1	0	0	1	42	26.25	←
0	2	16	1	0	1	0	1	0	43	26.88	←
0	2	16	1	0	1	0	1	1	44	27.50	←
0	2	16	1	0	1	1	0	0	45	28.13	←
0	2	16	1	0	1	1	0	1	46	28.75	←
0	2	16	1	0	1	1	1	0	47	29.38	←
0	2	16	1	0	1	1	1	1	48	30.00	←
0	2	8	0	1	0	1	1	1	24	30.00	←
0	2	8	0	1	1	0	0	0	25	31.25	←
0	2	8	0	1	1	0	0	1	26	32.50	←
0	2	8	0	1	1	0	1	0	27	33.75	←
0	2	8	0	1	1	0	1	1	28	35.00	←
0	2	8	0	1	1	1	0	0	29	36.25	←
0	2	8	0	1	1	1	0	1	30	37.50	←
0	2	8	0	1	1	1	1	0	31	38.75	←
0	2	8	0	1	1	1	1	1	32	40.00	←
0	2	8	1	0	0	0	0	0	33	41.25	←
0	2	8	1	0	0	0	0	1	34	42.50	←
0	2	8	1	0	0	0	1	0	35	43.75	←
0	2	8	1	0	0	0	1	1	36	45.00	←
0	2	8	1	0	0	1	0	0	37	46.25	←
0	2	8	1	0	0	1	0	1	38	47.50	←
0	2	8	1	0	0	1	1	0	39	48.75	←
0	2	8	1	0	0	1	1	1	40	50.00	←
0	2	8	1	0	1	0	0	0	41	51.25	←
0	2	8	1	0	1	0	0	1	42	52.50	←
0	2	8	1	0	1	0	1	0	43	53.75	←
0	2	8	1	0	1	0	1	1	44	55.00	←
0	2	8	1	0	1	1	0	0	45	56.25	←
0	2	8	1	0	1	1	0	1	46	57.50	←
0	2	8	1	0	1	1	1	0	47	58.75	←
0	2	8	1	0	1	1	1	1	48	60.00	←
0	2	4	0	1	0	1	1	1	24	60.00	←
0	2	4	0	1	1	0	0	0	25	62.50	←
0	2	4	0	1	1	0	0	1	26	65.00	←
0	2	4	0	1	1	0	1	0	27	67.50	←
0	2	4	0	1	1	0	1	1	28	70.00	←
0	2	4	0	1	1	1	0	0	29	72.50	←
0	2	4	0	1	1	1	0	1	30	75.00	←

**MainOSC = 20 MHz (3/3)**

<b>OUTBSEL</b>	<b>Mr</b>	<b>Par</b>	<b>N5</b>	<b>N4</b>	<b>N3</b>	<b>N2</b>	<b>N1</b>	<b>N0</b>	<b>Nr</b>	<b>f<sub>PPLLCLK</sub> Frequency (MHz)</b>	<b>f<sub>CPLLCLK</sub> Frequency (MHz)</b>
0	2	4	0	1	1	1	1	0	31	77.50	←
0	2	4	0	1	1	1	1	1	32	80.00	←
1	2	6	1	0	1	1	1	1	48	80.00	96.00 <sup>*1</sup>

Note 1. 96 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

**MainOSC = 24 MHz (1/2)**

<b>OUTBSEL</b>	<b>Mr</b>	<b>Par</b>	<b>N5</b>	<b>N4</b>	<b>N3</b>	<b>N2</b>	<b>N1</b>	<b>N0</b>	<b>Nr</b>	<b>f<sub>PPLLCLK</sub> Frequency (MHz)</b>	<b>f<sub>CPLLCLK</sub> Frequency (MHz)</b>
0	1	16	0	1	0	0	1	1	20	30.00	←
0	1	8	0	1	0	0	1	1	20	60.00	←
0	2	16	1	0	0	1	1	1	40	30.00	←
0	2	8	0	1	0	0	1	1	20	30.00	←
0	2	8	0	1	0	1	0	0	21	31.50	←
0	2	8	0	1	0	1	0	1	22	33.00	←
0	2	8	0	1	0	1	1	0	23	34.50	←
0	2	8	0	1	0	1	1	1	24	36.00	←
0	2	8	0	1	1	0	0	0	25	37.50	←
0	2	8	0	1	1	0	0	0	26	39.00	←
0	2	8	0	1	1	0	1	0	27	40.50	←
0	2	8	0	1	1	0	1	1	28	42.00	←
0	2	8	0	1	1	1	0	0	29	43.50	←
0	2	8	0	1	1	1	0	1	30	45.00	←
0	2	8	0	1	1	1	1	0	31	46.50	←
0	2	8	0	1	1	1	1	1	32	48.00	←
0	2	8	1	0	0	0	0	0	33	49.50	←
0	2	8	1	0	0	0	0	1	34	51.00	←
0	2	8	1	0	0	0	1	0	35	52.50	←
0	2	8	1	0	0	0	1	1	36	54.00	←
0	2	8	1	0	0	1	0	0	37	55.50	←
0	2	8	1	0	0	1	0	1	38	57.00	←
0	2	8	1	0	0	1	1	0	39	58.50	←
0	2	8	1	0	0	1	1	1	40	60.00	←
0	2	4	0	1	0	0	1	1	20	60.00	←
0	2	4	0	1	0	1	0	0	21	63.00	←
0	2	4	0	1	0	1	0	1	22	66.00	←
0	2	4	0	1	0	1	1	0	23	69.00	←
0	2	4	0	1	0	1	1	1	24	72.00	←
0	2	4	0	1	1	0	0	0	25	75.00	←
0	2	4	0	1	1	0	0	1	26	78.00	←
0	3	16	1	1	1	0	1	1	60	30.00	←
0	3	8	0	1	1	1	0	1	30	30.00	←
0	3	8	0	1	1	1	1	0	31	31.00	←
0	3	8	0	1	1	1	1	1	32	32.00	←

MainOSC = 24 MHz (2/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	f <sub>PPLLCLK</sub> Frequency (MHz)	f <sub>CPLLCLK</sub> Frequency (MHz)
0	3	8	1	0	0	0	0	0	33	33.00	←
0	3	8	1	0	0	0	0	1	34	34.00	←
0	3	8	1	0	0	0	1	0	35	35.00	←
0	3	8	1	0	0	0	1	1	36	36.00	←
0	3	8	1	0	0	1	0	0	37	37.00	←
0	3	8	1	0	0	1	0	1	38	38.00	←
0	3	8	1	0	0	1	1	0	39	39.00	←
0	3	8	1	0	0	1	1	1	40	40.00	←
0	3	8	1	0	1	0	0	0	41	41.00	←
0	3	8	1	0	1	0	0	1	42	42.00	←
0	3	8	1	0	1	0	1	0	43	43.00	←
0	3	8	1	0	1	0	1	1	44	44.00	←
0	3	8	1	0	1	1	0	0	45	45.00	←
0	3	8	1	0	1	1	0	1	46	46.00	←
0	3	8	1	0	1	1	1	0	47	47.00	←
0	3	8	1	0	1	1	1	1	48	48.00	←
0	3	8	1	1	0	0	0	0	49	49.00	←
0	3	8	1	1	0	0	0	1	50	50.00	←
0	3	8	1	1	0	0	1	0	51	51.00	←
0	3	8	1	1	0	0	1	1	52	52.00	←
0	3	8	1	1	0	1	0	0	53	53.00	←
0	3	8	1	1	0	1	0	1	54	54.00	←
0	3	8	1	1	0	1	1	0	55	55.00	←
0	3	8	1	1	0	1	1	1	56	56.00	←
0	3	8	1	1	1	0	0	0	57	57.00	←
0	3	8	1	1	1	0	0	1	58	58.00	←
0	3	8	1	1	1	0	1	0	59	59.00	←
0	3	8	1	1	1	0	1	1	60	60.00	←
0	3	4	0	1	1	1	0	1	30	60.00	←
0	3	4	0	1	1	1	1	0	31	62.00	←
0	3	4	0	1	1	1	1	1	32	64.00	←
0	3	4	1	0	0	0	0	0	33	66.00	←
0	3	4	1	0	0	0	0	1	34	68.00	←
0	3	4	1	0	0	0	1	0	35	70.00	←
0	3	4	1	0	0	0	1	1	36	72.00	←
0	3	4	1	0	0	1	0	0	37	74.00	←
0	3	4	1	0	0	1	0	1	38	76.00	←
0	3	4	1	0	0	1	1	0	39	78.00	←
0	3	4	1	0	0	1	1	1	40	80.00	←
1	2	6	1	0	0	1	1	1	40	80.00	96.00 <sup>*1</sup>
1	3	6	1	1	1	0	1	1	60	80.00	96.00 <sup>*1</sup>

Note 1. 96 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

### 10.4.3 Clock Selector Control Register

#### 10.4.3.1 WDTA0 Clock Domain C\_AWO\_WDTA

##### (1) CKSC\_AWDTAD\_CTL — C\_AWO\_WDTA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2000<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADCSID [1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.17 CKSC\_AWDTAD\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AWDTADCSID [1:0]	Clock Divider Setting for C_AWO_WDTA 00 <sub>B</sub> : Setting prohibited 01 <sub>B</sub> : LS IntOSC / 128 (default) 10 <sub>B</sub> : LS IntOSC / 1 11 <sub>B</sub> : Setting prohibited

#### CAUTION

Confirm that CKSC\_AWDTAD\_CTL is CKSC\_AWDTAD\_ACT before setting the CKSC\_AWDTAD\_CTL register.

## (2) CKSC\_AWDTAD\_ACT — C\_AWO\_WDTA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2008<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADACT [1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 <sup>*1</sup>	0 <sup>*1</sup>
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value in the inactive state. The value becomes 01<sub>B</sub> after entering the active state.

Table 10.18 CKSC\_AWDTAD\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AWDTADACT [1:0]	Clock divider for currently active C_AWO_WDTA

### (3) CKSC\_AWDTAD\_STPM — C\_AWO\_WDTA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2018<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTA DSTPM SK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.19 CKSC\_AWDTAD\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AWDTAD STPMSK	0: Clock domain C_AWO_WDTA is stopped in stand-by mode. 1: Clock domain C_AWO_WDTA is not stopped in stand-by mode.

### 10.4.3.2 TAUJ Clock Domain C\_AWO\_TAUJ

#### (1) CKSC\_ATAUJS\_CTL — C\_AWO\_TAUJ Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2100<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 10.20 CKSC\_ATAUJS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJSCSID [2:0]	Source Clock Setting for C_AWO_TAUJ* <sup>1</sup> 000 <sub>B</sub> : Disabled 001 <sub>B</sub> : HS IntOSC (default) 010 <sub>B</sub> : MainOSC 011 <sub>B</sub> : LS IntOSC 100 <sub>B</sub> : PPLLCLK2 Other than above: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

**(2) CKSC\_ATAUJS\_ACT — C\_AWO\_TAUJ Source Clock Active Register**

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2108<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.21 CKSC\_ATAUJS\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJSACT	Source clock for currently active C_AWO_TAUJ [2:0]

### (3) CKSC\_ATAUJD\_CTL — C\_AWO\_TAUJ Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2200<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 10.22 CKSC\_ATAUJD\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJDCSID [2:0]	Clock Divider Setting for C_AWO_TAUJ 000 <sub>B</sub> : Setting prohibited 001 <sub>B</sub> : CKSC_ATAUJS_CTL selection /1 (default) 010 <sub>B</sub> : CKSC_ATAUJS_CTL selection /2 011 <sub>B</sub> : CKSC_ATAUJS_CTL selection /4 100 <sub>B</sub> : CKSC_ATAUJS_CTL selection /8 Other than above: Setting prohibited

#### (4) CKSC\_ATAUJD\_ACT — C\_AWO\_TAUJ Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2208<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.23 CKSC\_ATAUJD\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJDACT [2:0]	Clock divider for currently active C_AWO_TAUJ

### (5) CKSC\_ATAUJD\_STPM — C\_AWO\_TAUJ Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2218<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJ DSTP MSK	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.24 CKSC\_ATAUJD\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset
0	ATAUJD STPMSK	0: Clock domain C_AWO_TAUJ is stopped in stand-by mode. Clock domain C_AWO_TAUJ is stopped by disabling the C_AWO_TAUJ source clock selection register (CKSC_ATAUJS_CTL = 0000 0000 <sub>H</sub> ). 1: Clock domain C_AWO_TAUJ is not stopped in stand-by mode. Clock domain C_AWO_TAUJ is not stopped by disabling the C_AWO_TAUJ source clock selection register (CKSC_ATAUJS_CTL = 0000 0001 <sub>H</sub> ).

### 10.4.3.3 RTCA Clock Domain C\_AWO\_RTCA

#### (1) CKSC\_ARTCAS\_CTL — C\_AWO\_RTCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by a power-up reset PURS.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2300<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.25 CKSC\_ARTCAS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ARTCASCSID [1:0]	Source Clock Setting for C_AWO_RTCA 00 <sub>B</sub> : Disable (default) 01 <sub>B</sub> : SubOSC* <sup>1</sup> 10 <sub>B</sub> : MainOSC* <sup>2</sup> 11 <sub>B</sub> : LS IntOSC

Note 1. Supported only for 144 pin and 176 pin products.

Note 2. To avoid supplying a clock signal at a frequency of 4 MHz or higher to the C\_AWO\_RTCA clock domain due to the CKSC\_ARTCAS\_CTL register setting, check that CKSC\_ARTCAD\_ACT = 0000 0000<sub>H</sub> (disabled) when the setting of CKSC\_ARTCAS\_CTL is 10<sub>B</sub> (MainOSC).

**(2) CKSC\_ARTCAS\_ACT — C\_AWO\_RTCA Source Clock Active Register**

This register is initialized by a power-up reset PURES.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2308<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.26 CKSC\_ARTCAS\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ARTCASACT [1:0]	Source clock for currently active C_AWO_RTCA

### (3) CKSC\_ARTCAD\_CTL — C\_AWO\_RTCA Clock Divider Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by a power-up reset PURS.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2400<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 10.27 CKSC\_ARTCAD\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ARTCADCSID [2:0]	Clock Divider Setting for C_AWO_RTCA 000 <sub>B</sub> : Disabled (default) 001 <sub>B</sub> : CKSC_ARTCAS_CTL selection /1 010 <sub>B</sub> : CKSC_ARTCAS_CTL selection /2 011 <sub>B</sub> : CKSC_ARTCAS_CTL selection /4 100 <sub>B</sub> : CKSC_ARTCAS_CTL selection /8 Other than above: Setting prohibited

**(4) CKSC\_ARTCAD\_ACT — C\_AWO\_RTCA Clock Divider Active Register**

This register is initialized by a power-up reset PURES.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2408<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.28 CKSC\_ARTCAD\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ARTCADACT [2:0]	Clock divider for currently active C_AWO_RTCA

### (5) CKSC\_ARTCAD\_STPM — C\_AWO\_RTCA Stop Mask Register

This register is initialized by a power-up reset PURES.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2418<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCA DSTPM SK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.29 CKSC\_ARTCAD\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ARTCAD STPMSK	0: Clock domain C_AWO_RTCA is stopped in stand-by mode. Clock domain C_AWO_RTCA is stopped by disabling the C_AWO_RTCA source clock selection register (CKSC_ARTCAS_CTL = 0000 0000 <sub>H</sub> ). 1: Clock domain C_AWO_RTCA is not stopped in stand-by mode. Clock domain C_AWO_RTCA is not stopped by disabling the C_AWO_RTCA source clock selection register (CKSC_ARTCAS_CTL = 0000 0001 <sub>H</sub> ).

#### 10.4.3.4 ADCA0 Clock Domain C\_AWO\_ADCA

##### (1) CKSC\_AADCAS\_CTL — C\_AWO\_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2500<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.30 CKSC\_AADCAS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCASCSID [1:0]	Source Clock Setting for C_AWO_ADCA <sup>*1</sup> 00 <sub>B</sub> : Disabled 01 <sub>B</sub> : HS IntOSC (default) 10 <sub>B</sub> : MainOSC 11 <sub>B</sub> : PPLLCLK2

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

## (2) CKSC\_AADCAS\_ACT — C\_AWO\_ADCA Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2508<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.31 CKSC\_AADCAS\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCASACT [1:0]	Source clock for currently active C_AWO_ADCA

### (3) CKSC\_AADCAD\_CTL — C\_AWO\_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

	<b>Access:</b> This register can be read or written in 32-bit units.															
	<b>Address:</b> FFF8 2600H															
	<b>Value after reset:</b> 0000 0001H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 10.32 CKSC\_AADCAD\_CTL Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCADCSID [1:0]	Clock Divider Setting for C_AWO_ADCA 00 <sub>B</sub> : Setting prohibited 01 <sub>B</sub> : CKSC_AADCAS_CTL selection /1 (default) 10 <sub>B</sub> : CKSC_AADCAS_CTL selection /2 <sup>1</sup> 11 <sub>B</sub> : Setting prohibited

Note 1. Make sure that the frequency of CKSC\_AADCA is no less than 8 MHz after division by 2.

#### (4) CKSC\_AADCAD\_ACT — C\_AWO\_ADCA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2608<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.33 CKSC\_AADCAD\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCADACT [1:0]	Clock divider for currently active C_AWO_ADCA

### (5) CKSC\_AADCAD\_STPM — C\_AWO\_ADCA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2618<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCA DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.34 CKSC\_AADCAD\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AADCAD STPMSK	0: Clock domain C_AWO_ADCA is stopped in stand-by mode. Clock domain C_AWO_ADCA is stopped by disabling the C_AWO_ADCA source clock selection register (CKSC_AADCAS_CTL = 0000 0000 <sub>H</sub> ). 1: Clock domain C_AWO_ADCA is not stopped in stand-by mode. Clock domain C_AWO_ADCA is not stopped by disabling the C_AWO_ADCA source clock selection register (CKSC_AADCAS_CTL = 0000 0000 <sub>H</sub> ).

### 10.4.3.5 FOUT Clock Domain C\_AWO\_FOUT

#### (1) CKSC\_AFOUTS\_CTL — C\_AWO\_FOUT Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register  
For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2700<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSCSID[2:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 10.35 CKSC\_AFOUTS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	AFOUTSCSID [2:0]	Source Clock Setting for C_AWO_FOUT <sup>*1</sup> 000 <sub>B</sub> : Disabled (default) 001 <sub>B</sub> : MainOSC 010 <sub>B</sub> : HS IntOSC 011 <sub>B</sub> : LS IntOSC 100 <sub>B</sub> : SubOSC <sup>*2</sup> 101 <sub>B</sub> : CPLLCLK2 110 <sub>B</sub> : PPLLCLK4 111 <sub>B</sub> : Setting prohibited

Note 1. 在进入待机模式之前，选择除了CPLLCLK2和PPLLCLK4之外的时钟  
Before transitioning to stand-by mode, select a source clock other than CPLLCLK2 and PPLLCLK4.

Note 2. Supported only for 144 pin and 176 pin products.

## (2) CKSC\_AFOUTS\_ACT — C\_AWO\_FOUT Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2708<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSACT[2:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.36 CKSC\_AFOUTS\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	AFOUTSACT [2:0]	Source clock for currently active C_AWO_FOUT

**(3) CKSC\_AFOUTS\_STPM — C\_AWO\_FOUT Stop Mask Register**

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2718<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUT SSTPM SK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

**CAUTION**

Do not change the value after reset of bit 1 from “1”.

**Table 10.37 CKSC\_AFOUTS\_STPM Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AFOUTS STPMSK	0: Clock domain C_AWO_FOUT is stopped in stand-by mode. 1: Clock domain C_AWO_FOUT is not stopped in stand-by mode.

### 10.4.3.6 CPU Clock Domain C\_ISO\_CPUCLK

#### (1) CKSC\_CPUCLKS\_CTL — C\_ISO\_CPUCLK Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A000<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.38 CKSC\_CPUCLKS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CPUCLKSCSID [1:0]	Source Clock Setting for C_ISO_CPUCLK <b>设置时钟源</b> 00 <sub>B</sub> : Setting prohibited 01 <sub>B</sub> : EMCLK (default) 10 <sub>B</sub> : MainOSC 11 <sub>B</sub> : CPLLCLK

#### CAUTION

The clock source selected for the C\_ISO\_CPUCLK clock domain should not be stopped by software.

**用于C\_ISO\_CPUCLK 时钟域的时钟源是不能被软件所停止的**

## (2) CKSC\_CPUCLKS\_ACT — C\_ISO\_CPUCLK Source Clock Active Register

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A008H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.39 CKSC\_CPUCLKS\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ACPUCLKSACT [1:0]	Source clock for currently active C_ISO_CPUCLK 获知是哪一个时钟源？
		01B: EMCLK (default) 10B: MainOSC 11B: CPLLCLK

### (3) CKSC\_CPUCLKD\_CTL — C\_ISO\_CPUCLK Clock **Divider** Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A100<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDCSID[2:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

**Table 10.40 CKSC\_CPUCLKD\_CTL Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CPUCLKDCSID [2:0]	Clock Divider Setting for C_ISO_CPUCLK 000 <sub>B</sub> : Setting prohibited 001 <sub>B</sub> : CKSC_CPUCLKS_CTL selection /1 (Default) 010 <sub>B</sub> : CKSC_CPUCLKS_CTL selection /2 011 <sub>B</sub> : CKSC_CPUCLKS_CTL selection /4 100 <sub>B</sub> : CKSC_CPUCLKS_CTL selection /8 Other than above: Setting prohibited

**(4) CKSC\_CPUCLKD\_ACT — C\_ISO\_CPUCLK Clock Divider Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A108H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDACT [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.41 CKSC\_CPUCLKD\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	CPUCLKDACT [2:0]	Clock divider for currently active C_ISO_CPUCLK 获知是哪种分频方式？ 001B: CKSC_CPUCLKS_CTL selection /1 (Default) 010B: CKSC_CPUCLKS_CTL selection /2 011B: CKSC_CPUCLKS_CTL selection /4 100B: CKSC_CPUCLKS_CTL selection /8

### 10.4.3.7 Peripheral Clock Domains C\_ISO\_PERI1 and C\_ISO\_PERI2

#### (1) CKSC\_IPERI1S\_CTL — C\_ISO\_PERI1 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A200H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.42 CKSC\_IPERI1S\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI1SCSID [1:0]	Source Clock Setting for C_ISO_PERI1 为C_ISO_PERI1选择时钟源 00 <sub>B</sub> : Disabled 01 <sub>B</sub> : CPUCLK2 (default) 10 <sub>B</sub> : PPLLCLK 11 <sub>B</sub> : Setting prohibited

**(2) CKSC\_IPERI1S\_ACT — C\_ISO\_PERI1 Source Clock Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A208H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.43 CKSC\_IPERI1S\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI1SACT [1:0]	Source clock for currently active C_ISO_PERI1

### (3) CKSC\_IPERI2S\_CTL — C\_ISO\_PERI2 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A300<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.44 CKSC\_IPERI2S\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI2SCSID [1:0]	Source Clock Setting for C_ISO_PERI2 00 <sub>b</sub> : Disabled 01 <sub>b</sub> : CPUCLK2 (default) 10 <sub>b</sub> : PPLLCLK2 11 <sub>b</sub> : Setting prohibited

#### (4) CKSC\_IPERI2S\_ACT — C\_ISO\_PERI2 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A308H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.45 CKSC\_IPERI2S\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI2SACT	Source clock for currently active C_ISO_PERI2 [1:0]

### 10.4.3.8 RLIN Clock Domains C\_ISO\_LIN

#### (1) CKSC\_ILINS\_CTL — C\_ISO\_LIN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A400H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSCSID[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.46 CKSC\_ILINS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ILINSCSID[1:0]	Source Clock Setting for C_ISO_LIN <sup>*1</sup> 00 <sub>B</sub> : Disabled 01 <sub>B</sub> : CPUCLK2 (default) 10 <sub>B</sub> : MainOSC 11 <sub>B</sub> : PPLLCLK2

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

**(2) CKSC\_ILINS\_ACT — C\_ISO\_LIN Source Clock Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A408H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSACT[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.47 CKSC\_ILINS\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ILINSACT[1:0]	Source clock for currently active C_ISO_LIN

### (3) CKSC\_ILIND\_CTL — C\_ISO\_LIN Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A800<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.48 CKSC\_ILIND\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ILINDCSID[1:0]	Clock Divider Setting for C_ISO_LIN 00 <sub>B</sub> : Setting prohibited 01 <sub>B</sub> : CKSC_ILINS_CTL selection /1 (default) 10 <sub>B</sub> : CKSC_ILINS_CTL selection /4 11 <sub>B</sub> : CKSC_ILINS_CTL selection /8

#### NOTE

The setting of this register is only applicable to RLIN30.

The settings 10<sub>B</sub> (CKSC\_ILINS\_CTL selection /4) and 11<sub>B</sub> (CKSC\_ILINS\_CTL selection /8) are only available in UART mode.

**(4) CKSC\_ILIND\_ACT — C\_ISO\_LIN Clock Divider Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A808H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDACT[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.49 CKSC\_ILIND\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ILINDACT[1:0]	Clock divider for currently active C_ISO_LIN

### (5) CKSC\_ILIND\_STPM — C\_ISO\_LIN Stop Mask Register

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A818H

**Value after reset:** 0000 0002H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILIND STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.50 CKSC\_ILIND\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ILINDSTPMSK	0: Clock domain C_ISO_LIN is stopped in stand-by mode. Clock domain C_ISO_LIN is stopped by disabling the C_ISO_LIN source clock selection register (CKSC_ILINS_CTL = 0000 0000H). 1: Clock domain C_ISO_LIN is not stopped in stand-by mode. Clock domain C_ISO_LIN is not stopped by disabling the C_ISO_LIN source clock selection register (CKSC_ILINS_CTL = 0000 0000H).

### 10.4.3.9 ADCA1 Clock Domain C\_ISO\_ADCA

#### (1) CKSC\_IADCAS\_CTL — C\_ISO\_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A500<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.51 CKSC\_IADCAS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IADCASCSID [1:0]	Source Clock Setting for C_ISO_ADCA 00 <sub>B</sub> : Disabled 01 <sub>B</sub> : HS IntOSC (default) 10 <sub>B</sub> : MainOSC 11 <sub>B</sub> : PLLCLK2

#### CAUTION

The CKSC\_IADCAS\_CTL register and the CKSC\_IADCAD\_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 2 to 4.8”.

- (1) Frequency [MHz] specified by the C\_ISO\_CPUCLK source clock selection register (CKSC\_CPUCLKS\_CTL) and C\_ISO\_CPUCLK clock divider selection register (CKSC\_CPUCLKD\_CTL)
- (2) Frequency [MHz] specified by the C\_ISO\_ADCA source clock selection register (CKSC\_IADCAS\_CTL) and the C\_ISO\_ADCA clock divider selection register (CKSC\_IADCAD\_CTL)

## (2) CKSC\_IADCAS\_ACT — C\_ISO\_ADCA Source Clock Active Register

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A508<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.52 CKSC\_IADCAS\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IADCASACT [1:0]	Source clock for currently active C_ISO_ADCA

### (3) CKSC\_IADCAD\_CTL — C\_ISO\_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A600H

**Value after reset:** 0000 0001H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.53 CKSC\_IADCAD\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IADCADCSID [1:0]	Clock Divider Setting for C_ISO_ADCA 00 <sub>B</sub> : Setting prohibited 01 <sub>B</sub> : CKSC_IADCAS_CTL selection /1 (default) 10 <sub>B</sub> : CKSC_IADCAS_CTL selection /2 <sup>1</sup> 11 <sub>B</sub> : Setting prohibited

Note 1. Make sure that the frequency of CKSC\_IADCA is no less than 8 MHz after division by 2.

#### CAUTION

The CKSC\_IADCAS\_CTL register and the CKSC\_IADCAD\_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 2 to 4.8”.

- (1) Frequency [MHz] specified by the C\_ISO\_CPUCLK source clock selection register (CKSC\_CPUCLKS\_CTL) and C\_ISO\_CPUCLK clock divider selection register (CKSC\_CPUCLKD\_CTL)
- (2) Frequency [MHz] specified by the C\_ISO\_ADCA source clock selection register (CKSC\_IADCAS\_CTL) and the C\_ISO\_ADCA clock divider selection register (CKSC\_IADCAD\_CTL)

**(4) CKSC\_IADCAD\_ACT — C\_ISO\_ADCA Clock Divider Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A608<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.54 CKSC\_IADCAD\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IADCADACT	Clock divider for currently active C_ISO_ADCA [1:0]

### 10.4.3.10 RS-CAN Clock Domains C\_ISO\_CAN and C\_ISO\_CANOSC

#### (1) CKSC\_ICANS\_CTL — C\_ISO\_CAN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A900<sub>H</sub>

**Value after reset:** 0000 0003<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSCSID[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.55 CKSC\_ICANS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANSCSID[1:0]	Source Clock Setting for C_ISO_CAN <sup>*1, *2</sup> 00 <sub>B</sub> : Disabled 01 <sub>B</sub> : MainOSC 10 <sub>B</sub> : CPLLCLK 11 <sub>B</sub> : CPUCLK (default)

Note 1. Before transitioning to stand-by mode, select a source clock other than CPLLCLK.

Note 2. Before shifting to STOP mode, select MainOSC.

## (2) CKSC\_ICANS\_ACT — C\_ISO\_CAN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 A908<sub>H</sub>

**Value after reset:** 0000 0003<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSACT[1:0]
	Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.56 CKSC\_ICANS\_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANSACT[1:0]	Source clock for currently active C_ISO_CAN

### (3) CKSC\_ICANS\_STPM — C\_ISO\_CAN Stop Mask Register

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 A918<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANS STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.57 CKSC\_ICANS\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANS STPMSK	0: Clock domain C_ISO_CAN is stopped in stand-by mode. 1: Clock domain C_ISO_CAN is not stopped in stand-by mode.

#### (4) CKSC\_ICANOSCD\_CTL — C\_ISO\_CANOSC Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 AA00<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCD CSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.58 CKSC\_ICANOSCD\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANOSCD CSID[1:0]	Clock Divider Setting for C_ISO_CANOSC* <sup>1</sup> 00 <sub>B</sub> : Disabled (default) 01 <sub>B</sub> : MainOSC /1 10 <sub>B</sub> : MainOSC /2 11 <sub>B</sub> : Setting prohibited

Note 1. Before shifting to STOP mode, select MainOSC/2.

**(5) CKSC\_ICANOSCD\_ACT — C\_ISO\_CANOSC Clock Divider Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 AA08<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.59 CKSC\_ICANOSCD\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANOSCDACT [1:0]	Clock divider for currently active C_ISO_CANOSC

### (6) CKSC\_ICANOSCD\_STPM — C\_ISO\_CANOSC Stop Mask Register

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 AA18<sub>H</sub>

**Value after reset:** 0000 0002<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANO SCDST PMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

#### CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 10.60 CKSC\_ICANOSCD\_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANOSCD STPMSK	0: Clock domain C_ISO_CANOSC is stopped in stand-by mode. 1: Clock domain C_ISO_CANOSC is not stopped in stand-by mode.

### 10.4.3.11 CSI Clock Domain C\_ISO\_CSI

#### (1) CKSC\_ICSIS\_CTL — C\_ISO\_CSI Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 AB00<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISSCID[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.61 CKSC\_ICSIS\_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICSISSCID[1:0]	Source Clock Setting for C_ISO_CSI 00 <sub>B</sub> : Disabled 01 <sub>B</sub> : CPUCLK (default) 10 <sub>B</sub> : PPLLCLK 11 <sub>B</sub> : Setting prohibited

**(2) CKSC\_ICSIS\_ACT — C\_ISO\_CSI Source Clock Active Register**

This register is initialized by all reset sources (ISORES).

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 AB08<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSSIACT[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 10.62 CKSC\_ICSIS\_ACT Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICSSIACT[1:0]	Source clock for currently active C_ISO_CSI

## 10.5 Clock Domain Setting Method

### 10.5.1 Clock Domain Setting

#### 10.5.1.1 Overview of Clock Selector Register

The clock selector for a clock domain C\_AWO\_<name>/C\_ISO\_<name> can be controlled by the following registers:

- **Source clock selection registers**

These registers select the clock to be used as the domain clock from the available source clocks.  
这些寄存器是从可获取的时钟源当中选择时钟作为某时钟域的时钟

- AWO source clock selection: CKSC\_A<name>S\_CTL
- ISO source clock selection: CKSC\_I<name>S\_CTL

- **Clock divider selection registers**

These registers specify the clock division ratio for the selected source clock.  
这些寄存器是用来指定所选时钟源的分频系数

- AWO clock division ratio: CKSC\_A<name>D\_CTL
- ISO clock division ratio: CKSC\_I<name>D\_CTL

- **Source clock active registers and clock divider active registers**

These registers return the currently active source clock selection and division ratio, respectively.  
这些寄存器返回当前已经选择的时钟和分频系数

- AWO source clock active register/clock divider active register: CKSC\_A<name>S\_ACT/  
CKSC\_A<name>D\_ACT
- ISO source clock active register/clock divider active register: CKSC\_I<name>S\_ACT/  
CKSC\_I<name>D\_ACT

**NOTE**

并不是所有的时钟选择器都会提供上面提到的控制功能

- Not all clock selectors provide all of the control functions described above.
- The symbol “I”, which indicates the power domain, is not added to the names of registers within clock domain C\_ISO\_CPUCLK.

### 10.5.1.2 Setting Procedure for Clock Domain 设置时钟域的过程

Procedure of setting up clock domain is described as below:

1. Set up a source clock **设置时钟源**
  - Select a source clock. (CKSC\_A<name>S\_CTL, CKSC\_I<name>S\_CTL) **选择时钟源**
  - Confirm completion of selection. (CKSC\_A<name>S\_ACT, CKSC\_I<name>S\_ACT)\*<sup>1</sup> **确认选择成功**
2. Set up a clock divider **设置分频系数**
  - Select a clock divider. (CKSC\_A<name>D\_CTL, CKSC\_I<name>D\_CTL) **选择分频系数**
  - Confirm completion of selection. (CKSC\_A<name>D\_ACT, CKSC\_I<name>D\_ACT)\*<sup>2</sup> **确认分频系数成功**

**Note 1.** Continue processing after CKSC\_A<name>S\_ACT and CKSC\_I<name>S\_ACT are updated with the new values written to CKSC\_A<name>S\_CTL and CKSC\_I<name>S\_CTL.

**Note 2.** Continue processing after CKSC\_A<name>D\_ACT and CKSC\_I<name>D\_ACT are updated with the new values written to CKSC\_A<name>D\_CTL and CKSC\_I<name>D\_CTL.

#### CAUTION

**在操作这些设置之前，必须确定被选择的时钟源是正常工作的**  
**The source clock to be selected must be operating before performing these settings.**  
**在时钟源停止后，设置后的行为和性能是不能被保证的**  
**The behavior and performance are not guaranteed when setup is made while the source clock is stopped.**  
**在时钟被应用于该模块之前，不能访问外设模块**  
**Access to a peripheral module is prohibited while the clock is not supplied to the module.**

### 10.5.2 Stopping the Clock in Stand-By Mode

In stand-by mode, clock domain C\_AWO\_<name>/C\_ISO\_<name> can be configured to stop or continue its clock CKSCLK\_A<name>/CKSCLK\_I<name> in response to clock stop requests from the stand-by controller. **在待机模式下，时钟域可以被配置成用来停止或者及继续其时钟，这样来相应来自待机控制器的请求**

The clock stop mask registers are used to determine the operation status of the clock in stand-by mode: **时钟停止屏蔽寄存器用来决定在待机模式下时钟的工作状态**

- **CKSC\_A<name>\_STPM.A<name>STPMSK/CKSC\_I<name>\_STPM.I<name>STPMSK = 0:**  
 The STOP request signal is not masked, so the domain clock CKSCLK\_A<name>/CKSCLK\_I<name> is stopped during stand-by mode.  
 If the domain clock was in operation before transition to stand-by mode, it is automatically re-started after wake-up from stand-by mode.  
 If there is another clock domain which selects same source clock and its stop mask setting is set to 1 (CKSC\_A<name>\_STPM.A<name>STPMSK/CKSC\_I<name>\_STPM.I<name>STPMSK = 1), the source clock will continue operation in stand-by mode. The CPU clock domain C\_ISO\_CPUCLK is always stopped in stand-by mode.
- **CKSC\_A<name>\_STPM.A<name>STPMSK/CKSC\_I<name>\_STPM.I<name>STPMSK = 1:**  
 The STOP request signal is masked, so CKSCLK\_A<name>/CKSCLK\_I<name> continues to operate during stand-by.  
 The source clock selected for the target clock domain will also continue to operate in standby mode.  
**Supply of clock signals to the clock domains in the ISO area will be stopped in DeepSTOP mode.**

### 10.5.3 Clock Domain Settings

The following table shows the selectable source clocks, frequency division ratios, and registers to be used for each clock domain.

Table 10.63 List of Selectable Clocks (1/2)

Clock Domain	Domain Clock	Source Clock Selection Register		Clock Divider Selection Register		Maximum Frequency	Target Unit
C_AWO_WDTA	CKSCLK_AWDTA	—	LS IntOSC	CKSC_AWDTAD_CTL	1/1	240 kHz	WDTA0
					1/128		
C_AWO_TAUJ	CKSCLK_ATAUJ	CKSC_ATAUJS_CTL	MainOSC	CKSC_ATAUJD_CTL	1/1	40 MHz	TAUJ0
			HS IntOSC		1/2		
			LS IntOSC		1/4		
			PPLLCLK2		1/8		
			Disable		—		
C_AWO_RTCA	CKSCLK_ARTCA	CKSC_ARTCAS_CTL	MainOSC	CKSC_ARTCAD_CTL	1/1	4 MHz	RTCA0
			LS IntOSC		1/2		
			SubOSC <sup>5</sup>		1/4		
			Disable		1/8		
			—		Disable		
C_AWO_ADCA	CKSCLK_AADC	CKSC_AADCAS_CTL	MainOSC	CKSC_AADCAD_CTL	1/1	40 MHz	ADCA0
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				
C_AWO_FOUT	CKSCLK_AFOUT	CKSC_AFOUTS_CTL	MainOSC	—	1/1	24 MHz	FOUT
			HS IntOSC				
			LS IntOSC				
			SubOSC <sup>5</sup>				
			CPLLCLK2				
			PPLLCLK4				
C_ISO_CPUCLK	CKSCLK_CPUCLK	CKSC_CPUCLKS_CTL	MainOSC	CKSC_CPUCLKD_CTL	1/1	96 / 80 MHz <sup>3</sup>	CPU subsystem (MEMC) <sup>1</sup>
			CPLLCLK		1/2		
			EMCLK		1/4		
			—		1/8		
C_ISO_PERI1	CKSCLK_IPERI1	CKSC_IPERI1S_CTL	PPLLCLK	—	1/1	80 MHz	TAUD0 TAUJ1 ENCA0 TAPA0 PIC0
			CPUCLK2				
			Disable				
C_ISO_PERI2	CKSCLK_IPERI2	CKSC_IPERI2S_CTL	CPUCLK2	—	1/1	48 / 40 MHz <sup>4</sup>	TAUBn PWBA <sub>n</sub> PWGA <sub>n</sub> PWSAn
			PPLLCLK2				
			Disable				
C_ISO_LIN	CKSCLK_ILIN	CKSC_ILINS_CTL	MainOSC	CKSC_ILIND_CTL <sup>2</sup>	1/1	48 / 40 MHz <sup>4</sup>	RLIN2m RLIN3n
			CPUCLK2		1/4		
			PPLLCLK2		1/8		
			Disable		—		
C_ISO_ADCA	CKSCLK_IADC	CKSC_IADCAS_CTL	MainOSC	CKSC_IADCAD_CTL	1/1	40MHz	ADCA1
			HS IntOSC		1/2		
			PPLLCLK2		—		
			Disable				

Table 10.63 List of Selectable Clocks (2/2)

Clock Domain	Domain Clock	Source Clock Selection Register	Clock Divider Selection Register		Maximum Frequency	Target Unit
C_ISO_CAN	CKSCLK_ICAN	CKSC_ICANS_CTL	MainOSC	—	1/1	96 / 80 MHz <sup>3</sup>
			CPLLCLK			
			CPUCLK			
			Disable			
C_ISO_CANOSC	CKSCLK_ICANOSC	—	MainOSC	CKSC_ICANOSCD_CTL	1/1	24 MHz
					1/2	
					Disable	
C_ISO_CSI	CKSCLK_ICSI	CKSC_ICSI_S_CTL	PPLLCLK	—	1/1	96 / 80 MHz <sup>3</sup>
			CPUCLK			
			Disable			

**Note:** The items written in **bold** are the default clocks for each register.

Note 1. The MEMC clock is set based on C\_ISO\_CPUCLK using the option bytes.

Note 2. The setting of this register is only applicable to RLIN30. The settings 1/4 and 1/8 are only available in UART mode.

Note 3. 96 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

Note 4. 48 MHz is supported only by ADVANCED/PREMIUM/Gateway-1MB.

Note 5. Supported only for 144 pin and 176 pin products.

### CAUTION

To stop the source clock selected for the clock domain before transitioning to STOP/DeepSTOP mode, select “Disable” for that clock domain in advance. Do not stop the source clock of a clock domain for which “Disable” cannot be selected while functions are operating on that clock domain. To stop the clock source selected for the domain by transitioning to STOP/DeepSTOP mode, “Disable” does not need to be selected. Instead of the setting, select stop of clock domain during stand-by mode by the stop mask register.

## 10.6 Frequency Output Function (FOUT) 频率输出功能

允许输出时钟作为外部信号

The frequency output function (FOUT) allows to output the clock as the external signal. Furthermore, the frequency can be divided by the clock divider before it is output.  
进一步来讲，频率在输出之前可以被分频

### 10.6.1 Functional Overview

Figure 10.9 shows the configuration of the frequency output function.

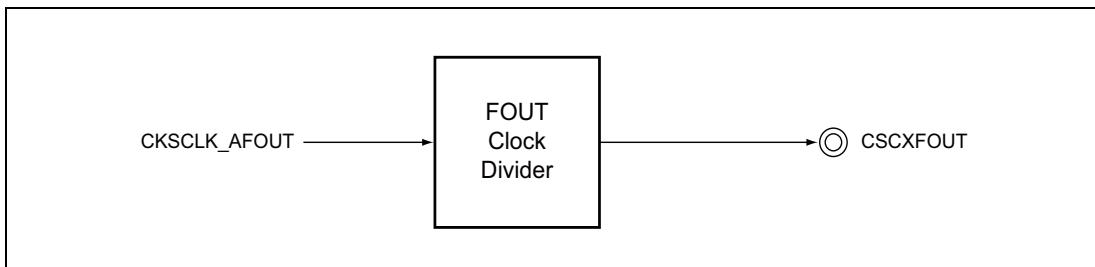


Figure 10.9 Frequency Output Function

The clock output function outputs the CKSCLK\_AFOUT clock divided by 1 to 63 through the clock divider from CSCXFOUT. Division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register. Clock output frequency  $f_{CSCXFOUT}$  is expressed by the following equation:

$$f_{CSCXFOUT} = (\text{CKSCLK\_AFOUT clock frequency}) / N$$

Clock output starts when, after CKSCLK\_AFOUT is set and the clock output for the pin function is selected, division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register.

分频系数即N可以随时更改，更改后就可以使能

When a new division ratio is written to the FOUTDIV.FOUTDIV[5:0] bits, it becomes effective in synchronization with the CSCXFOUT output clock. Accordingly, the division ratio can be changed even while the CSCXFOUT clock is operating. The clock output is stopped by writing 000H to the FOUTDIV[5:0] bits.

写入000H既可以停止时钟的输出

## 10.6.2 Registers

### 10.6.2.1 List of Registers

The FOUT registers are listed in the following table.

**Table 10.64 CSCXFOUT Clock Divider Registers**

Register Name	Symbol	Address
Clock division ratio register	FOUTDIV	FFF8 2800 <sub>H</sub>
Clock divider status register	FOUTSTAT	FFF8 2804 <sub>H</sub>

### 10.6.2.2 Clock Supply

The clock supply to the CSCXFOUT is shown in the following table.

**Table 10.65 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name	Description
CSCXFOUT	PCLK	CPUCLK2	Bus clock (Register access) 总线时钟
	CKSCLK_AFOUT	CKSCLK_AFOUT	Clock source of FOUT clock divider FOUT时钟分频器的时钟源

### 10.6.2.3 FOUTDIV — Clock Division Ratio Register

This register defines the clock division ratio.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).  
除非进入DeepSTOP mode，此寄存器可以被所有的复位源初始化

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 2800<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	FOUTDIV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.66 FOUTDIV Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	FOUTDIV[5:0]	Clock Division Ratio N 00 <sub>H</sub> : Clock output is stopped 01 <sub>H</sub> : N = 1 02 <sub>H</sub> : N = 2 ... 3E <sub>H</sub> : N = 62 3F <sub>H</sub> : N = 63

#### 10.6.2.4 FOUTSTAT — Clock Divider Status Register

表征时钟输出的状态

This register indicates the clock output status.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

除非进入DeepSTOP mode，此寄存器可以所有的复位源初始化

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFF8 2804<sub>H</sub>

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.67 FOUTSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUTCLKACT	Clock Divider Output Active 时钟分频器激活的状态 0: Frequency output is stopped. 1: Frequency output is active.
0	FOUTSYNC	Clock Divider Synchronized 时钟分频器是否已经同步 0: The clock divider is in the process of synchronization. 1: The clock divider is stable (or stopped).

## 10.7 Clock Monitor A (CLMA) 时钟检测器

The following sections describe clock monitor A (CLMA).

**第一章节介绍指定MCU的属性，包含通道数目，寄存器地址，输入/输出信号的名称**

The first section describes the attributes specific to the RH850/F1L microcontrollers, including the number of channels, register base addresses, and input/output signal names. The ensuing sections describe the functions relevant to all operations.

**接下来的部分介绍与操作相关的功能**

### 10.7.1 Features of RH850/F1L CLMA

#### 10.7.1.1 Number of Channels

The RH850/F1L microcontrollers incorporate CLMA with the following number of channels.

Table 10.68 Number of Channels

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of channels				3		
Name				CLMA0, CLMA1, CLMA2		

#### 10.7.1.2 Register Base Addresses

The CLMA base addresses are listed in the following table.

The CLMA register addresses are expressed as offsets from the base addresses.

Table 10.69 Register Base Addresses

Base Address Name	Base Address
<CLMA0_base>	FFF8 C000 <sub>H</sub>
<CLMA1_base>	FFF8 D000 <sub>H</sub>
<CLMA2_base>	FFF8 E000 <sub>H</sub>

#### 10.7.1.3 Clock Supply

The clocks monitored by CLMA and the CLMA sampling clocks are indicated below.

Table 10.70 Clock Supply

Channel Name	Unit Clock Name	Clock Name
CLMA0	CLMATMON (monitored clock)	HS IntOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK2
CLMA1	CLMATMON (monitored clock)	MainOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK2
CLMA2	CLMATMON (monitored clock)	CPLLCLK
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK2

#### 10.7.1.4 Reset Sources

The reset sources of the CLMA are listed in the following table. The CLMA are initialized by these reset sources. **CLMA可以被这些复位源所初始化**

**Table 10.71 Reset Sources**

Channel Name	Reset Source
CLMA0	Reset sources other than transition to the DeepSTOP mode (AWORES)
CLMA1	Reset sources other than transition to the DeepSTOP mode (AWORES)
CLMA2	All reset source (ISORES)
Common Registers (CLMATEST, CLMATESTS)	Reset sources other than transition to the DeepSTOP mode (AWORES)

#### 10.7.1.5 Internal Input/Output Signals

The internal input/output signals of CLMA are listed in the following table.  
**CLMA的内部输入输出信号如下所示**

**Table 10.72 Internal Input/Output Signals**

Unit Signal Name	Description	Connection
CLMATRES	CLMA0 error reset output	Reset controller (CLMA0RES)
CLMATRES	CLMA1 error reset output	Reset controller (CLMA1RES)
CLMATRES	CLMA2 error reset output	Reset controller (CLMA2RES)

#### 10.7.2 Overview

##### 10.7.2.1 Functional Overview

**CLMA在被监测的时钟里面探测频率异常。它使用CLMASMP去监测输入时钟CLMAMON的频率是否在指定的范围内。一旦检测到不正常的时钟，他就会输出reset 请求信号**

Clock monitor CLMA detects frequency abnormalities in the monitored clock. It uses sampling clock CLMASMP to monitor whether the frequency of input clock CLMAMON is within a specific range. Upon detection of an abnormal clock, it outputs a reset request signal.

The main components of the clock monitor are shown in **Figure 10.10.**

**时钟检测器的组成如下所示**

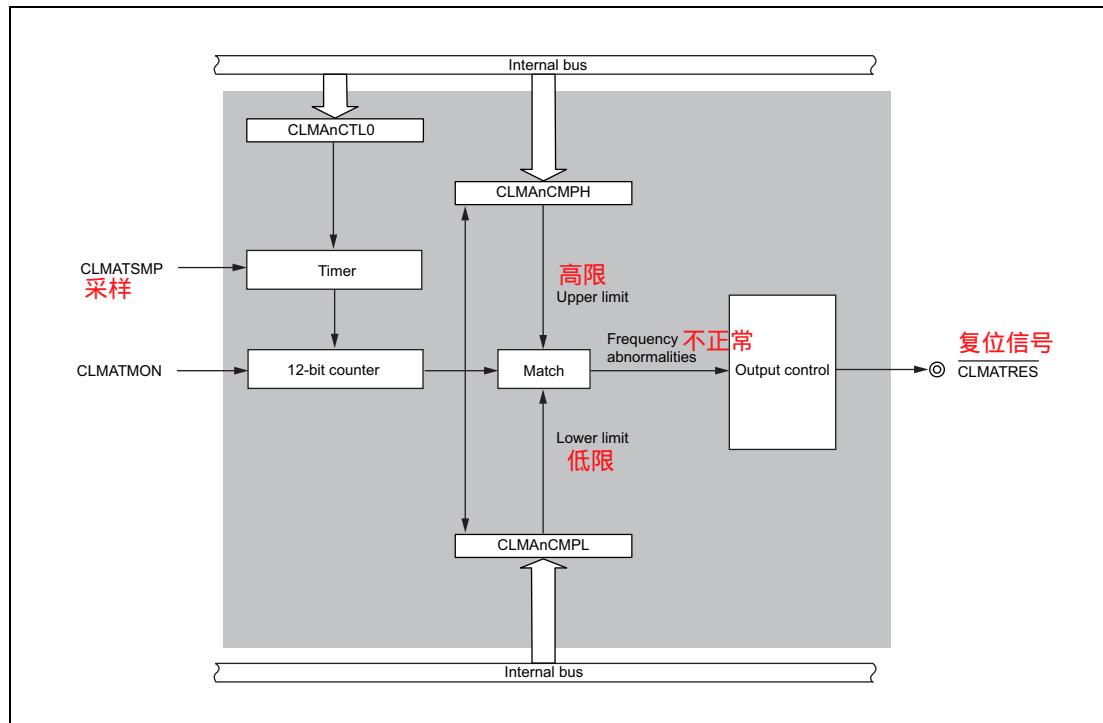


Figure 10.10 Block Diagram of the Clock Monitor A

**NOTE**

Once enabled, only a reset can disable CLMA. 一旦使能，是由复位才能关闭CLMA

### 10.7.3 Enabling CLMA

Clock monitoring is started by the clock monitor when CLMAnCTL0.CLMAAnCLME = 1.

当被检测的时钟由于寄存器的操作或者进入待机模式而停止，其对应的时钟检测器也会自动禁止。When the monitored clock is stopped by register operation or stand-by mode, the corresponding clock monitor is automatically disabled. After the monitored clock starts oscillation again and becomes stable, the clock monitor also starts operation. 当被检测的时钟重新运行并且稳定，时钟检测器同样会重新工作。

Since CLMA2 is initialized on return from DeepSTOP mode, the CLMA2 register must be set again before further operation is started. 从DeepSTOP模式恢复的时候，CLMA2被初始化，在进行操作之前CLMA2必须被重新设置。

### 10.7.4 Functions

#### 10.7.4.1 Detection of Abnormal Clock Frequencies

##### Detection Method 探测方法

1. CLMA counts the rising edges of the monitored clock CLMATMON within 16 cycles of the sampling clock CLMATSMP and then compares the counter value with the specified thresholds:
  - CLMAnCMPL.CLMAAnCMPL[11:0] defines the lower threshold. 低限
  - CLMAnCMPH.CLMAAnCMPH[11:0] defines the upper threshold. 高限
2. When CLMATMON stops or its frequency is lower than the limit, the counter falls below CLMAnCMPL.CLMAAnCMPL[11:0]. 如果CLMATMON停止或者其频率低于限制，计数器会低于低限。
3. When the frequency of CLMATMON is higher than the limit, the counter exceeds CLMAnCMPH.CLMAAnCMPH[11:0]. 如果CLMATMON高于限制，那么计数器就会超过高限。

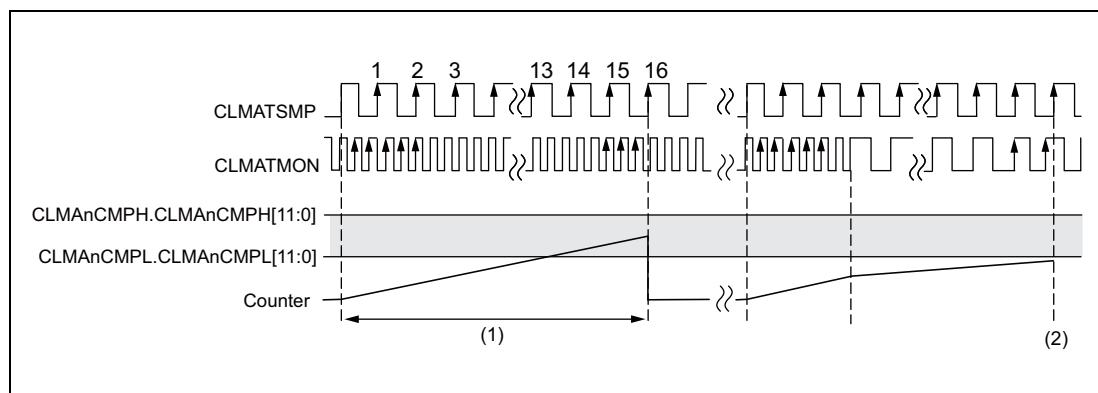


Figure 10.11 Example:  $f_{CLMATMON}$  is Lower than the Specified Limit

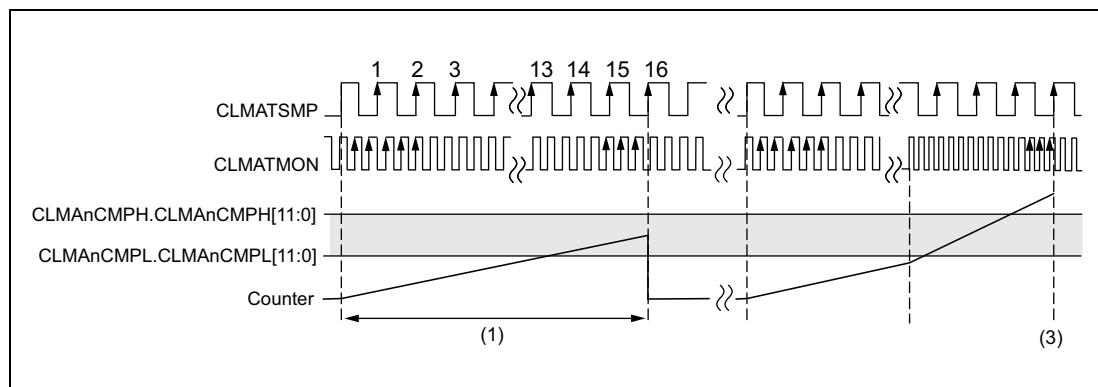


Figure 10.12 Example:  $f_{CLMATMON}$  is Higher than the Specified Limit

**NOTE**

Even if  $f_{CLMATMON}$  exceeds or falls below the specified limits during a sampling interval, the counter might be within the valid range.

Abnormal  $f_{CLMATMON}$  is detected after one sampling interval.

**(1) Calculation method of the thresholds CLMAAnCMPL.CLMAAnCMPL[11:0] and CLMAAnCMPH.CLMAAnCMPH[11:0]**

The compare registers CLMAAnCMPL and CLMAAnCMPH are configured with the minimum and maximum number of clock cycles of CLMATMON that are assumed to be valid within 16 cycles of the sampling clock CLMATSMP.

The expected number of clock cycles is denoted by N.

$$\frac{16}{f_{CLMATSMP}} = \frac{N}{f_{CLMATMON}}$$

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMP}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMP, the threshold values can be calculated by the following formulas:

$$\text{Lower threshold} = N_{\min}$$

$$= \frac{f_{CLMATMON(\min)}}{f_{CLMATSMP(\max)}} \times 16 - 1$$

$$\text{Upper threshold} = N_{\max}$$

$$= \frac{f_{CLMATMON(\max)}}{f_{CLMATSMP(\min)}} \times 16 + 1$$

**NOTE**

The jitter of the PLL is covered by "+1" and "-1" in the formulas.

**Example:**

When  $f_{CLMATSMP} = 240$  kHz ( $\pm 8\%$ ) and  $f_{CLMATMON} = 16$  MHz ( $\pm 5\%$ ), the recommended threshold values are as follows:

$$N_{\min} = 15,200 / 259.2 \times 16 - 1 \quad 16\text{MHz}^* (1-5\%) = 15200\text{kHz}$$

$$= 937.27$$

$$\text{CLMAAnCMPL} = 937 = 03A9_H$$

$$N_{\max} = 16,800 / 220.8 \times 16 + 1 \quad 16\text{MHz}^* (1+5\%) = 16800\text{kHz}$$

$$= 1218.39$$

$$\text{CLMAAnCMPH} = 1219 = 04C3_H$$

### Minimum thresholds

The following restrictions must be taken into account:

- $\text{CLMAnCMPL} \geq 0001_{\text{H}}$
- $\text{CLMAnCMPH} \geq \text{CLMAnCMPL} + 0003_{\text{H}}$

### (2) Definition of the initial value input to the threshold registers

The values after reset of the threshold registers are set so that the maximum frequency deviation of the monitored clock is allowed: threshold寄存器复位后的值要被设置，这样才能允许被监测时钟出现最大的差值

- $\text{CLMAnCMPL}[11:0] = 001_{\text{H}}$
- $\text{CLMAnCMPH}[11:0] = 3FF_{\text{H}}$  为什么是3FF？而不是FFF呢？？？？？？？？？？？？

#### 10.7.4.2 Notification of Abnormal Clock Frequency

If  $f_{\text{CLMATMON}}$  exceeds the upper threshold or falls below the lower threshold, this is indicated as follows: 如果 $f_{\text{CLMATMON}}$ 超过高限或者低于低限，会出现下面的情况

1. The reset request signal  $\overline{\text{CLMATRES}}$  is set to low level. 复位请求信号被设置为低电平
2. The system reset (AWORES and ISORES) is generated and CLMAn is reset. 产生系统复位，并且CLMAn被复位

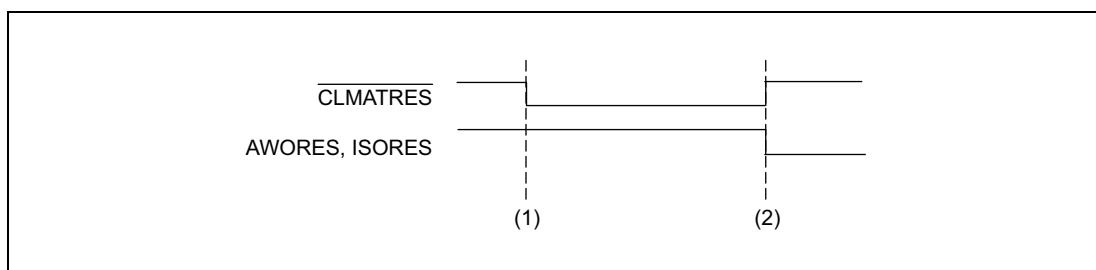


Figure 10.13 Error Request Signal Output if  $f_{\text{CLMATMON}}$  Exceeds Upper Threshold

#### CAUTION

For usage notes for CLMAn abnormality detection, see Section 10.7.6, Usage Notes for CLMAn.

## 10.7.5 Registers

### 10.7.5.1 List of Registers

The following table lists the CLMA registers.

<CLMAn\_base> is defined in **Section 10.7.1.2, Register Base Addresses**.  
 <CLMA0\_base> 0xFFFF C000  
 <CLMA1\_base> 0xFFFF D000  
 <CLMA2\_base> 0xFFFF E000

**Table 10.73 List of Clock Monitor Registers**

Module Name	Register Name	Symbol	Address
CLMAn	CLMAn control register 0	CLMAnCTL0	<CLMAn_base> + 00H
CLMAn	CLMAn compare register L	CLMAnCMPL	<CLMAn_base> + 08H
CLMAn	CLMAn compare register H	CLMAnCMPH	<CLMAn_base> + 0CH
CLMAn	CLMA test register	CLMATEST	FFF8 C100H
CLMAn	CLMA test status register	CLMATESTS	FFF8 C104H
CLMAn	CLMAn emulation register 0	CLMAnEMU0	<CLMAn_base> + 18H

### 10.7.5.2 CLMAnCTL0 — CLMAn Control Register 0

This register enables the clock monitor CLMAn.

The correct write sequence using the CLMAnPCMD register is required in order to update this register. For details, see **Section 4, Write-Protected Registers**.

**Access:** This register can be read or written in 8-bit units.

**Address:** <CLMAn\_base> + 00H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMAnCLME
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

**Table 10.74 CLMAnCTL0 Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CLMAnCLME	Enables or disables the clock monitor. 0: CLMAn is disabled. 1: CLMAn is enabled.

#### CAUTION

After the CLMAnCTL0.CLMAnCLME bit is set to 1, writing 0 to this bit is ignored. The only condition for clearing the bit is a reset (AWORES, ISORES). In addition, the bit is cleared when CLMATEST.RESCLM bit is set to 1 during self-test of CLMAn.

### 10.7.5.3 CLMAAnCMPL — CLMAAn Compare Register H

This register specifies the upper limit of frequency.

It can only be written when CLMAAn is disabled (CLMAAnCTL0.CLMAAnCLME = 0).

For details, see **Section 10.7.4.1 (1) Calculation method of the thresholds CLMAAnCMPL.CLMAAnCMPL[11:0] and CLMAAnCMPLH.CLMAAnCMPLH[11:0]**.

**Access:** This register can be read or written in 16-bit units.

**Address:** <CLMAAn\_base> + 0C<sub>H</sub>

**Value after reset:** 03FF<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAAnCMPLH[11:0]											
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.75 CLMAAnCMPLH Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAAnCMPLH [11:0]	Specifies the upper threshold. <ul style="list-style-type: none"> <li>The recommended value is <math>f_{CLMATMON}(\max) / f_{CLMATSMP}(\min) \times 16 + 1</math>.</li> <li>The minimum value is CLMAAnCMPL + 0003<sub>H</sub>.</li> </ul>

#### 10.7.5.4 CLMAnCMPL — CLMAn Compare Register L

This register specifies the lower limit of frequency.

It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME=0).

For details, see **Section 10.7.4.1 (1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

**Access:** This register can be read or written in 16-bit units.

**Address:** <CLMAn\_base> + 08H

**Value after reset:** 0001H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—												CLMAnCMPL[11:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W											

**Table 10.76 CLMAnCMPL Register Contents**

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPL [11:0]	Specifies the lower threshold. <ul style="list-style-type: none"> <li>• The recommended value is <math>f_{CLMATMON}(\text{min}) / f_{CLMATSMP}(\text{max}) \times 16 - 1</math></li> <li>• The minimum value is 0001H.</li> </ul>

### 10.7.5.5 CLMATEST — CLMA Test Register

This register is used to control self-test of CLMA0, CLMA1, and CLMA2.

The correct write sequence using the PROTCMDCLMA register is required in order to update this register. For details, see **Section 4, Write-Protected Registers**.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 C100<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CLMA2 TESEN	CLMA1 TESEN	CLMA0 TESEN	ERR MSK	MONCL KMSK	RES CLM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 10.77 CLMATEST Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CLMA2TESEN	CLMA2 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
4	CLMA1TESEN	CLMA1 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
3	CLMA0TESEN	CLMA0 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
2	ERRMSK <sup>*1</sup>	CLMA Test Reset Signal Mask Setting 0: Signal generation enabled 1: Signal generation disabled (masked)
1	MONCLKMSK <sup>*1</sup>	Monitor Clock Mask Setting 0: Monitor clock enabled 1: Monitor clock disabled (masked)
0	RESCLM <sup>*1</sup>	CLMA Test Reset Signal Control 0: Reset released 1: Reset executed

Note 1. These bits are effective when the CLMA<sub>n</sub>TESEN bit is 1.

### 10.7.5.6 CLMATESTS — CLMA Test Status Register

This register is used to confirm the self-test result of CLMA0, CLMA1, and CLMA2.

<b>Access:</b>	This register is a read-only register that can be read in 32-bit units.															
<b>Address:</b>	$FFF8\ C104_H$															
<b>Value after reset:</b>	$0000\ 0000_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CLMA2 ERRS	CLMA1 ERRS	CLMA0 ERRS	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.78 CLMATESTS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	CLMA2ERRS	CLMA2 Error Status 0: Errors are not detected 1: Errors are detected
1	CLMA1ERRS	CLMA1 Error Status 0: Errors are not detected 1: Errors are detected
0	CLMA0ERRS	CLMA0 Error Status 0: Errors are not detected 1: Errors are detected

### 10.7.5.7 CLMAnEMU0 — CLMAn Emulation Register 0

This register is used to set pseudo flags for emulation.

This register can be accessed only during a break by debugger and is reset by a break release.

**Access:** This register can be read or written in 8-bit units.

**Address:** <CLMAn\_base> + 18H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CLMAnSLFST	CLMAnSLSLW
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 10.79 CLMAnEMU0 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CLMAnSLFST	Specifies whether $f_{CLMATMON}$ is assumed to be high. 0: CLMATMON is assumed to be within the normal frequency range. 1: CLMATMON is assumed to exceed the upper threshold.
0	CLMAnSLSLW	Specifies whether $f_{CLMATMON}$ is assumed to be low. 0: CLMATMON is assumed to be within the normal frequency range. 1: CLMATMON is assumed to fall below the lower threshold.

#### CAUTION

It is prohibited to emulate a low and high CLMATMON at the same time. Thus CLMAnEMU0 must not be set to 03H.

### 10.7.6 Usage Notes for CLMAn

不要使用被CLMAn监测不正常的时钟，否则其行为和性能是不能被保证的。

Do not use a clock for which CLMAn detected an abnormality. The behavior and performance are not guaranteed if such a clock is used. When CLMA0 has detected a clock abnormality, modifying any of the clock domain settings is prohibited. **如果CLMA0监测到某个时钟不正常，不允许修改时钟域中的所有东西**

**Table 10.80** provides usage notes for each CLMAn.

**Table 10.80 Usage Notes for CLMAn**

Monitor Clock	CPU Clock after CLMA Reset Release	Note
HS IntOSC (CLMA0)	EMCLK <sup>*1</sup>	<p>Set ROSCE.ROSCDISTRG to 1.<sup>*1</sup></p> <p>Do not set the control registers of the MainOSC and the PLL.</p> <p>After CLMA0RES occurs, modifying any of the clock domain settings is prohibited.</p>
MainOSC (CLMA1)	EMCLK (= HS IntOSC)	<p>Do not set the control registers of the MainOSC and the PLL.</p> <p>After occurrence of the CLMA1RES, do not select a clock whose source clock is the MainOSC or the PLL.</p>
PLL (CLMA2)	EMCLK (= HS IntOSC)	<p>Do not set the control registers of the PLL.</p> <p>After occurrence of the CLMA2RES, do not select a clock whose source clock is the PLL.</p>

Note 1. The state of the EMCLK after reset by the CLMA0RES depends on the state of HS IntOSC oscillation. In the case the HS IntOSC is completely stopped, the LS IntOSC is supplied as the EMCLK. In the case the HS IntOSC continues oscillation but the frequency is outside of the lower threshold and the upper threshold of CLMA0, the HS IntOSC is still supplied as EMCLK. After setting the ROSCE.ROSCDISTRG bit to 1, the EMCLK is switched from the HS IntOSC to the LS IntOSC.

## Section 11 Stand-By Controller (STBC)

This section describes the functions of the stand-by controller (STBC), the registers, and various stand-by modes.

### 11.1 Functions

#### 11.1.1 Types of Stand-By Mode

The RH850/F1L supports HALT mode and the following four stand-by modes:

- HALT mode

HALT mode can be entered from normal RUN mode by performing the CPU instruction “HALT”. This stops the CPU operation, while all clocks other than the CPU clock continue to operate and power continues to be supplied.

For details, see the *RH850G3K User’s Manual: Software*.

- STOP mode

In STOP mode, the clock supply to a certain clock domain can be stopped.

STOP mode is entered when the STBC0STPT.STBC0STPTRG bit is set to 1.

The clock supply to clock domains can continue even in STOP mode by setting

CKSC\_xxx\_STPM.xxxSTPMSK = 1. For details on the CKSC\_xxx\_STPM register, see **Section 10, Clock Controller**.

- DeepSTOP mode

In order to further reduce power consumption, the power supply of the Isolated area can be switched off.

DeepSTOP mode is entered when the STBC0PSC.STBC0DISTRG is set to 1.

- Cyclic RUN mode

In this mode, only the CPU, peripherals on Always-ON area, and RLIN3 can operate.

The CPU executes the instructions in the retention RAM.

In DeepSTOP mode, the mode transitions to Cyclic RUN mode when wake-up factor 2 is generated, and in Cyclic STOP mode, the mode transitions to Cyclic RUN mode when either wake-up factor 1 or wake-up factor 2 is generated.

- Cyclic STOP mode

This mode stops the CPU in addition to the functions stopped in Cyclic RUN mode.

This mode is entered by setting the STBC0STPT.STBC0STPTRG bit to 1 in Cyclic RUN mode.

#### NOTE

The mode in which CPU is running is called RUN mode.

## 11.1.2 Wake-Up Control

### 11.1.2.1 Wake-Up Factors for Stand-By Modes

The stand-by controller can initiate return from stand-by mode by the following wake-up factors.

**Table 11.1 Wake-Up Factor 1**

Wake-Up Factor	Unit	STOP →RUN	DeepSTOP →RUN	Cyclic RUN →RUN <sup>*1</sup>	Cyclic STOP →RUN <sup>*1</sup>
TNMI	Port	✓	✓	✓	✓
WDTA0NMI	WDTA0	✓	✓	✓	✓
INTLVIL <sup>*3</sup>	LVI	✓	✓	✓	✓
INTP0	Port	✓	✓	✓	✓
INTP1	Port	✓	✓	✓	✓
INTP2	Port	✓	✓	✓	✓
INTWDTA0	WDTA0	✓	✓	✓	✓
INTP3	Port	✓	✓	✓	✓
INTP4	Port	✓	✓	✓	✓
INTP5	Port	✓	✓	✓	✓
INTP10	Port	✓	✓	✓	✓
INTP11	Port	✓	✓	✓	✓
WUTRG1	LPS	✓	✓	✓	✓
INTTAUJ0I0	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓	✓	✓
WUTRG0	LPS	✓	✓	✓	✓
INTP6	Port	✓	✓	✓	✓
INTP7	Port	✓	✓	✓	✓
INTP8	Port	✓	✓	✓	✓
INTP12	Port	✓	✓	✓	✓
INTP9	Port	✓	✓	✓	✓
INTP13	Port	✓	✓	✓	✓
INTP14	Port	✓	✓	✓	✓
INTP15	Port	✓	✓	✓	✓
INTRTCA01S	RTCA0	✓	✓	✓	✓
INTRTCA0AL	RTCA0	✓	✓	✓	✓
INTRTCA0R	RTCA0	✓	✓	✓	✓
INTDCUTDI	JTAG	✓	✓	✓	✓
INTKR0	KR0	✓	—	—	—
INTRCANGRECC <sup>*2</sup>	RSCAN0	✓	—	—	—
INTRCAN0REC <sup>*2</sup>	RSCAN0	✓	—	—	—
INTRCAN1REC <sup>*2</sup>	RSCAN0	✓	—	—	—
INTRCAN2REC <sup>*2</sup>	RSCAN0	✓	—	—	—
INTRCAN3REC <sup>*2</sup>	RSCAN0	✓	—	—	—
INTRCAN4REC <sup>*2</sup>	RSCAN0	✓	—	—	—
INTRCAN5REC <sup>*2</sup>	RSCAN0	✓	—	—	—

- Note 1. The mode returns to RUN mode via DeepSTOP mode. When the transition from Cyclic STOP mode to Cyclic RUN mode is made by Wake-Up Factor 1, if the transition to DeepSTOP mode by STBC0PSC.STBC0DISTRG is made without clearing Wake-Up Factor 1, the transition to RUN is made.
- Note 2. By using the INTP external interrupt assigned to the alternate-function pin shared with the CAN reception pin, wake-up from stand-by modes such as DeepSTOP mode is possible. As the trigger for waking up from DeepSTOP mode, use a pin of port P0 or P1, which is assigned to the AWO area.
- Note 3. Cannot be cleared while REGVCC is below the reference voltage (REGVCC (min.) < VLVI). To clear the wake-up factor flag (WUF0[2]), WUFC0[2] must be set while WUFMSK0[2] = 1 and REGVCC is above the reference voltage (REGVCC (min.) > VLVI). The INTLVIH interrupt can be used to check that REGVCC is above the reference voltage.

**Table 11.2 Wake-Up Factor 2**

Wake-Up Factor	Unit	DeepSTOP → Cyclic RUN	Cyclic STOP → Cyclic RUN
INTADCA0I0	ADCA0	—	✓*1
INTADCA0I1	ADCA0	—	✓*1
INTADCA0I2	ADCA0	—	✓*1
INTRLIN30	RLIN30	—	✓
INTTAUJ0I0	TAUJ0	✓	✓
INTTAUJ0I1	TAUJ0	✓	✓
INTTAUJ0I2	TAUJ0	✓	✓
INTTAUJ0I3	TAUJ0	✓	✓
INTRLIN31	RLIN31	—	✓
INTRLIN32	RLIN32	—	✓
INTRTCA01S	RTCA0	✓	✓
INTRTCA0AL	RTCA0	✓	✓
INTRTCA0R	RTCA0	✓	✓
INTRLIN33	RLIN33	—	✓
INTRLIN34	RLIN34	—	✓
INTRLIN35	RLIN35	—	✓

Note 1. These wake-up factors are only available in LPS analog input mode.

#### CAUTION

For the pins of the function used for the wake-up factors from DeepSTOP mode, use the multiplexed functions of the ports assigned to the AWO area.

### 11.1.2.2 Setting of Wake-Up Factors

Wake-up factors for returning from stand-by modes are controlled by the following stand-by controller registers:

- Wake-up factor registers: WUF0, WUF20, WUF\_ISO0

Upon occurrence of an effective wake-up factor, the associated wake-up factor flag is set to 1. By checking these registers and their flags, it is possible to identify the wake-up factor.

- Wake-up factor mask registers: WUFMSK0, WUFMSK20, WUFMSK\_ISO0

Each bit of these registers is assigned to a certain wake-up factor. Wake-up by this factor is enabled if its mask bit is set to 0. Wake-up factors assigned to both wake-up factor 1 and 2 should not be enabled at the same time.

- Wake-up factor clear registers: WUFC0, WUFC20, WUFC\_ISO0

By setting the applicable bits in these registers to 1, the Wake-up factor bit (WUFy) in the Wake-up factor registers (WUF0, WUF20, WUF\_ISO0) can be cleared.

#### NOTE

The Wake-up factor flags in the Wake-up factor registers (WUF0, WUF20, and WUF\_ISO0) only indicate the occurrence of wake-up factor. These flags do not indicate a transition from stand-by mode to normal mode.

The assignment of the wake-up factors to the control register bits and status register bits are shown in the following tables.

For details about the wake-up control and status registers, see **Section 11.2.2.3, WUF0/WUF20/WUF\_ISO0 — Wake-Up Factor Registers**, **Section 11.2.2.4, WUFMSK0/WUFMSK20/WUFMSK\_ISO0 — Wake-Up Factor Mask Registers**, and **Section 11.2.2.5, WUFC0/WUFC20/WUFC\_ISO0 — Wake-Up Factor Clear Registers**.

Table 11.3 Wake-Up Factor 1 Register Assignment

Wake-Up Factor	Register Bit Assignment	Unit	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
TNMI	WUFO[0]	WUFMSK0[0]	WUFC0[0]	Port	✓	✓	✓	✓
WDTAONMI	WUFO[1]	WUFMSK0[1]	WUFC0[1]	WDTA0	✓	✓	✓	✓
INTLVIL	WUFO[2]	WUFMSK0[2]	WUFC0[2]	LVI	✓	✓	✓	✓
INTP0	WUFO[5]	WUFMSK0[5]	WUFC0[5]	Port	✓	✓	✓	✓
INTP1	WUFO[6]	WUFMSK0[6]	WUFC0[6]	Port	✓	✓	✓	✓
INTP2	WUFO[7]	WUFMSK0[7]	WUFC0[7]	Port	✓	✓	✓	✓
INTWDTA0	WUFO[8]	WUFMSK0[8]	WUFC0[8]	WDTA0	✓	✓	✓	✓
INTP3	WUFO[9]	WUFMSK0[9]	WUFC0[9]	Port	✓	✓	✓	✓
INTP4	WUFO[10]	WUFMSK0[10]	WUFC0[10]	Port	✓	✓	✓	✓
INTP5	WUFO[11]	WUFMSK0[11]	WUFC0[11]	Port	✓	✓	✓	✓
INTP10	WUFO[12]	WUFMSK0[12]	WUFC0[12]	Port	✓	✓	✓	✓
INTP11	WUFO[13]	WUFMSK0[13]	WUFC0[13]	Port	✓	✓	✓	✓
WUTRG1	WUFO[14]	WUFMSK0[14]	WUFC0[14]	LPS	✓	✓	✓	✓
INTTAUJ0I0	WUFO[15]	WUFMSK0[15]	WUFC0[15]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I1	WUFO[16]	WUFMSK0[16]	WUFC0[16]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I2	WUFO[17]	WUFMSK0[17]	WUFC0[17]	TAUJ0	✓	✓	✓	✓
INTTAUJ0I3	WUFO[18]	WUFMSK0[18]	WUFC0[18]	TAUJ0	✓	✓	✓	✓
WUTRG0	WUFO[19]	WUFMSK0[19]	WUFC0[19]	LPS	✓	✓	✓	✓
INTP6	WUFO[20]	WUFMSK0[20]	WUFC0[20]	Port	—	—	✓	✓
INTP7	WUFO[21]	WUFMSK0[21]	WUFC0[21]	Port	—	—	✓	✓
INTP8	WUFO[22]	WUFMSK0[22]	WUFC0[22]	Port	—	—	✓	✓
INTP12	WUFO[23]	WUFMSK0[23]	WUFC0[23]	Port	—	—	✓	✓
INTP9	WUFO[24]	WUFMSK0[24]	WUFC0[24]	Port	—	—	—	✓
INTP13	WUFO[25]	WUFMSK0[25]	WUFC0[25]	Port	—	—	—	✓
INTP14	WUFO[26]	WUFMSK0[26]	WUFC0[26]	Port	—	—	—	✓
INTP15	WUFO[27]	WUFMSK0[27]	WUFC0[27]	Port	—	—	—	✓
INTRTCA01S	WUFO[28]	WUFMSK0[28]	WUFC0[28]	RTCA0	—	—	—	✓
INTRTCA0AL	WUFO[29]	WUFMSK0[29]	WUFC0[29]	RTCA0	—	—	—	✓
INTRTCA0R	WUFO[30]	WUFMSK0[30]	WUFC0[30]	RTCA0	—	—	—	✓
INTDCUTDI	WUFO[31]	WUFMSK0[31]	WUFC0[31]	JTAG	✓	✓	✓	✓
INTKRO	WUF_ISO0[1]	WUFMSK_ISO0[1]	WUFC_ISO0[1]	KR0	✓	✓	✓	✓
INTRCANGRECC	WUF_ISO0[2]	WUFMSK_ISO0[2]	WUFC_ISO0[2]	RSCAN0	✓	✓	✓	✓
INTRCAN0REC	WUF_ISO0[3]	WUFMSK_ISO0[3]	WUFC_ISO0[3]	RSCAN0	✓	✓	✓	✓
INTRCAN1REC	WUF_ISO0[4]	WUFMSK_ISO0[4]	WUFC_ISO0[4]	RSCAN0	—	✓	✓	✓
INTRCAN2REC	WUF_ISO0[5]	WUFMSK_ISO0[5]	WUFC_ISO0[5]	RSCAN0	—	✓	✓	✓
INTRCAN3REC	WUF_ISO0[6]	WUFMSK_ISO0[6]	WUFC_ISO0[6]	RSCAN0	—	—	—	*1 ✓
INTRCAN4REC	WUF_ISO0[7]	WUFMSK_ISO0[7]	WUFC_ISO0[7]	RSCAN0	—	—	—	*1 *2 ✓
INTRCAN5REC	WUF_ISO0[8]	WUFMSK_ISO0[8]	WUFC_ISO0[8]	RSCAN0	—	—	—	*1 *2 ✓

Note 1. Supported only for F1L for Gateway

Note 2. Supported only for Code Flash 1.5 M and 2 M products

Table 11.4 Wake-Up Factor 2 Register Assignment

Wake-Up Factor	Register Bit Assignment			Unit	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
INTADCA0I0	WUF20[0]	WUFMSK20[0]	WUFC20[0]	ADCA0	✓	✓	✓	✓	✓	✓
INTADCA0I1	WUF20[1]	WUFMSK20[1]	WUFC20[1]	ADCA0	✓	✓	✓	✓	✓	✓
INTADCA0I2	WUF20[2]	WUFMSK20[2]	WUFC20[2]	ADCA0	✓	✓	✓	✓	✓	✓
INTRLIN30	WUF20[3]	WUFMSK20[3]	WUFC20[3]	RLIN30	✓	✓	✓	✓	✓	✓
INTTAUJ0I0	WUF20[4]	WUFMSK20[4]	WUFC20[4]	TAUJ0	✓	✓	✓	✓	✓	✓
INTTAUJ0I1	WUF20[5]	WUFMSK20[5]	WUFC20[5]	TAUJ0	✓	✓	✓	✓	✓	✓
INTTAUJ0I2	WUF20[6]	WUFMSK20[6]	WUFC20[6]	TAUJ0	✓	✓	✓	✓	✓	✓
INTTAUJ0I3	WUF20[7]	WUFMSK20[7]	WUFC20[7]	TAUJ0	✓	✓	✓	✓	✓	✓
INTRLIN31	WUF20[8]	WUFMSK20[8]	WUFC20[8]	RLIN31	—	✓	✓	✓	✓	✓
INTRLIN32	WUF20[9]	WUFMSK20[9]	WUFC20[9]	RLIN32	—	—	✓	✓	✓	✓
INTRTCA01S	WUF20[10]	WUFMSK20[10]	WUFC20[10]	RTCA0	—	—	—	—	✓	✓
INTRTCA0AL	WUF20[11]	WUFMSK20[11]	WUFC20[11]	RTCA0	—	—	—	—	✓	✓
INTRTCA0R	WUF20[12]	WUFMSK20[12]	WUFC20[12]	RTCA0	—	—	—	—	✓	✓
INTRLIN33	WUF20[13]	WUFMSK20[13]	WUFC20[13]	RLIN33	—	—	—	✓	✓	✓
INTRLIN34	WUF20[14]	WUFMSK20[14]	WUFC20[14]	RLIN34	—	—	—	—	✓	✓
INTRLIN35	WUF20[15]	WUFMSK20[15]	WUFC20[15]	RLIN35	—	—	—	—	✓	✓

### 11.1.3 On-Chip Debug Wake-Up

The On-Chip Debug unit (OCD) generates a wake-up event while the microcontroller runs the application program in the following cases:

- The debugger issues a stop request
- A breakpoint is hit

In either case all stand-by modes are terminated, provided the OCD debug event is enabled as a wake-up factor via the WUFMSK0 register.

#### CAUTION

If the OCD wake-up event is disabled, it is not possible to wake up from stand-by modes via an On-chip debugger request.

The OCD wake-up event can be enabled as a wake-up factor for all stand-by modes by setting WUFMSK0[31] = 0.

When the hot plug-in function is used, make sure to enable the OCD wake-up event and return from stand-by mode by INTDCUTDI interrupt.

## 11.1.4 I/O Buffer Control

This section describes the behavior of the I/O buffers during various stand-by modes.

The port groups in the Isolated area support the I/O buffer hold state.

The port groups in the Always-ON area remain their state before entering stand-by mode.

For details on the port group assignment to the Isolated area and to the Always-ON area, see **Section 38, Power Supply and Power Domains**.

### 11.1.4.1 I/O Buffer Hold State

During the I/O buffer hold state, the I/O buffers maintains the state it was in before entering this state.

Therefore, no external or internal signal can change the state of the I/O buffer until the I/O buffer hold state is terminated.

### 11.1.4.2 I/O Buffers during STOP Mode

In STOP mode, the I/O buffers remain their state before entering STOP mode (I/O buffer hold state is not entered).

### 11.1.4.3 I/O Buffers during DeepSTOP Mode

In DeepSTOP mode, the I/O buffers of port groups in the Isolated area transition to I/O buffer hold state.

After wake-up from DeepSTOP mode, the I/O buffers remain in I/O buffer hold state until cancel the state by software. To cancel I/O buffer hold state, follow the steps shown below.

1. Re-configure the peripheral or port function.
2. Set IOHOLD.IOHOLD = 0.

The following table is a summary of the I/O buffer in Isolated area during standby mode and after wakeup.

**Table 11.5    Buffer Operation during Standby Mode and after WakeUp (I/O Buffers in the Isolated Area)**

	Before Standby	During Standby	After Wakeup
STOP mode	Normal operation		
DeepSTOP mode	Normal operation	I/O buffer hold state	I/O buffer hold state * <sup>1</sup>

Note 1. Set the IOHOLD.IOHOLD bit to "0" to release the I/O buffer hold state.

The port groups in the Always-ON area don't support I/O buffer hold state. They continue operation and remain its state before entering DeepSTOP mode. In the case an alternative function of modules in Isolated area is assigned to the pin in the Always-ON area, the state of the I/O buffer may change in the transition to the DeepSTOP mode due to initialization of the modules in the Isolated area by ISORES. To avoid this behavior, it is recommended to change to function of modules in Always-ON area (e. g. Port mode) before entering DeepSTOP mode.

### 11.1.5 Transition to Stand-By Mode

The figure below shows transition between RUN mode and stand-by mode.

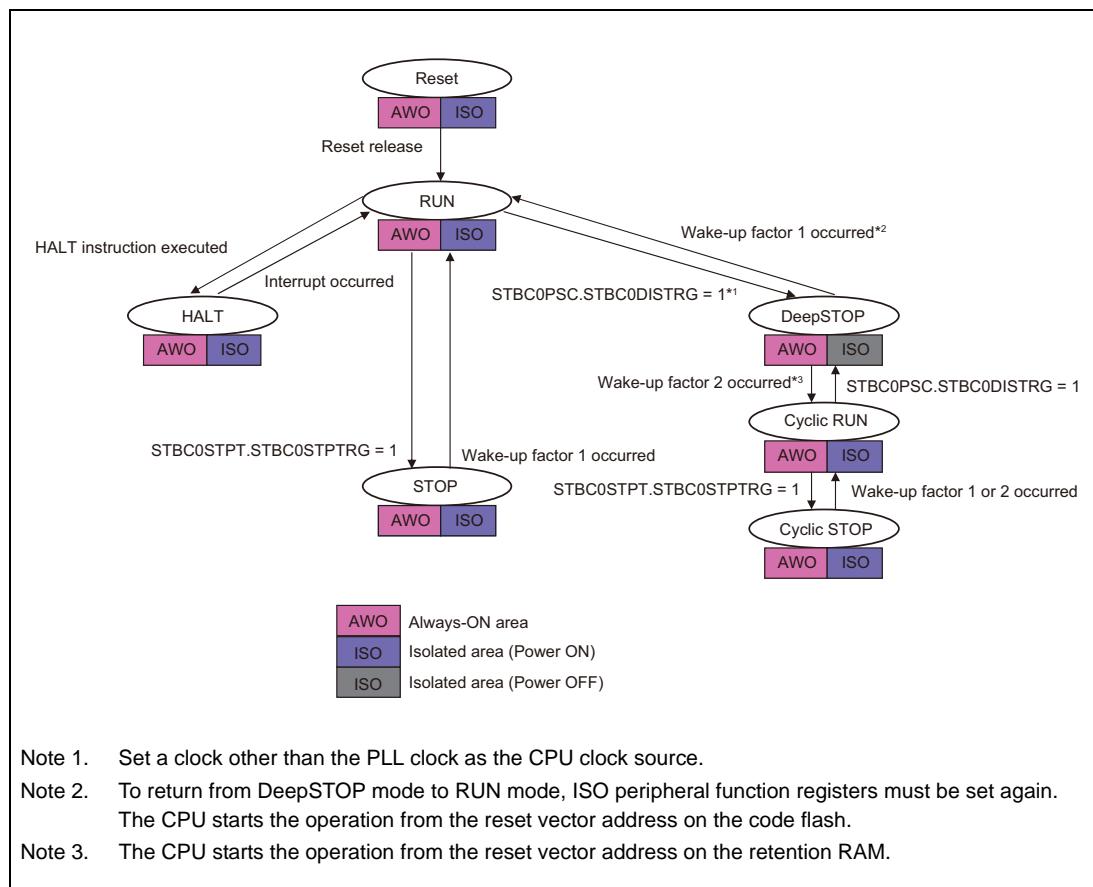


Figure 11.1 Transition to Stand-By Mode

### 11.1.6 Clock Supply

The clock supply to the standby controller is shown in the following table.

Table 11.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Standby controller	Register access	EMCLK

## 11.2 Registers

### 11.2.1 List of Registers

The following table lists the stand-by controller registers.

**Table 11.7 List of Registers**

Module Name	Register Name	Symbol	Address
STBC	Power save control register	STBC0PSC	FFF8 0100 <sub>H</sub>
	Power stop trigger register	STBC0STPT	FFF8 0110 <sub>H</sub>
	Wake-up factor registers	WUF0	FFF8 0400 <sub>H</sub>
		WUF20	FFF8 0520 <sub>H</sub>
		WUF_ISO0	FFF8 8110 <sub>H</sub>
	Wake-up factor mask registers	WUFMSK0	FFF8 0404 <sub>H</sub>
		WUFMSK20	FFF8 0524 <sub>H</sub>
		WUFMSK_ISO0	FFF8 8114 <sub>H</sub>
	Wake-up factor clear registers	WUFC0	FFF8 0408 <sub>H</sub>
		WUFC20	FFF8 0528 <sub>H</sub>
		WUFC_ISO0	FFF8 8118 <sub>H</sub>
	I/O buffer hold control register	IOHOLD	FFF8 0B00 <sub>H</sub>

## 11.2.2 Details of Stand-By Controller Control Registers

### 11.2.2.1 STBC0PSC — Power Save Control Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.

For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

**Access:** This register can be read or written in 32-bit units.

**Address:** FFF8 0100<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0 DISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 11.8 STBC0PSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	STBC0DISTRG	0: No effect 1: Transition to DeepSTOP mode
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

### 11.2.2.2 STBC0STPT — Power Stop Trigger Register

The correct write sequence using the PROTCMD0 register is required in order to update this register.  
For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

	<b>Access:</b>	This register can be read or written in 32-bit units.															
	<b>Address:</b>	FFF8 0110 <sub>H</sub>															
	<b>Value after reset:</b>	0000 0000 <sub>H</sub>															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	STBC0 STPTR G
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.9 STBC0STPT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STBC0STPTRG	0: No effect 1: Transition to STOP mode <ul style="list-style-type: none"> <li>– In RUN mode: Transition to STOP mode</li> <li>– In Cyclic RUN mode: Transition to Cyclic STOP mode</li> </ul>

### 11.2.2.3 WUF0/WUF20/WUF\_ISO0 — Wake-Up Factor Registers

These registers indicate the generation of wake-up factors.

WUF0 and WUF20 are initialized by all reset sources except the transition to DeepSTOP mode (AWORES). WUF\_ISO0 is initialized by all reset sources (ISORES).

**Access:** These registers are read-only registers that can be read in 32-bit units.

**Address:** WUF0: FFF8 0400<sub>H</sub>  
WUF20: FFF8 0520<sub>H</sub>  
WUF\_ISO0: FFF8 8110<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUF31	WUF30	WUF29	WUF28	WUF27	WUF26	WUF25	WUF24	WUF23	WUF22	WUF21	WUF20	WUF19	WUF18	WUF17	WUF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF09	WUF08	WUF07	WUF06	WUF05	WUF04	WUF03	WUF02	WUF01	WUF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.10 WUF0/WUF20/WUF\_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFy	Indicates the generation of a wake-up factor. 0: Wake-up factor is not generated 1: Wake-up factor is generated

#### NOTE

While the WUFMSKY bit in the wake-up factor mask register is 1, WUFy is not set to 1 at the generation of a wake-up factor.

#### Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 11.3, Wake-Up Factor 1 Register Assignment** and **Table 11.4, Wake-Up Factor 2 Register Assignment**.

The bit to which a wake-up factor is not assigned is read as the value “0”.

### 11.2.2.4 WUFMSK0/WUFMSK20/WUFMSK\_ISO0 — Wake-Up Factor Mask Registers

These registers enable wake-up factors.

WUFMSK0 and WUFMSK20 are initialized by all reset sources except the transition to DeepSTOP mode (AWORES). WUFMSK\_ISO0 is initialized by all reset sources (ISORES).

**Access:** These registers can be read or written in 32-bit units.

**Address:** WUFMSK0: FFF8 0404<sub>H</sub>  
WUFMSK20: FFF8 0524<sub>H</sub>  
WUFMSK\_ISO0: FFF8 8114<sub>H</sub>

**Value after reset:** WUFMSK0: FFFF FFFF<sub>H</sub>  
WUFMSK20: FFFF FFFF<sub>H</sub>  
WUFMSK\_ISO0: FFFF FFFF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFM SK31	WUFM SK30	WUFM SK29	WUFM SK28	WUFM SK27	WUFM SK26	WUFM SK25	WUFM SK24	WUFM SK23	WUFM SK22	WUFM SK21	WUFM SK20	WUFM SK19	WUFM SK18	WUFM SK17	WUFM SK16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFM SK15	WUFM SK14	WUFM SK13	WUFM SK12	WUFM SK11	WUFM SK10	WUFM SK09	WUFM SK08	WUFM SK07	WUFM SK06	WUFM SK05	WUFM SK04	WUFM SK03	WUFM SK02	WUFM SK01	WUFM SK00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.11 WUFMSK0/WUFMSK20/WUFMSK\_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFMSK <sub>y</sub>	Enables/disables a wake-up factor. 0: Wake-up factor is enabled 1: Wake-up factor is disabled

#### NOTE

While the WUFMSK<sub>y</sub> bit is 1, WUFY of the wake-up factor register is not set to 1 at the generation of a wake-up factor.

#### Wake-Up factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 11.3, Wake-Up Factor 1 Register Assignment** and **Table 11.4, Wake-Up Factor 2 Register Assignment**.

When writing to these registers, write the value “1” to the bits to which wake-up factors are not assigned.

### 11.2.2.5 WUFC0/WUFC20/WUFC\_ISO0 — Wake-Up Factor Clear Registers

These registers clear the WUFY bits in the wake-up factor registers.

**Access:** These registers are write-only registers that can be written in 32-bit units.

**Address:** WUFC0: FFF8 0408<sub>H</sub>  
WUFC20: FFF8 0528<sub>H</sub>  
WUFC\_ISO0: FFF8 8118<sub>H</sub>

**Value after reset:** Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUFC 31	WUFC 30	WUFC 29	WUFC 28	WUFC 27	WUFC 26	WUFC 25	WUFC 24	WUFC 23	WUFC 22	WUFC 21	WUFC 20	WUFC 19	WUFC 18	WUFC 17	WUFC 16	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUFC 15	WUFC 14	WUFC 13	WUFC 12	WUFC 11	WUFC 10	WUFC 09	WUFC 08	WUFC 07	WUFC 06	WUFC 05	WUFC 04	WUFC 03	WUFC 02	WUFC 01	WUFC 00	
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 11.12 WUFC0/WUFC20/WUFC\_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFCy	Clears the wake-up factor bit WUFY in the wake-up factor registers. 0: WUFY is not modified 1: WUFY is cleared

#### Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 11.3, Wake-Up Factor 1 Register Assignment** and **Table 11.4, Wake-Up Factor 2 Register Assignment**.

When writing to these registers, write the value “0” to the bits to which wake-up factors are not assigned.

### 11.2.2.6 IOHOLD — I/O Buffer Hold Control Register

This register specifies the hold state of the I/O buffer in DeepSTOP mode. The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

	Access: This register can be read or written in 32-bit units.															
	Address: FFF8 0B00 <sub>H</sub>															
	Value after reset: 0000 0000 <sub>H</sub>															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOHOLD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.13 IOHOLD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IOHOLD	0: I/O hold state is released 1: I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is prohibited. To release the I/O hold state after the wake-up, set this bit to 0 by software.

## 11.3 Mode Transition

This section explains the mode transition procedures.

### 11.3.1 STOP Mode

In STOP mode, most of the clock supplies to the Always-ON area and the Isolated area are stopped. The clock stop mask registers control clock supply to related clock domains in stand-by mode. Stop all of the peripheral functions before transition to STOP mode if the clock supply to the function will be stopped in STOP mode.

The transition procedure (example) to STOP mode is shown below.

#### Preparation for STOP mode

- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag ( $IC_{xxx}.RF_{xxx} = 0$ ).
  - Mask the interrupts for non-wake-up factors ( $IC_{xxx}.MK_{xxx} = 1$ ).
  - Release the masks of the interrupts for wake-up factors ( $IC_{xxx}.MK_{xxx} = 0$ ).
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUF0/WUFC\_ISO0 registers).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK\_ISO0 registers).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK\_ISO0 registers).
- Set the clock stop mask register to select the clock domains to be stopped and the ones to continue operating (using the CKSC\_xxx\_STPM.xxxxxSTPMSK bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the ROSCSTPMSK bit in the ROSCSTPM register).

#### Start of STOP mode

Set the STBC0STPTRG bit in the STBC0STPT register to 1 to transition to STOP mode.

#### End of STOP mode

When a wake-up factor is generated, the microcontroller returns from STOP mode.

#### Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0, WUF\_ISO0).

When an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.

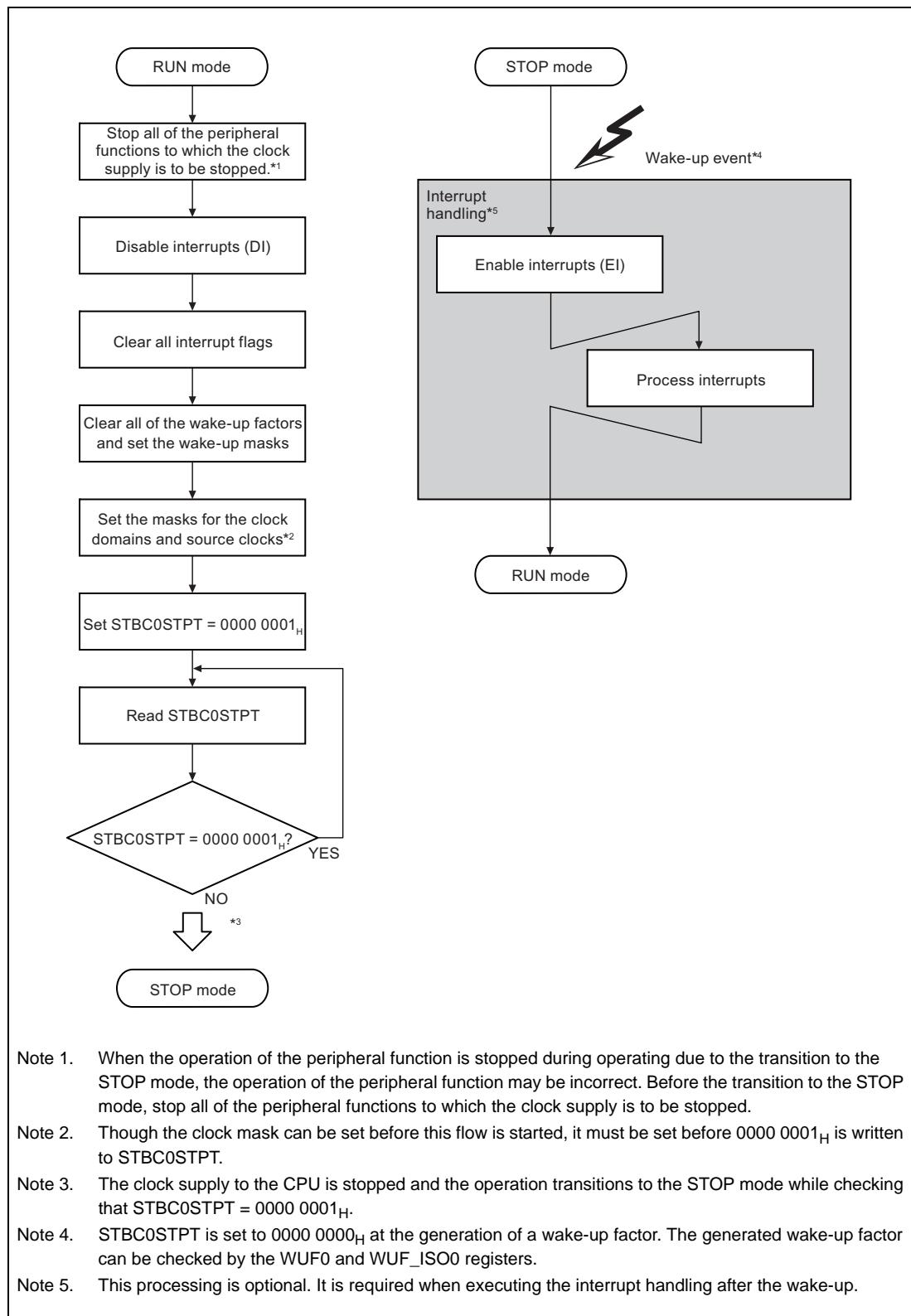


Figure 11.2 Example of STOP Mode Transition

### 11.3.2 DeepSTOP Mode

In DeepSTOP mode, the clock supply to all areas and the power supply to the Isolated area are stopped. However, clock supply to the peripheral functions in the AWO (Always-ON) area can be continued by setting the clock stop mask register.

Select the clock other than the PLL as the CPU operating clock, before the transition to DeepSTOP mode.

The transition procedure (example) to DeepSTOP mode is shown below.

#### Preparation for DeepSTOP mode

- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupts for non-wake-up factors (ICxxx.MKxxx = 1).
  - Release the masks of the interrupts for wake-up factors (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUF0 and WUFC20 registers).
  - Mask the non-wake-up factor (the WUFMSK0 and WUFMSK20 registers).
  - Release the masks of the wake-up factors (the WUFMSK0 and WUFMSK20 registers).

#### CAUTION

**When a wake-up factor is assigned to both wake-up factor 1 registers and wake-up factor 2 registers, it can be used only in one of them.**

- Set the clock stop mask register to select the clock domains to be stopped and the ones to continue operating (using the CKSC\_xxx\_STPM.xxxxSTPMSK bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the ROSCSTPMSK bit in the ROSCSTPM register).

#### Start of DeepSTOP mode

Set the STBC0DISTRG bit in the STBC0PSC register to 1 to transition to DeepSTOP mode.

#### End of DeepSTOP mode

When a wake-up factor is generated, the microcontroller returns from DeepSTOP mode.

### Wake-up handling

- When returned from DeepSTOP mode due to wake-up factor 1, the microcontroller starts the operation from the reset vector address.

If one of the following interrupts has been generated before recovery from DeepSTOP mode to RUN mode, the microcontroller restart operation from the exception handler address:

- FENMI: FENMI handler address ( $E0_H$ )
- FEINT: FEINT handler address ( $F0_H$ )

Note that the General-purpose registers, Primary Local RAM and Secondary Local RAM are undefined value after return from DeepSTOP mode.

- The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0).
- The ports in the Isolated area maintain the I/O buffer hold state.

Release the I/O buffer hold state by executing the following steps:

1. Re-configure the peripheral functions and port functions.
  2. Set IOHOLD.IOHOLD = 0.
- To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. Then, when an interrupt is enabled by the CPU instruction ‘EI’, the generated wake-up interrupt will be executed.

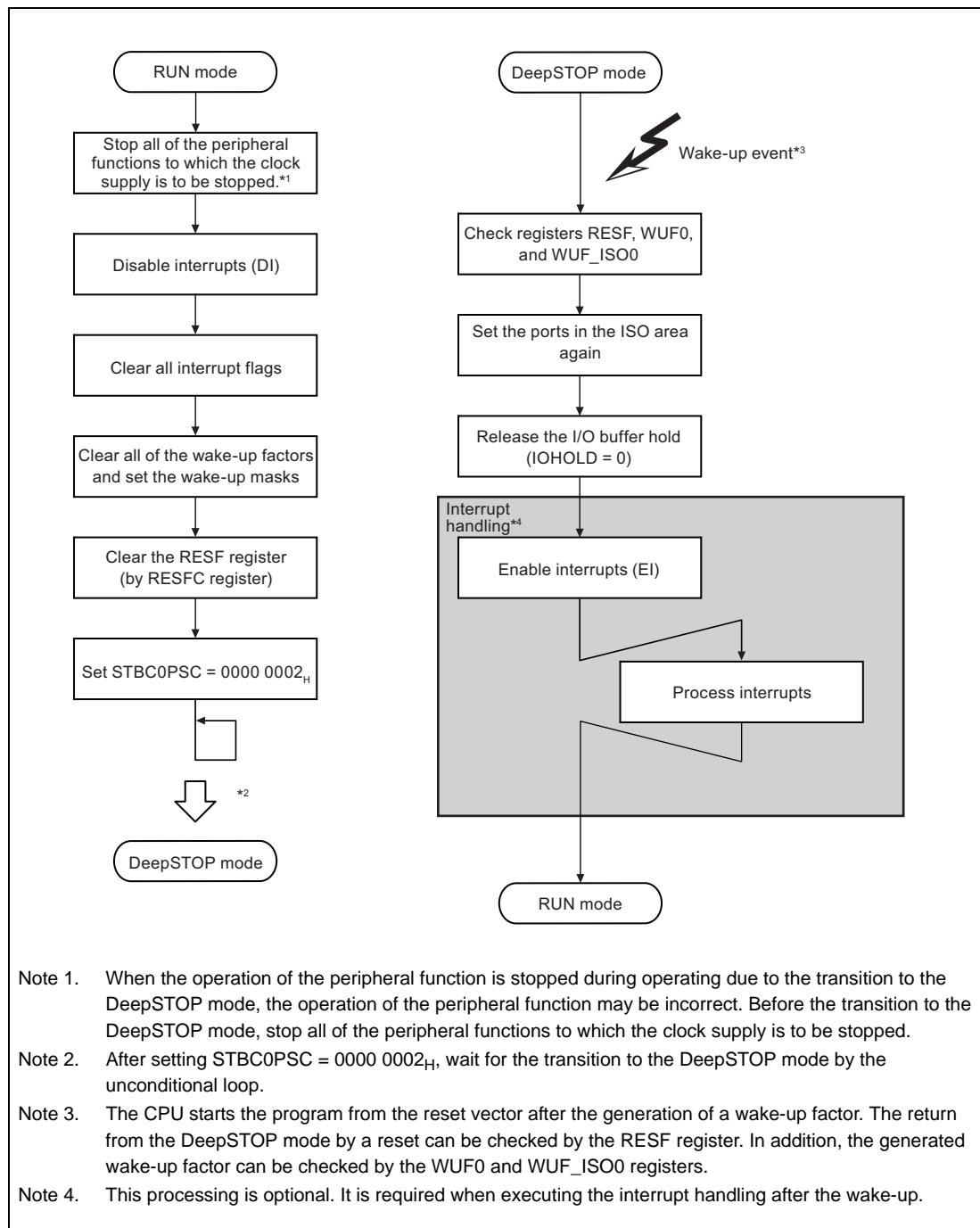


Figure 11.3 Example of DeepSTOP Mode Transition

### 11.3.3 Cyclic RUN Mode

In Cyclic RUN mode, the functions except the CPU, AWO area peripheral functions, and RLIN3 are stopped. In this mode, PLL and Flash Memory are not available. The transition procedure (example) to Cyclic RUN mode is shown below.

#### Preparation of Cyclic RUN mode

Allocate the program for Cyclic RUN mode to the Retention RAM. The reset vector base address in Cyclic RUN mode is set to the first address of the Retention RAM (FEE0 0000<sub>H</sub>).

The instruction to transition to DeepSTOP mode should be arranged in the interrupt exception handler or a polling routine of interrupt request which is used as the source of returning to the RUN mode.

For details on the exception vector, see the *RH850G3K User's Manual: Software*.

#### CAUTION

**Do not change the PSW.EBV bit from its value after reset in Cyclic RUN mode (Do not set the PSW.EBV bit to 1 in Cyclic RUN mode).**

- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK20 register).
  - Release the masks of the wake-up factors (the WUFMSK20 register).
- Transition to DeepSTOP mode. For details on the transition to DeepSTOP mode, see **Section 11.3.2, DeepSTOP Mode**.

#### Start of Cyclic RUN mode

The operation transitions to Cyclic RUN mode from DeepSTOP mode at the generation of wake-up factor 2. The microcontroller starts operation from the reset vector address of Cyclic RUN mode (the first address of the Retention RAM (FEE0 0000<sub>H</sub>)).

If one of the following interrupts has been generated during recovery from DeepSTOP mode to Cyclic RUN mode, the microcontroller restart operation from the exception handler address:

- FENMI: FENMI handler address in Cyclic RUN mode (FEE00000<sub>H</sub> + E0<sub>H</sub>)
- FEINT: FEINT handler address in Cyclic RUN mode (FEE00000<sub>H</sub> + F0<sub>H</sub>)

Note that the General-purpose registers, Primary Local RAM and Secondary Local RAM are undefined value after the transition to Cyclic RUN mode from DeepSTOP mode.

The operation transitions to Cyclic RUN mode from Cyclic STOP mode at the generation of wake-up factors 1 and 2.

#### End of Cyclic RUN mode

The Cyclic RUN mode ends at the transition to the Cyclic STOP mode by setting the STBC0STPT.STBC0STPTRG bit to 1, or at the transition to the DeepSTOP mode by setting the STBC0PSC.STBC0DISTRG bit to 1.

#### Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF20).

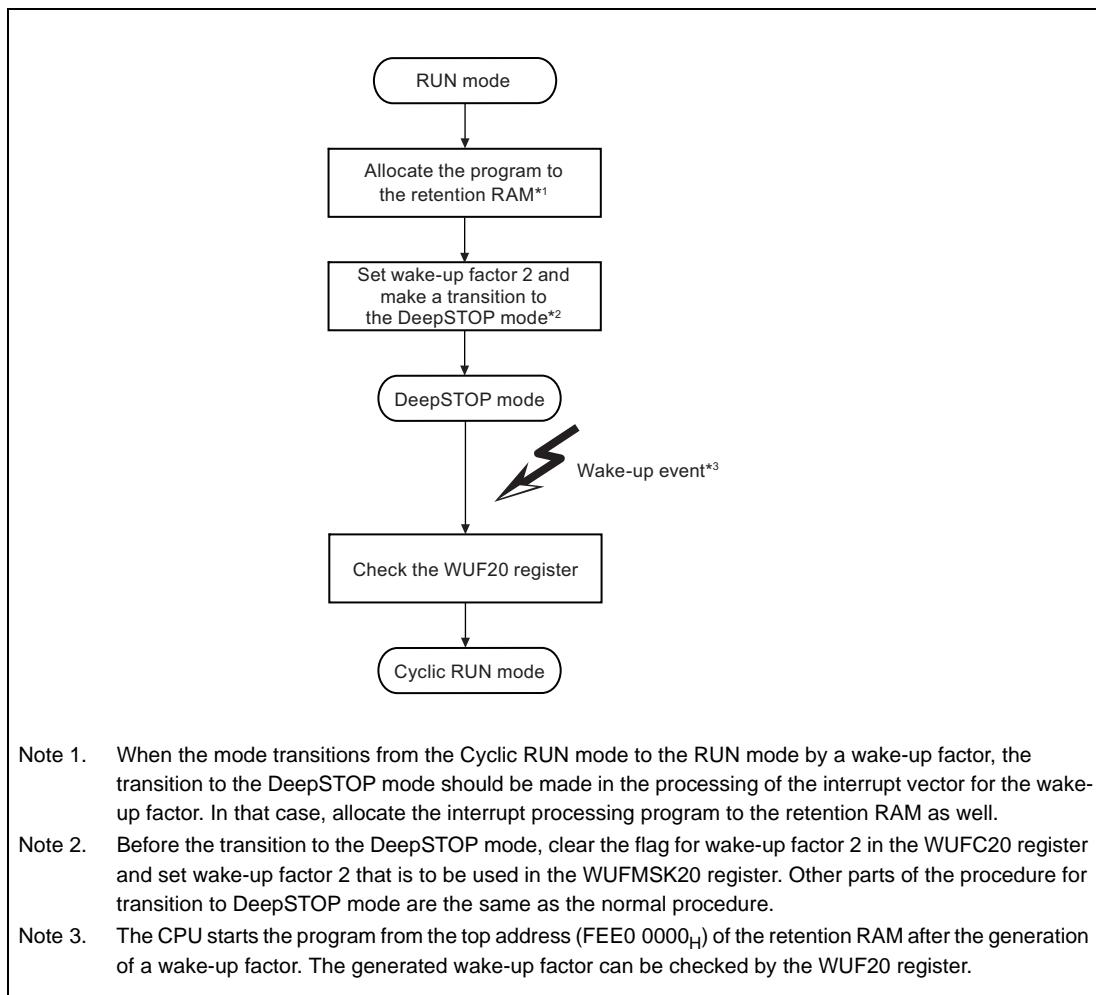


Figure 11.4 Example of Cyclic RUN Mode Transition

### 11.3.4 Cyclic STOP Mode

In Cyclic STOP mode, the functions except the AWO area peripheral function and RLIN3 are stopped.

The transition procedure (example) to Cyclic STOP mode is shown below.

#### Preparation for Cyclic STOP mode

- Transition to Cyclic RUN mode.
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUF0/WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK20 register).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK20 register).

#### Start of Cyclic STOP mode

Set the STBC0STPT.STBC0STPTR bit to 1 to transition to Cyclic STOP mode.

#### End of Cyclic STOP mode

The operation transitions to Cyclic RUN mode at the generation of wake-up factor 1 or 2.

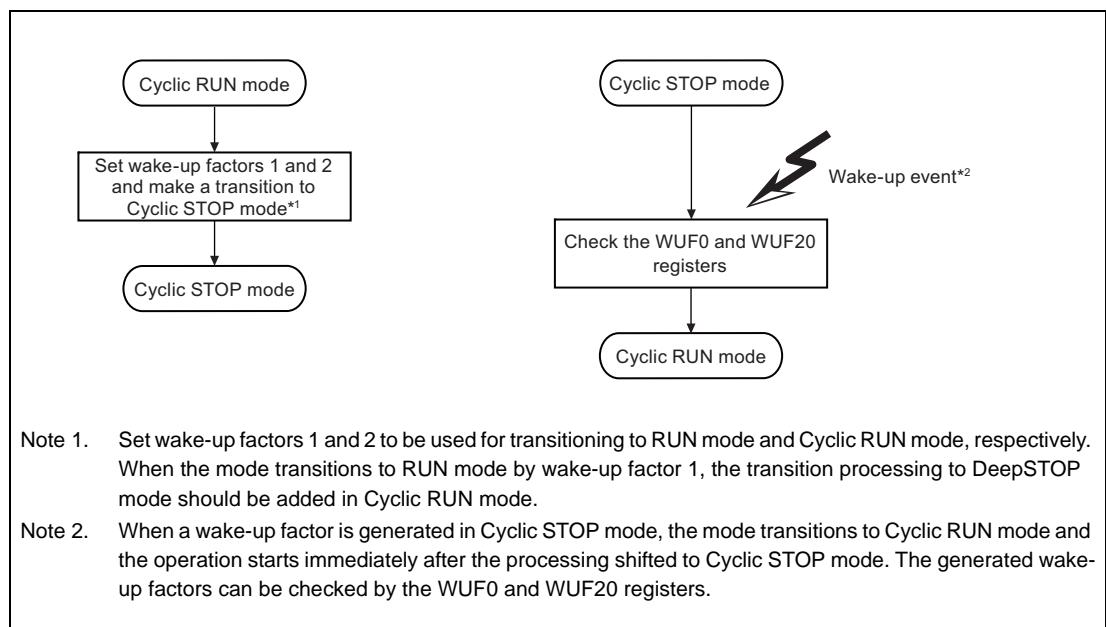


Figure 11.5 Example of Cyclic STOP Mode Transition

## 11.4 Writing to the Stand-By Controller Related Registers

The following stand-by controller registers are write-protected registers.

- STBC0PSC register
- STBC0STPT register
- IOHOLD register

The write-protected registers are protected against the illegal writing due to an incorrect program operation.

For details on the write-protected sequence, see **Section 4, Write-Protected Registers**.

## 11.5 Cautions when Using Stand-By Modes

### 11.5.1 Cautions Concerning Transitioning to DeepSTOP Mode When Using a Debugger

When using a debugger, executing a program that causes the mode to transition to DeepSTOP mode immediately after the program is started may cause improper communication between the OCD emulator and microcontroller because the microcontroller will enter DeepSTOP mode before the preparations for communication between the OCD emulator and microcontroller are completed.

The communication preparation period depends on the OCD emulator's host PC environment and the operating frequency of the microcontroller, so when performing debugging that causes the program to enter DeepSTOP mode immediately after the program starts, insert a wait between reset release and the DeepSTOP execution instruction so that the debugger starts normally.

In DeepSTOP mode, the debugging controller stops. For return from DeepSTOP mode by the debugger, see **Section 11.1.3, On-Chip Debug Wake-Up**.

## Section 12 Low-Power Sampler (LPS)

This section contains a generic description of the low-power sampler (LPS).

The first part in this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the LPS.

### 12.1 Features of RH850/F1L LPS

#### 12.1.1 Number of Units

This microcontroller has the following number of LPS units.

**Table 12.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	1	1
Name	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)			

**Table 12.2 Unit Configurations and Channels**

Unit Name LPSn	Channels per Unit	Function	Channel Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
LPS0	1	Digital port input m for port polling	DPINm	3 ch	8 ch	12 ch	17 ch	24 ch	24 ch
		Analog input m for A/D converter	ADCA0lm	8 ch	10 ch	11 ch	16 ch	16 ch	16 ch

**Table 12.3 Indices**

Index	Description
n	Throughout this section, the individual LPS units are identified by the index "n" (n = 0).
m	Throughout this section, the number of digital port input channels for LPS port polling is indicated by the index "m" (m = 0 to 23) and the number of analog input channels for A/D converter is indicated by the index "m" (m = 0 to 15)
k	The external multiplexer select output signal for digital port is indicated by the index "k".
x	LPS sequence start trigger input signal is indicated by the index "x".

#### NOTE

Descriptions of functions and registers in this section are based on the maximum configurations. Adjust the indices in the text to the proper value for each product. When writing a value to a register that will result in writing to bits outside the range of the index for the product you are using, write the value after reset to these bits.

The following table shows the values indicated by the indices of each product.

**Table 12.4 Indices of Products**

Indices of Each Product	
48 pins	64 pins, 80 pins, 100 pins, 144 pins, 176 pins
k = —	k = 0 to 2
x = 0 to 3	x = 0 to 3

### 12.1.2 Register Base Address

The LPS base address is shown in the following table.

LPS register addresses are given as offsets from the base address.

**Table 12.5 Register Base Address**

Base Address Name	Base Address
<LPS0_base>	FFBC 2000 <sub>H</sub>

### 12.1.3 Clock Supply

The LPS clock supply is shown in the following table.

If the operation request signal for the low-power sampler (LPS) is at the active level, the clock for clock domains for which the HS IntOSC is selected also operates.

To stop the function of a clock domain, set the target clock domain to “disabled” before making a transition to standby mode.

**Table 12.6 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
LPSn	Register access	EMCLK

### 12.1.4 Interrupt Request

The LPS interrupt requests are listed in the following table.

**Table 12.7 Interrupt Requests**

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
<b>LPS0</b>			
INTCWEND	Port polling end interrupt (LPS)	104	—
INTADCA0I0 <sup>*1</sup>	ADCA0 SG1 end interrupt	10	4 (channels 0 to 7)
INTADCA0I1 <sup>*1</sup>	ADCA0 SG2 end interrupt	11	5 (channels 0 to 7)
INTADCA0I2 <sup>*1</sup>	ADCA0 SG3 end interrupt	12	6 (channels 0 to 7)

Note 1. These signals are output from ADCA0.

### 12.1.5 Reset Sources

The LPS reset sources are shown in the following table. LPS is initialized by the reset source.

**Table 12.8 Reset Sources**

Unit Name	Reset Source
LPS0	All reset sources except transition to DeepSTOP mode (AWORES)

### 12.1.6 External Input/Output Signals

External input/output signals of LPS are listed below.

**Table 12.9 External Input/Output Signals**

Unit Signal Name	Description	Alternative Port Pin Signal
<b>LPS0</b>		
DPO	Port output signal for digital input	DPO
DPSELk	External multiplexer select output signal for digital port	SELDPk
DPINm	Digital port input signal	DPINm
APO	Port output signal for analog input	APO
ADCA0Im	ADCA input channel signal	ADCA0Im

#### CAUTION

When the P0\_0 pin is used as DPO, note that the port P0\_0 outputs low-level as RESETOUT function during reset and after release from the reset.

For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

### 12.1.7 Internal Input/Output Signals

Internal input/output signals for connecting the LPS and STBC or the LPS and TAUJ are listed below.

**Table 12.10 Internal Input/Output Signals**

Unit Signal Name	Description	Connected to
WUTRG0	LPS wake-up source trigger 0 output signal	STBC
WUTRG1	LPS wake-up source trigger 1 output signal	STBC
INTTAUJ0Ix	LPS sequence start trigger x input signal	TAUJ0

## 12.2 Overview

### 12.2.1 Functional Overview

To monitor the external input without consuming CPU resources, the low-power sampler (LPS) can check the digital input ports and analog input ports without using the CPU. The figure below shows a connection example between the main components of the LPS and the external circuit.

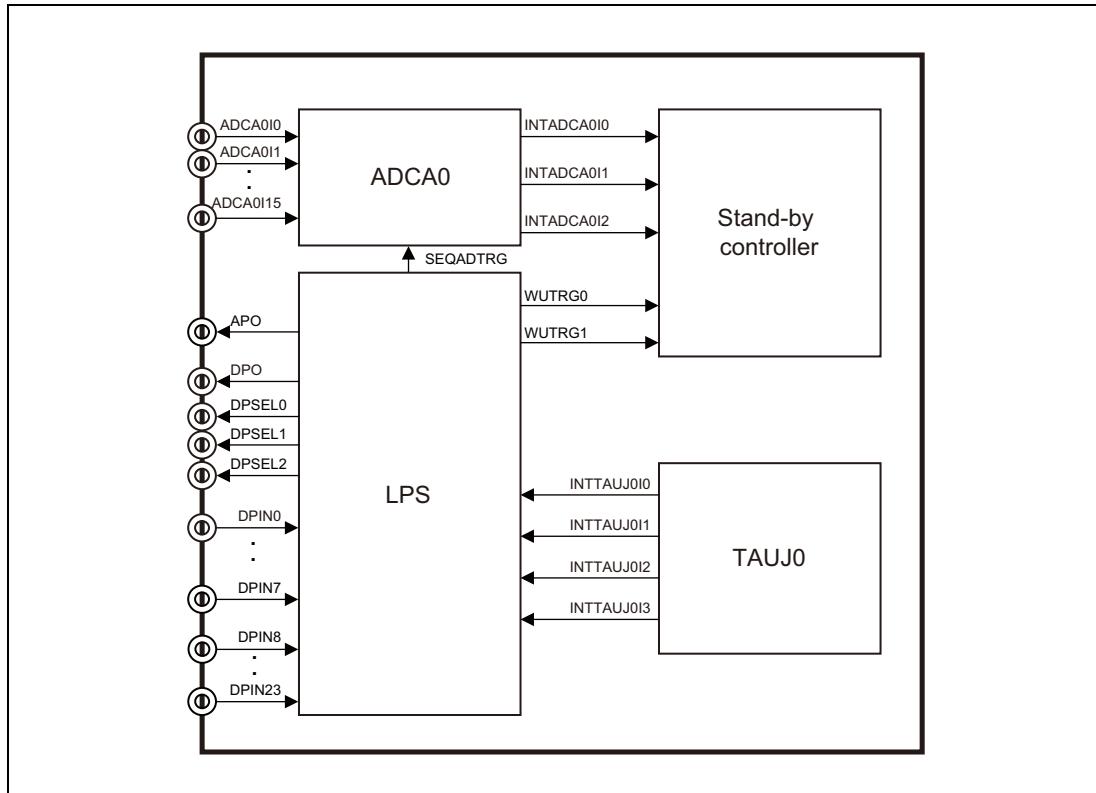


Figure 12.1 Block Diagram of the LPS

#### CAUTION

DPSEL2 to DPSEL0 are assigned to the same pins as DPIN10 to DPIN8 as alternative-function. They cannot be used simultaneously.

## 12.3 Registers

### 12.3.1 List of Registers

LPS registers are listed in the following table.

For details about <LPS0\_base>, see **Section 12.1.2, Register Base Address**.

Table 12.11 List of Registers

Module Name	Register Name	Symbol	Address
—	LPS control register	SCTRLR	<LPS0_base> + 00H
—	Event flag register	EVFR	<LPS0_base> + 04H
—	DPIN select register 0	DPSELRO	<LPS0_base> + 08H
—	DPIN select register M	DPSELRM	<LPS0_base> + 0CH
—	DPIN select register H	DPSELRH	<LPS0_base> + 10H
—	DPIN data set register 0	DPDSR0	<LPS0_base> + 14H
—	DPIN data set register M	DPDSRM	<LPS0_base> + 18H
—	DPIN data set register H	DPDSRH	<LPS0_base> + 1CH
—	DPIN data input monitor register 0	DPDIMR0	<LPS0_base> + 20H
—	DPIN data input monitor register 1	DPDIMR1	<LPS0_base> + 24H
—	DPIN data input monitor register 2	DPDIMR2	<LPS0_base> + 28H
—	DPIN data input monitor register 3	DPDIMR3	<LPS0_base> + 2CH
—	DPIN data input monitor register 4	DPDIMR4	<LPS0_base> + 30H
—	DPIN data input monitor register 5	DPDIMR5	<LPS0_base> + 34H
—	DPIN data input monitor register 6	DPDIMR6	<LPS0_base> + 38H
—	DPIN data input monitor register 7	DPDIMR7	<LPS0_base> + 3CH
—	Count value register	CNTVAL	<LPS0_base> + 40H
—	LPS operation status register	SOSTR	<LPS0_base> + 44H

### 12.3.2 SCLR — LPS Control Register

This register is used to configure the LPS.

**Access:** This register can be read or written in 32-bit units.

**Address:** <LPS0\_base> + 00H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						

Table 12.12 SCLR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	NUMDP[2:0]	These bits specify the number of times the port is read in digital input mode. If two or more times are specified, the external multiplexer is controlled by the DPSEL[2:0] pins. The bits for which comparison is enabled in the DPSELRO, DPSELRM, and DPSELRH registers are compared regardless of the repeat number setting, and WUTRG will be generated according to the results.
	NUMDP[2:0]	Number of Times the Port Is Read
	000 <sub>B</sub>	One time
	001 <sub>B</sub>	Two times
	010 <sub>B</sub>	Three times
	011 <sub>B</sub>	Four times
	100 <sub>B</sub>	Five times
	101 <sub>B</sub>	Six times
	110 <sub>B</sub>	Seven times
	111 <sub>B</sub>	Eight times
3, 2	TJIS[1:0]	These bits should be set before the TAUJ0 and sequence operations are started (when the SCLR.DPEN bit = 0, the SCLR.ADEN bit = 0, and the SOSTR.SOF bit = 0). (When changing the SCLR.DPEN bit and the SCLR.ADEN bit, write the same value to these bits.)
1	ADEN	Sequence Start Trigger Select 00: INTTAUJ0I0 01: INTTAUJ0I1 10: INTTAUJ0I2 11: INTTAUJ0I3 These bits should be set before the sequence operation is started (when the SCLR.DPEN bit = 0, the SCLR.ADEN bit = 0, and the SOSTR.SOF bit = 0). (When changing the SCLR.DPEN bit and the SCLR.ADEN bit, write the same value to these bits.)

**Table 12.12 SCTRL Register Contents (2/2)**

Bit Position	Bit Name	Function
0	DPEN	0: Digital input mode is disabled 1: Digital input mode is enabled

### 12.3.3 EVFR — Event Flag Register

This register indicates the result of comparing the data sequentially captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers.

**Access:** This register can be read or written in 32-bit units.

**Address:** <LPS0\_base> + 04H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DINEVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.13 EVFR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DINEVF	This bit indicates the result of comparing the data captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers.  Read: 0: The result of comparison is a match. 1: The result of comparison is a mismatch.  Write: 0: Clear the bit. 1: Prohibited.  This bit is set to 1 when a mismatch is detected even in one bit. Only 0 can be written to clear this bit.

### 12.3.4 DPSELRL0 — DPIN Select Register 0

This register specifies the compare target bits in the DPDSR0 and DPDIMR0 registers.

Write to the DPSELRL0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

**Access:** This register can be read or written in 32-bit units.

**Address:** <LPS0\_base> + 08H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0EN_23	D0EN_22	D0EN_21	D0EN_20	D0EN_19	D0EN_18	D0EN_17	D0EN_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0EN_15	D0EN_14	D0EN_13	D0EN_12	D0EN_11	D0EN_10	D0EN_9	D0EN_8	D0EN_7	D0EN_6	D0EN_5	D0EN_4	D0EN_3	D0EN_2	D0EN_1	D0EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 12.14 DPSELRL0 Register Contents**

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0EN_n (n = 23 to 0)	These bits enable or disable comparing each bit of the first data captured at the digital input pins and stored in the DPDIMR0 register with the comparison target data in the DPDSR0 register. 0: Disables comparison. 1: Enables comparison.

### 12.3.5 DPSELRM — DPIN Select Register M

This register specifies the compare target bits in the DPDSRM and DPDIMRm ( $m = 4$  to  $1$ ) registers.

Write to the DPSELRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

**Access:** DPSELRM can be read or written in 32-bit units.  
 DPSELRML and DPSELRMH can be read or written in 16-bit units.  
 DPSELR1, DPSELR2, DPSELR3, and DPSELR4 can be read or written in 8-bit units.

**Address:** DPSELRM: <LPS0\_base> + 0C<sub>H</sub>  
 DPSELRML: <LPS0\_base> + 0C<sub>H</sub>, DPSELRMH: <LPS0\_base> + 0E<sub>H</sub>  
 DPSELR1: <LPS0\_base> + 0C<sub>H</sub>, DPSELR2: <LPS0\_base> + 0D<sub>H</sub>, DPSELR3: <LPS0\_base> + 0E<sub>H</sub>,  
 DPSELR4: <LPS0\_base> + 0F<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4EN_7	D4EN_6	D4EN_5	D4EN_4	D4EN_3	D4EN_2	D4EN_1	D4EN_0	D3EN_7	D3EN_6	D3EN_5	D3EN_4	D3EN_3	D3EN_2	D3EN_1	D3EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2EN_7	D2EN_6	D2EN_5	D2EN_4	D2EN_3	D2EN_2	D2EN_1	D2EN_0	D1EN_7	D1EN_6	D1EN_5	D1EN_4	D1EN_3	D1EN_2	D1EN_1	D1EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.15 DPSELRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fifth data captured at the digital input pins and stored in the DPDIMR4 register with the comparison target data in the DPDSR4 register. 0: Disables comparison. 1: Enables comparison.
23 to 16	D3EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fourth data captured at the digital input pins and stored in the DPDIMR3 register with the comparison target data in the DPDSR3 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D2EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the third data captured at the digital input pins and stored in the DPDIMR2 register with the comparison target data in the DPDSR2 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D1EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the second data captured at the digital input pins and stored in the DPDIMR1 register with the comparison target data in the DPDSR1 register. 0: Disables comparison. 1: Enables comparison.

### 12.3.6 DPSELRH — DPIN Select Register H

This register specifies the compare target bits in the DPDSRH and DPDIMR $m$  ( $m = 7$  to  $5$ ) registers.

Write to the DPSELRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

**Access:** DPSELRH can be read or written in 32-bit units.  
 DPSELRHL and DPSELRHH can be read or written in 16-bit units.  
 DPSELR5, DPSELR6, and DPSELR7 can be read or written in 8-bit units.

**Address:** DPSELRH: <LPS0\_base> + 10<sub>H</sub>  
 DPSELRHL: <LPS0\_base> + 10<sub>H</sub>, DPSELRHH: <LPS0\_base> + 12<sub>H</sub>  
 DPSELR5: <LPS0\_base> + 10<sub>H</sub>, DPSELR6: <LPS0\_base> + 11<sub>H</sub>, DPSELR7: <LPS0\_base> + 12<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7EN_7	D7EN_6	D7EN_5	D7EN_4	D7EN_3	D7EN_2	D7EN_1	D7EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6EN_7	D6EN_6	D6EN_5	D6EN_4	D6EN_3	D6EN_2	D6EN_1	D6EN_0	D5EN_7	D5EN_6	D5EN_5	D5EN_4	D5EN_3	D5EN_2	D5EN_1	D5EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.16 DPSELRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the eighth data captured at the digital input pins and stored in the DPDIMR7 register with the compare target data in the DPDSR7 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D6EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the seventh data captured at the digital input pins and stored in the DPDIMR6 register with the compare target data in the DPDSR6 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D5EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the sixth data captured at the digital input pins and stored in the DPDIMR5 register with the compare target data in the DPDSR5 register. 0: Disables comparison. 1: Enables comparison.

### 12.3.7 DPDSR0 — DPIN Data Set Register 0

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR0 register.

Write to the DPDSR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

**Access:** This register can be read or written in 32-bit units.

**Address:** <LPS0\_base> + 14<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0_23	D0_22	D0_21	D0_20	D0_19	D0_18	D0_17	D0_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0_15	D0_14	D0_13	D0_12	D0_11	D0_10	D0_9	D0_8	D0_7	D0_6	D0_5	D0_4	D0_3	D0_2	D0_1	D0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 12.17 DPDSR0 Register Contents**

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0_n (n = 23 to 0)	Data to be compared with the first digital port input (DPINm)

### 12.3.8 DPDSRM — DPIN Data Set Register M

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR4 to DPDIMR1 registers.

Write to the DPDSRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

**Access:** DPDSRM can be read or written in 32-bit units.  
 DPDSRML and DPDSRMH can be read or written in 16-bit units.  
 DPDSR1, DPDSR2, DPDSR3, and DPDSR4 can be read or written in 8-bit units.

**Address:** DPDSRM: <LPS0\_base> + 18<sub>H</sub>  
 DPDSRML: <LPS0\_base> + 18<sub>H</sub>, DPDSRMH: <LPS0\_base> + 1A<sub>H</sub>  
 DPDSR1: <LPS0\_base> + 18<sub>H</sub>, DPDSR2: <LPS0\_base> + 19<sub>H</sub>, DPDSR3: <LPS0\_base> + 1A<sub>H</sub>,  
 DPDSR4: <LPS0\_base> + 1B<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4_7	D4_6	D4_5	D4_4	D4_3	D4_2	D4_1	D4_0	D3_7	D3_6	D3_5	D3_4	D3_3	D3_2	D3_1	D3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2_7	D2_6	D2_5	D2_4	D2_3	D2_2	D2_1	D2_0	D1_7	D1_6	D1_5	D1_4	D1_3	D1_2	D1_1	D1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 12.18 DPDSRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4_n (n = 7 to 0)	Data to be compared with the fifth digital port input (DPINm)
23 to 16	D3_n (n = 7 to 0)	Data to be compared with the fourth digital port input (DPINm)
15 to 8	D2_n (n = 7 to 0)	Data to be compared with the third digital port input (DPINm)
7 to 0	D1_n (n = 7 to 0)	Data to be compared with the second digital port input (DPINm)

### 12.3.9 DPDSRH — DPIN Data Set Register H

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR7 to DPDIMR5 registers.

Write to the DPDSRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

**Access:** DPDSRH can be read or written in 32-bit units.  
 DPDSRHL and DPDSRHH can be read or written in 16-bit units.  
 DPDSR5, DPDSR6, and DPDSR7 can be read or written in 8-bit units.

**Address:** DPDSRH: <LPS0\_base> + 1C<sub>H</sub>  
 DPDSRHL: <LPS0\_base> + 1C<sub>H</sub>, DPDSRHH: <LPS0\_base> + 1E<sub>H</sub>  
 DPDSR5: <LPS0\_base> + 1C<sub>H</sub>, DPDSR6: <LPS0\_base> + 1D<sub>H</sub>, DPDSR7: <LPS0\_base> + 1E<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7_7	D7_6	D7_5	D7_4	D7_3	D7_2	D7_1	D7_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6_7	D6_6	D6_5	D6_4	D6_3	D6_2	D6_1	D6_0	D5_7	D5_6	D5_5	D5_4	D5_3	D5_2	D5_1	D5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.19 DPDSRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7_n (n = 7 to 0)	Data to be compared with the eighth digital port input (DPINm)
15 to 8	D6_n (n = 7 to 0)	Data to be compared with the seventh digital port input (DPINm)
7 to 0	D5_n (n = 7 to 0)	Data to be compared with the sixth digital port input (DPINm)

### 12.3.10 DPDIMR0 — DPIN Data Input Monitor Register 0

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $23$ )) in digital input mode. DPDIMR0 stores the data acquired for the first time.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:**  $<\text{LPS0\_base}> + 20_{\text{H}}$

**Value after reset:**  $0000\ 0000_{\text{H}}$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DOM_2 3	DOM_2 2	DOM_2 1	DOM_2 0	DOM_1 9	DOM_1 8	DOM_1 7	DOM_1 6
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DOM_1 5	DOM_1 4	DOM_1 3	DOM_1 2	DOM_11	DOM_1 0	DOM_9	DOM_8	DOM_7	DOM_6	DOM_5	DOM_4	DOM_3	DOM_2	DOM_1	DOM_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.20 DPDIMR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	D0M_n ( $n = 23$ to 0)	The first digital port input (DPINm) data

### 12.3.11 DPDIMR1 — DPIN Data Input Monitor Register 1

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode or MIX mode. DPDIMR1 stores the data acquired for the second time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:**  $<\text{LPS0\_base}> + 24_{\text{H}}$

**Value after reset:**  $00_{\text{H}}$

Bit	7	6	5	4	3	2	1	0
	D1M_7	D1M_6	D1M_5	D1M_4	D1M_3	D1M_2	D1M_1	D1M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.21 DPDIMR1 Register Contents

Bit Position	Bit Name	Function
7 to 0	D1M_n ( $n = 7$ to 0)	The second digital port input (DPINm) data

### 12.3.12 DPDIMR2 — DPIN Data Input Monitor Register 2

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode or MIX mode. DPDIMR2 stores the data acquired for the third time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 28<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	D2M_7	D2M_6	D2M_5	D2M_4	D2M_3	D2M_2	D2M_1	D2M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.22 DPDIMR2 Register Contents

Bit Position	Bit Name	Function
7 to 0	D2M_n (n = 7 to 0)	The third digital port input (DPINm) data

### 12.3.13 DPDIMR3 — DPIN Data Input Monitor Register 3

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode or MIX mode. DPDIMR3 stores the data acquired for the fourth time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 2C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	D3M_7	D3M_6	D3M_5	D3M_4	D3M_3	D3M_2	D3M_1	D3M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.23 DPDIMR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	D3M_n (n = 7 to 0)	The fourth digital port input (DPINm) data

### 12.3.14 DPDIMR4 — DPIN Data Input Monitor Register 4

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode or MIX mode. DPDIMR4 stores the data acquired for the fifth time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 30<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	D4M_7	D4M_6	D4M_5	D4M_4	D4M_3	D4M_2	D4M_1	D4M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.24 DPDIMR4 Register Contents

Bit Position	Bit Name	Function
7 to 0	D4M_n (n = 7 to 0)	The fifth digital port input (DPINm) data

### 12.3.15 DPDIMR5 — DPIN Data Input Monitor Register 5

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode or MIX mode. DPDIMR5 stores the data acquired for the sixth time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 34<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	D5M_7	D5M_6	D5M_5	D5M_4	D5M_3	D5M_2	D5M_1	D5M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.25 DPDIMR5 Register Contents

Bit Position	Bit Name	Function
7 to 0	D5M_n (n = 7 to 0)	The sixth digital port input (DPINm) data

### 12.3.16 DPDIMR6 — DPIN Data Input Monitor Register 6

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode or MIX mode. DPDIMR6 stores the data acquired for the seventh time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 38<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	D6M_7	D6M_6	D6M_5	D6M_4	D6M_3	D6M_2	D6M_1	D6M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.26 DPDIMR6 Register Contents

Bit Position	Bit Name	Function
7 to 0	D6M_n (n = 7 to 0)	The seventh digital port input (DPINm) data

### 12.3.17 DPDIMR7 — DPIN Data Input Monitor Register 7

This register stores the data which the LPS acquired from the digital port input (DPINm ( $m = 0$  to  $7$ )) in multiplexer mode. DPDIMR7 stores the data acquired for the eighth time.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 3C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	D7M_7	D7M_6	D7M_5	D7M_4	D7M_3	D7M_2	D7M_1	D7M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.27 DPDIMR7 Register Contents

Bit Position	Bit Name	Function
7 to 0	D7M_n (n = 7 to 0)	The eighth digital port input (DPINm) data

### 12.3.18 CNTVAL — Count Value Register

This register specifies the stabilization time of the external circuits (digital signal source and analog signal source).

- In digital mode

The time from when the DPO output is set to 1 to the time when the port input is acquired for the first time

- In analog mode

The time from when the APO output is set to 1 to the time when the LPS outputs the A/D conversion trigger to the ADCA0

Write to the CNTVAL register before the sequence operation is started (when the SOSTR.SOF bit = 0).

#### CAUTION

In analog mode, make sure to secure the stabilization time longer than 1  $\mu$ s for the stabilization of the A/D converter.

**Access:** This register can be read or written in 16-bit units.

**Address:** <LPS0\_base> + 40<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT17	CNT16	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT07	CNT06	CNT05	CNT04	CNT03	CNT02	CNT01	CNT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.28 CNTVAL Register Contents

Bit Position	Bit Name	Function
15 to 8	CNT1n (n = 7 to 0)	These bits set the stabilization time of the external circuit (analog signal source). Stabilization time = $(1/f_{RH}) \times 16 \times CNT1n$ (set value)
7 to 0	CNT0n (n = 7 to 0)	These bits set the stabilization time of the external circuit (digital signal source). Stabilization time = $(1/f_{RH}) \times 16 \times CNT0n$ (set value)

### 12.3.19 SOSTR — LPS Operation Status Register

This register indicates the operating state of the LPS.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <LPS0\_base> + 44H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SOF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.29 SOSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	SOF	LPS Operation Status Flag 0: Initial state before the occurrence of the start trigger 1: LPS operation is in progress (after the start trigger occurs) If the start trigger occurs while the SOF bit is set to 1 (during the LPS operation), the start trigger is canceled.

## 12.4 Digital Input Mode

With the digital input port DPINm and the externally connected multiplexer, up to 64 input ports can be monitored as shown in **Table 12.30, Combination of Monitored Ports**.

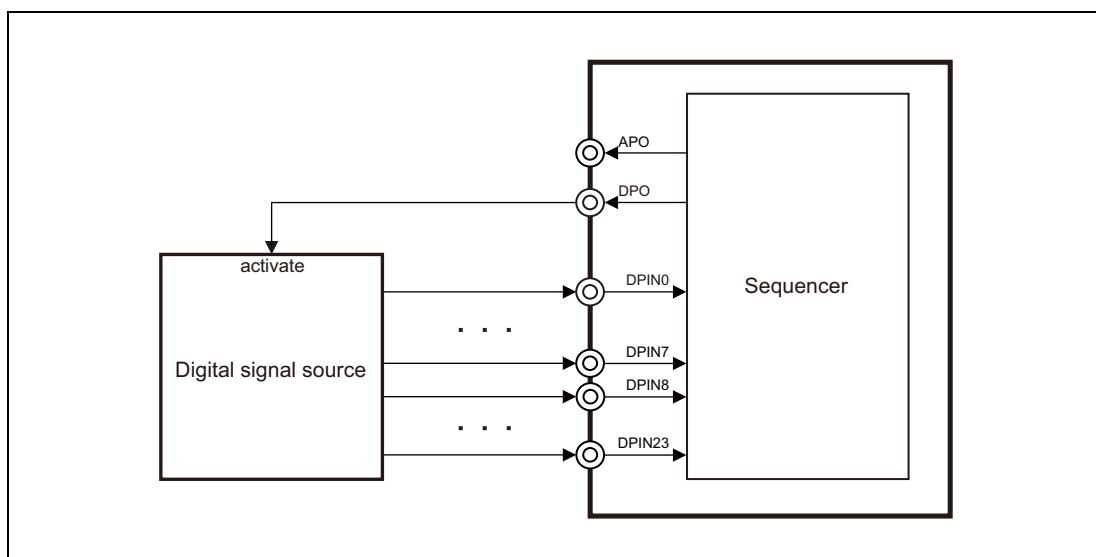
Port DPSELk is used to switch the external multiplexer. The DPSELk output is switched for the number of times specified in the SCTRL register.

TAUJ0 is used to set the timing to check the value input to the port.

**Table 12.30 Combination of Monitored Ports**

Combination (Number of Ports x Number of Checks)	Ports Used	Total Number
Direct mode When input ports are checked simultaneously without using the external multiplexer Up to 24 ports × 1	DPIN23 to DPIN0	Up to 24
Multiplexer mode When input ports are checked by using a small number of pins and the external multiplexer Up to 8 ports × 8	DPIN7 to DPIN0 DPSEL2 to DPSEL0	Up to 64
MIX mode When input ports are checked using a combination of the above two modes Up to 14 ports × 1 + Up to 7 ports × 7	DPIN7 to DPIN0 DPIN16 to DPIN11 DPSEL2 to DPSEL0*1	Up to 63

Note 1. DPIN16 to DPIN11 and DPIN7 are checked only for the first time. DPIN10 to DPIN8 cannot be used because they are shared with DPSEL2 to DPSEL0.



**Figure 12.2 Direct Mode Connection Example**

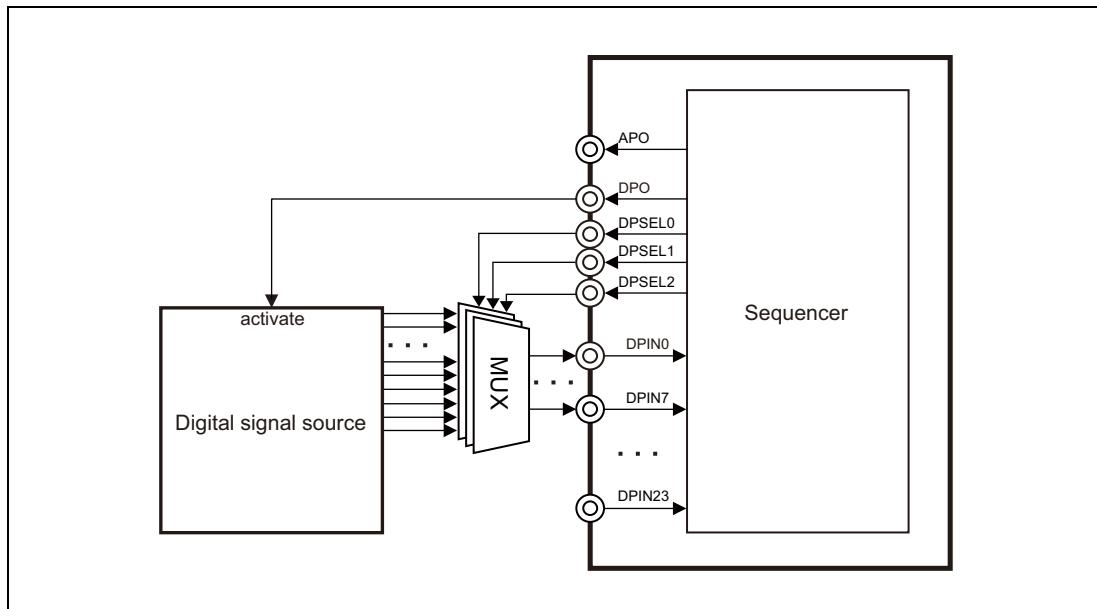


Figure 12.3 Multiplexer Mode Connection Example

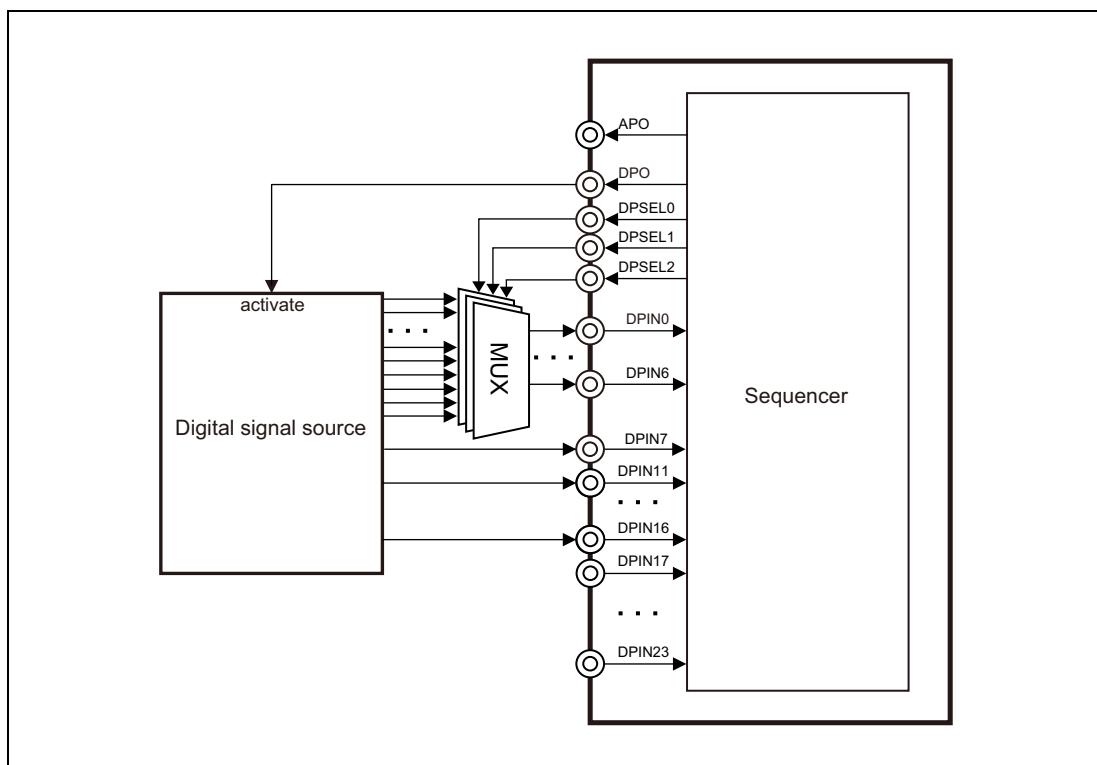


Figure 12.4 MIX Mode Connection Example

#### CAUTION

DPSEL2 to DPSEL0 are assigned to the same pins as DPIN10 to DPIN8 as alternative-function. They cannot be used simultaneously.

### Preparation

- Set NUMDP[2:0] and TJIS[1:0] bits in the SCTLR register to specify the number of times the port is to be read, and the TAUJ0 interrupt to be used as sequence start trigger.
- Set TAUJ0 to interval timer mode.
- Set the wait time of the digital signal source by using the lower 8 bits in the CNTVAL register.
- Set expected values in the DPDSR0, DPDSRM and DPDSRH registers.
- Set the ports to be checked in the DPSELR0, DPSELRM, and DPSELRH registers.

### Start

- Start the TAUJ0.
- Set the SCTLR.DPEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJ0. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in stand-by mode, it can only resume operation while the sequencer is running.

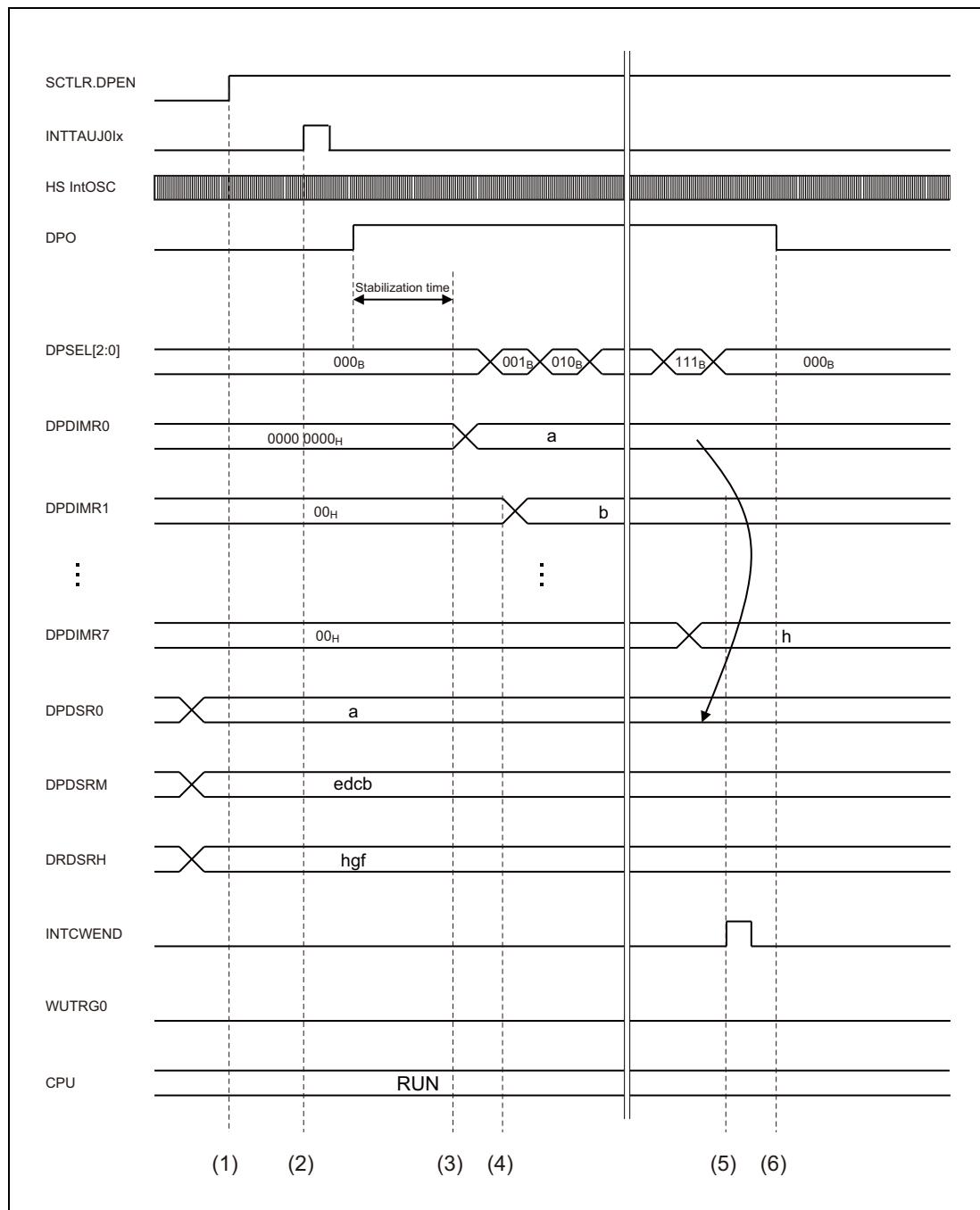
Upon completion of checking all ports that have been set, the INTCWEND interrupt occurs. In addition, if the input value of the port is different from the expected value set by the DPDSR0, DPDSRM, or DPDSRH register, the wake-up factor WUTRG0 occurs. The following figures show an example of the operation in digital input mode.

### Stop

To stop the LPS operation in Digital Input Mode (by changing the SCTLR.DPEN bit setting from 1 to 0), follow the procedure shown below. In this example, the P0\_0 pin is used as DPO.

1. Set the port register to specify low level output on the pin (P0.P0\_0 = 0).
2. Change the setting for the P0\_0 pin from the alternative port mode to the port mode (PMC0.PMC0\_0 = 0).
3. Set SCTLR.DPEN = 0.

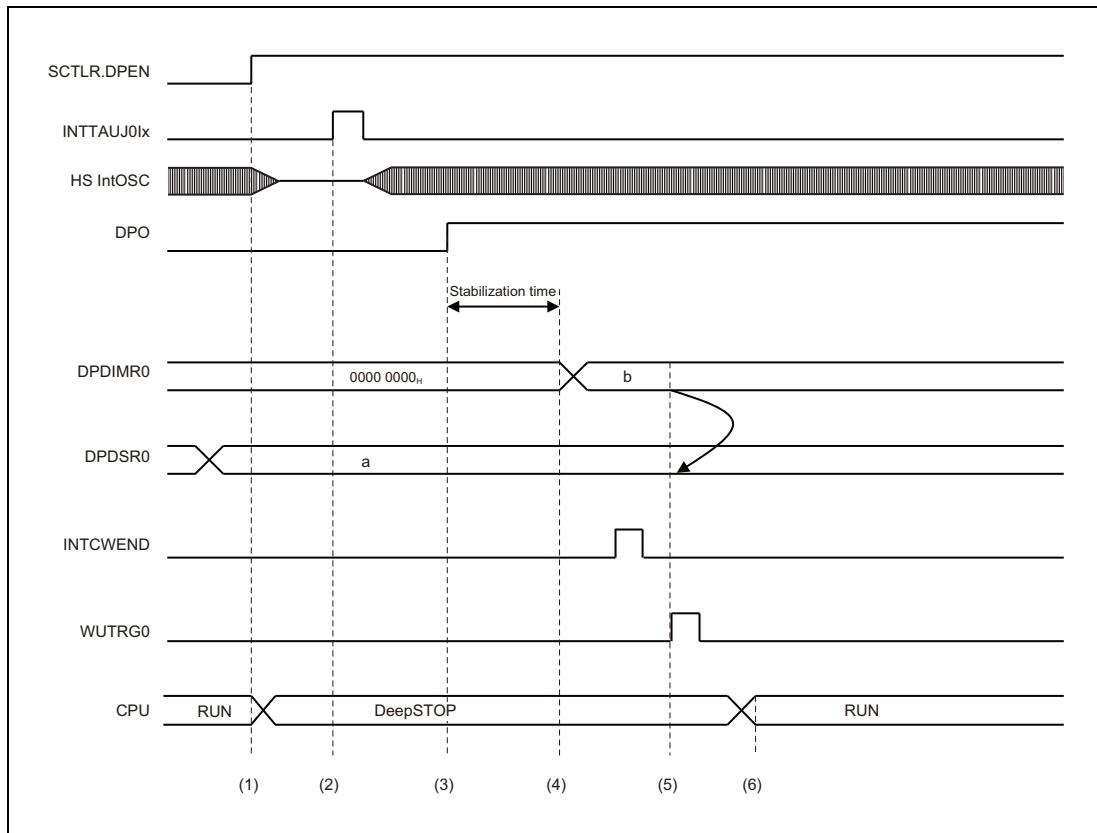
**Note 1.** The above procedure applies when the P0\_0 pin is used as DPO. If the P0\_2 pin is used as DPO, specify the P0\_2 pin settings in the same way.



**Figure 12.5 Operation of Digital Input Mode (8 Ports × 8) when the Input Value is not Changed (RUN Mode)**

- (1) Set the SCTLR.DPEN bit to 1 by software to enable the digital input mode of the LPS.
- (2) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the sequencer outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.
- (3) After the completion of the signal source stabilization, the LPS stores the DPIN[7:0] input value to the DPDIMR0 register and increments the DPSEL[2:0] pins to switch the external multiplexer.
- (4) After the switching of the DPSEL[2:0] pins, the LPS stores the values in the DPDIMRn registers in order from DPDIMR1 and continues to increment the DPSEL[2:0] pins.

- (5) After the value is stored up to the DPDIMR7 register, the INTCWEND interrupt is generated and the value is compared with the expected value set in the DPDSR0, DPDSRM, and DPDSRH registers.
- (6) When the value is not different from the expected value, the wake-up factor WUTRG0 is not generated. The LPS stops the DPO output and returns to the waiting state for the trigger.



**Figure 12.6 Operation of Digital Input Mode (24 Ports x 1) when the Input Value is Changed (DeepSTOP Mode)**

- (1) Set the STBC0PSC.STBC0DISTRG bit to 1 to transition to the DeepSTOP mode, while the SCTLR.DPEN bit is set to 1 by software to enable the digital input mode of the LPS.
- (2) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the LPS enables the HS IntOSC to start the oscillation.
- (3) After the completion of the HS IntOSC stabilization time, the LPS outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.
- (4) After the completion of the signal source stabilization, the LPS stores the DPIN[23:0] input value to the DPDIMR0 register and the INTCWEND interrupt is generated.
- (5) The value stored in the DPDIMR0 register is compared with the expected value set in the DPDSR0 register. When the value is different from the expected value, the wake-up factor WUTRG0 is generated.
- (6) The CPU returns to RUN mode at the generation of WUTRG0. The DPO pin is driven high until the EVFR.DINEVF bit is cleared to 0 by software.

## 12.5 Analog Input Mode

The analog input port ADCA0Im ( $m = 0$  to  $15$ ) can be monitored.

TAUJ0 is used to set the timing to check the value input to the port.

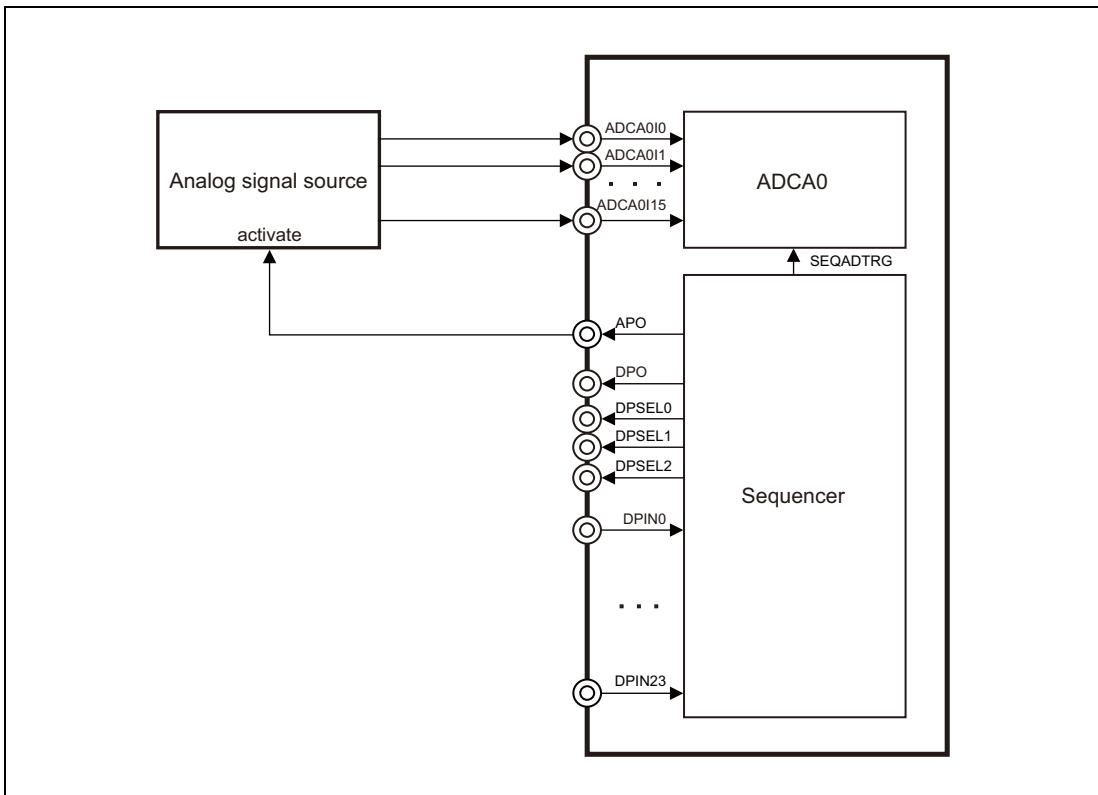


Figure 12.7 Analog Input Mode Connection Example

### Preparation

- Set TJIS[1:0] bits in the SCLTR register to specify the TAUJ0 interrupt to be used as sequence start trigger.
- Set TAUJ0 to interval timer mode.
- Set the wait time of the analog signal source by using the upper 8 bits in the CNTVAL register.
- Set the ADCA0.

### CAUTIONS

- When the LPS is in use, the A/D conversion completion interrupt (INT\_SGx) should be output after the conversion of all channels of the LPS has been completed. The setting is as follows.
  - Set the ADIE bit in virtual channel register j (ADCA0VCRj) to 0 (a scan group x end interrupt (INT\_SGx) is not generated when A/D conversion for virtual channel j ends in SGx.).
  - Set the ADIE bit in the scan group x control register (ADCA0SGCRx) to 1 (INT\_SGx is output when the scan for SGx ends).
- Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[1:0] bits to the completion of A/D conversion for all channels of the LPS, only proceed with A/D conversion for PWM-Diag.

3. Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[1:0] bits to the completion of A/D conversion for all channels of LPS, do not forcibly end A/D conversion by using the ADCA0ADHALTR.HALT bit.
4. When the LPS is in use, do not use the following modes.
  - Continuous scan mode (the setting ADCA0SGCRx.SCANMD = 1 is prohibited)
  - Multicycle scan mode with 2 or more cycles  
(the settings ADCA0SGMCYCRx.MCYC = 01<sub>B</sub> and 11<sub>B</sub> are prohibited)
  - Channel repeat mode with 2 or more cycles  
(the settings ADCA0SGCRx.SCT = 01<sub>B</sub> and 10<sub>B</sub> are prohibited)

### Start

- Start the TAUJ0.
- Set the SCTLR.ADEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJ0. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in stand-by mode, it can only resume operation while the sequencer is running.

To detect whether the analog input value differs from the expected value, use the A/D error interrupt request (INTADCA0ERR) of the A/D converter.

In addition, if the analog input value is different from the expected value, the wake-up factor WUTRG1 occurs.

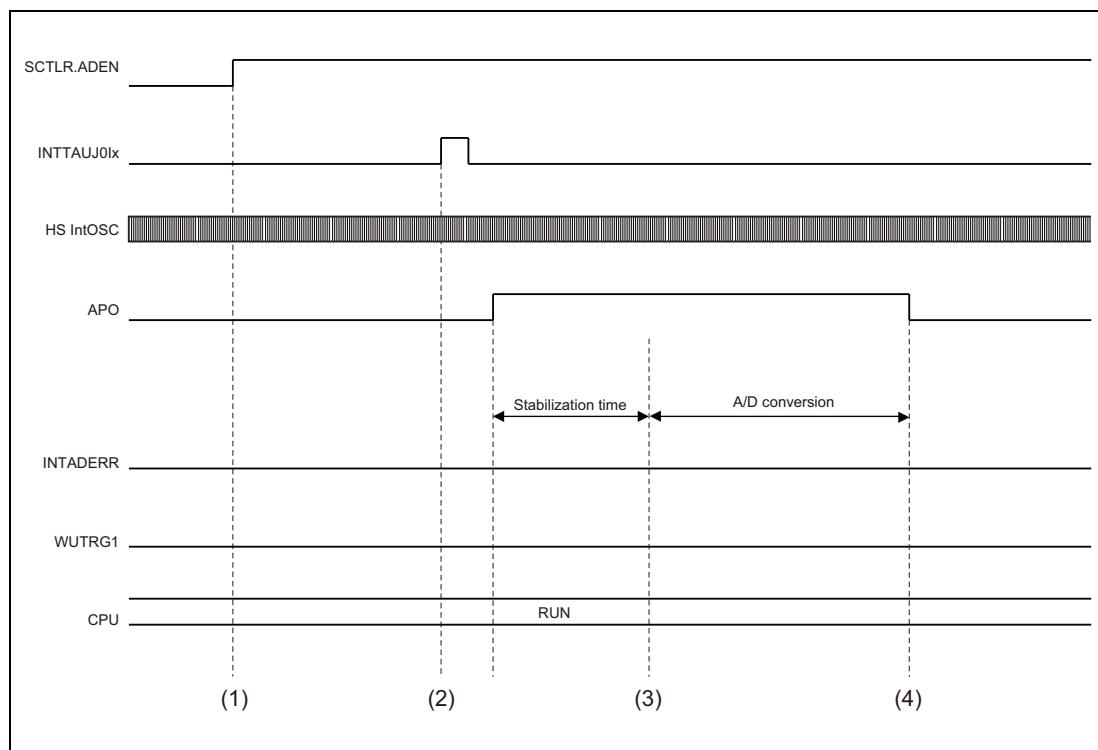
For details on the A/D error interrupt request (INTADCA0ERR), see **Section 29.4.13, A/D Error Interrupt Request<sup>\*1</sup>**. The following figures show an example of the operation in analog input mode.

**Note 1.** In **Section 29 A/D Converter (ADCA)**, the name of the A/D error interrupt request is described as “INT\_ADE”.

### Stop

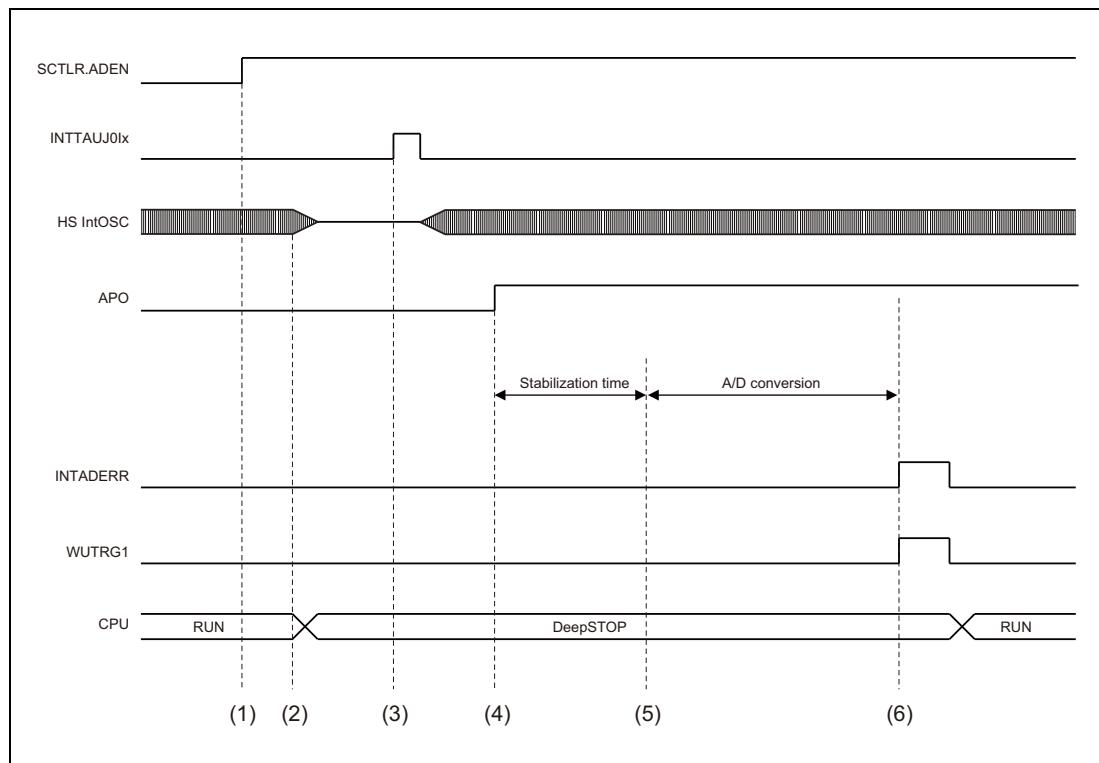
To stop the LPS operation in Analog Input Mode (by changing the SCTLR.ADEN bit setting from 1 to 0), follow the procedure shown below. Note that the P0\_1 pin is used as APO.

1. Set the port register to specify low level output on the pin (P0.P0\_1 = 0).
2. Change the setting for the P0\_1 pin from the alternative port mode to the port mode (PMC0.PMC0\_1 = 0).
3. Set SCTLR.ADEN = 0.



**Figure 12.8 Operation of Analog Input Mode when the Conversion Result is within the Expected Range (RUN Mode)**

- (1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTRLR.ADEN bit to 1 to enable the analog input mode of the LPS.
- (2) When the INTTAUJ0Ix interrupt specified by the SCTRLR.TJS bit is generated, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source. Set the stabilization time not less than 1  $\mu$ s.
- (3) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCA0Im ( $m = 0$  to 15), set in the A/D converter scan group, is started.
- (4) When the INTADCA0ERR interrupt is not generated as a result of A/D conversion, the LPS halts the A/D converter and resets the APO pin.



**Figure 12.9 Operation of Analog Input Mode when the Conversion Result is not within the Expected Range (DeepSTOP Mode)**

- (1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.
- (2) Set the STBC0PSC.STBC0DISTRG bit to 1 by software to transition to the DeepSTOP mode.
- (3) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the LPS enables the HS IntOSC to start the oscillation.
- (4) After the completion of the HS IntOSC stabilization, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source.
- (5) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCA0Im ( $m = 0$  to 15), set in the A/D converter scan group, is started.
- (6) When the INTADCA0ERR interrupt is generated as a result of A/D conversion, the wake-up factor WUTRG1 is generated and the CPU returns to RUN mode. The APO pin is driven high until the upper limit/lower limit error flag of the A/D converter is cleared to 0 by software.

## Section 13 External Memory Access Controller (MEMC)

This section contains a generic description of the external memory access controller (MEMC).

The first part in this section describes the features specific to RH850/F1L microcontrollers, including register base addresses and input/output signals. The ensuing sections describe the functions and registers of MEMC.

### 13.1 Features of MEMC

#### 13.1.1 Products that Incorporate MEMC

MEMC is incorporated in the following products.

**Table 13.1 Products that Incorporate MEMC**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	—	—	—	—	—	1
Unit Name	—	—	—	—	—	MEMC $n$ ( $n = 0$ )

**Table 13.2 Indices**

Index	Description
n	Throughout this section, the individual MEMC units are identified by the index “n” ( $n = 0$ ).
x	Throughout this section, chip select areas are identified by “x” ( $x = 0$ to 3). For example, the external memory area of CS $x$ is described as the CS $x$ area.

**Table 13.3 External Memory Access Functions**

Control Signal Name	Number of Control Signals
Address output	20 bits
Chip select output	4
Data bus	8/16 bits

Note 1. The lower 16 bits of the address output are multiplexed with the data bus.

#### 13.1.2 Register Base Address

The MEMC base address is shown in the following table.

The MEMC register addresses are expressed as offsets from the base address.

**Table 13.4 Register Base Address**

Base Address Name	Base Address
<MEMC0_base>	FFFF 8200 <sub>H</sub>

### 13.1.3 Clock Supply

The MEMC clock supply is shown in the following table. The maximum frequency of MEMC0CLK is 40 MHz.

**Table 13.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
MEMC	MEMC0CLK	CKSCLK_CPUCLK* <sup>1</sup>

Note 1. CKSCLK\_CPUCLK/2 or CKSCLK\_CPUCLK/4 is selected as the clock using the option bytes.

### 13.1.4 Reset Sources

The MEMC reset sources are shown below. MEMC is initialized by the following reset sources.

**Table 13.6 Reset Sources**

Unit Name	Reset Source
MEMC	All reset sources (ISORES)

### 13.1.5 External Input/Output Signals

External input/output signals of MEMC are listed below.

**Table 13.7 External Input/Output Signals**

Unit Signal Name	Description	Alternative Port Pin Signal Name
<b>MEMC</b>		
MEMC0A[19:16]	Address bus output signal	MEMC0A[19:16]
MEMC0AD[15:0]	Address/data bus I/O signal	MEMC0AD[15:0]
MEMC0ASTB	Address strobe output signal	MEMC0ASTB
MEMC0BEN[1:0]	Byte enable output signal	MEMC0BEN[1:0]
MEMC0CLK	Bus clock output signal	MEMC0CLK
MEMC0CS[3:0]	Chip select output signal	MEMC0CS[3:0]
MEMC0RD	Read strobe output signal	MEMC0RD
MEMC0WAIT	External wait request input signal	MEMC0WAIT
MEMC0WR	Write strobe output signal	MEMC0WR

Note 1. For the port pins that are used as MEMC, set the output driving ability to high (PDSCn\_m = 1).

## 13.2 Overview

The external memory access controller provides four chip select areas, and the bus width and wait time are selectable in each chip select area.

Moreover, the bus clock (CPU clock/2, CPU clock/4) is selectable using the option bytes. For details, see **Section 35.12, Usage Notes**.

### 13.2.1 Functional Overview

The main features of the external memory access controller:

- Multiplexed bus mode
- Four chip select areas
- Data bus width can be selected individually for each chip select area (8 or 16 bits)
- The data endian format can be set individually for each chip select area.
- Various wait functions can be set individually for each chip select area.
- External wait on SRAM access cycles

#### 13.2.1.1 Multiplexed Bus

This is an operation mode that connects address output and data input/output to external memory using the same signal line, making it possible to reduce the number of pins required for external memory connection.

#### 13.2.1.2 Chip Select Output Function

The external bus area of the memory space is divided into four chip select areas, and a chip select signal can be output for each chip select area. The allocation of these chip select areas is fixed by the system and cannot be changed through programming.

#### 13.2.1.3 Bus Sizing Function

The bus size can be selected from 8 bits or 16 bits for each chip select area. When data that exceeds the selected bus width is accessed, the data is divided into sizes smaller than the bus width by using the bus sizing function.

#### 13.2.1.4 Data Endian Setting Function

The data endian (little endian/big endian) can be specified for each chip select area.

#### 13.2.1.5 Programmable Wait Setting Functions

This microcontroller has the following wait functions, which can be set for each chip select area.

- Programmable data wait
- Data hold wait
- Address setup wait
- Address hold wait
- Idle cycle (after read cycle)

### 13.2.1.6 External Wait Function

When accessing SRAM data in a write cycle, waits of any width can be inserted externally from the MEMC0WAIT pin. The MEMC0WAIT pin is sampled just before the data output cycle, and the data latch timing can be delayed by any amount.

## 13.3 Registers

### 13.3.1 List of Registers

The MEMC registers are listed in the following table.

For details on <MEMCn\_base>, see **Section 13.1.2, Register Base Address**.

Table 13.8 List of Registers

Module Name	Register Name	Symbol	Address
MEMC	Data bus width configuration register	BSC	<MEMCn_base> + 00 <sub>H</sub>
	Data endian configuration register	DEC	<MEMCn_base> + 02 <sub>H</sub>
	Data wait configuration register	DWC	<MEMCn_base> + 08 <sub>H</sub>
	Data hold wait configuration register	DHC	<MEMCn_base> + 0C <sub>H</sub>
	Address wait configuration register	AWC	<MEMCn_base> + 10 <sub>H</sub>
	Idle cycle configuration register	ICC	<MEMCn_base> + 14 <sub>H</sub>

**NOTE**

Writing to the MEMC registers is enabled only in supervisor mode(PSW.UM = 0).

### 13.3.2 BSC — Data Bus Width Configuration Register

The BSC register is used to set the data bus width of the external bus for each chip select area.

**Access:** This register can be read or written in 16-bit units.

**Address:** <MEMCn\_base> + 00H

**Value after reset:** 0055H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

**Table 13.9 BSC Register Contents**

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 4, 2, 0	BSx	Bus Size Setting These bits set the bus width of each chip select area. 0: 8 bits 1: 16 bits

The relationship between each chip select area and the control bit is shown below.

**Table 13.10 Relationship between BSx Bit and Chip Select Area**

Chip Select Area	BSx Bit
CS3 area	BS3
CS2 area	BS2
CS1 area	BS1
CS0 area	BS0

### 13.3.3 DEC — Data Endian Configuration Register

The DEC register is used to set the endianness of the external bus.

**Access:** This register can be read or written in 16-bit units.

**Address:** <MEMCn\_base> + 02H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

**Table 13.11 DEC Register Contents**

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 4, 2, 0	DEx	Data Endian Setting These bits set the endian of each chip select area. 0: Little endian 1: Big endian

The relationship between each chip select area and the control bit is shown below.

**Table 13.12 Relationship between DEx Bit and Chip Select Area**

Chip Select Area	DEx Bit
CS3 area	DE3
CS2 area	DE2
CS1 area	DE1
CS0 area	DE0

### 13.3.4 DWC — Data Wait Configuration Register

The DWC register is used to set the number of data wait states for the external bus.

<b>Access:</b>	This register can be read or written in 16-bit units.															
<b>Address:</b>	<MEMCn_base> + 08H															
<b>Value after reset:</b>	7777H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 13.13 DWC Register Contents

Bit Position	Bit Name	Function	
15, 11, 7, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.	
14 to 12, 10 to 8, 6 to 4, 2 to 0	DWx2, DWx1, DWx0	Data Wait Setting These bits set the number of data wait states for each chip select area.	
DWx2	DWx1	DWx0	Number of Data Wait States
0	0	0	Setting prohibited
0	0	1	1 clock cycle
0	1	0	2 clock cycles
0	1	1	3 clock cycles
1	0	0	4 clock cycles
1	0	1	5 clock cycles
1	1	0	6 clock cycles
1	1	1	7 clock cycles

The relationship between each chip select area and the control bit is shown below.

Table 13.14 Relationship between DWx2 to DWx0 Bits and Chip Select Area

Chip Select Area	DWx2 to DWx0 Bits
CS3 area	DW32, DW31, DW30
CS2 area	DW22, DW21, DW20
CS1 area	DW12, DW11, DW10
CS0 area	DW02, DW01, DW00

### 13.3.5 DHC — Data Hold Wait Configuration Register

The DHC register is used to set the number of data hold waits for the external bus.

By setting this register, a data hold wait of "DHC register setting + one cycle" is inserted in a write cycle.

**Access:** This register can be read or written in 16-bit units.

**Address:** <MEMCn\_base> + 0C<sub>H</sub>

**Value after reset:** 0055<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	DH3	—	DH2	—	DH1	—	DH0
Value after reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 13.15 DHC Register Contents

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 4, 2, 0	DHx	Data Hold Wait Setting These bits set the number of data hold wait states for each chip select area. 0: No data hold wait 1: 1 clock cycle

The relationship between each chip select area and the control bit is shown below.

Table 13.16 Relationship between DHx Bits and Chip Select Area

Chip Select Area	DHx Bits
CS3 area	DH3
CS2 area	DH2
CS1 area	DH1
CS0 area	DH0

### 13.3.6 AWC — Address Wait Configuration Register

The AWC register is used to set the address wait period of the external bus for each chip select area.

<b>Access:</b>	This register can be read or written in 16-bit units.															
<b>Address:</b>	<MEMCn_base> + 10H															
<b>Value after reset:</b>	5555H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Table 13.17 AWC Register Contents

Bit Position	Bit Name	Function
15, 13, 11, 9, 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14, 10, 6, 2	AHWx	Address Hold Wait Setting These bits set the number of address hold wait states for each chip select area. 0: No address hold wait 1: 1 clock
12, 8, 4, 0	ASWx	Address Setup Wait Setting These bits set the number of address setup wait states for each chip select area. 0: No address setup wait 1: 1 clock

The relationship between each chip select area and the control bit is shown below.

Table 13.18 Relationship between ASWx / AHWx Bits and Chip Select Area

Chip Select Area	ASWx and AHWx Bits
CS3 area	ASW3, AHW3
CS2 area	ASW2, AHW2
CS1 area	ASW1, AHW1
CS0 area	ASW0, AHW0

### 13.3.7 ICC — Idle Cycle Configuration Register

The ICC register is used to set the number of idle cycles of the external bus. The number of idle cycles can be set for each chip select area.

**Access:** This register can be read or written in 16-bit units.

**Address:** <MEMCn\_base> + 14H

**Value after reset:** 1111H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	RIC3	—	—	—	RIC2	—	—	—	RIC1	—	—	—	RIC0
Value after reset	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Table 13.19 ICC Register Contents

Bit Position	Bit Name	Function
15 to 13, 11 to 9, 7 to 5, 3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12, 8, 4, 0	RICx	Idle Cycle Setting after Read Cycle These bits set the number of idle cycles after a read cycle for each chip select area. 0: No idle cycle 1: 1 clock

## 13.4 Functions

### 13.4.1 Bus Control Functions

#### 13.4.1.1 Chip Select Output Function

The connected external memory area is divided into and managed in 4 chip select areas as shown in **Figure 13.1, External Memory Map**.

When a bus cycle is generated for the external bus, this microcontroller makes the  $\overline{\text{MEMC0CS[3:0]}}$  output pin corresponding to the access target address active (low level), along with outputting the access target address from the  $\text{MEMC0A[19:16]}$  and  $\text{MEMC0AD[15:0]}$ .

The various settings for the external bus, such as the bus size and number of wait/idle states, can all be made for each chip select area.

By using these functions, different types of memory can be connected to each chip select area.

The allocation of the chip select areas is fixed by the system and cannot be changed through programming.

The memory map is shown below.

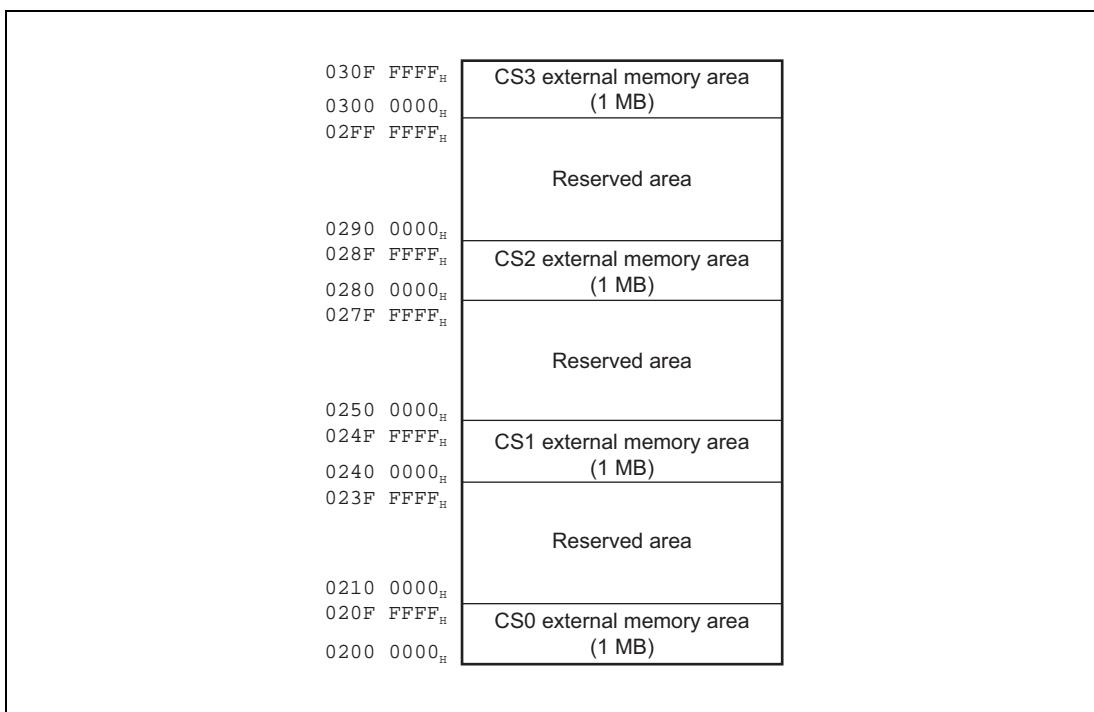


Figure 13.1 External Memory Map

### 13.4.1.2 Bus Sizing Function

Access requests from the CPU (or DMA) are executed after being divided in accordance with the bit width of the external bus of the access destination.

The bit width of the external bus can be selected from 16 and 8 bits for each chip select area by setting the BSC register.

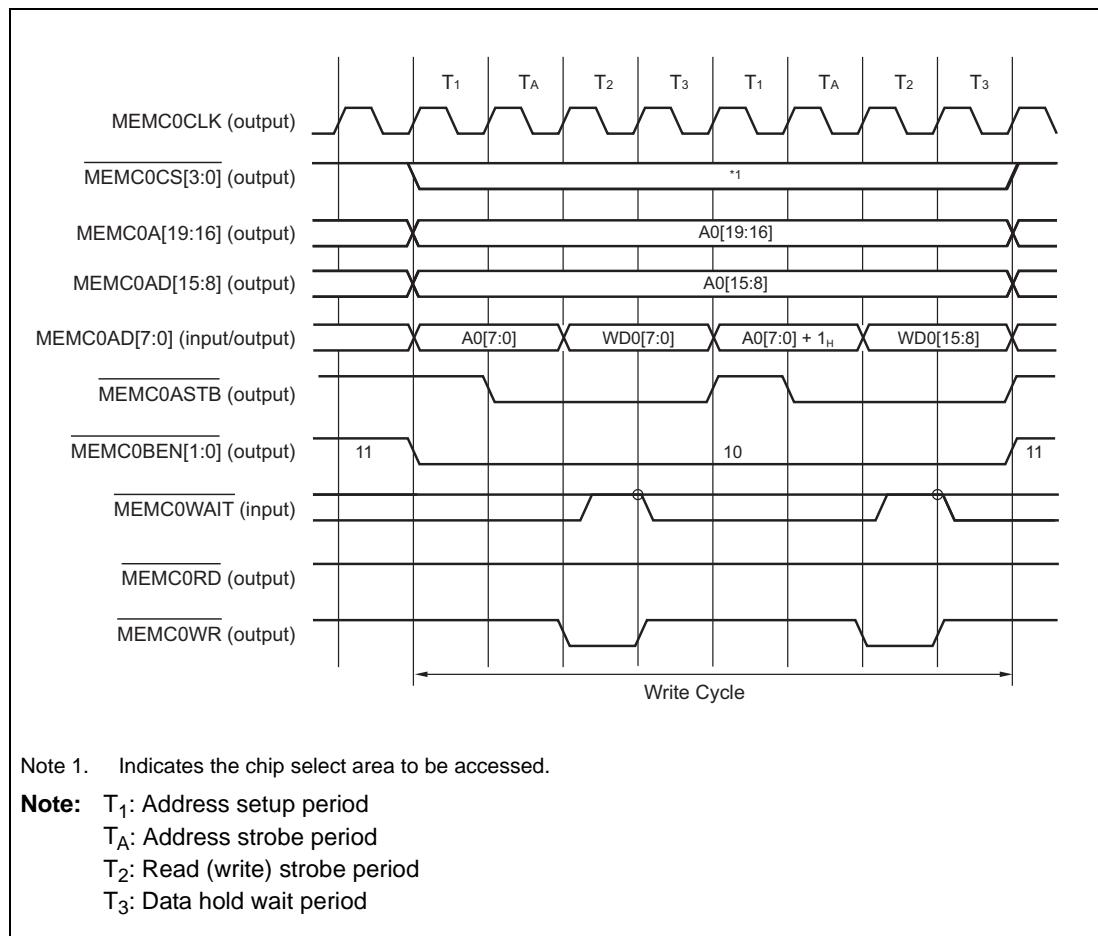


Figure 13.2 Bus Cycles when Making a 16-Bit Write Access to an 8-Bit External Bus

### 13.4.1.3 Data Endian Setting Function

Either little endian or big endian can be selected as the data endian of the external bus interface. This setting can be made for each chip select area with the DEC register.

#### CAUTIONS

1. In this microcontroller, instruction fetch operation with big endian is not supported. Assembler and debugger were made for little endian.
2. Misaligned access in big endian format is not supported.

### 13.4.2 Wait Functions

Wait functions are listed below.

Table 13.20 Wait Functions

Wait Function	Data Wait		Data Hold Wait	Address Wait	Idle
	Programmable	External Wait			
Read	✓	✓	—	✓	✓
Write	✓	✓	✓	✓	—
Setting registers	DWC	—	DHC	AWC	ICC
Max. number of waits	7	—	1	1	1

### 13.4.2.1 Programmable Data Wait Function

This wait function is used to delay the data latch timing by extending the read strobe and write strobe periods.

This function is enabled during data transfer.

Up to 7 cycles can be inserted.

Data waits can be set for each chip select area using the DWC register.

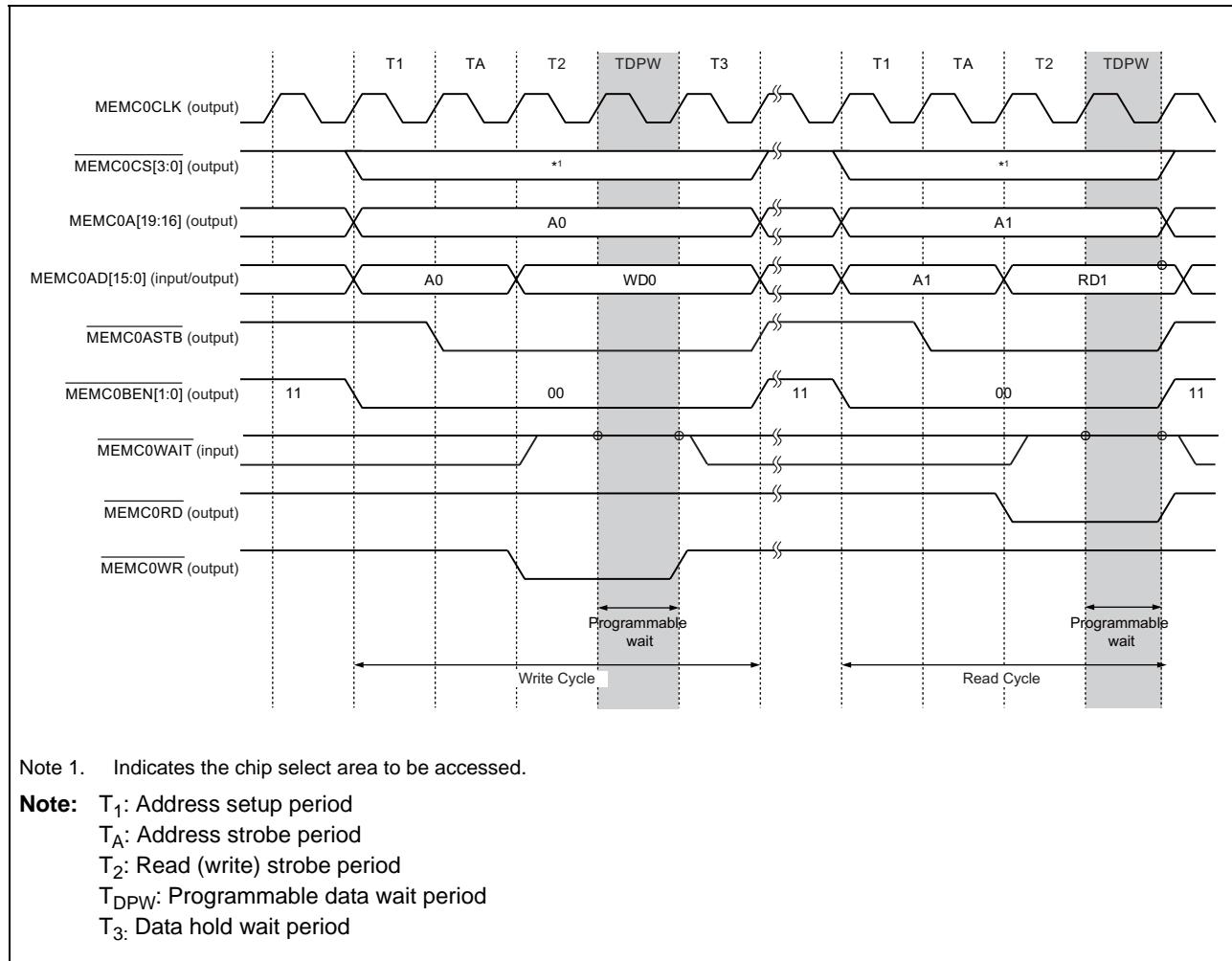


Figure 13.3 Bus Cycles when Inserting Programmable Data Waits

### 13.4.2.2 External Wait Function

Data waits of any length can be inserted from the  $\overline{\text{MEMC}0\text{WAIT}}$  pin.

The  $\overline{\text{MEMC}0\text{WAIT}}$  pin input level is sampled immediately after completion of the  $T_2$  cycle and the  $T_{DPW}$ ,  $T_{DEW}$  cycles.

Data wait cycles obtained by ORing the programmable data wait set by the data wait control register (DWC register) and the external wait set by the  $\overline{\text{MEMC}0\text{WAIT}}$  pin input, are inserted.

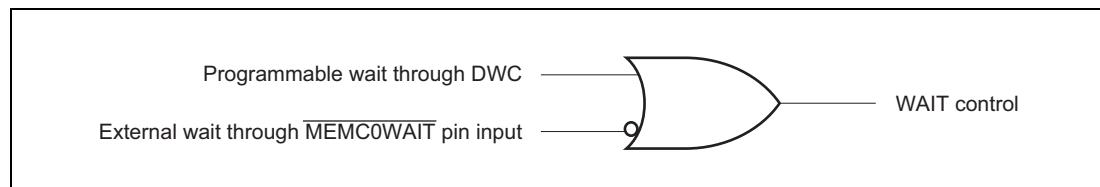


Figure 13.4 Internal Data Wait Generator

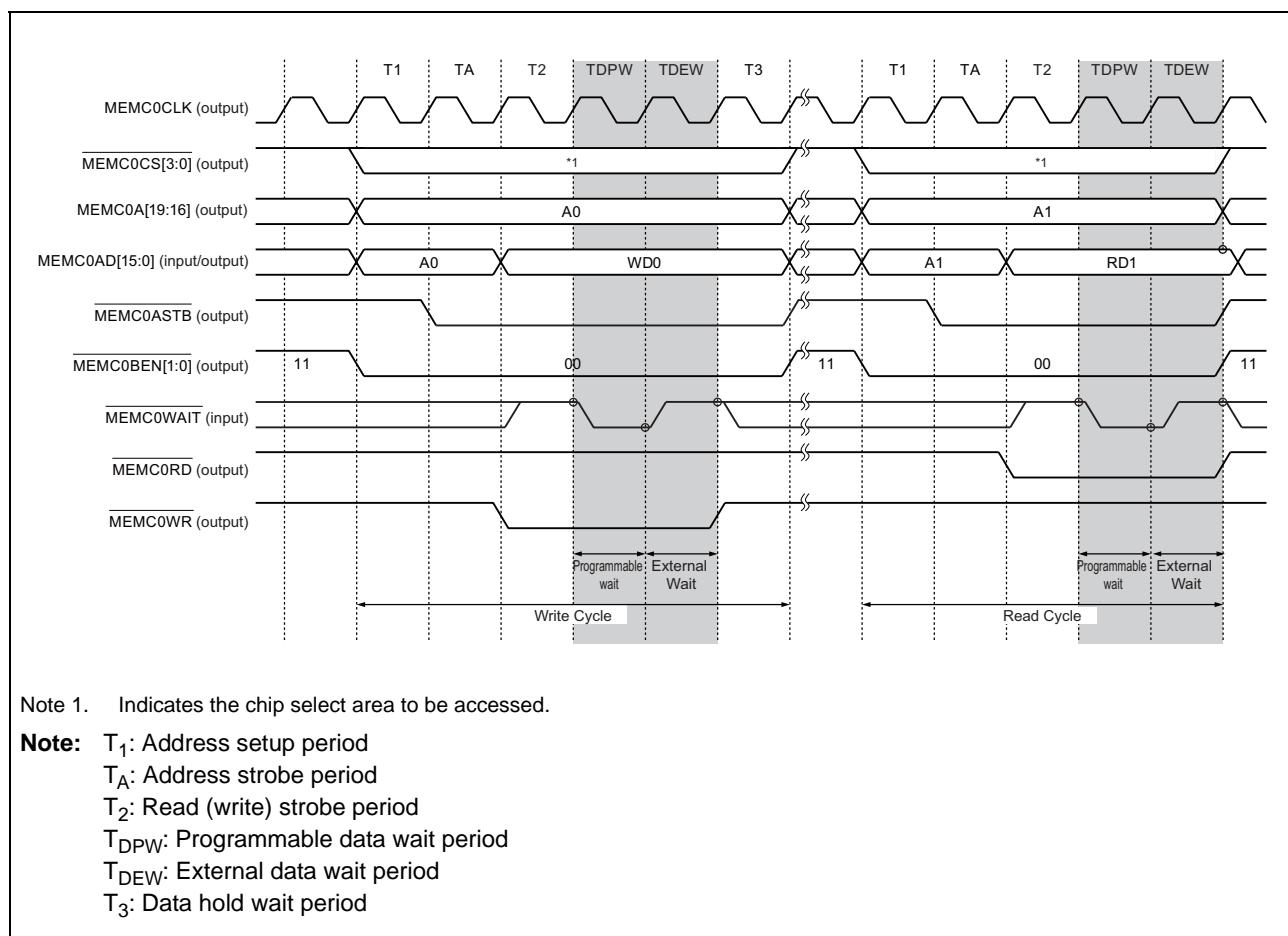


Figure 13.5 Bus Cycles when Inserting Programmable Data Waits and External Waits (while DWC = 1)

### 13.4.2.3 Data Hold Wait Function

This function inserts a wait in the state following the rising edge of the write strobe signal in order to secure the hold time for the data write strobe.

This microcontroller always inserts 1 data hold wait state upon occurrence of a write cycle. This data hold wait can be extended by up to 1 cycle by setting the DHC register, allowing insertion of 2 cycles.

The number of cycles to extend the data hold wait can be set for each chip select area with the DHC register.

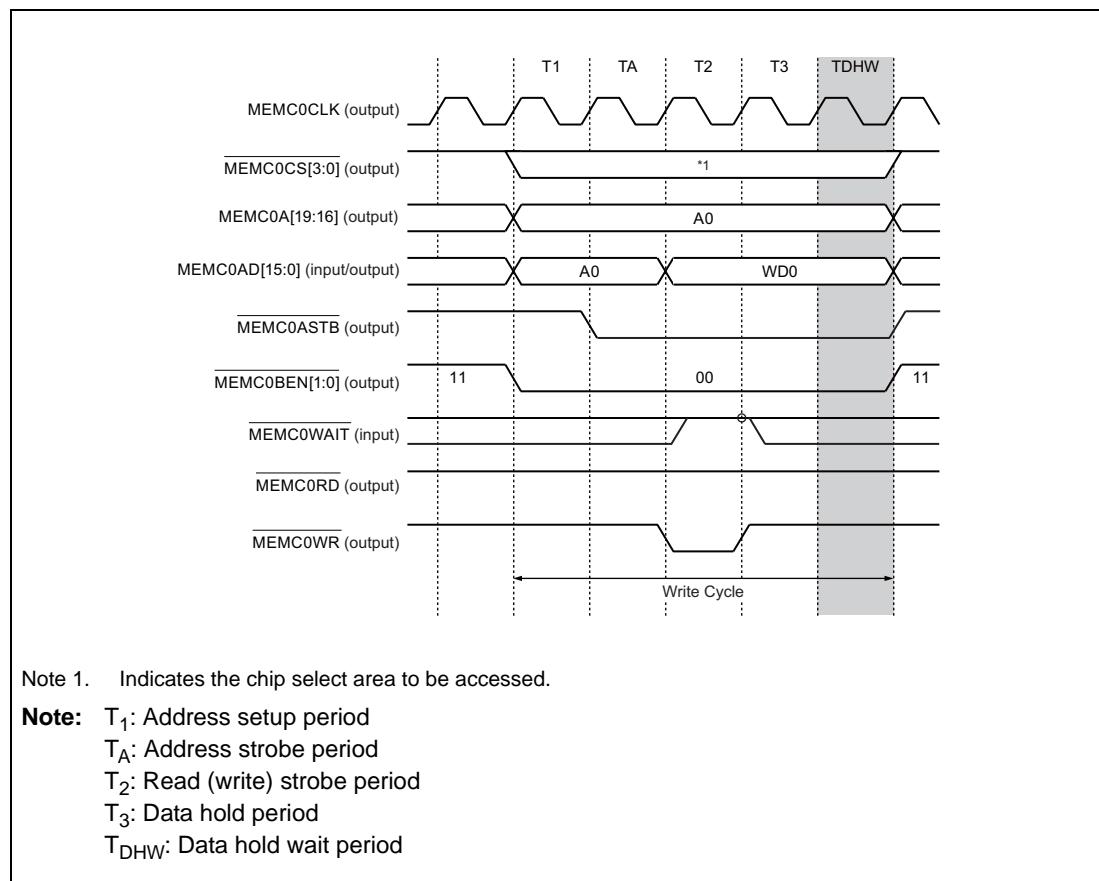


Figure 13.6 Bus Cycles when Inserting Data Hold Waits

#### 13.4.2.4 Address Setup Wait and Address Hold Wait Functions

The address setup wait function inserts a wait before the address transfer state in order to secure the setup time for the address strobe.

The address hold wait function inserts a wait in order to secure the hold time for the address strobe.

Up to 1 cycle can be inserted.

Waits can be set for each chip select area by using the AWC register.

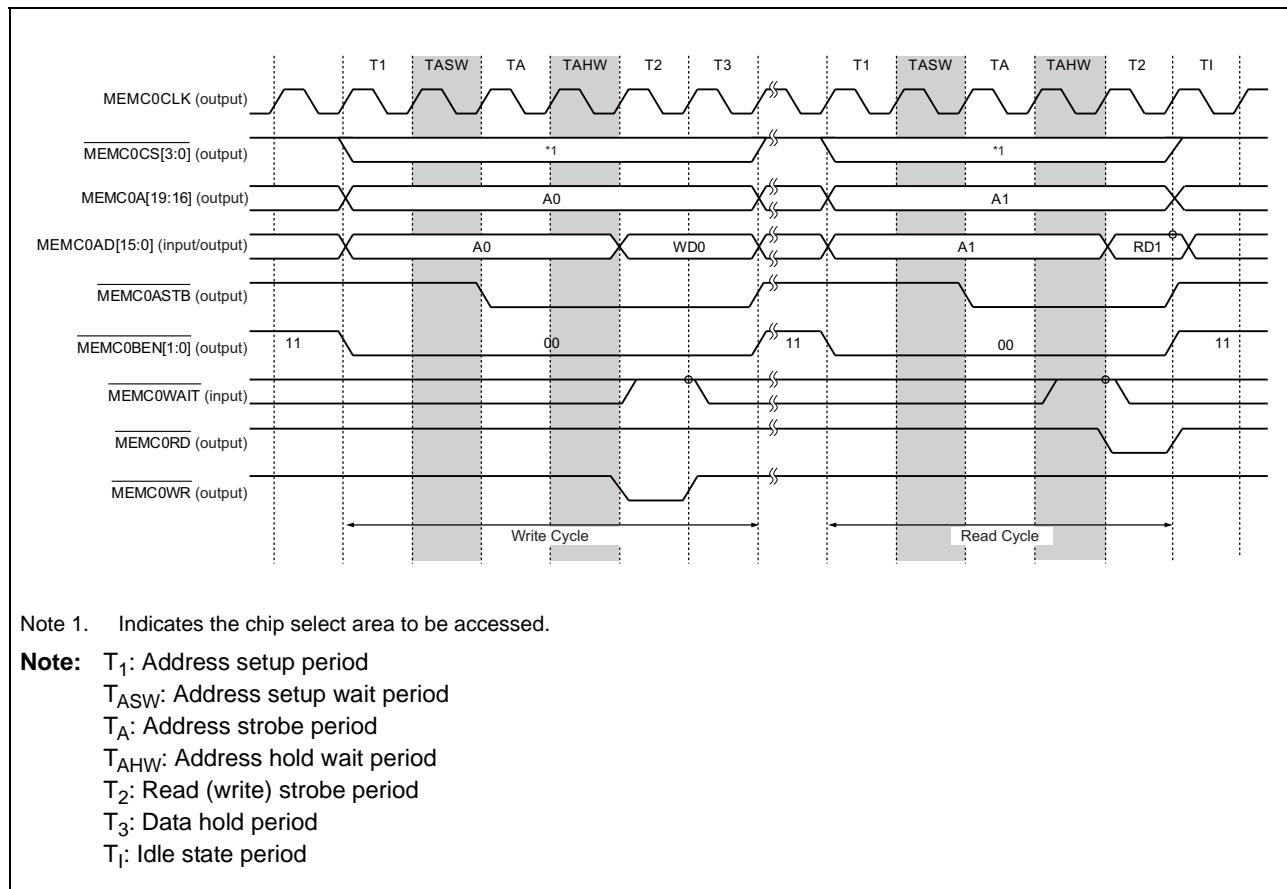


Figure 13.7 Bus Cycles when Inserting Address Setup Waits and Address Hold Waits

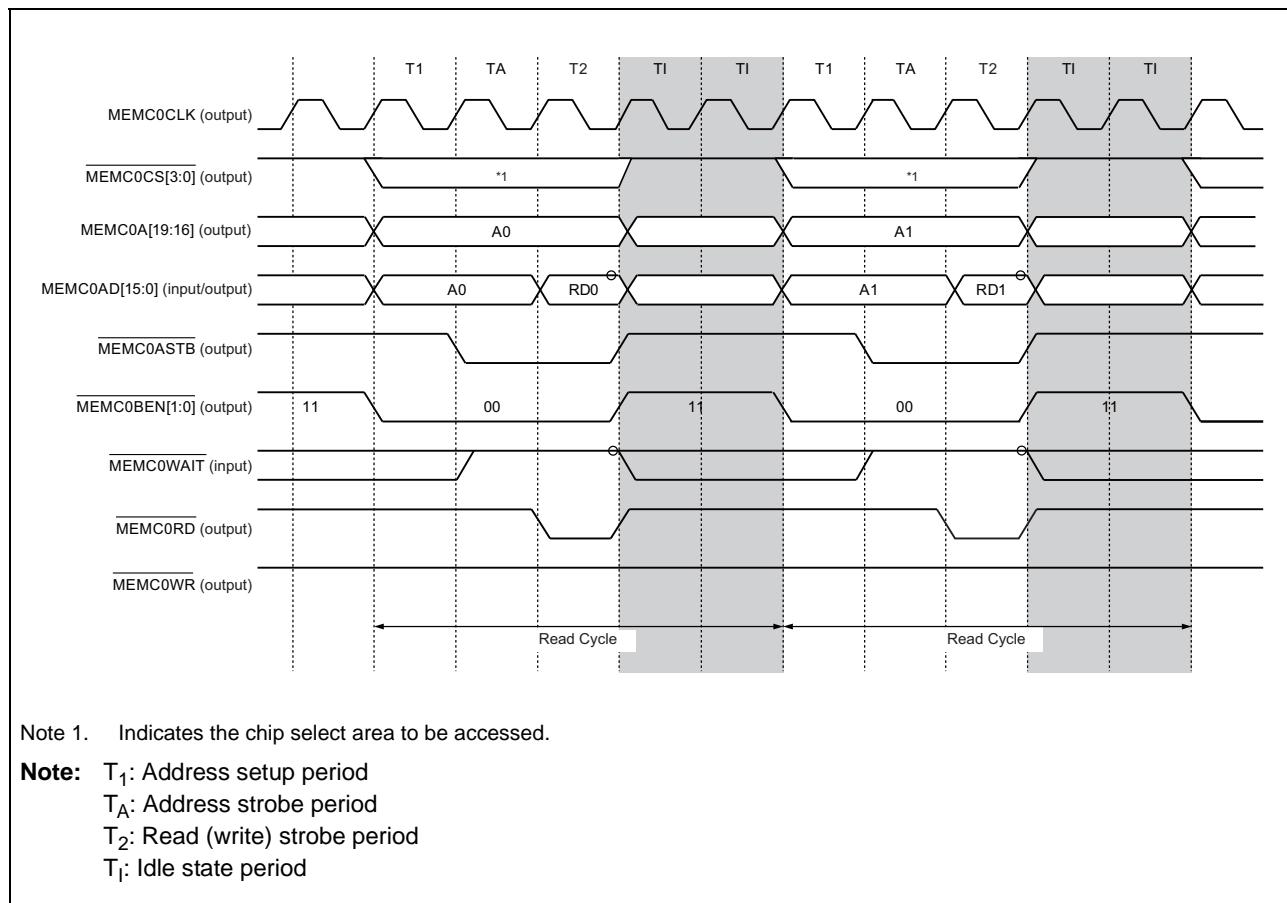
### 13.4.2.5 Idle Insertion Function

This function inserts an idle state after the last state of each cycle in order to prevent bus conflicts between cycles.

This function can be set independently after a read cycle for each chip select area by setting the ICC register.

#### CAUTION

**The interval between the completion of a bus cycle and the generation of the next bus cycle from the CPU (or DMA) is 1 cycle, regardless of the idle cycle setting. Therefore, a 1-cycle interval occurs between bus cycles even if idle states are not inserted.**



Note 1. Indicates the chip select area to be accessed.

**Note:** T<sub>1</sub>: Address setup period

T<sub>A</sub>: Address strobe period

T<sub>2</sub>: Read (write) strobe period

T<sub>I</sub>: Idle state period

Figure 13.8 Bus Cycles when Inserting Idle States

### 13.4.3 Memory Connection Example

#### 13.4.3.1 SRAM Connection Example

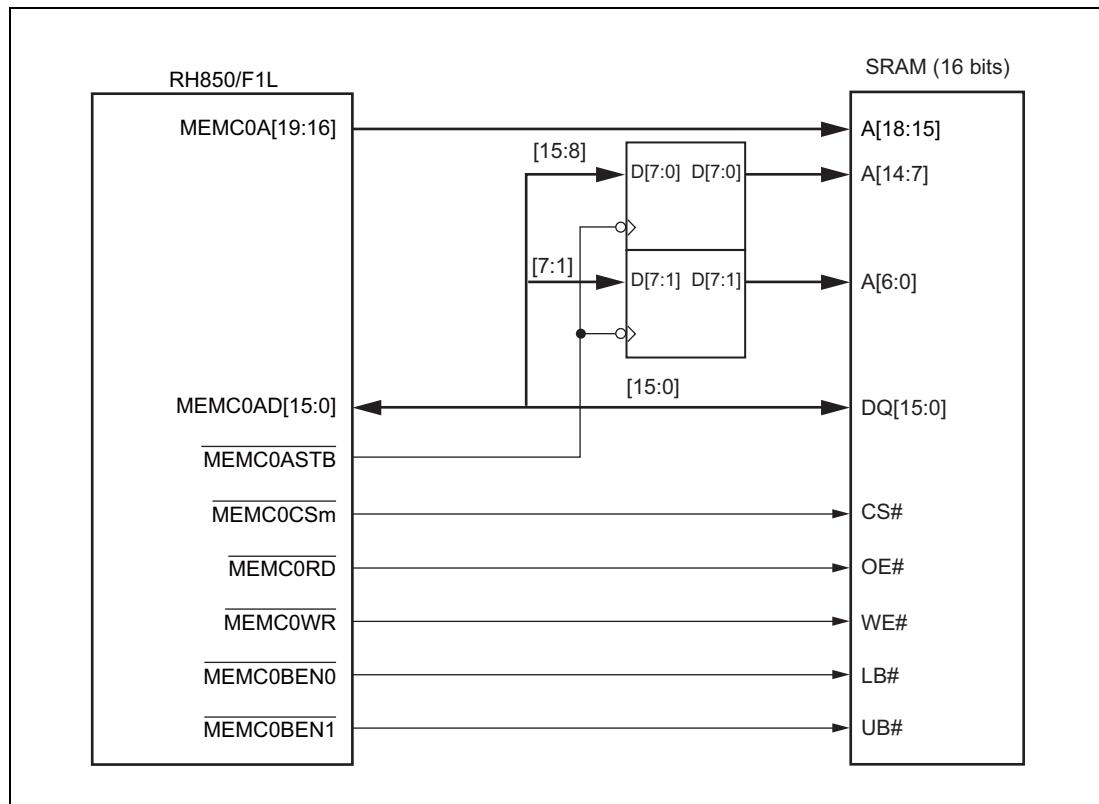


Figure 13.9 SRAM (16 Bits) Connection Example

### 13.4.4 Data Access Flow

The data transfer flow to the external memory differs according to the data width, specified endian, external bus width, and start address.

For misaligned access, the CPU performs the division and coupling of data. Whether the access is misaligned or not depends on the data width and start address.

#### Misaligned

As the accesses listed below are misaligned, the CPU divides the cycles.

- When half-word (16-bit) data is read from/written to an odd address
- When word (32-bit) data is read from/written to an address that is not a multiple of 4

**Table 13.21 Misaligned Access Conditions**

Access Conditions		Cycles Divided by the CPU		
Data Width	Address	1st	2nd	3rd
16 bits	$2n + 1$	8-bit access to $2n + 1$	8-bit access to $2n + 2$	—
32 bits	$4n + 1$	8-bit access to $4n + 1$	16-bit access to $4n + 2$	8-bit access to $4n + 4$
32 bits	$4n + 2$	16-bit access to $4n + 2$	16-bit access to $4n + 4$	—
32 bits	$4n + 3$	8-bit access to $4n + 3$	16-bit access to $4n + 4$	8-bit access to $4n + 6$

The data flows for each condition are shown on the following pages.

### 13.4.4.1 Data Flow for Byte Access (for Reading and Writing)

Table 13.22 Data Flow for Byte Access (Little Endian)

Address to be Accessed	Data Transfer Flow	
	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n		
4n + 1		
4n + 2		
4n + 3		

Note: E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.23 Data Flow for Byte Access (Big Endian)

Address to be Accessed	Data Transfer Flow	
	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n		
4n + 1		
4n + 2		
4n + 3		

**Note:** E: Internal bus

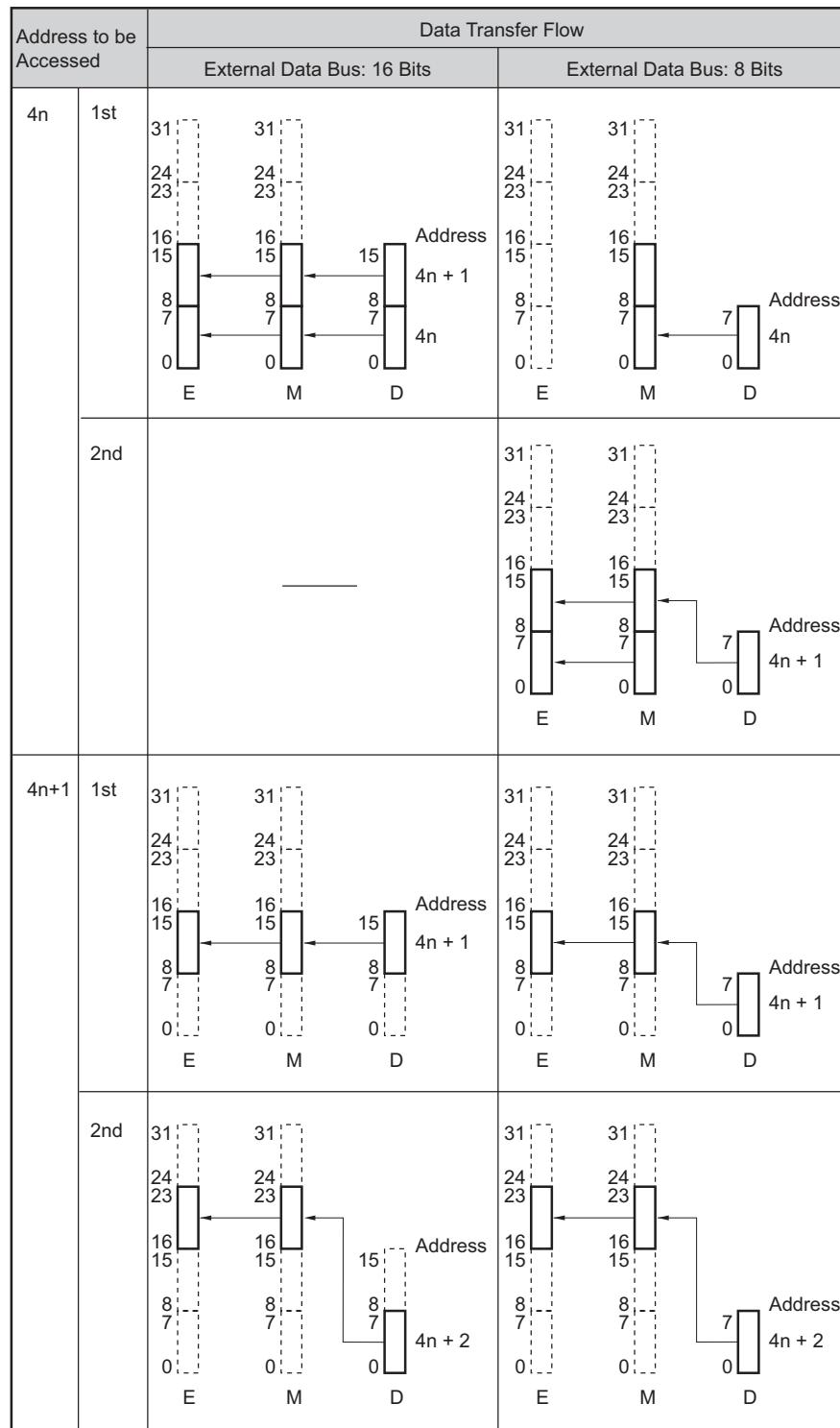
M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

### 13.4.4.2 Data Flow for Half-Word Read Access

Table 13.24 Data Flow for Half-Word Read Access (Little Endian) (1/2)



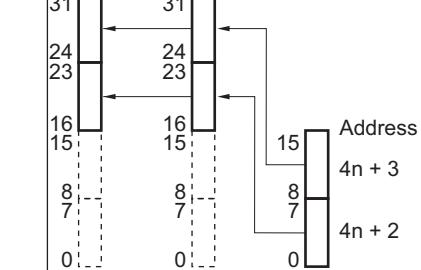
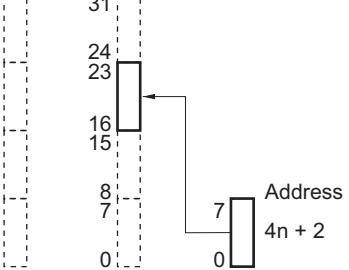
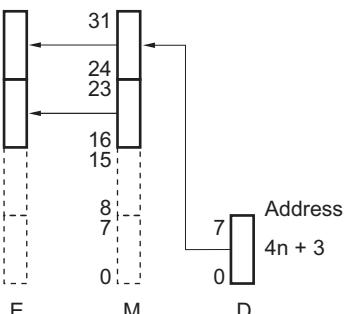
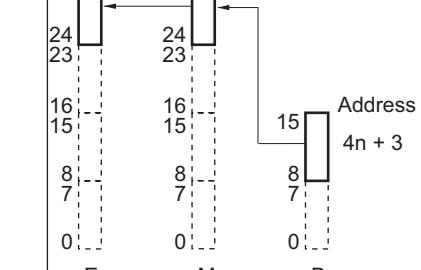
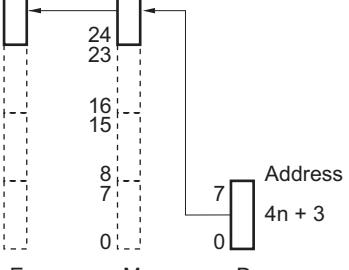
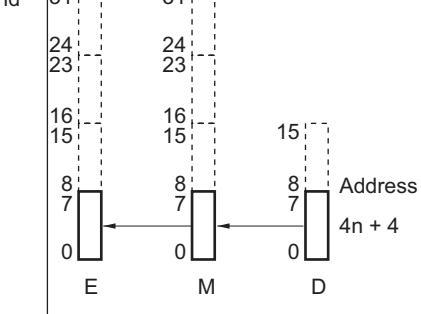
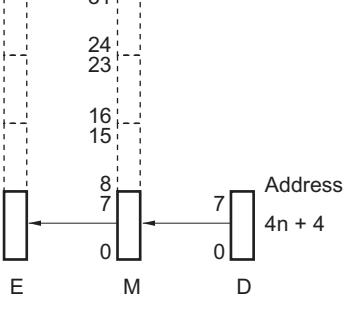
**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.24 Data Flow for Half-Word Read Access (Little Endian) (2/2)

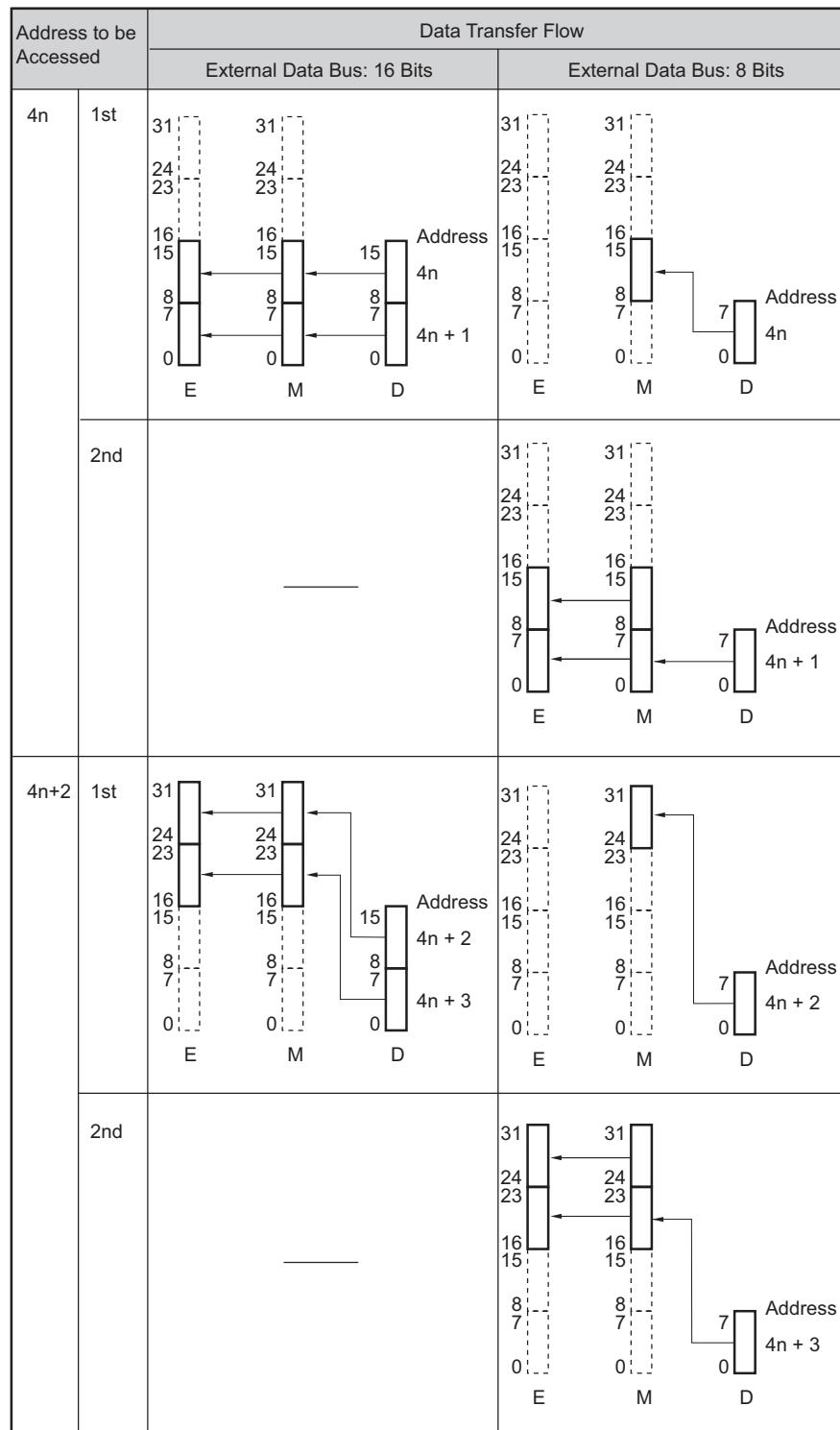
Address to be Accessed		Data Transfer Flow			
		External Data Bus: 16 Bits		External Data Bus: 8 Bits	
4n+2	1st				
	2nd				
4n+3	1st				
	2nd				

Note: E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

**Table 13.25 Data Flow for Half-Word Read Access (Big Endian)****Note 1.** E: Internal bus

M: Memory controller data buffer

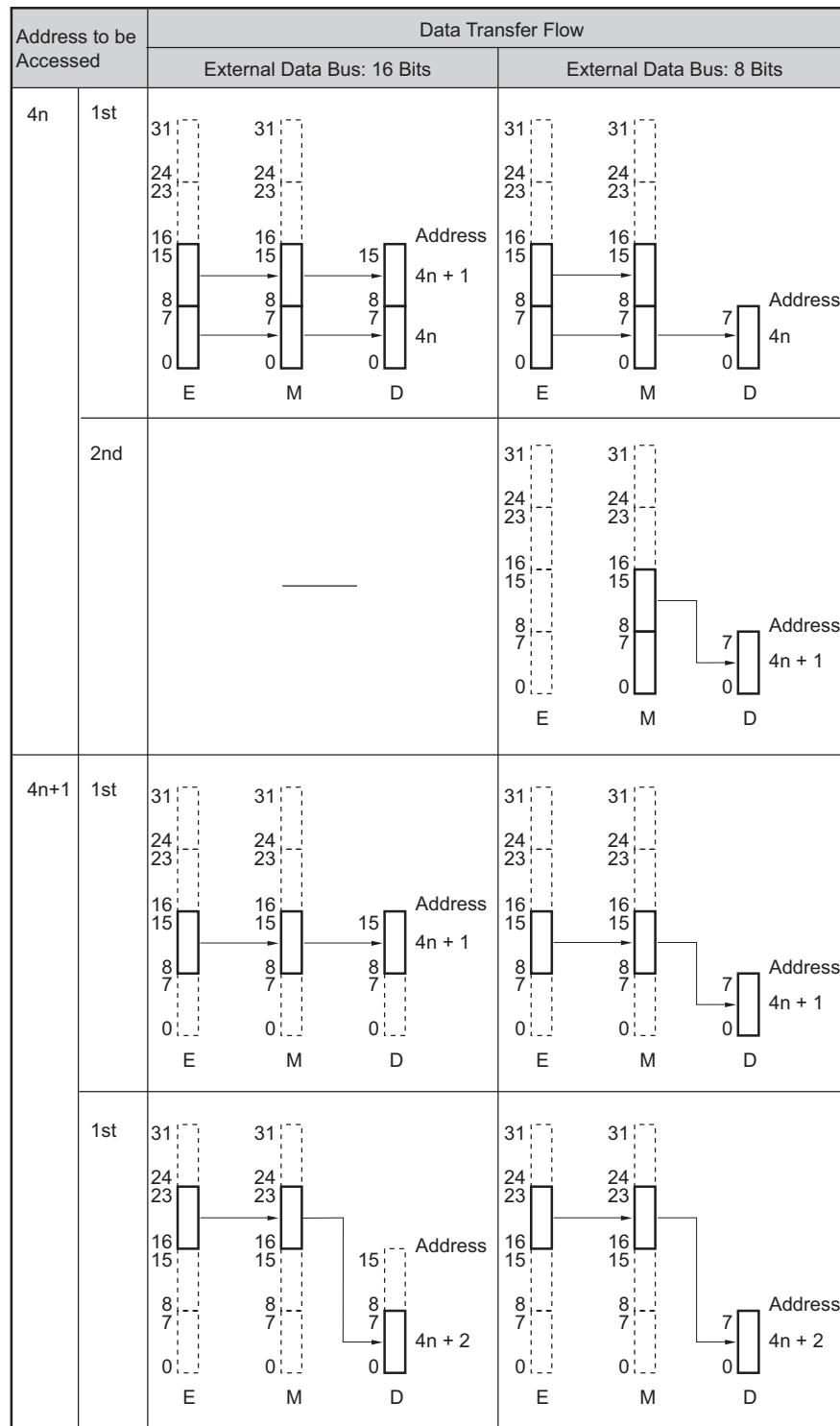
D: External data bus

n = 0, 1, 2, 3, ...

**Note 2.** Accessing an address starting with 4n + 1 or 4n + 3 is prohibited.

### 13.4.4.3 Data Flow for Half-Word Write Access

Table 13.26 Data Flow for Half-Word Write Access (Little Endian) (1/2)



**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.26 Data Flow for Half-Word Write Access (Little Endian) (2/2)

Address to be Accessed		Data Transfer Flow			
		External Data Bus: 16 Bits		External Data Bus: 8 Bits	
4n+2	1st				
	2nd				
4n+3	1st				
	2nd				

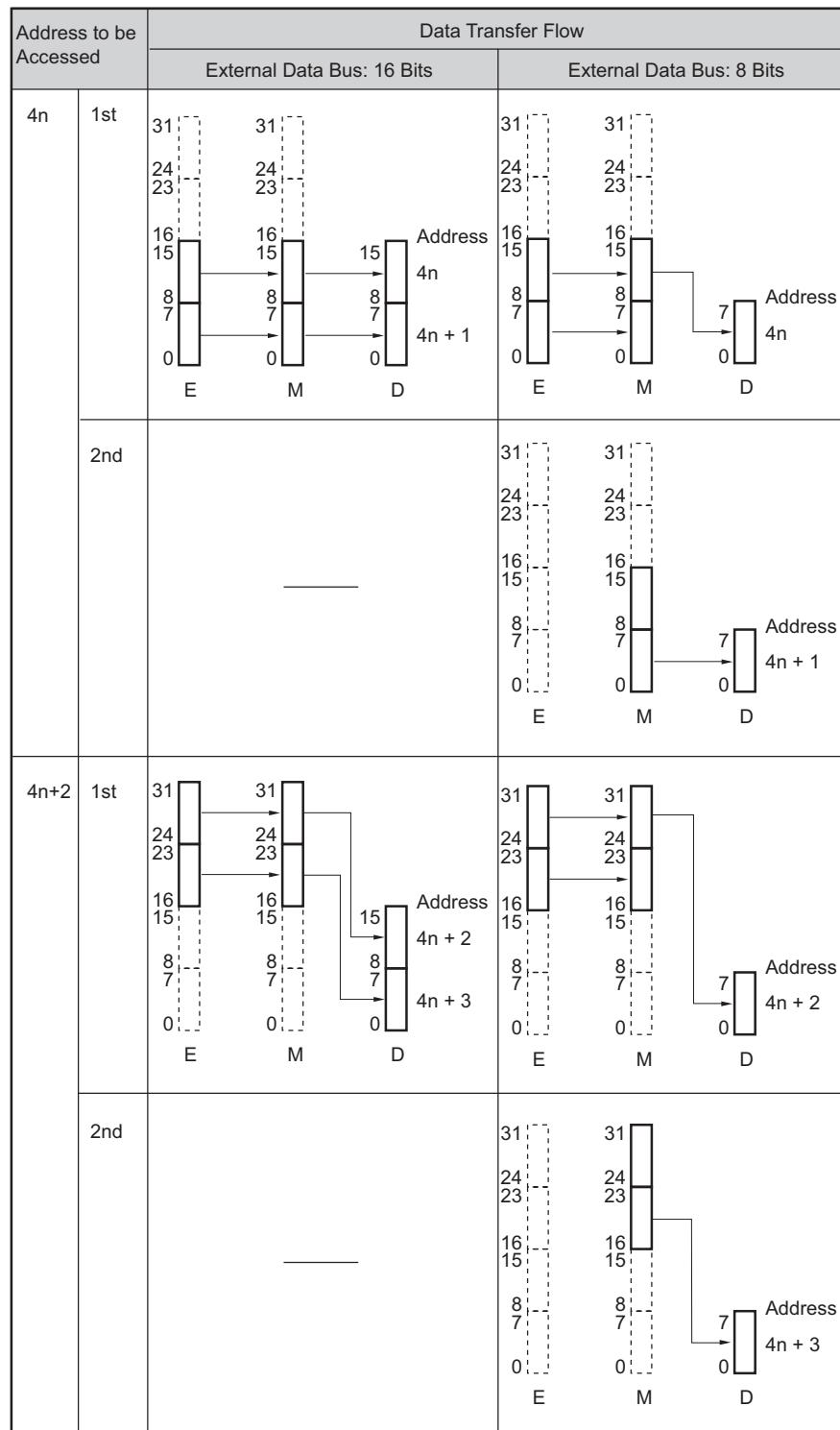
**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.27 Data Flow for Half-Word Write Access (Big Endian)



**Note 1.** E: Internal bus  
 M: Memory controller data buffer  
 D: External data bus  
 $n = 0, 1, 2, 3, \dots$

**Note 2.** Accessing an address starting with  $4n + 1$  or  $4n + 3$  is prohibited.

#### 13.4.4.4 Data Flow for Word Read Access

Table 13.28 Data Flow for Word Read Access (Little Endian) (1/4)

Address to be Accessed		Data Transfer Flow		
		External Data Bus: 16 Bits		External Data Bus: 8 Bits
4n	1st			
	2nd			
	3rd			
	4th			

**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.28 Data Flow for Word Read Access (Little Endian) (2/4)

Address to be Accessed		Data Transfer Flow					
		External Data Bus: 16 Bits			External Data Bus: 8 Bits		
4n+1	1st						
	2nd						
	3rd						
	4th						

**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.28 Data Flow for Word Read Access (Little Endian) (3/4)

Address to be Accessed		Data Transfer Flow		
		External Data Bus: 16 Bits		External Data Bus: 8 Bits
4n+2	1st			
	2nd			
	3rd			
	4th			

**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

 $n = 0, 1, 2, 3, \dots$

Table 13.28 Data Flow for Word Read Access (Little Endian) (4/4)

Address to be Accessed		Data Transfer Flow					
		External Data Bus: 16 Bits			External Data Bus: 8 Bits		
4n+3	1st						
	2nd						
	3rd						
	4th						

**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

 $n = 0, 1, 2, 3, \dots$

Table 13.29 Data Flow for Word Read Access (Big Endian)

Address to be Accessed		Data Transfer Flow			
		External Data Bus: 16 Bits		External Data Bus: 8 Bits	
4n	1st				
	2nd				
	3rd				
	4th				

**Note 1.** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

**Note 2.** Accessing an address starting with 4n + 1, 4n + 2, or 4n + 3 is prohibited.

### 13.4.4.5 Data Flow for Word Write Access

Table 13.30 Data Flow for Word Write Access (Little Endian)(1/4)

Address to be Accessed		Data Transfer Flow		
		External Data Bus: 16 Bits		External Data Bus: 8 Bits
4n	1st			
2nd				
3rd				
4th				

**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.30 Data Flow for Word Write Access (Little Endian) (2/4)

Address to be Accessed		Data Transfer Flow					
		External Data Bus: 16 Bits			External Data Bus: 8 Bits		
4n+1	1st						
	2nd						
	3rd						
	4th						

Note: E: Internal bus

M: Memory controller data buffer

D: External data bus

n = 0, 1, 2, 3, ...

Table 13.30 Data Flow for Word Write Access (Little Endian) (3/4)

Address to be Accessed		Data Transfer Flow		
		External Data Bus: 16 Bits		External Data Bus: 8 Bits
4n+2	1st			
	2nd			
	3rd			
	4th			

**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

 $n = 0, 1, 2, 3, \dots$

Table 13.30 Data Flow for Word Write Access (Little Endian) (4/4)

Address to be Accessed		Data Transfer Flow					
		External Data Bus: 16 Bits			External Data Bus: 8 Bits		
4n+3	1st						
	2nd						
	3rd						
	4th						

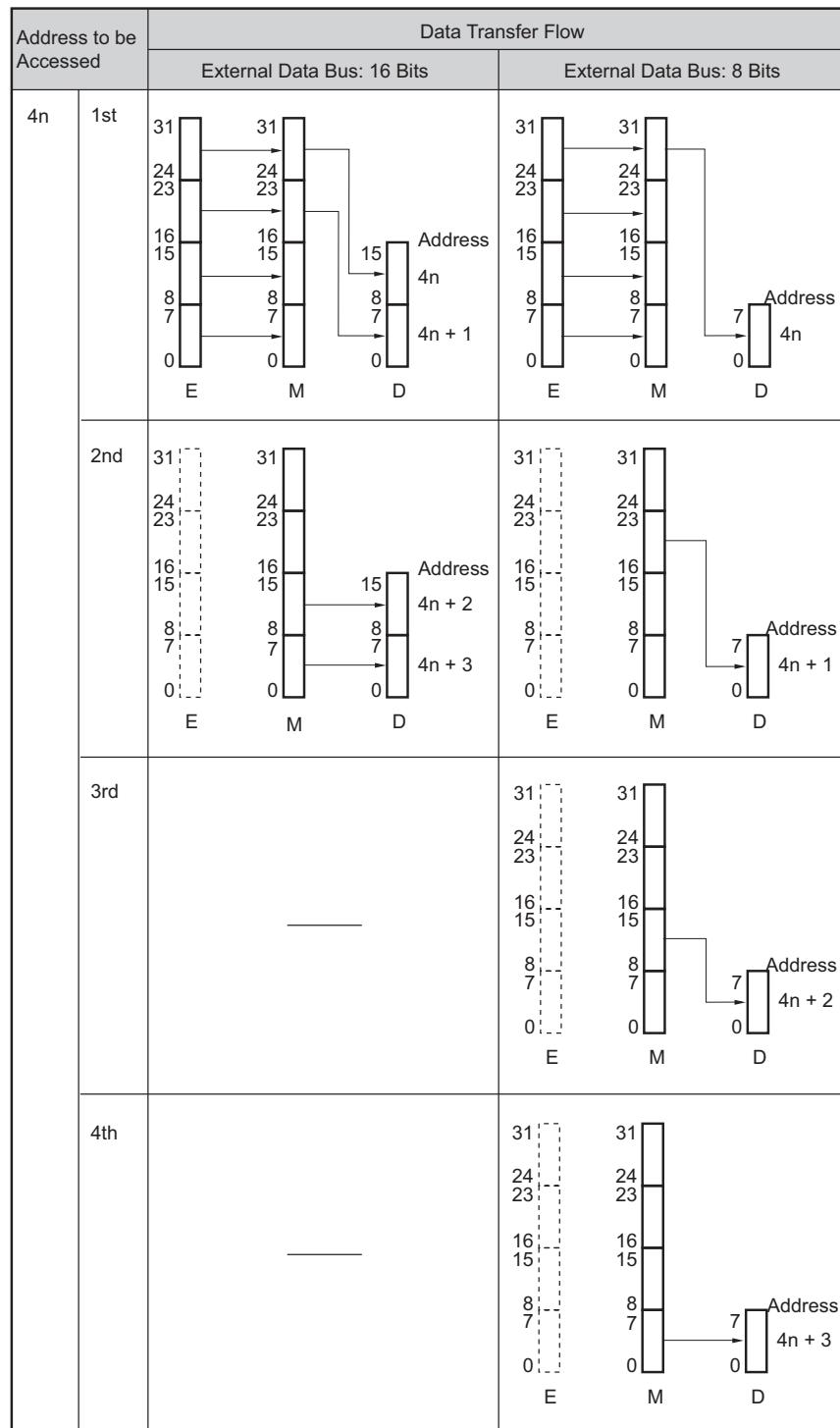
**Note:** E: Internal bus

M: Memory controller data buffer

D: External data bus

 $n = 0, 1, 2, 3, \dots$

Table 13.31 Data Flow for Word Write Access (Big Endian)



**Note 1.** E: Internal bus  
 M: Memory controller data buffer  
 D: External data bus  
 $n = 0, 1, 2, 3, \dots$

**Note 2.** Accessing an address starting with  $4n + 1$ ,  $4n + 2$ , or  $4n + 3$  is prohibited.

## Section 14 Clocked Serial Interface G (CSIG)

This section contains a generic description of the Clocked Serial Interface G (CSIG).

The first part in this section describes the features specific to RH850/F1L such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the CSIG.

### 14.1 Features of RH850/F1L CSIG

#### 14.1.1 Number of Units

This microcontroller has the following number of CSIG units.

Each CSIG unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

**Table 14.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	2	2
Name	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0, 1)	CSIGn (n = 0, 1)

**Table 14.2 Index**

Index	Description
n	Throughout this section, the individual CSIG units are identified by the index “n” (n = 0, 1); for example, CSIGnCTL0 is the CSIGn control register 0.

#### 14.1.2 Register Base Address

CSIG base addresses are listed in the following table.

CSIG register addresses are given as offsets from the base addresses.

**Table 14.3 Register Base Addresses**

Base Address Name	Base Address
<CSIG0_base>	FFDB 0000 <sub>H</sub>
<CSIG1_base>	FFDB 2000 <sub>H</sub>

#### 14.1.3 Clock Supply

The CSIG clock supply is shown in the following table.

**Table 14.4 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
CSIGn	PCLK	CKSCLK_ICSI
	Register access	CKSCLK_ICSI

#### 14.1.4 Interrupt Requests

CSIG interrupt requests are listed in the following table.

**Table 14.5 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>CSIG0</b>			
INTCSIGTIC	Communication status interrupt	19,110	8 (channels 0 to 7)
INTCSIGTIR	Receive status interrupt	20,111	9 (channels 0 to 7)
INTCSIGTIRE	Communication error interrupt	49	—
<b>CSIG1</b>			
INTCSIGTIC	Communication status interrupt	215	45 (channels 8 to 15)
INTCSIGTIR	Receive status interrupt	216	46 (channels 8 to 15)
INTCSIGTIRE	Communication error interrupt	217	—

#### 14.1.5 Reset Sources

CSIG reset sources are listed in the following table. CSIG is initialized by these reset sources.

**Table 14.6 Reset Sources**

Unit Name	Reset Source
CSIGn	All reset sources (ISORES)

### 14.1.6 External Input/Output Signals

External input/output signals of CSIG are listed below.

**Table 14.7 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal Name
<b>CSIG0</b>		
CSIGTSCK	Serial clock signal	CSIG0SC
CSIGTSI	Serial data input signal	CSIG0SI
CSIGTSO	Serial data output signal	CSIG0SO
CSIGTSSI	Slave select input signal	CSIG0SSI
CSIGTRYI	Ready / busy input signal	CSIG0RYI
CSIGTRYO	Ready / busy output signal	CSIG0RYO
<b>CSIG1</b>		
CSIGTSCK	Serial clock signal	CSIG1SC
CSIGTSI	Serial data input signal	CSIG1SI
CSIGTSO	Serial data output signal	CSIG1SO
CSIGTSSI	Slave select input signal	CSIG1SSI
CSIGTRYI	Ready / busy input signal	CSIG1RYI
CSIGTRYO	Ready / busy output signal	CSIG1RYO

Note 1. For the port pins that are used as CSIGnSO and CSIGnSC, set the output driving ability to high (PDSCn\_m = 1).

### 14.1.7 Data Consistency Check

The port and the alternative function for data consistency check of CSIGnSO (CSIGTSO) output are shown in the following table. See **Section 14.5.10, Error Detection** for details on data consistency checking.

**Table 14.8 Port Pins for Data Consistency Checking**

Unit Signal Name	Port Pin Name	Alternative Function
<b>CSIG0</b>		
CSIGTSO	P0_13	ALT_OUT4
	P10_6	ALT_OUT2
<b>CSIG1</b>		
CSIGTSO	P11_9	ALT_OUT1

## 14.2 Overview

### 14.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode or slave mode is selectable.
- Slave select input signal ( $\overline{\text{CSIGTSSI}}$ ) is available.
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
- Clock phases and data phases are selectable.
- Data transfer with MSB first or LSB first is selectable.
- Transfer data length is selectable from 7 to 16 bits in 1-bit units
- Built-in EDL (extended data length) function for transferring more than 16 bits of data.
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, overrun)
- Three different interrupt request signals (INTCSIGTIC, INTCSIGTIR, INTCSIGTIRE)
- Built-in LBM (Loop Back Mode) function for self-test

### 14.2.2 Functional Overview Description

The CSIG uses three signals for communication:

- Transmission clock CSIGTCK (output in master mode, input in slave mode)
- Serial data output signal CSIGTSO
- Serial data input signal CSIGTSI

The CSIGNCTL2 register is used to select whether the CSIG should be operated in master mode or slave mode.

Additional signals can be used for external control and monitoring:

- CSIGTSSI: Slave select input signal
- CSIGTRYO: Ready/busy output signal (handshake signal)
- CSIGTRYI: Ready/busy input signal (handshake signal)

Data transmission is bit-wise and serial and synchronous to the transmission clock.

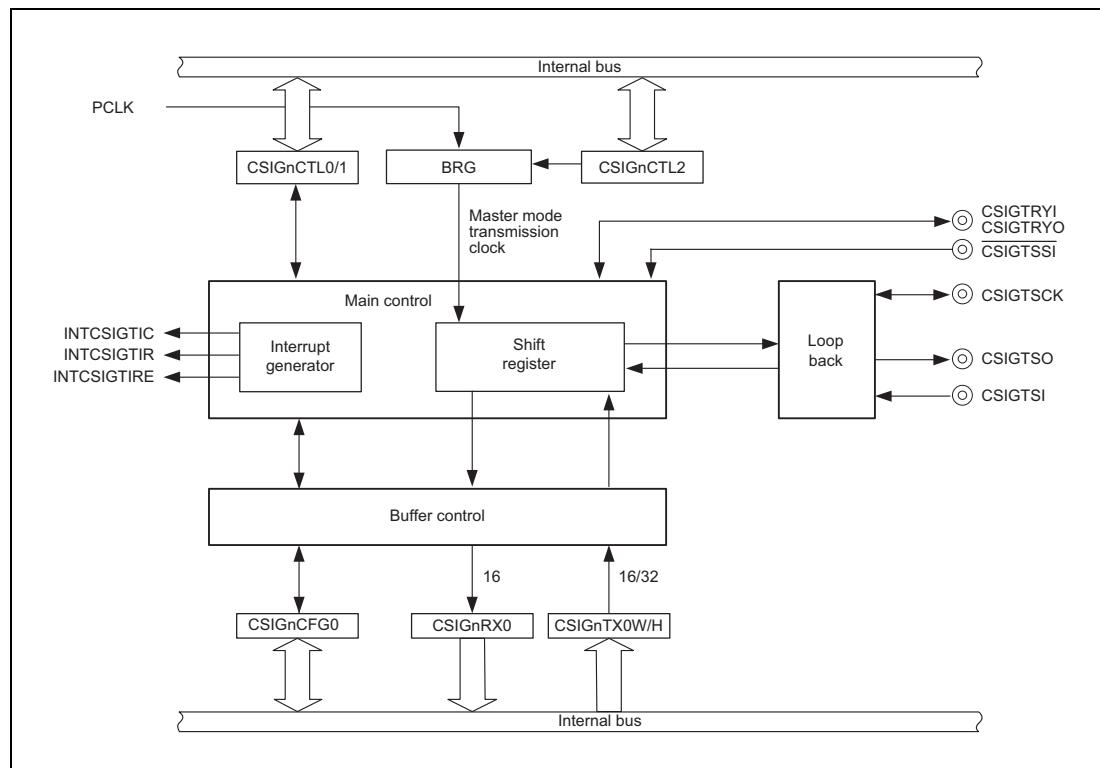
The following table shows the most important registers for setting up CSIG.

**Table 14.9 Main Registers of CSIG**

Register	Function
CSIGnCTL0	Provides and stops operating clock and enables/disables data transmission and data reception.
CSIGnCTL1	Controls options such as interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc.
CSIGnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIGnCFG0	Configures the communication protocol.

### 14.2.3 Block Diagram

The following block diagram shows the main components of the CSIG.



**Figure 14.1 CSIG Block Diagram**

In master mode, the transmission clock CSIGTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

## 14.3 Registers

### 14.3.1 List of Registers

CSIG registers are listed in the following table.

For details on <CSIGn\_base>, see **Section 14.1.2, Register Base Address**.

Table 14.10 List of Registers

Module Name	Register Name	Abbreviation	Address
CSIGn	CSIGn control register 0	CSIGnCTL0	<CSIGn_base> + 0000 <sub>H</sub>
CSIGn	CSIGn control register 1	CSIGnCTL1	<CSIGn_base> + 0010 <sub>H</sub>
CSIGn	CSIGn control register 2	CSIGnCTL2	<CSIGn_base> + 0014 <sub>H</sub>
CSIGn	CSIGn status register 0	CSIGnSTR0	<CSIGn_base> + 0004 <sub>H</sub>
CSIGn	CSIGn status clear register 0	CSIGnSTCR0	<CSIGn_base> + 0008 <sub>H</sub>
CSIGn	CSIGn Receive-only mode control register 0	CSIGnBCTL0	<CSIGn_base> + 1000 <sub>H</sub>
CSIGn	CSIGn configuration register 0	CSIGnCFG0	<CSIGn_base> + 1010 <sub>H</sub>
CSIGn	CSIGn transmission register 0 for word access	CSIGnTX0W	<CSIGn_base> + 1004 <sub>H</sub>
CSIGn	CSIGn transmission register 0 for half word access	CSIGnTX0H	<CSIGn_base> + 1008 <sub>H</sub>
CSIGn	CSIGn reception register 0	CSIGnRX0	<CSIGn_base> + 100C <sub>H</sub>
CSIGn	CSIGn emulation register	CSIGnEMU	<CSIGn_base> + 0018 <sub>H</sub>

### 14.3.2 CSIGNCTL0 — CSIGN Control Register 0

This register controls the operation clock, and enables or disables transmission/reception.

**Access:** This register can be read or written in 8-bit or 1-bit units.

**Address:** <CSIGN\_base> + 0000H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	CSIGNPWR	CSIGNTXE	CSIGNRXE	—	—	—	—	CSIGNMBS
Value after reset	0	0	0	0	0	0	0	0

Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 14.11 CSIGNCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIGNPWR	Controls operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIGNPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. Clock supply to internal circuits stops. If CSIGNPWR is cleared during communication, ongoing communication is aborted. In this case, communication setting must be started from the beginning.
6	CSIGNTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIGNRXE	Enables/disables reception. 0: Reception disabled 1: Reception enabled
4 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CSIGNMBS	This bit must be set to 1 (the value after reset is 0).

#### CAUTION

When configuring this register, see Table 14.24, List of Cautions when Configuring the Registers.

### 14.3.3 CSIGNCTL1 — CSIGN Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake function, and slave select function.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIGN\_base> + 0010H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIGN CKR	CSIGN SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGN EDLE	—	CSIGN DCS	—	CSIGN LBM	CSIGN SIT	CSIGN HSE	CSIGN SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 14.12 CSIGNCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	CSIGNCKR	CSIGTSCK clock inversion function 0: Default level of CSIGTSCK is high. 1: Default level of CSIGTSCK is low. The CSIGNCKR bit is used in combination with the CSIGNCFG0.CSIGNDAP bit. For details, see <b>Section 14.3.8, CSIGNCFG0 — CSIGN Configuration Register 0</b> .
16	CSIGNSLIT	Selects the timing of interrupt INTCSIGTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt generation when CSIGNTX0W/H is empty and available for storing the next data. For details, see <b>14.4.2, INTCSIGTIC (Communication Status Interrupt)</b>
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	CSIGNEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, see <b>14.5.5.2, Data Length Selection with Extended Data Length</b> .
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSIGNDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, see <b>14.5.10.1, Data Consistency Check</b> .
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIGNLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated Loop-back mode can be set only in master mode. Set this bit to 0 in slave mode. For details, see <b>Section 14.5.9, Loop-Back Mode</b> .

**Table 14.12 CSIGnCTL1 Register Contents (2/2)**

Bit Position	Bit Name	Function
2	CSIGnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, see <b>Section 14.4.1, Interrupt Delay</b> .
1	CSIGnHSE	Enables/disables handshake function. 0: Handshake function disabled 1: Handshake function enabled For details, see <b>Section 14.5.8, Handshake Function</b> .
0	CSIGnSSE	Enables/disables slave select function. 0: Input signal CSIGTSSI disabled. 1: Input signal CSIGTSSI enabled. If the slave select function is not used, this bit must be set to 0 (see also <b>Section 14.5.2, Master/Slave Connections</b> ).

Details about CSIGnCTL1.CSIGnSSE:

**Table 14.13 Operation of the Slave Select Function during Reception**

CSIGnCTL0. CSIGnRXE	CSIGnCTL1. CSIGnSSE	CSIGTSSI	Receive Operation
0	—	—	Reception disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

**Table 14.14 Operation of the Slave Select Function during Transmission**

CSIGnCTL0. CSIGnTXE	CSIGnCTL1. CSIGnSSE	CSIGTSSI	Transmit Operation
0	—	—	Transmission disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

#### CAUTION

When configuring this register, see **Table 14.24, List of Cautions when Configuring the Registers**.

#### 14.3.4 CSIGNCTL2 — CSIGN Control Register 2

This register selects the communication clock.

**Access:** This register can be read or written in 16-bit units.

**Address:** <CSIGN\_base> + 0014H

**Value after reset:** E000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	CSIGNPRS[2:0]		—	CSIGNBRS[11:0]													
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R	R/W												

Table 14.15 CSIGNCTL2 Register Contents

Bit Position	Bit Name	Function																																				
15 to 13	CSIGNPRS [2:0]	Selects the value of the prescaler.																																				
		<table border="1"> <thead> <tr> <th>CSIGN PRS2</th> <th>CSIGN PRS1</th> <th>CSIGN PRS0</th> <th>Prescaler Output (PRSOUT)</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>PCLK (master mode)</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>PCLK / 2 (master mode)</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>PCLK / 4 (master mode)</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>PCLK / 8 (master mode)</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>PCLK / 16 (master mode)</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>PCLK / 32 (master mode)</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>PCLK / 64 (master mode)</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>External clock via CSIGTSCK (slave mode)</td></tr> </tbody> </table>	CSIGN PRS2	CSIGN PRS1	CSIGN PRS0	Prescaler Output (PRSOUT)	0	0	0	PCLK (master mode)	0	0	1	PCLK / 2 (master mode)	0	1	0	PCLK / 4 (master mode)	0	1	1	PCLK / 8 (master mode)	1	0	0	PCLK / 16 (master mode)	1	0	1	PCLK / 32 (master mode)	1	1	0	PCLK / 64 (master mode)	1	1	1	External clock via CSIGTSCK (slave mode)
CSIGN PRS2	CSIGN PRS1	CSIGN PRS0	Prescaler Output (PRSOUT)																																			
0	0	0	PCLK (master mode)																																			
0	0	1	PCLK / 2 (master mode)																																			
0	1	0	PCLK / 4 (master mode)																																			
0	1	1	PCLK / 8 (master mode)																																			
1	0	0	PCLK / 16 (master mode)																																			
1	0	1	PCLK / 32 (master mode)																																			
1	1	0	PCLK / 64 (master mode)																																			
1	1	1	External clock via CSIGTSCK (slave mode)																																			
12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																				
11 to 0	CSIGNBRS [11:0]	Selects the transfer clock frequency. Settings of the CSIGNBRS[11:0] bits are valid only in master mode. They are ignored in slave mode.																																				
		<table border="1"> <thead> <tr> <th>CSIGNBRS [11:0]</th> <th>Transfer Clock Frequency of CSIGTSCK</th> </tr> </thead> <tbody> <tr><td>0</td><td>BRG is stopped</td></tr> <tr><td>1</td><td>PCLK / (2<sup>a</sup> × 1 × 2)</td></tr> <tr><td>2</td><td>PCLK / (2<sup>a</sup> × 2 × 2)</td></tr> <tr><td>3</td><td>PCLK / (2<sup>a</sup> × 3 × 2)</td></tr> <tr><td>4</td><td>PCLK / (2<sup>a</sup> × 4 × 2)</td></tr> <tr><td>...</td><td>...</td></tr> <tr><td>4095</td><td>PCLK / (2<sup>a</sup> × 4095 × 2)</td></tr> </tbody> </table>	CSIGNBRS [11:0]	Transfer Clock Frequency of CSIGTSCK	0	BRG is stopped	1	PCLK / (2 <sup>a</sup> × 1 × 2)	2	PCLK / (2 <sup>a</sup> × 2 × 2)	3	PCLK / (2 <sup>a</sup> × 3 × 2)	4	PCLK / (2 <sup>a</sup> × 4 × 2)	...	...	4095	PCLK / (2 <sup>a</sup> × 4095 × 2)																				
CSIGNBRS [11:0]	Transfer Clock Frequency of CSIGTSCK																																					
0	BRG is stopped																																					
1	PCLK / (2 <sup>a</sup> × 1 × 2)																																					
2	PCLK / (2 <sup>a</sup> × 2 × 2)																																					
3	PCLK / (2 <sup>a</sup> × 3 × 2)																																					
4	PCLK / (2 <sup>a</sup> × 4 × 2)																																					
...	...																																					
4095	PCLK / (2 <sup>a</sup> × 4095 × 2)																																					

**Note:** a = 0 to 6 (value set by CSIGNPRS[2:0])

#### CAUTION

When configuring this register, see **Table 14.24, List of Cautions when Configuring the Registers.**

### 14.3.5 CSIGNSTR0 — CSIGN Status Register 0

This register indicates the status of the CSIG.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <CSIGN\_base> + 0004H

**Value after reset:** 0000 0010H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.16 CSIGNSTR0 Register Contents (1/2)

Bit Position	Bit Name	Function												
31 to 8	Reserved	When read, the value after reset is returned.												
7	CSIGNTSF	<p>Transfer Status Flag            0: Idle state            1: Communication is in progress or being prepared            The timing to set or clear this bit is as follows:</p> <table border="1"> <thead> <tr> <th>Master Mode</th> <th>Timing to Set</th> <th>Timing to Clear</th> </tr> </thead> <tbody> <tr> <td>Tx-only mode</td> <td>Writing to transmission register</td> <td>Within a half clock cycle from the last serial clock edge</td> </tr> <tr> <td>Tx/Rx mode</td> <td></td> <td></td> </tr> <tr> <td>Rx-only mode</td> <td>Reading from reception register</td> <td></td> </tr> </tbody> </table>	Master Mode	Timing to Set	Timing to Clear	Tx-only mode	Writing to transmission register	Within a half clock cycle from the last serial clock edge	Tx/Rx mode			Rx-only mode	Reading from reception register	
Master Mode	Timing to Set	Timing to Clear												
Tx-only mode	Writing to transmission register	Within a half clock cycle from the last serial clock edge												
Tx/Rx mode														
Rx-only mode	Reading from reception register													
6 to 4	Reserved	When read, the value after reset is returned.												
3	CSIGNDCE	<p>Data Consistency Check Error Flag            0: No data consistency check error detected            1: Data consistency check error detected            This bit is cleared by writing 1 to CSIGNSTCR0.CSIGNDCEC.            When setting to 1 due to data consistency error detection and clearing to 0 by CSIGNSTCR0.CSIGNDCEC occur simultaneously, setting to 1 due to data consistency error detection takes precedence.            This bit is initialized when CSIGNCTL0.CSIGNPWR changes from 0 to 1, or from 1 to 0.</p>												
2	Reserved	When read, the value after reset is returned.												

**Table 14.16 CSIGNSTR0 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
1	CSIGNPE	<p>Parity Error Flag 0: No parity error detected 1: Parity error detected</p> <p>This bit is cleared by writing 1 to CSIGNSTCR0.CSIGNPEC. When setting to 1 due to parity error detection and clearing to 0 by writing to CSIGNSTCR0.CSIGNPEC occur simultaneously, setting to 1 due to parity error detection takes precedence.</p> <p>This bit is initialized when CSIGNCTL0.CSIGNPWR changes from 0 to 1, or from 1 to 0.</p>
0	CSIGNOVE	<p>Overrun Error Flag 0: No overrun error detected 1: Overrun error detected</p> <p>This bit is cleared by writing 1 to CSIGNSTCR0.CSIGNOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIGNSTCR0.CSIGNOVEC occur simultaneously, setting to 1 due to overrun error detection takes precedence.</p> <p>This bit is initialized by the value of CSIGNCTL0.CSIGNPWR changes from 0 to 1, or from 1 to 0.</p>

**CAUTION**

When configuring this register, see **Table 14.24, List of Cautions when Configuring the Registers.**

### 14.3.6 CSIGnSTCR0 — CSIGn Status Clear Register 0

This register clears the status flags of the CSIGnSTR0 status register.

**Access:** This register can be read or written in 16-bit units.  
When read, the value  $0000_H$  is always returned.

**Address:** <CSIGn\_base> +  $0008_H$

**Value after reset:**  $0000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn DCEC	—	CSIGn PEC	CSIGn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 14.17 CSIGnSTCR0 Register Contents

Bit Position	Bit Name	Function
15 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIGnDCEC	Controls the data consistency error flag clear command. 0: No operation. Read value is always 0. 1: Clears the data consistency check error flag (CSIGnSTR0.CSIGnDCE).
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIGnPEC	Controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clears the parity error flag (CSIGnSTR0.CSIGnPCE).
0	CSIGnOVEC	Controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clears the overrun error flag (CSIGnSTR0.CSIGnOVE).

### 14.3.7 CSIGnBCTL0 — CSIGn Rx-Only Mode Control Register 0

This register enables/disables the data transfer in Rx-only mode.

**Access:** This register can be read or written in 8-bit or 1-bit units.

**Address:** <CSIGn\_base> + 1000H

**Value after reset:** 01H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CSIGnSCE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Table 14.18 CSIGnBCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CSIGnSCE	Disables/enables the start of the next data reception by reading CSIGnRX0. 0: Next reception disabled 1: Next reception enabled For details, see <b>Section 14.5.4.2, Receive-Only Mode.</b>

#### CAUTION

When configuring this register, see **Table 14.24, List of Cautions when Configuring the Registers.**

### 14.3.8 CSIGNCFG0 — CSIGN Configuration Register 0

This register configures the communication protocols such as data length, parity, transfer direction, clock phase, and data phase.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIGN\_base> + 1010H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGNPS[1:0]		CSIGNDLS[3:0]		—	—	—	—	—	—	CSIGN DIR	—	CSIGN DAP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.19 CSIGNCFG0 Register Contents (1/2)

Bit Position	Bit Name	Function				
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.				
29, 28	CSIGNPS[1:0]	Specifies parity.				
		CSIGN PS1	CSIGN PS0	Transmission	Reception	
		0	0	Does not transmit any parity bit.	Does not wait for reception of the parity bit.	
		0	1	Adds a parity bit fixed to 0.	Waits for reception of the parity bit but does not evaluate it.	
		1	0	Adds the odd parity bit.	Waits for the odd parity bit.	
		1	1	Adds the even parity bit.	Waits for the even parity bit.	
27 to 24	CSIGNDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits				
		<b>CAUTION</b>				
		Do not set bits CSIGNCFG0.CSIGNDLS[3:0] to values 1 to 6 when the extended data length function is disabled with bit CSIGNCTL1.CSIGNEDLE set to 0. Transmitting two consecutive data with a data length of less than 7 bits is prohibited.				
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.				
18	CSIGNDIR	Selects the serial data direction. 0: Data is transmitted/received with MSB first 1: Data is transmitted/received with LSB first				
17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.				

Table 14.19 CSIGnCFG0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	CSIGnDAP	Data Phase Selection Used in conjunction with CSIGnCTL1.CSIGnCKR bit to select the data phase. Refer to the following table for the selectable clock phases and data phases.
		<b>CSIGnCTL</b> <b>1.CSIGnC</b> <b>KR</b> <b>CSIGn</b> <b>DAP</b>
		<b>Clock and Data Phase Selection</b>
		0      0
		0      1
		1      0
		1      1
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

**CAUTION**

When configuring this register, see **Table 14.24, List of Cautions when Configuring the Registers.**

### 14.3.9 CSIGnTX0W — CSIGn Transmission Register 0 for Word Access

This register stores the transmission data and specifies the extended data length.

<b>Access:</b>	This register can be read or written in 32-bit units.															
<b>Address:</b>	<CSIGn_base> + 1004H															
<b>Value after reset:</b>	0000 0000H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGn EDL	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															CSIGnTX[15:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.20 CSIGnTX0W Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	CSIGnEDL	Specifies the extended data length. 0: Normal operation 1: Extended data length enabled The associated data is transmitted as 16-bit data. This bit can only be set if CSIGnCTL1.CSIGnEDLE = 1.
28 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CSIGnTX[15:0]	Data to be transmitted

#### CAUTION

When configuring this register, see Table 14.24, List of Cautions when Configuring the Registers.

### 14.3.10 CSIGnTX0H — CSIGn Transmission Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of CSIGnTX0W register.

The settings specified by the upper 16 bits of CSIGnTX0W are applied to the transmission.

**Access:** This register can be read or written in 16-bit units.

**Address:** <CSIGn\_base> + 1008<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnTX[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.21 CSIGnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnTX[15:0]	Data to be transmitted

**CAUTION**

When configuring this register, see Table 14.24, List of Cautions when Configuring the Registers.

### 14.3.11 CSIGnRX0 — CSIGn Reception Register 0

This register stores the received data.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <CSIGn\_base> + 100C<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnRX[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.22 CSIGnRX0 Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnRX[15:0]	Received Data These bits are initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When reading, the values of these bits must be read at least 1 clock before the generation of CSIGTIR interrupt.

**CAUTION**

When configuring this register, see Table 14.24, List of Cautions when Configuring the Registers.

### 14.3.12 CSIGNEMU — CSIGN Emulation Register

This register controls operation by SVSTOP.

**Access:** This register can be read or written in 8-bit or 1-bit units.  
Write to this register when EPC.SVSTOP = 0.

**Address:** <CSIGN\_base> + 0018<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	CSIGNSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 14.23 CSIGNEMU Register Contents

Bit Position	Bit Name	Function
7	CSIGNSVSDIS	Selects whether to continue or stop transmit/receive operation during debugging. <ul style="list-style-type: none"> <li>When the EPC.SVSTOP bit is set to 0: Continues transmit/receive operation regardless of the setting of this bit.</li> <li>When the EPC.SVSTOP bit is set to 1: 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.</li> </ul>
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

#### CAUTION

When configuring this register, see Table 14.24, List of Cautions when Configuring the Registers.

### 14.3.13 List of Cautions

Table 14.24 List of Cautions when Configuring the Registers

Register Name	Bit Name	Cautions
CSIGnCTL0	CSIGnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIGnCTL0	CSIGnTXE CSIGnRXE	Do not modify any of these bits while CSIGnCTL0.CSIGnPWR = 0. (These bits can be modified simultaneously with the CSIGnCTL0.CSIGnPWR bit.) Do not modify these bits while CSIGnSTR0.CSIGnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIGnCTL0	CSIGnMBS	When writing, be sure to set to this bit to 1. (The value after reset is "0".) This bit must be modified simultaneously with CSIGnCTL0.CSIGnPWR bit.
CSIGnCTL1	CSIGnCKR	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnSLIT CSIGnEDLE CSIGnDCS CSIGnHSE	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnLBM	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of this bit is prohibited in slave mode.
CSIGnCTL1	CSIGnSSE	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIGnCTL1	CSIGnSIT	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIGnCTL2	CSIGnPRS[2:0] CSIGnBRS[11:0]	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of the maximum transfer clock frequency is as follows. <ul style="list-style-type: none"> <li>• Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)</li> <li>• Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)</li> </ul>
CSIGnSTR0	CSIGnTSF	Writing to this bit is prohibited, and only reading is permitted.
CSIGnSTR0	CSIGnDCE CSIGnPE CSIGnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1 → 0.
CSIGnBCTL0	CSIGnSCE	Write to this bit before CSIGnRX0 is read. Fix the CSIGnSCE bit to 0 when the transfer mode is transmit mode or transmit/receive mode.
CSIGnCFG0	CSIGnPS[1:0] CSIGnDLS[3:0] CSIGnDIR CSIGnDAP	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnTX0W	CSIGnEDL	This bit is valid only when CSIGnCTL1.CSIGnEDLE = 1.
CSIGnTX0W CSIGnTX0H		Write access to these bits are prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnRX0		These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1 → 0. Read access to this bit is prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnEMU	CSIGnSVSDIS	Modification of this bit is prohibited while SVSTOP = 1.

## 14.4 Interrupt Sources

CSIG can generate the following interrupts:

- INTCSIGTIC (communication status interrupt)
- INTCSIGTIR (reception status interrupt)
- INTCSIGTIRE (communication error interrupt)

### 14.4.1 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half cycle of the transmission clock CSIGTSCK. This function is not available in slave mode.

The delay is specified by setting CSIGnCTL1.CSIGnSIT = 1. (The setting of the CSIGnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIGnCTL1.CSIGnSIT = 1 (interrupt delay enabled), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000<sub>B</sub> (data length 8 bits).

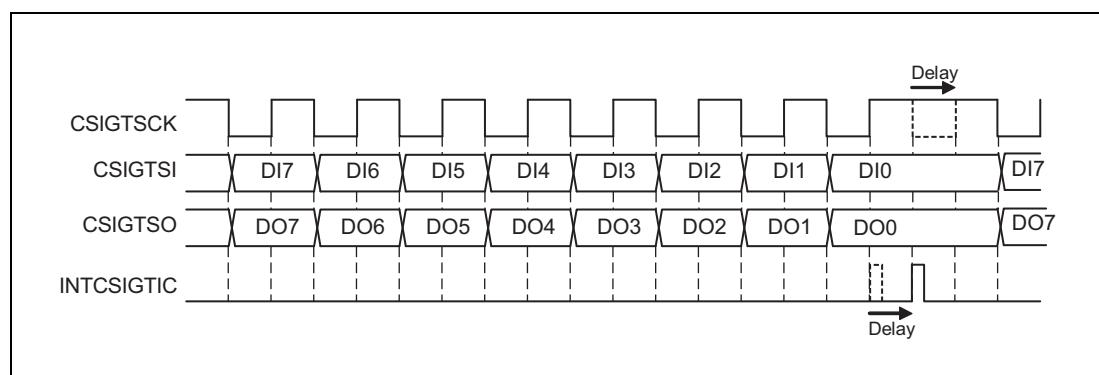
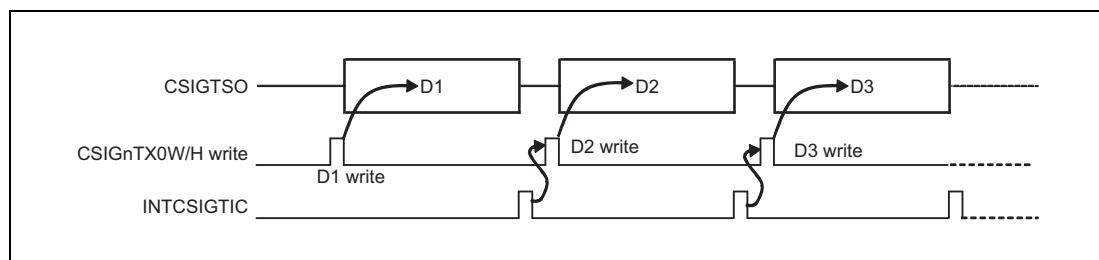


Figure 14.2 Interrupt Delay Function (CSIGnCTL1.CSIGnSIT = 1)

#### 14.4.2 INTCSIGTIC (Communication Status Interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGnTX0W or CSIGnTX0H.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), CSIGnCFG0.CSIGnDLS[3:0] = 1000<sub>B</sub> (data length 8 bits), and CSIGnCTL1.CSIGnSLIT = 0 (normal interrupt timing).

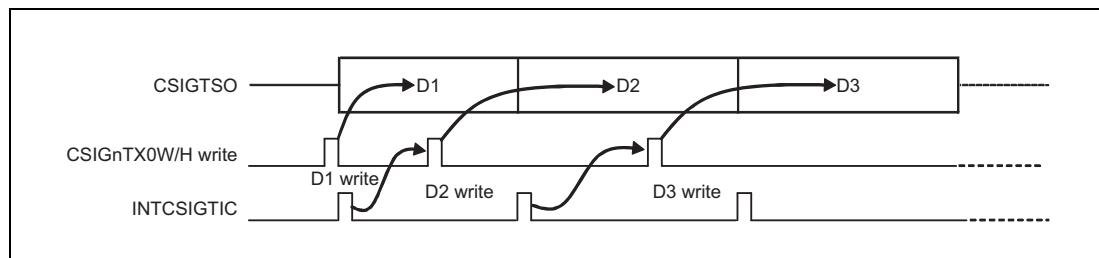


**Figure 14.3 Generation of INTCSIGTIC after Communication (CSIGnCTL1.CSIGnSLIT = 0)**

However, INTCSIGTIC can also be set up to occur when the CSIGnTX0W/H register is empty and available for receiving the next data. This is specified by setting CSIGnCTL1.CSIGnSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.



**Figure 14.4 Generation of INTCSIGTIC at the Beginning of Communication**

### 14.4.3 INTCSIGTIR (Reception Status Interrupt)

This interrupt is generated in receive-only and transmit/receive mode after data has been received and is available in the reception register. It can be used to trigger a DMA for reading the received data from CSIGnRX0 register.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000<sub>B</sub> (data length 8 bits).

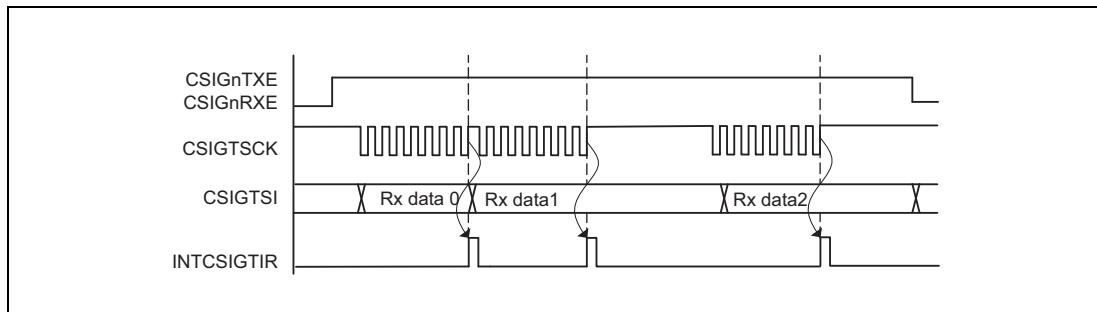


Figure 14.5 Generation of INTCSIGTIR

### 14.4.4 INTCSIGTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 14.25 Data Error Types

Error Type	Communication Status After Error Interrupt	Note
Parity error	Interrupt is generated and communication continues.	—
Data consistency error	Interrupt is generated and communication continues.	—
Overrun error <sup>*1</sup>	When CSIGnCTL1.CSIGnHSE = 0 (handshake function disabled) in slave mode, interrupt is generated and communication continues.	When CSIGnCTL1.CSIGnHSE = 1 (handshake function enabled) in slave mode, communication stops due to the handshake. An interrupt is not generated and an overrun error does not occur.

Note 1. In master mode, overrun errors do not occur.  
In slave mode, communication cannot be stopped.

The type of error that caused the generation of INTCSIGTIRE is indicated in register CSIGnSTR0.

For details about the various error types, see **Section 14.5.10, Error Detection**.

## 14.5 Operation

### 14.5.1 Master/Slave Mode

Whether CSIG operates in master mode or in slave mode depends on the setting of bits CSIGnCTL2.CSIGnPRS[2:0]. If master mode is selected, the source of the transmission clock must be selected as well.

#### 14.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the built-in baud rate generator (BRG) and supplied to the slave via signal CSIGTSCK.

Master mode is enabled by setting bits CSIGnCTL2.CSIGnPRS[2:0] to values other than  $111_B$ . In master mode, the frequency of BRG can be configured by setting bits CSIGnCTL2.CSIGnPRS[2:0] and bits CSIGnCTL2.CSIGnBRS[11:0].

The default level of CSIGTSCK depends on the CSIGTSCK clock inversion function bit; it is high when CSIGnCTL1.CSIGnCKR = 0, and is low when CSIGnCTL1.CSIGnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

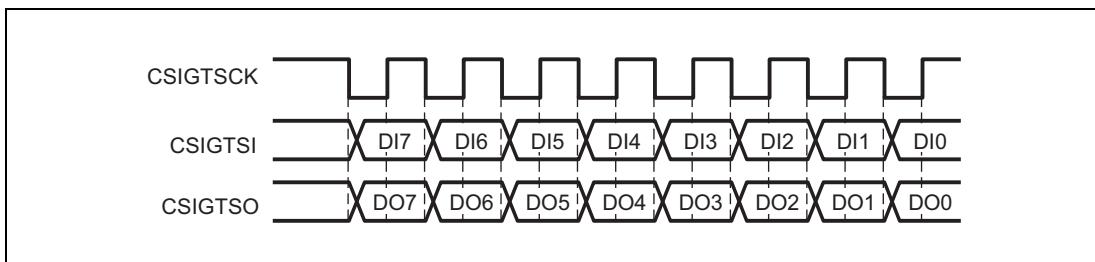


Figure 14.6 Transmission/Reception in Master Mode

### 14.5.1.2 Slave Mode

In slave mode, another device is the communication master. The external clock is supplied via the signal CSIGTSCK. Transmit/receive operation starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111<sub>B</sub>.

#### NOTE

When using slave mode, disable the baud rate generator (BRG) by setting bits CSIGnCTL2.CSIGnBRS[11:0] to 000<sub>H</sub>.

The example below shows the communication in slave mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

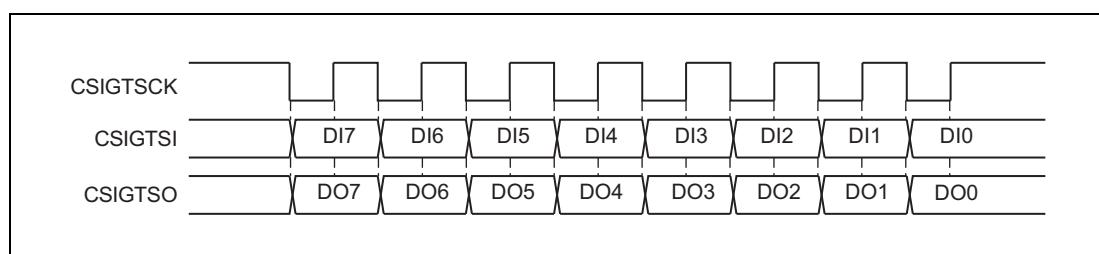


Figure 14.7 Transmission/Reception in Slave Mode

## 14.5.2 Master/Slave Connections

### 14.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

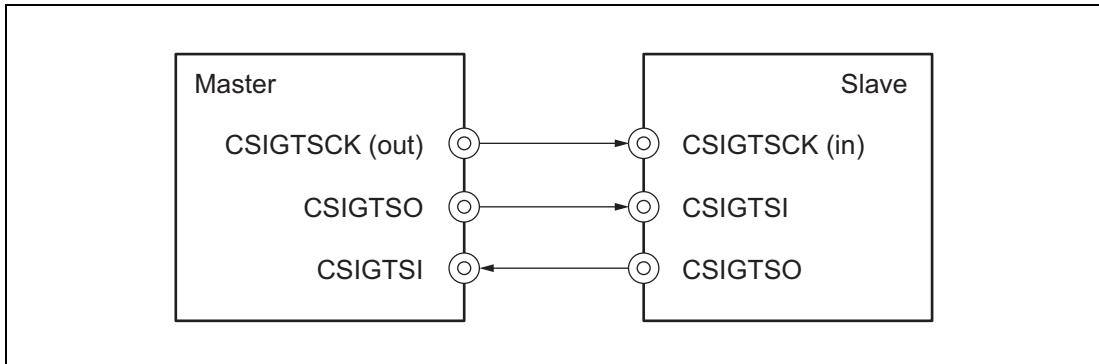


Figure 14.8 Direct Master/Slave Connection

### 14.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this case, the master must provide one slave select (SS) signal to each of the slaves. This signal is connected to the slave select input CSIGTSSI of the slave.

The CSIGTSSI signal can be enabled or disabled by the CSIGNCTL1.CSIGnSSE bit.

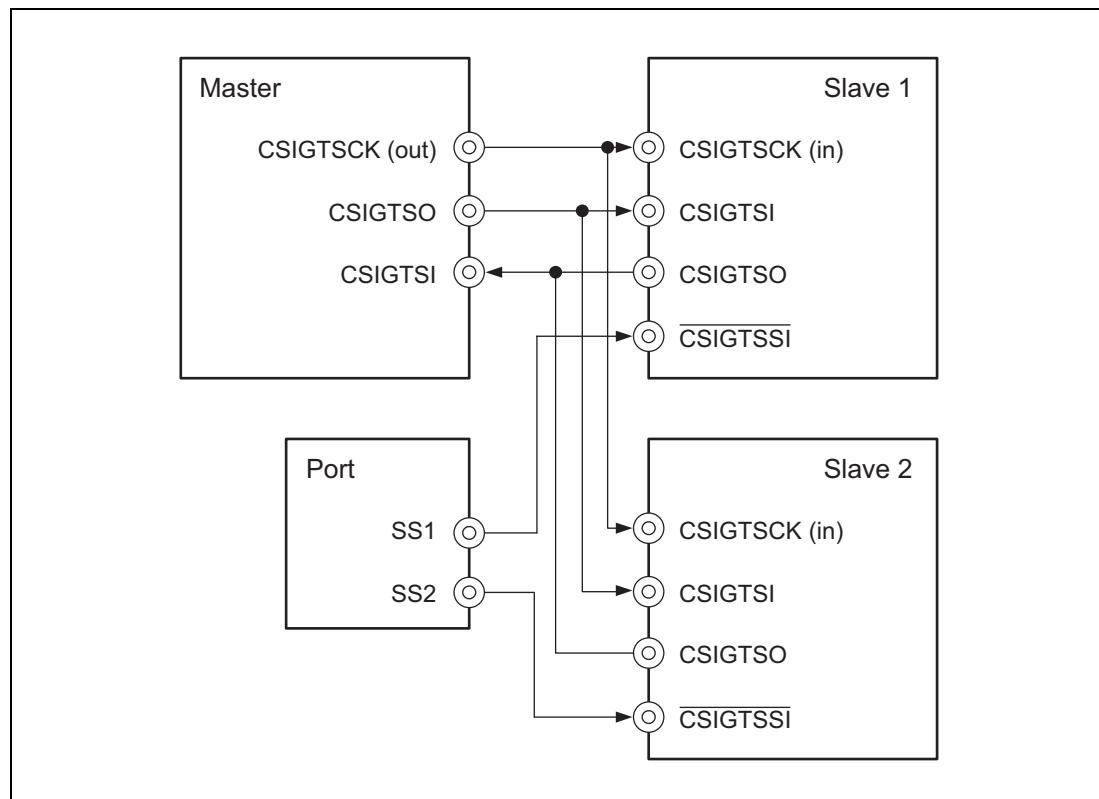


Figure 14.9 Master to Multiple Slaves Connection

A slave is selected (enabled) when its CSIGTSSI signal is low.

If it is not selected, the slave will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set (CSIGNCTL0.CSIGnTXE = 1), the CSIGTSO output buffer of the slave which is not selected is disabled and set to input mode in order to avoid interference with the outputs of other selected slaves.

### 14.5.3 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the CSIGnPRS[2:0] and CSIGnBRS[11:0] bits in the CSIGnCTL2 register.

The following figure shows a block diagram of the BRG.

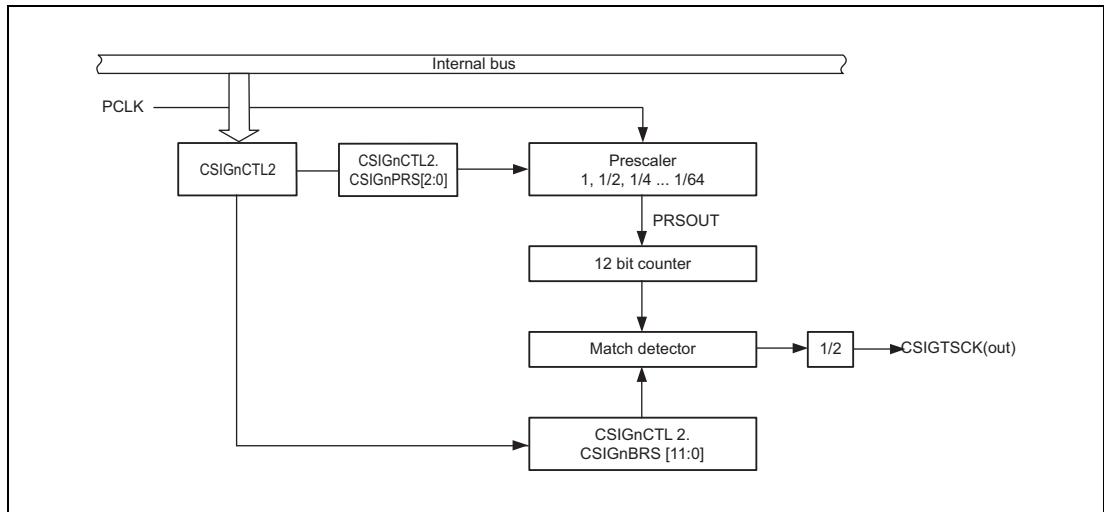


Figure 14.10 BRG Block Diagram

Setting CSIGnCTL2.CSIGnBRS[11:0] to 000<sub>H</sub> disables the BRG.

#### Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

$$\text{Transfer clock frequency (CSIGTSCK)} = \text{PCLK} / (\text{division ratio of PCLK}) = \text{PCLK} / (2^\alpha \times k \times 2)$$

where:

$$\alpha = \text{CSIGnCTL2.CSIGnPRS[2:0]} = 0 \text{ to } 6$$

$$k = \text{CSIGnCTL2.CSIGnBRS[11:0]} = 1 \text{ to } 4095$$

#### Transfer clock frequency upper and lower limits

When specifying the transfer clock frequency, please note the following:

- For the maximum transfer clock frequency of this product in master mode or slave mode, refer to the CSIG timing shown in the electrical characteristics. In addition, in either mode, specify a frequency within the defined range.
- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)

#### 14.5.4 Data Transfer Modes

##### 14.5.4.1 Transmit-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 places CSIG in transmit-only mode. Transmission starts when data to be transmitted is written in the CSIGnTX0W or CSIGnTX0H register.

##### CAUTION

**When the mode transitions from one of the receive modes to transmit-only mode, the data in the CSIGnRX0 buffer becomes undefined after completion of the first transmission.**

**Therefore, the reception register CSIGnRX0 has to be read before switching to transmit-only mode.**

##### 14.5.4.2 Receive-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in receive-only mode.

In master mode, reception starts when dummy data is read from the CSIGnRX0 register. All subsequent receptions are triggered by reads from the CSIGnRX0 register, as long as CSIGnBCTL0.CSIGnSCE = 1.

Moreover, CSIGnBCTL0.CSIGnSCE has to be set to 0 before reading the last received data from CSIGnRX0.

The recommended procedure is:

1. Set CSIGnBCTL0.CSIGnSCE = 1.
2. Read CSIGnRX0 (dummy data).
3. Wait for the reception interrupt INTCSIGTIR.
4. Read CSIGnRX0 (received data).

In case more data receptions follow at step 3, continue to read until all data is received.

Before reading the last received data from CSIGnRX0, set CSIGnBCTL0.CSIGnSCE = 0.

In slave mode, reception starts when the communication clock CSIGTSCK is received from the master. In this case, it is not necessary to read data from the CSIGnRX0 register of the slave.

##### NOTE

In slave mode, any previously received data must be read from the reception register CSIGnRX0 in order to avoid any overwrite situation.

##### 14.5.4.3 Transmit/Receive Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when the data to be transmitted is written to the CSIGnTX0W or CSIGnTX0H register.

### 14.5.5 Data Length Selection

#### 14.5.5.1 Data Length Selection without Extended Length

Transmission data length is selectable from 7 to 16 bits using the CSIGnDLS[3:0] bits in the CSIGnCFG0 register. The examples below show the communication with MSB first (CSIGnCFG0.CSIGnDIR = 0):

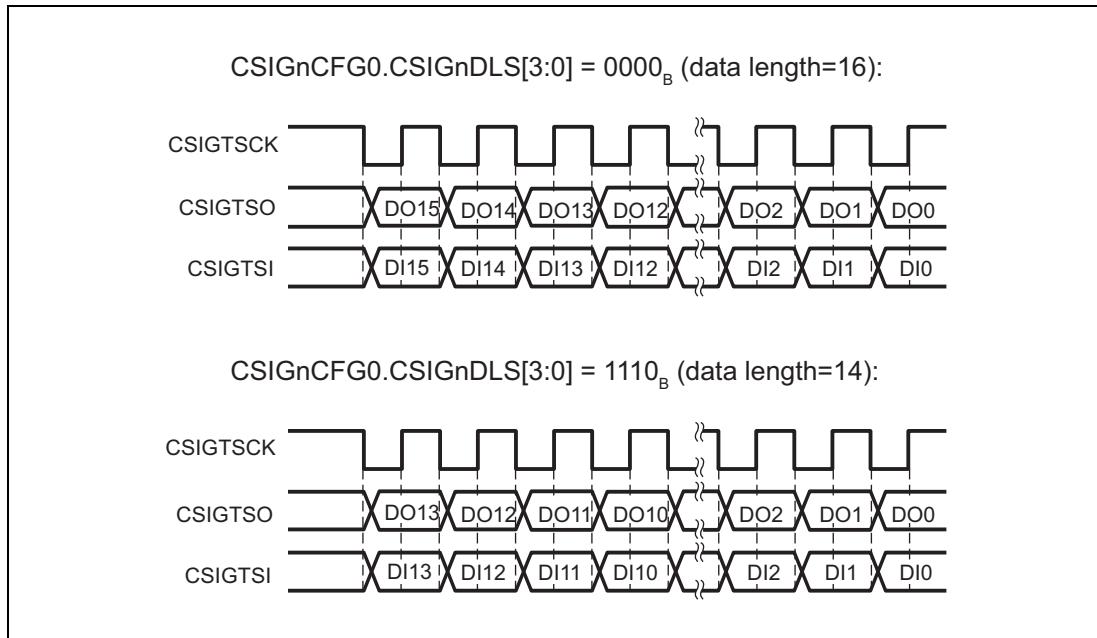


Figure 14.11 Data Length Selection Function

#### 14.5.5.2 Data Length Selection with Extended Data Length

If the data to be transmitted/received exceeds 16 bits, the extended data length (EDL) function can be used.

The EDL function is enabled by setting the CSIGnCTL1.CSIGnEDLE bit to 1.

The following describes how the EDL function works and how to specify the EDL setting.

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIGnCFG0.CSIGnDLS[3:0] bits.
- Set the CSIGnTX0W.CSIGnEDL bit to 1 to transmit the 16-bit blocks. In this case, the data written to the CSIGnTX0W register is sent as 16-bit data regardless of the setting of the CSIGnCFG0.CSIGnDLS[3:0] bits.
- The transfer completes after the data with the specified data length (the remainder when CSIGnTX0W.CSIGnEDL = 0) has been sent.

### Example

Example of sending 40 bits of data,  $123456789A_H$ :

40 bits are split into  $2 \times 16$  bits plus 8 bits.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] =  $8_D$ .
- To transmit the data  $123456789A_H$  with MSB first, write the following sequence to CSIGnTX0W:
  - $2000\ 1234_H$  (CSIGnTX0W.CSIGnEDL = 1)
  - $2000\ 5678_H$  (CSIGnTX0W.CSIGnEDL = 1)
  - $0000\ 009A_H$  (CSIGnTX0W.CSIGnEDL = 0)

The following figure illustrates the timing.

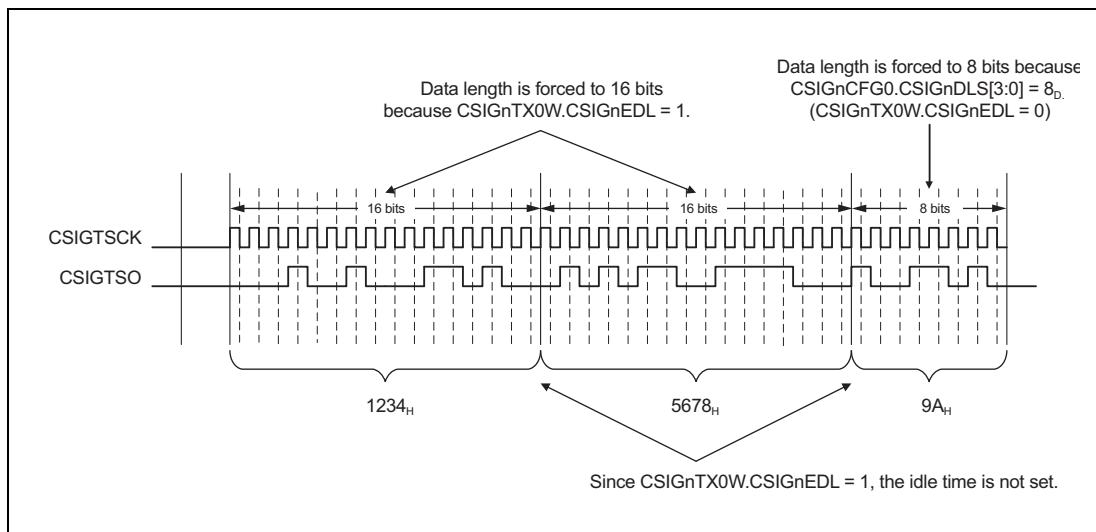


Figure 14.12 EDL Timing Diagram

### NOTES

1. Data length with less than 7 bits can be set only when EDL mode is used.
2. It is not possible to transmit two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. Example for setting the data direction:
  - Data to be transmitted:  $123456_H$
  - MSB first:  
Set CSIGnCFG0.CSIGnDIR to 0.  
Write  $2000\ 1234_H$  to CSIGnTX0W (EDL bit = 1).  
Write  $0000\ 0056_H$  to CSIGnTX0W (EDL bit = 0).
  - LSB first:  
Set CSIGnCFG0.CSIGnDIR to 1.  
Write  $2000\ 3456_H$  to CSIGnTX0W (EDL bit = 1).  
Write  $0000\ 0012_H$  to CSIGnTX0W (EDL bit = 0).
5. EDL mode cannot be used in the slave mode configured for receive-only mode.  
(CSIGnCTL2.CSIGnPRS[2:0] =  $111_B$ , CSIGnCTL0.CSIGnTxE = 0,  
CSIGnCTL0.CSIGnRxE = 1)

### 14.5.6 Serial Data Direction Selection Function

The serial data direction is selectable using the CSIGNDIR bit in the CSIGNCFG0 register. The examples below show the communication for 8-bit data (CSIGNCFG0.CSIGnDLS[3:0] = 1000<sub>B</sub>):

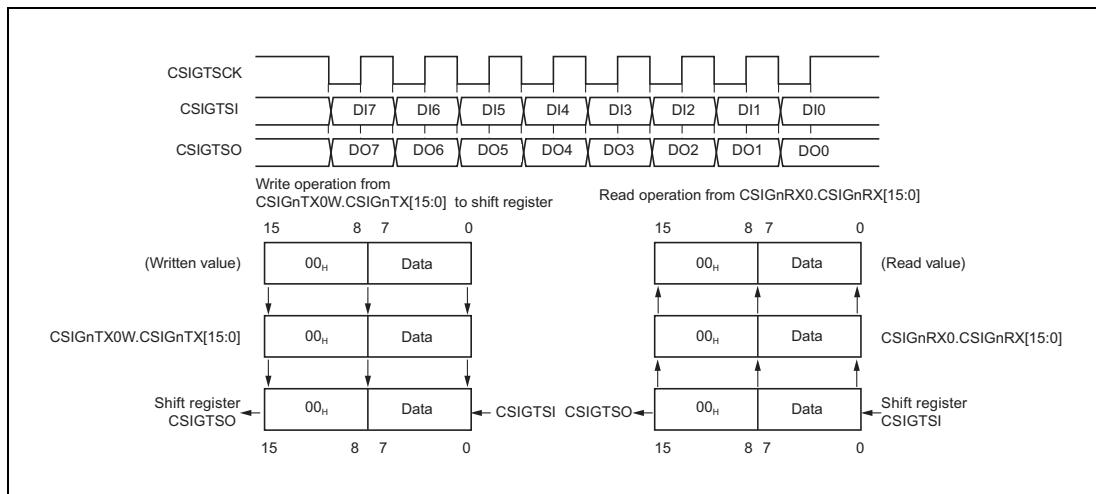


Figure 14.13 Serial Data Direction Select Function — MSB First (CSIGNDIR = 0)

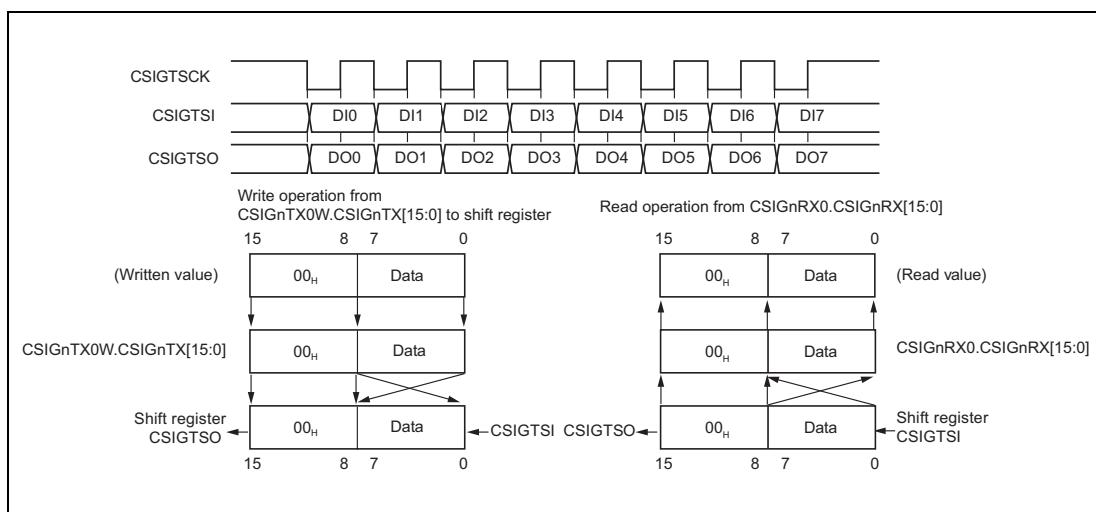
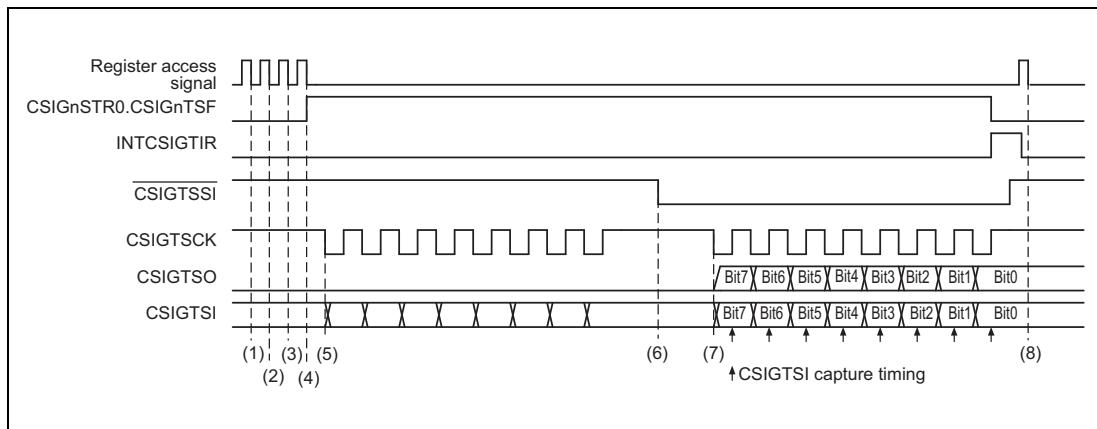


Figure 14.14 Serial Data Direction Select Function — LSB First (CSIGNDIR = 1)

### 14.5.7 Communication in Slave Mode

The following figure illustrates the communication signals and timing in slave mode.



**Figure 14.15 Rx/Tx Communication Timing in Slave Mode**

1. Slave mode is selected ( $\text{CSIGnCTL2.CSIGnPRS[2:0]} = 111_B$ ), the  $\overline{\text{CSIGTSSI}}$  signal is enabled ( $\text{CSIGnCTL1.CSIGnSSE} = 1$ ) and the clock phase is set to the high level ( $\text{CSIGnCTL1.CSIGnCKR} = 0$ ).
2. Data length is 8 bits ( $\text{CSIGnCFG0.CSIGnDLS[3:0]} = 1000_B$ ).  
Data direction is set to MSB first ( $\text{CSIGnCFG0.CSIGnDIR} = 0$ ).
3. CSIG is set to transmit/receive mode ( $\text{CSIGnCTL0.CSIGnPWR} = 1$ ,  $\text{CSIGnCTL0.CSIGnTXE} = 1$ ,  $\text{CSIGnCTL0.CSIGnRXE} = 1$ ).
4. When transfer data is written to the transmission register  $\text{CSIGnTX0H}$ , the transfer status flag  $\text{CSIGnSTR0.CSIGnTSF}$  is automatically set and the CSIG waits until signal  $\overline{\text{CSIGTSSI}}$  goes low.
5. While signal  $\overline{\text{CSIGTSSI}}$  is high, transmission/reception is not started even if the serial clock is input. CSIGTSO retains the values and input at CSIGTSI is ignored.
6. As soon as  $\overline{\text{CSIGTSSI}}$  falls to low level, CSIGTSO is enabled.
7. If the serial clock is input to the CSIG while  $\overline{\text{CSIGTSSI}}$  is low, transfer data is sent to CSIGTSO in synchronization with the serial clock, and simultaneously data is received from CSIGTSI.
8. The register  $\text{CSIGnRX0}$  is read.

### 14.5.8 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIGnCTL1.CSIGnHSE bit. For handshake, the signals CSIGTRYI and CSIGTRYO are used.

The busy timing depends on the setting of the data phase selection bit CSIGnCFG0.CSIGnDAP.

#### 14.5.8.1 Slave Mode

If CSIGnCTL1.CSIGnHSE = 1, a low-level CSIGTRYO signal is output when the slave becomes busy. This happens when previously received data is still in the CSIGnRX0 register, and the new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume a data length of 8 bits.

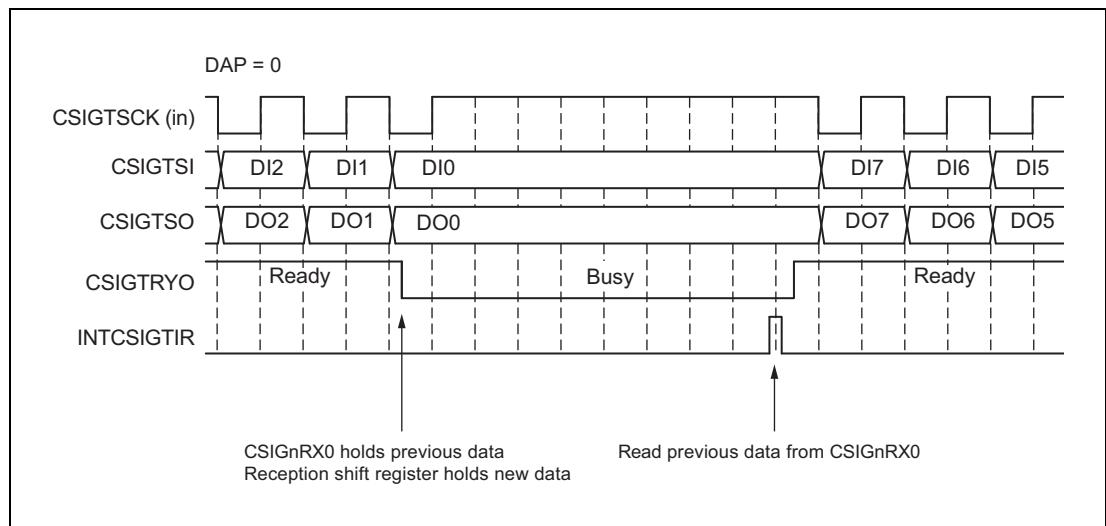


Figure 14.16 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

While the slave is busy, the master has to wait (i.e. suspend the transmission clock). The slave sets CSIGTRYO to high (“ready”) as soon as the read from the reception register CSIGnRX0 completes.

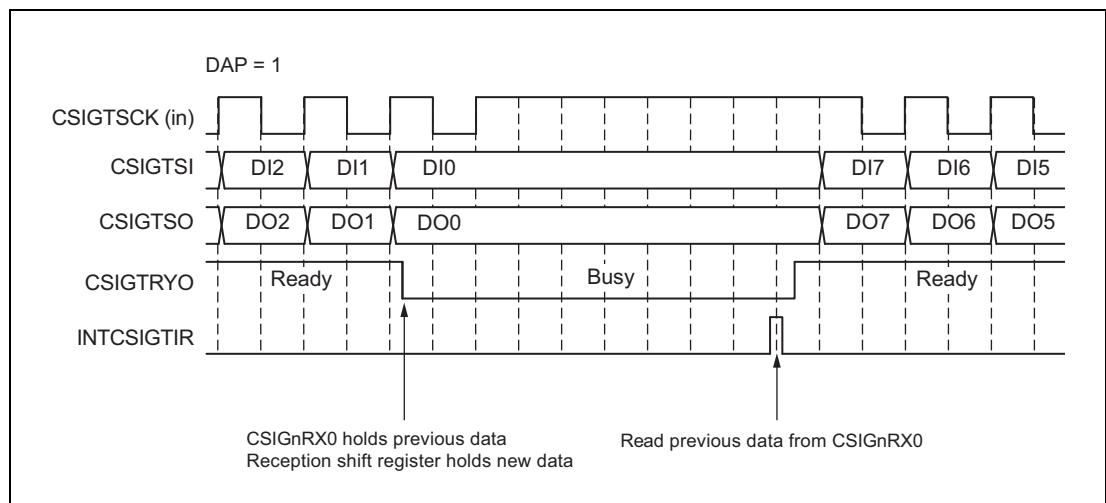


Figure 14.17 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

### 14.5.8.2 Master Mode

When the master detects low level of the CSIGTRYI while CSIGnCTL1.CSIGnHSE = 1, subsequent transfers are put on hold, the master goes into wait state and suspends the CSIGTSCK clock.

The CSIGTRYI level is checked at each half clock cycle of CSIGTSCK.

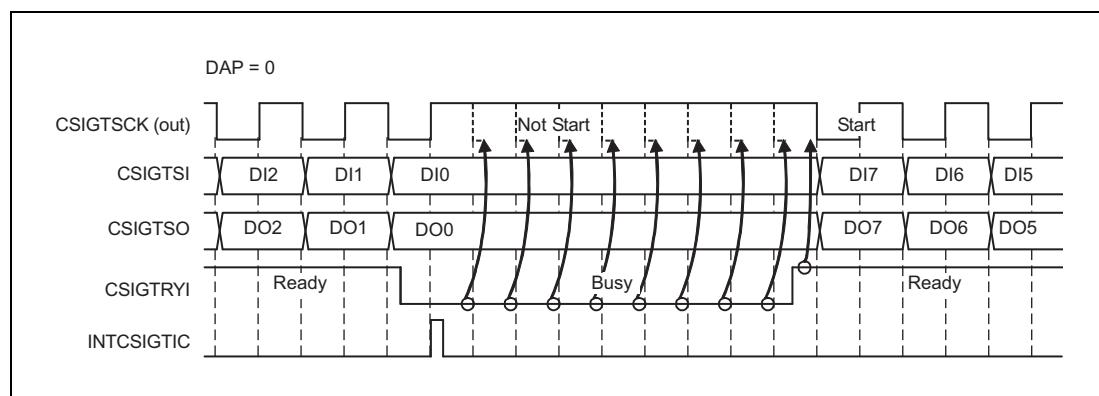


Figure 14.18 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 0)

If the CSIGTRYI low signal from the slave is received while data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIGTRYI becomes high (slave is “ready”).

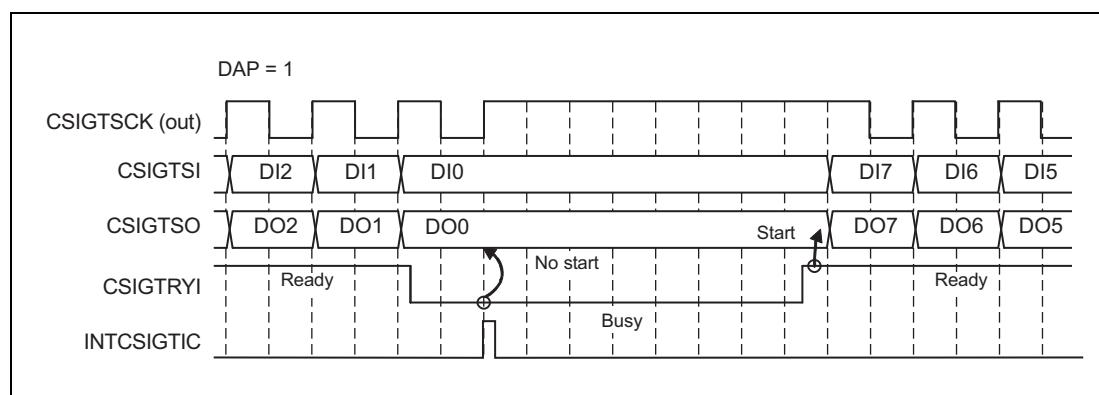


Figure 14.19 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 1)

#### CAUTION

If multiple slaves are connected, the master must only detect the CSIGTRYI signal of the slave it has selected for communication.

CSIGTRYI of the master must be pulled down by the slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer will continue until it completes.

### 14.5.9 Loop-Back Mode

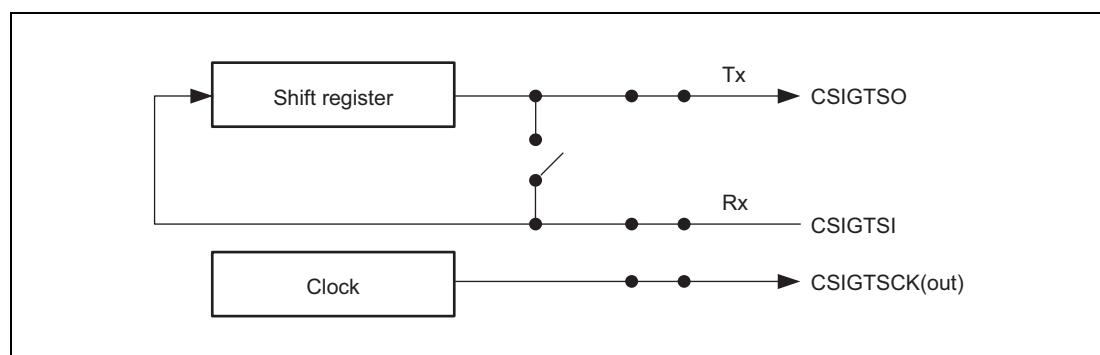
Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIGnCTL1.CSIGnLBM = 1), the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIGTCK, CSIGTSO, and CSIGTSI are disconnected from the ports. In addition, the CSIGTSO output level is fixed to low, and CSIGTCK is set to reset level (High). The rest of CSIG works as in normal operation.

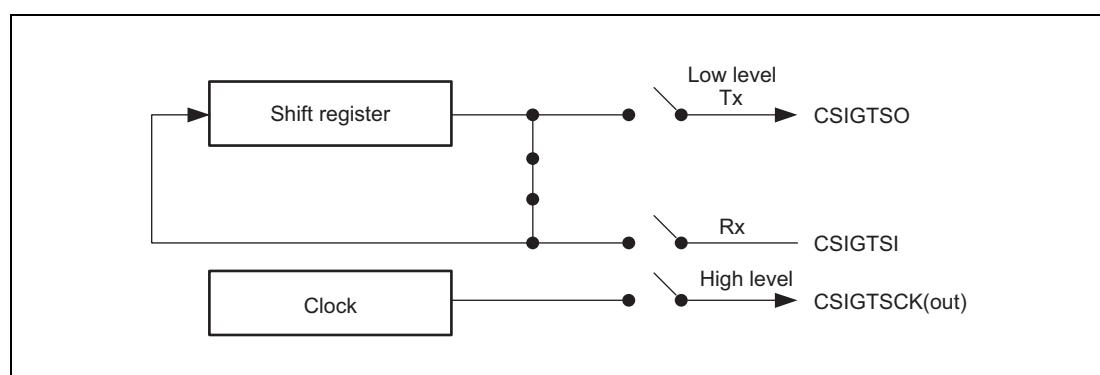
In order to test the CSIG, set the loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data.

**Table 14.26 Output Level of Pins**

Pin	Output Level
CSIGTCK(out)	High level
CSIGTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIGTRYO	Normal function (Low level)



**Figure 14.20 Normal Operation**



**Figure 14.21 Operation in Loop-Back Mode**

### 14.5.10 Error Detection

CSIG can detect three error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)

Data consistency error and parity error check functions can be individually enabled or disabled.

If one of these errors is detected, the interrupt INTCSIGTIRE is generated.

#### 14.5.10.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as an output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIGnCTL1.CSIGnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn\_m = 1 for CSIGTSO). It will not be enabled if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When the data consistency check is enabled, the data transferred from CSIGnTX0W or CSIGnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIGTSO are captured and their logical interpretation is written to the corresponding shift register.

After completion of the transmission, the data sent is compared with the original transmission data.

Mismatch is considered as a data consistency error and:

- Interrupt INTCSIGTIRE is generated.
- The CSIGnSTR0.CSIGnDCE bit is set.

The data consistency check function is illustrated in the following block diagram.

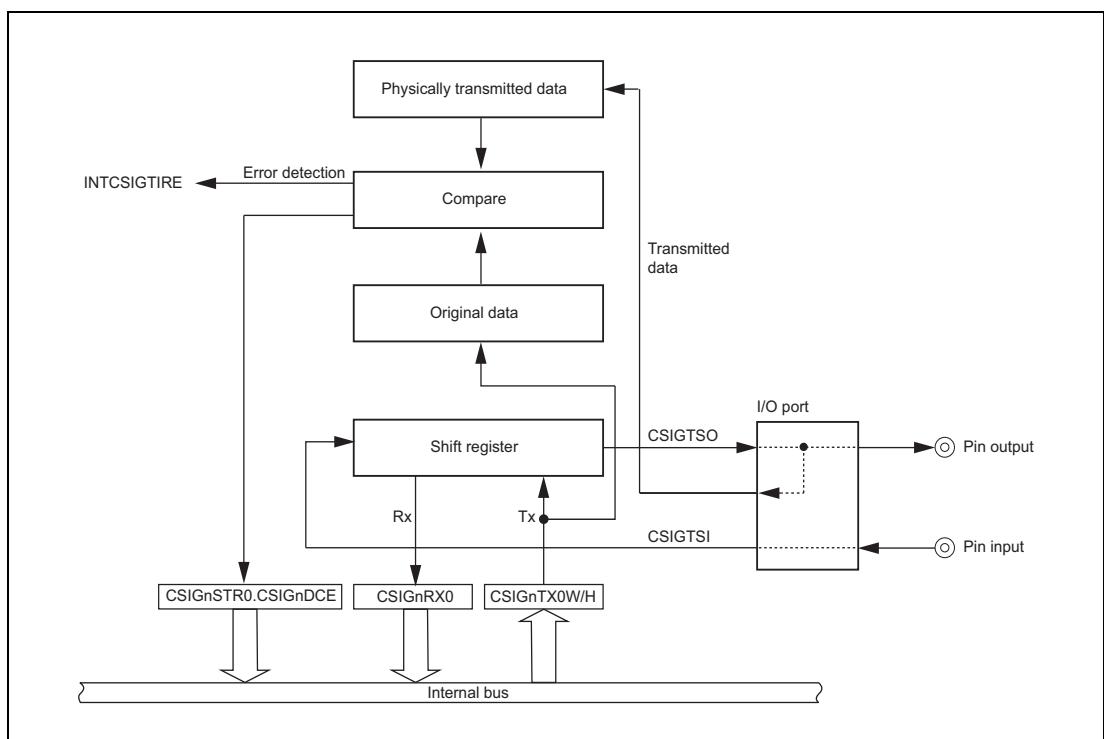


Figure 14.22 Functional Block Diagram of the Data Consistency Check

#### 14.5.10.2 Parity Check

Parity is a common mean to detect a single bit error during data transmission. CSIG can append a parity bit after the last data bit (even if extended data length is used).

The use and type of parity is specified by CSIGNCFG0.CSIGNPS[1:0].

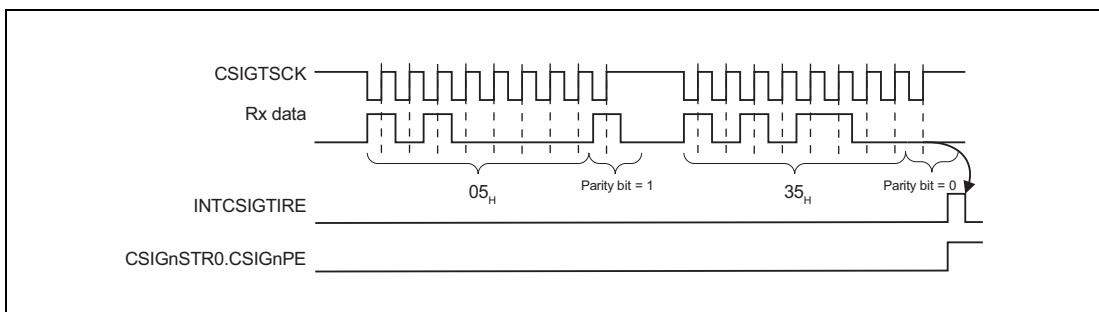
Parity check is enabled if CSIGNCFG0.CSIGNPS[1] = 1.

The parity bit is checked after reception is complete. If a parity error occurs:

- Interrupt INTCSIGTIRE is generated.
- The CSIGNSTR0.CSIGNPE bit is set.

The following figure shows an example.

Data length is 8 bits. The data transmitted is  $05_{\text{H}}$  and  $35_{\text{H}}$ . Parity type is odd.



**Figure 14.23 Parity Check Example**

For the first 8 bits, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the second 8 bits, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last data bit.

#### 14.5.10.3 Overrun Error

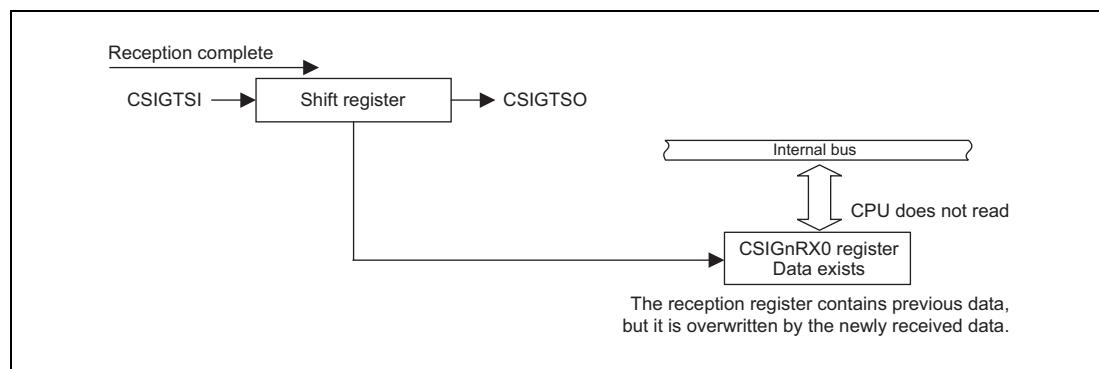
This error occurs when previously received data still resides in the reception register CSIGNRX0 because it has not been read, and new data is received.

The overrun error is not generated if data reception is disabled (CSIGNCTL0.CSIGNRXE = 0).

If overrun occurs:

- Interrupt INTCSIGTIRE is generated
- The CSIGNSTR0.CSIGNOVE bit is set
- Data in the CSIGNRX0 register is overwritten and communication continues.

The following figure illustrates the overrun error detection function.

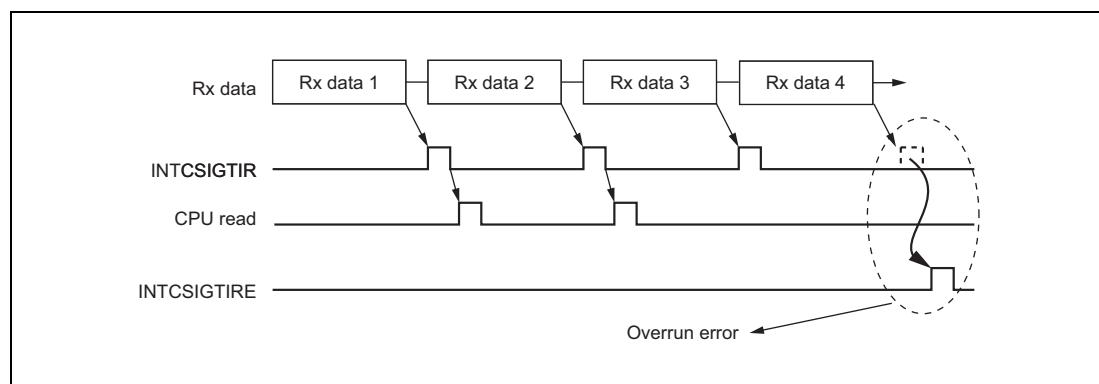


**Figure 14.24 Overrun Error Detection**

The following figure illustrates an example where:

- Rx data 3 is not read
- Rx data 4 is received, and data is overwritten.

Thus an overrun error occurs.



**Figure 14.25 Overrun Error Detection - Example**

#### NOTE

An overrun error can be avoided by using the handshake.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

For details see **Section 14.5.8, Handshake Function**.

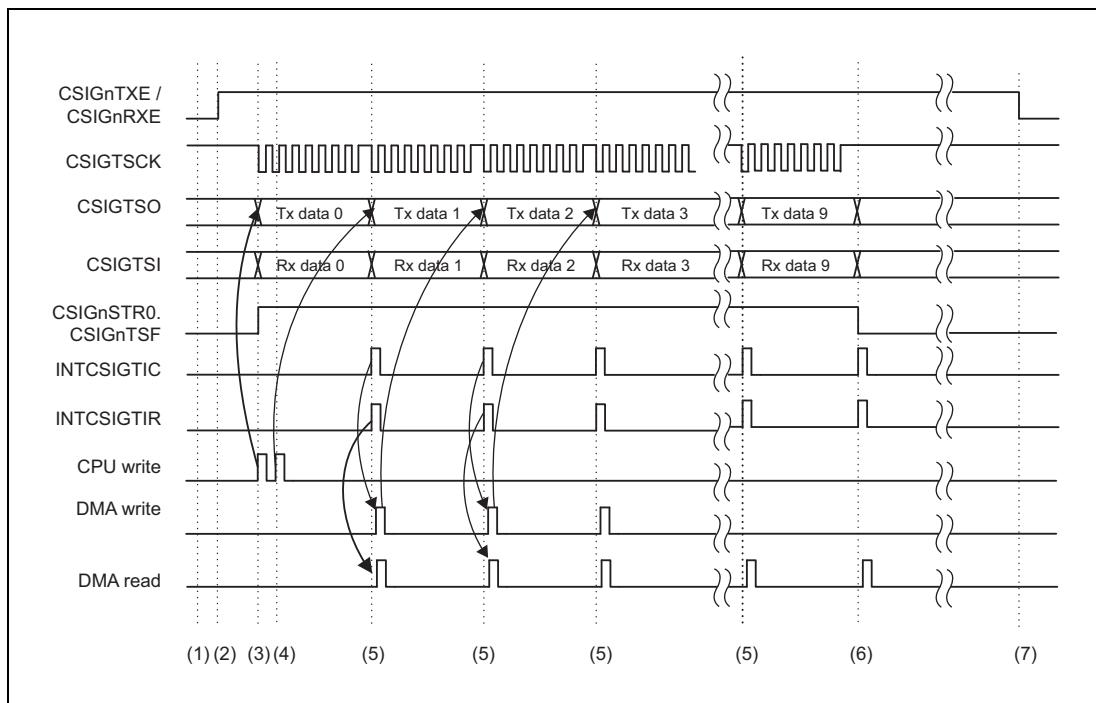
## 14.6 Operating Procedures

### 14.6.1 Master Mode Transmission/Reception by DMA

This section describes an example of performing the transmission/reception in master mode in combination with a DMA.

The following instructions are based on the assumption that:

- Transmission data length is 8 bits ( $\text{CSIGnCFG0.CSIGnDLS[3:0]} = 1000_B$ )
- MSB is transmitted first ( $\text{CSIGnCFG0.CSIGnDIR} = 0$ )
- INTCSIGTIC interrupt is generated at the end of the transfer ( $\text{CSIGnCTL1.CSIGnSLIT} = 0$ )
- Normal clock and data phase ( $\text{CSIGnCTL1.CSIGnCKR} = 0$ ,  $\text{CSIGnCFG0.CSIGnDAP} = 0$ )
- The number of data is 10 (0 to 9)



**Figure 14.26 Communication in Master Mode**

#### Procedure:

1. Configure the communication protocol in register CSIGnCFG0. Specify the interrupt timing and operation mode by setting the corresponding bits of the CSIGnCTL1 register and CSIGnCTL2 register.
2. In the CSIGnCTL0 register, set CSIGnPWR = 1 (enable clock), CSIGnTXE = 1 (enable transmission), and CSIGnRXE = 1 (enable reception).
3. Write the first data to be sent to the transmission register CSIGnTX0H. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIGnTX0H. Writing the second data immediately after the first one avoids unnecessary delays between the data.

5. After the transmission or reception of each data, INTCSIGTIC or INTCSIGTIR interrupt is generated. INTCSIGTIC indicates that the next data can be written to CSIGnTX0H. INTCSIGTIR indicates that the reception register CSIGnRX0 must be read. In this example, CPU write and DMA write are assumed as equivalent.
6. No more write action is required after the transmission of data 8 completes. Data 9 (the last data) has been written after the transmission of data 7. However, the reception register CSIGnRX0 must be read after the reception of data 8 and 9 completes.
7. Finally, to disable the transmit/receive operation, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE. When no communication is taking place, set CSIGnCTL0.CSIGnPWR to “0” to minimize the power consumption of the CSIGn.

## Section 15 Clocked Serial Interface H (CSIH)

This section contains a generic description of the Clocked Serial Interface H (CSIH).

The first part in this section describes the features specific to RH850/F1L, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of CSIH.

### 15.1 Features of RH850/F1L CSIH

#### 15.1.1 Number of Units

This microcontroller has the following number of CSIH units.

Each CSIH unit has one channel interface.

**Table 15.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of units	1	1	3	4	4	4
Name	CSIHn (n = 0)	CSIHn (n = 0)	CSIHn (n = 0 to 2)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)

**Table 15.2 Indices**

Index	Description
n	Throughout this section, the individual CSIH units are identified by the index "n" (n = 0 to 3); for example, CSIHnCTL0 is the CSIHn control register 0.
x	CSIHn has a maximum of 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index "x"; that is, CSx denotes a non-specified chip select signal.
y	A variable used for explanation is identified by the index "y"; for example, CSIHnBRSy is the baud rate setting register of CSIHn.

The following table shows values indicated by the indices of each product.

**Table 15.3 Indices of Products**

Indices of Each Product				
48 pins	64 pins	80 pins	100 pins	144 pins, 176 pins
For the value of x, see Table 15.4, Number of Chip Select Signals.				
y = 0 to 3	y = 0 to 3	y = 0 to 3	y = 0 to 3	y = 0 to 3

The numbers of chip select signals for each of the CSIH units are listed in the following table.

**Table 15.4 Number of Chip Select Signals**

Unit Name	Chip Select Index					
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
CSIH0	CSx (x = 0, 1)	CSx (x = 0 to 3)	CSx (x = 0 to 7)			
CSIH1	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH2	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH3	—	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 3)

### 15.1.2 Register Base Address

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses.

**Table 15.5 Register Base Addresses**

Base Address Name	Base Address
<CSIH0_base>	FFD8 0000 <sub>H</sub>
<CSIH1_base>	FFD8 2000 <sub>H</sub>
<CSIH2_base>	FFD8 4000 <sub>H</sub>
<CSIH3_base>	FFD8 6000 <sub>H</sub>

### 15.1.3 Clock Supply

The CSIH clock supply is shown in the following table.

**Table 15.6 Clock Supply**

Unit Name	Unit Clock Name	Internal Clock Signal
CSIHn	PCLK	CKSCLK_ICSI
	Register access	CKSCLK_ICSI

### 15.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

**Table 15.7 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>CSIH0</b>			
INTCSIHTIC	Communication status interrupt	21	6 (channels 8 to 15)
INTCSIHTIR	Receive status interrupt	22	7 (channels 8 to 15)
INTCSIHTIRE	Communication error interrupt	23	—
INTCSIHTIJC	Job completion interrupt	24	8 (channels 8 to 15)
<b>CSIH1</b>			
INTCSIHTIC	Communication status interrupt	8, 108	28 (channels 0 to 7)
INTCSIHTIR	Receive status interrupt	9, 109	29 (channels 0 to 7)
INTCSIHTIRE	Communication error interrupt	19, 110	—
INTCSIHTIJC	Job completion interrupt	20, 111	30 (channels 0 to 7)
<b>CSIH2</b>			
INTCSIHTIC	Communication status interrupt	0, 124	25 (channels 8 to 15)
INTCSIHTIR	Receive status interrupt	29, 125	26 (channels 8 to 15)
INTCSIHTIRE	Communication error interrupt	30, 126	—
INTCSIHTIJC	Job completion interrupt	31, 127	27 (channels 8 to 15)
<b>CSIH3</b>			
INTCSIHTIC	Communication status interrupt	1, 150	41 (channels 0 to 7)
INTCSIHTIR	Receive status interrupt	5, 151	42 (channels 0 to 7)
INTCSIHTIRE	Communication error interrupt	6, 152	—
INTCSIHTIJC	Job completion interrupt	7, 153	43 (channels 0 to 7)

### 15.1.5 Reset Sources

CSIH reset sources are listed in the following table. CSIH is initialized by these reset sources.

**Table 15.8 Reset Sources**

Unit Name	Reset Source
CSIHn	All reset sources (ISORES)

### 15.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

**Table 15.9 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal Name
<b>CSIH0</b>		
CSIHTSCK	Serial clock signal	CSIH0SC <sup>*2</sup>
CSIHTSI	Serial data input signal	CSIH0SI
CSIHTSSI	Slave select input signal	CSIH0SSI
CSIHTRYI	Ready/busy input signal	CSIH0RYI
CSIHTSO	Serial data output signal	CSIH0SO <sup>*2</sup>
CSIHTRYO	Ready/busy output signal	CSIH0RYO
CSIHTCSS[7:0] <sup>*1</sup>	Chip select signal	CSIH0CSS[7:0] <sup>*1</sup>
<b>CSIH1</b>		
CSIHTSCK	Serial clock signal	CSIH1SC <sup>*2</sup>
CSIHTSI	Serial data input signal	CSIH1SI
CSIHTSSI	Slave select input signal	CSIH1SSI
CSIHTRYI	Ready/busy input signal	CSIH1RYI
CSIHTSO	Serial data output signal	CSIH1SO <sup>*2</sup>
CSIHTRYO	Ready/busy output signal	CSIH1RYO
CSIHTCSS[5:0] <sup>*1</sup>	Chip select signal	CSIH1CSS[5:0] <sup>*1</sup>
<b>CSIH2</b>		
CSIHTSCK	Serial clock signal	CSIH2SC <sup>*2</sup>
CSIHTSI	Serial data input signal	CSIH2SI
CSIHTSSI	Slave select input signal	CSIH2SSI
CSIHTRYI	Ready/busy input signal	CSIH2RYI
CSIHTSO	Serial data output signal	CSIH2SO <sup>*2</sup>
CSIHTRYO	Ready/busy output signal	CSIH2RYO
CSIHTCSS[5:0] <sup>*1</sup>	Chip select signal	CSIH2CSS[5:0] <sup>*1</sup>
<b>CSIH3</b>		
CSIHTSCK	Serial clock signal	CSIH3SC <sup>*2</sup>
CSIHTSI	Serial data input signal	CSIH3SI
CSIHTSSI	Slave select input signal	CSIH3SSI
CSIHTRYI	Ready/busy input signal	CSIH3RYI
CSIHTSO	Serial data output signal	CSIH3SO <sup>*2</sup>
CSIHTRYO	Ready/busy output signal	CSIH3RYO
CSIHTCSS[3:0] <sup>*1</sup>	Chip select signal	CSIH3CSS[3:0] <sup>*1</sup>

Note 1. For the number of chip select signals, see **Table 15.4, Number of Chip Select Signals**.

Note 2. For the port pins that are used as CSIHiSO and CSIHiSC, set the output driving ability to high (PDSCn\_m = 1).

#### CAUTION

When the P0\_0 pin is used as CSIH0SSI, the P0\_0 pin outputs a low-level RESETOUT signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

### 15.1.7 Data Consistency Check

The port and the alternative function for data consistency check of CSI<sub>n</sub>SO (CSIHTSO) output are shown in the following table. See **Section 15.5.12, Error Detection** for details on data consistency checking.

**Table 15.10 Port Pins for Data Consistency Checking**

Unit Signal Name	Port Pin Name	Alternative Function
<b>CSIH0</b>		
CSIHTSO	P0_3	ALT_OUT4
<b>CSIH1</b>		
CSIHTSO	P0_5	ALT_OUT3
	P10_2	ALT_OUT5
<b>CSIH2</b>		
CSIHTSO	P11_2	ALT_OUT1
<b>CSIH3</b>		
CSIHTSO	P11_6	ALT_OUT3

## 15.2 Overview

### 15.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode or slave mode is selectable
- Multiple slaves configuration and RCB (Recessive Configuration for Broadcasting) are possible since there are to eight configurable chip select output signals
- Slave select input signal ( $\overline{\text{CSIHTSSI}}$ ) is usable
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
- Clock phases and data phases are selectable
- Data transfer with MSB first or LSB first is selectable
- Transfer data length is selectable from 2 to 16 bits in 1-bit units
- Built-in EDL (extended data length) function for transferring more than 16 bits of data
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, time-out, overflow, and overrun)
- Support of job concept
- 128 words I/O buffer memory
- Direct access mode or memory mode (FIFO, dual buffer, and transmit-only buffer) is selectable
- Four different interrupt request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIRE, INTCSIHTIJC)
- Built-in LBM (Loop Back Mode) function for self-test
- CPU-controlled high-priority communication function
- Enforced chip select idle setting
- Built-in RCB (Recessive Configuration for Broadcasting) bit
- Built-in JOB enable control bit for AUTOSAR

### 15.2.2 Functional Overview Description

The CSIH uses three signals for communication:

- Transmission clock CSIHTSCK (output in master mode, input in slave mode)
- Data output signal CSIHTSO
- Data input signal CSIHTSI

Additional signals are available for external control and monitoring:

- CSIHTSSI: Slave select input signal
- CSIHTRYO: Ready/busy output signal (handshake signal)
- CSIHTRYI: Ready/busy input signal (handshake signal)
- CSIHTCSS[7:0]: Chip select signals

Data transmission is bit-wise and serial, and performed synchronously with the transmission clock.

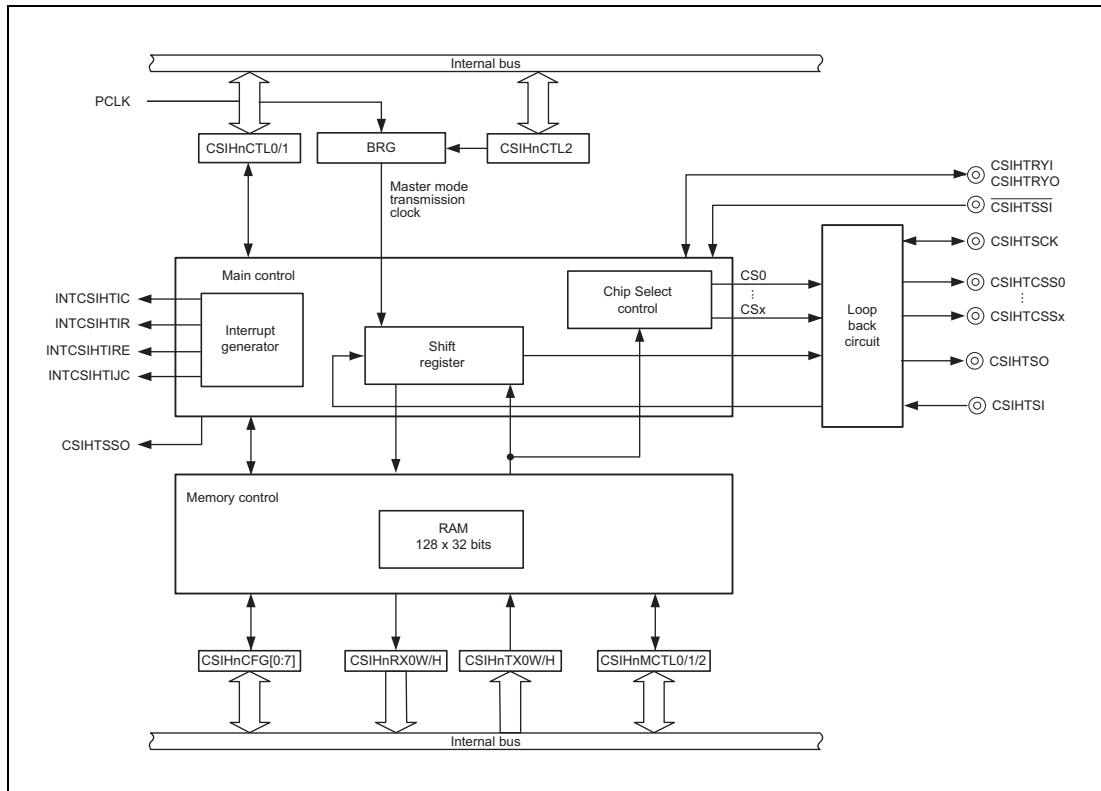
The following table shows the most important registers for setting up the CSIH.

**Table 15.11 Main Registers of CSIH**

Register	Function
CSIHnCTL0	Enables/disables serial clock, and permits/ prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables (bypasses) buffering.
CSIHnCTL1	Controls options such as interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc.
CSIHnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIHnBRSy	Specifies the transfer clock frequency for each chip select signal.
CSIHnMCTL0	Selects memory mode and specifies the time-out value.
CSIHnMCTL1	Controls the memory in FIFO mode.
CSIHnMCTL2	Controls the memory in dual buffer mode.
CSIHnCFGx	Configures the communication protocol for each chip select signal.

### 15.2.3 Block Diagram

The following block diagram shows the main components of the CSIH.



**Figure 15.1** CSIH Block Diagram

In master mode, the transmission clock CSIHTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

The built-in memory can be configured as FIFO, dual buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self-test.

#### NOTE

This section describes the following modes:

- The “operating mode” is either master mode or slave mode. Only in master mode can the master control communicate with several slaves (for details, see **Section 15.5.1, Operating Modes (Master/Slave)**).
- The “job mode” is related to the AUTOSAR job concept (for details, see **Section 15.5.3.3, Job Concept**).
- The “memory mode” accords with the various configurations of the associated buffer memory (for details, see **Section 15.5.6, CSIH Buffer Memory**).
- The “data transfer mode” specifies the type of communication – transmit-only, receive-only, or transmit/receive (for details, see **Section 15.5.7, Data Transfer Modes**).

## 15.3 Registers

### 15.3.1 List of Registers

CSIH registers are listed in the following table.

For details about <CSIHn\_base>, see **Section 15.1.2, Register Base Address**.

Table 15.12 List of Registers

Module	Register	Symbol	Address
CSIHn	CSIHn control register 0	CSIHnCTL0	<CSIHn_base> + 0000 <sub>H</sub>
CSIHn	CSIHn control register 1	CSIHnCTL1	<CSIHn_base> + 0010 <sub>H</sub>
CSIHn	CSIHn control register 2	CSIHnCTL2	<CSIHn_base> + 0014 <sub>H</sub>
CSIHn	CSIHn status register 0	CSIHnSTR0	<CSIHn_base> + 0004 <sub>H</sub>
CSIHn	CSIHn status clear register 0	CSIHnSTCR0	<CSIHn_base> + 0008 <sub>H</sub>
CSIHn	CSIHn memory control register 0	CSIHnMCTL0	<CSIHn_base> + 1040 <sub>H</sub>
CSIHn	CSIHn memory control register 1	CSIHnMCTL1	<CSIHn_base> + 1000 <sub>H</sub>
CSIHn	CSIHn memory control register 2	CSIHnMCTL2	<CSIHn_base> + 1004 <sub>H</sub>
CSIHn	CSIHn memory read/write pointer register 0	CSIHnMRWP0	<CSIHn_base> + 1018 <sub>H</sub>
CSIHn	CSIHn configuration register 0	CSIHnCFG0	<CSIHn_base> + 1044 <sub>H</sub>
CSIHn	CSIHn configuration register 1	CSIHnCFG1	<CSIHn_base> + 1048 <sub>H</sub>
CSIHn	CSIHn configuration register 2	CSIHnCFG2	<CSIHn_base> + 104C <sub>H</sub>
CSIHn	CSIHn configuration register 3	CSIHnCFG3	<CSIHn_base> + 1050 <sub>H</sub>
CSIHn	CSIHn configuration register 4	CSIHnCFG4	<CSIHn_base> + 1054 <sub>H</sub>
CSIHn	CSIHn configuration register 5	CSIHnCFG5	<CSIHn_base> + 1058 <sub>H</sub>
CSIHn	CSIHn configuration register 6	CSIHnCFG6	<CSIHn_base> + 105C <sub>H</sub>
CSIHn	CSIHn configuration register 7	CSIHnCFG7	<CSIHn_base> + 1060 <sub>H</sub>
CSIHn	CSIHn transmit data register 0 for word access	CSIHnTX0W	<CSIHn_base> + 1008 <sub>H</sub>
CSIHn	CSIHn transmit data register 0 for half word access	CSIHnTX0H	<CSIHn_base> + 100C <sub>H</sub>
CSIHn	CSIHn receive data register 0 for word access	CSIHnRX0W	<CSIHn_base> + 1010 <sub>H</sub>
CSIHn	CSIHn receive data register 0 for half word access	CSIHnRX0H	<CSIHn_base> + 1014 <sub>H</sub>
CSIHn	CSIHn emulation register	CSIHnEMU	<CSIHn_base> + 0018 <sub>H</sub>
CSIHn	CSIHn baud rate setting register 0	CSIHnBRS0	<CSIHn_base> + 1068 <sub>H</sub>
CSIHn	CSIHn baud rate setting register 1	CSIHnBRS1	<CSIHn_base> + 106C <sub>H</sub>
CSIHn	CSIHn baud rate setting register 2	CSIHnBRS2	<CSIHn_base> + 1070 <sub>H</sub>
CSIHn	CSIHn baud rate setting register 3	CSIHnBRS3	<CSIHn_base> + 1074 <sub>H</sub>

### 15.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock, enables/disables transmission/reception, and enables/disables the memory allocated for transmission and/or reception. It forces the stop of communication at the end of the current job.

**Access:** This register can be read or written in 8-bit or 1-bit units.

**Address:** <CSIHn\_base> + 0000H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	CSIHnPWR	CSIHnTXE	CSIHnRXE	—	—	—	CSIHnJOBE	CSIHnMBS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W

**Table 15.13 CSIHnCTL0 Register Contents**

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. Clock supply to internal circuits stops.  If CSIHnPWR is cleared (to 0) during communication, ongoing communication is immediately aborted. In this case, the communication setting must be reconfigured.
6	CSIHnTXE	Enables/disables transmission. 0: Disables transmission. 1: Enables transmission.
5	CSIHnRXE	Enables/disables reception. 0:Disables reception. 1: Enables reception.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIHnJOBE	Stops communication at the end of the current job (communication ends if data is written to the transmission buffer when CSIHnTX0W.CSIHnEOJ = 1 (job completion)). 0: Communication stop is not requested. 1: Stops communication. This bit can be used to abort an ongoing job. This bit is cleared to 0 automatically. Even if this bit is set to 1, the read value is always 0. In FIFO mode, the pointer must be cleared by setting CSIHnSTCR0.CSIHnPCT = 1 before the next communication is started
0	CSIHnMBS	Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIH memory is used for transmission and/or reception data. 1: Direct access mode CSIH memory is bypassed.

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake functionality, and job mode. It selects the active output level of each chip select signal and the behavior of the chip select signals after the transfer of the final data.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIHn\_base> + 0010H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CSIHn SLRS	—	—	—	—	—	CSIHn PHE	CSIHn CKR	CSIHn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn CSL7	CSIHn CSL6	CSIHn CSL5	CSIHn CSL4	CSIHn CSL3	CSIHn CSL2	CSIHn CSL1	CSIHn CSL0	CSIHn EDLE	CSIHn JE	CSIHn DCS	CSIHn CSR1	CSIHn LBM	CSIHn SIT	CSIHn HSE	CSIHn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.14 CSIHnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CSIHnSLRS	Sets the internal synchronization timing for receive data input. 0: Rising edge of PCLK 1: Falling edge of PCLK For differences by the setting, see Data Sheet.
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	CSIHnPHE	Sets the CPU-controlled priority-based communication function. 0: The CPU-controlled high-priority communication function is disabled. 1: The CPU-controlled high-priority communication function is enabled. To enable the CPU-controlled high-priority communication function, set this bit to 1 and set CSIHnJE = 1. This bit can only be set in transmit-only buffer mode.
17	CSIHnCKR	CSIHTSCK Clock Inversion Function 0: The default level of CSIHTSCK is high 1: The default level of CSIHTSCK is low For details, see <b>Section 15.3.11, CSIHnCFGx — CSIHn Configuration Register x</b> .
16	CSIHnSLIT	Selects the timing of interrupt INTCSIHTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: As soon as the contents of the CSIHnTX0W/H register are transferred to the shift register, an interrupt is generated (this function is activated only in direct access mode/transmit-only buffer mode). For details, see <b>Section 15.4.3, INTCSIHTIC (Communication Status Interrupt)</b> .
15 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHTCSSx). 0: Chip select is active low. 1: Chip select is active high. For details, see <b>Section 15.5.3, Chip Selection (CS) Features</b> .
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Disables extended data length mode. 1: Enables extended data length mode. For details, see <b>Section 15.5.8.2, Data Length Greater than 16 Bits</b> .

**Table 15.14 CSIhnCTL1 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
6	CSIhnJE	<p>Enables/disables job mode. 0: Disables job mode. 1: Enables job mode.</p> <p>For details, see <b>Section 15.5.3.3, Job Concept</b>. The CSIhnCTL0.CSIhnJOBE, CSIhnTX0W.CSIhnEOJ, and CSIhnTX0W.CSIhnCIRE bits are enabled only when CSIhnJE = 1. Setting this bit in slave mode is prohibited. In addition, to enable the CPU-controlled high-priority communication function, set this bit to 1 as well as CSIhnPHE = 1.</p>
5	CSIhnDCS	<p>Enables/disables data consistency check. 0: Disables data consistency check. 1: Enables data consistency check.</p> <p>For details, see <b>Section 15.5.12.1, Data Consistency Check</b>.</p>
4	CSIhnCSRI	<p>Defines chip select signal behavior after last data transfer. 0: Chip select signal retains the active level. 1: Chip select signal returns to the inactive level.</p> <p>The last data is determined at the interrupt timing in direct access mode or FIFO mode. When CSIhnCTL1.CSIhnSLIT = 1, the last data is determined in direct access mode.</p>
3	CSIhnLBM	<p>Controls loop-back mode (LBM). 0: Deactivates loop-back mode. 1: Activates loop-back mode.</p> <p>For details, see <b>Section 15.5.13, Loop-Back Mode</b>.</p>
2	CSIhnSIT	<p>Selects interrupt delay mode. 0: No delay is generated. 1: Half clock delay is generated for all interrupts.</p> <p>This bit is only valid in master mode. In slave mode, no delay is generated.</p> <p>For details, see <b>Section 15.4.2, Interrupt Delay</b>.</p>
1	CSIhnHSE	<p>Enables/disables the handshake function. 0: Disables the handshake function. 1: Enables the handshake function.</p> <p>For details see <b>Section 15.5.11, Handshake Function</b>.</p>
0	CSIhnSSE	<p>Enables/disables the slave select function. 0: Input signal <u>CSIHTSSI</u> is disabled. 1: Input signal <u>CSIHTSSI</u> is recognized.</p> <p>If the slave select function is not used, this bit must be set to 0 (see also <b>Section 15.5.2, Master/Slave Connections</b>).</p>

Details about CSIhnCTL1.CSIhnSSE are shown in the following tables.

**Table 15.15 Operation of the Slave Select Function during Reception**

<b>CSIhnCTL0. CSIhnRXE</b>	<b>CSIhnCTL1. CSIhnSSE</b>	<b>CSIHTSSI</b>	<b>Receive Operation</b>
0	—	—	Reception is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

**Table 15.16 Operation of the Slave Select Function during Transmission**

<b>CSIhnCTL0. CSIhnTXE</b>	<b>CSIhnCTL1. CSIhnSSE</b>	<b>CSIHTSSI</b>	<b>Transmit Operation</b>
0	—	—	Transmission is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

**CAUTION**

When setting this register, see **Table 15.32, Notes on Setting Registers.**

### 15.3.4 CSI<sub>n</sub>CTL2 — CSI<sub>n</sub> Control Register 2

This register selects operating mode and the reference clock value, and specifies the transfer clock frequency.

For details see **Section 15.5.5, Transmission Clock Selection.**

**Access:** This register can be read or written in 16-bit units.

**Address:** <CSI<sub>n</sub>\_base> + 0014<sub>H</sub>

**Value after reset:** E000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				CSI <sub>n</sub> PRS[2:0]	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 15.17 CSI<sub>n</sub>CTL2 Register Contents**

Bit Position	Bit Name	Function																																				
15 to 13	CSI <sub>n</sub> PRS[2:0]	These bits select the operation mode and the reference clock value.																																				
		<table border="1"> <thead> <tr> <th>CSI<sub>n</sub>PRS2</th> <th>CSI<sub>n</sub>PRS1</th> <th>CSI<sub>n</sub>PRS0</th> <th>Selection of Reference Clock (PRSOUT)</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>PCLK (Master mode)</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>PCLK/2 (Master mode)</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>PCLK/4 (Master mode)</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>PCLK/8 (Master mode)</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>PCLK/16 (Master mode)</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>PCLK/32 (Master mode)</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>PCLK/64 (Master mode)</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>External clock via CSIHTSCK(in) (Slave mode)</td></tr> </tbody> </table>	CSI <sub>n</sub> PRS2	CSI <sub>n</sub> PRS1	CSI <sub>n</sub> PRS0	Selection of Reference Clock (PRSOUT)	0	0	0	PCLK (Master mode)	0	0	1	PCLK/2 (Master mode)	0	1	0	PCLK/4 (Master mode)	0	1	1	PCLK/8 (Master mode)	1	0	0	PCLK/16 (Master mode)	1	0	1	PCLK/32 (Master mode)	1	1	0	PCLK/64 (Master mode)	1	1	1	External clock via CSIHTSCK(in) (Slave mode)
CSI <sub>n</sub> PRS2	CSI <sub>n</sub> PRS1	CSI <sub>n</sub> PRS0	Selection of Reference Clock (PRSOUT)																																			
0	0	0	PCLK (Master mode)																																			
0	0	1	PCLK/2 (Master mode)																																			
0	1	0	PCLK/4 (Master mode)																																			
0	1	1	PCLK/8 (Master mode)																																			
1	0	0	PCLK/16 (Master mode)																																			
1	0	1	PCLK/32 (Master mode)																																			
1	1	0	PCLK/64 (Master mode)																																			
1	1	1	External clock via CSIHTSCK(in) (Slave mode)																																			
12 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																				

In master mode, the following bits are used to set the transfer clock frequency:

CSI<sub>n</sub>CTL2.CSI<sub>n</sub>PRS[2:0], CSI<sub>n</sub>CFGx.CSI<sub>n</sub>BRSS[1:0],  
CSI<sub>n</sub>BRSy.CSI<sub>n</sub>BRS[11:0]

In addition, any of the four different transfer clock frequency settings that are specified by the CSI<sub>n</sub>BRSy.CSI<sub>n</sub>BRS[11:0] bits is selected for each chip select signal. To select the transfer clock frequency setting for each chip select signal, use the CSI<sub>n</sub>CFGx.CSI<sub>n</sub>BRSS[1:0] bits.

The following table shows the relationship between CSIHnCFGx.CSIHnBRSS[1:0] and CSIHnBRSy.CSIHnBRS[11:0].

CSIHnCFGx. CSIHnBRSS[1:0]	Transfer Clock Frequency Setting Bit to be Selected
00	CSIHnBRS0.CSIHnBRS[11:0]
01	CSIHnBRS1.CSIHnBRS[11:0]
10	CSIHnBRS2.CSIHnBRS[11:0]
11	CSIHnBRS3.CSIHnBRS[11:0]

The following table shows the relationship between the transfer clock frequency and the transfer clock frequency setting (CSIHnBRSy[11:0]) selected by the CSIHnBRSS[1:0] bits when the bit value of the CSIHnPRS[2:0] bits is  $\alpha$ .

CSIHnBRSy[11:0]	Transfer clock frequency
0	BRG stopped
1	PCLK / ( $2^\alpha \times 1 \times 2$ )
2	PCLK / ( $2^\alpha \times 2 \times 2$ )
3	PCLK / ( $2^\alpha \times 3 \times 2$ )
4	PCLK / ( $2^\alpha \times 4 \times 2$ )
...	...
4095	PCLK / ( $2^\alpha \times 4095 \times 2$ )

When a time-out error is used in slave mode, the clock selected by this setting is used. In slave mode, the CSIHnPRS[2:0] bits are set to 111<sub>B</sub>. In this case, the prescaler has the same setting as when the CSIHnPRS[2:0] bits are set to 000<sub>B</sub>. When using a time-out error, set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000<sub>H</sub>.

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers.**

### 15.3.5 CSIhnSTR0 — CSIhn Status Register 0

This register indicates the status of CSIH.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <CSIhn\_base> + 0004<sub>H</sub>

**Value after reset:** 0000 0010<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIhnSRP[7:0]										CSIhnSPF[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIhn TMOE	CSIhn OFE	—	—	—	—	—	CSIhn HPST	CSIhn TSF	—	CSIhn FLF	CSIhn EMF	CSIhn DCE	—	CSIhn PE	CSIhn OVE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.18 CSIhnSTR0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	CSIhnSRP[7:0]	Indicates the number of received data in FIFO mode.
		<b>CSIhnSRP[7:0]</b> Description
		00 <sub>H</sub> Number of received data packets (0 to 128)
		... ...
		80 <sub>H</sub>
		Other than the above Undefined
These bits are cleared by CSIhnSTCR0.CSIhnPCT. In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00 <sub>H</sub> . In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIhnMCTL2.CSIhnND[7:0].		
23 to 16	CSIhnSPF[7:0]	Indicates the number of unsent data in FIFO mode. (The number of data written by the CPU is the number of sent data.)
		<b>CSIhnSPF[7:0]</b> Description
		00 <sub>H</sub> Number of unsent data (0 to 128)
		... ...
		80 <sub>H</sub>
		Other than the above Undefined
These bits are cleared by CSIhnSTCR0.CSIhnPCT. In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00 <sub>H</sub> . In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIhnMCTL2.CSIhnND[7:0].		

Table 15.18 CSIhnSTR0 Register Contents (2/3)

Bit Position	Bit Name	Function																																				
15	CSIhnTMOE	<p>Time-out Error Flag in FIFO Mode Indicates whether a time-out error was detected in FIFO mode. 0: No time-out error was detected. 1: A time-out error was detected.</p> <p>For details, see <b>Section 15.5.12.3, Time-Out Error.</b> This bit is cleared by CSIhnSTCR0.CSIhnTMOEC.</p> <p>When setting to 1 by time-out error detection and clearing to 0 by CSIhnSTCR0.CSIhnTMOEC occur simultaneously, setting to 1 takes precedence over clearing to 0.</p> <p>This bit is also initialized when CSIhnCTL0.CSIhnPWR is changed from 0 to 1 or from 1 to 0.</p>																																				
14	CSIhnOFE	<p>Overflow Error Flag in FIFO mode Indicates whether an overflow error was detected in FIFO mode. 0: No overflow error was detected. 1: An overflow error was detected.</p> <p>For details, see <b>Section 15.5.12.4, Overflow Error.</b> This bit is cleared by CSIhnSTCR0.CSIhnOFEC.</p> <p>When setting to 1 by overflow error detection and clearing to 0 by CSIhnSTCR0.CSIhnOFEC occur simultaneously, setting to 1 takes precedence over clearing to 0.</p> <p>This bit is also initialized when CSIhnCTL0.CSIhnPWR is changed from 0 to 1 or from 1 to 0.</p>																																				
13 to 9	Reserved	When read, the value after reset is returned.																																				
8	CSIhnHPST	<p>Communication Priority Indication Flag 0: Indicates low-priority communication is in progress. 1: Indicates high-priority communication is in progress.</p> <p>This bit always reads 0 if CPU-controlled high-priority communication is disabled (CSIhnCTL1.CSIhnPHE = 0).</p>																																				
7	CSIhnTSF	<p>Transfer Status Flag 0: Idle state 1: Communication is in progress or being prepared.</p> <p>The timing to set or clear this bit is as follows:</p> <table border="1"> <thead> <tr> <th rowspan="2">Master Mode</th> <th colspan="2">Timing to Set</th> <th rowspan="2">Timing to Clear</th> </tr> <tr> <th>Direct Access Mode, FIFO Mode</th> <th>Dual Buffer Mode, Transmit-Only Buffer Mode</th> </tr> </thead> <tbody> <tr> <td>Transmit-only mode</td> <td>Data is written to a transmit register (CSIhnTX0W/ CSIhnTX0H)</td> <td>CSIhnMCTL2.CSIHnBTST bit is set</td> <td>Within a half clock cycle the last serial clock edge</td> </tr> <tr> <td>Transmit/receive mode</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Receive-only mode</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th rowspan="2">Slave Mode</th> <th colspan="2">Timing to Set</th> <th rowspan="2">Timing to Clear</th> </tr> <tr> <th>Direct Access Mode, FIFO Mode</th> <th>Dual Buffer Mode, Transmit-Only Buffer Mode</th> </tr> </thead> <tbody> <tr> <td>Transmit-only mode</td> <td>Data is written to a transmit register (CSIhnTX0W/ CSIhnTX0H)</td> <td>Bit CSIhnMCTL2.CSIHnBTST is set</td> <td>Within a half clock cycle the last serial clock edge</td> </tr> <tr> <td>Transmit/receive mode</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Receive-only mode</td> <td>Input timing of CSIHTSCK</td> <td></td> <td></td> </tr> </tbody> </table>	Master Mode	Timing to Set		Timing to Clear	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit-Only Buffer Mode	Transmit-only mode	Data is written to a transmit register (CSIhnTX0W/ CSIhnTX0H)	CSIhnMCTL2.CSIHnBTST bit is set	Within a half clock cycle the last serial clock edge	Transmit/receive mode				Receive-only mode				Slave Mode	Timing to Set		Timing to Clear	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit-Only Buffer Mode	Transmit-only mode	Data is written to a transmit register (CSIhnTX0W/ CSIhnTX0H)	Bit CSIhnMCTL2.CSIHnBTST is set	Within a half clock cycle the last serial clock edge	Transmit/receive mode				Receive-only mode	Input timing of CSIHTSCK		
Master Mode	Timing to Set			Timing to Clear																																		
	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit-Only Buffer Mode																																				
Transmit-only mode	Data is written to a transmit register (CSIhnTX0W/ CSIhnTX0H)	CSIhnMCTL2.CSIHnBTST bit is set	Within a half clock cycle the last serial clock edge																																			
Transmit/receive mode																																						
Receive-only mode																																						
Slave Mode	Timing to Set		Timing to Clear																																			
	Direct Access Mode, FIFO Mode	Dual Buffer Mode, Transmit-Only Buffer Mode																																				
Transmit-only mode	Data is written to a transmit register (CSIhnTX0W/ CSIhnTX0H)	Bit CSIhnMCTL2.CSIHnBTST is set	Within a half clock cycle the last serial clock edge																																			
Transmit/receive mode																																						
Receive-only mode	Input timing of CSIHTSCK																																					
6	Reserved	When read, the value after reset is returned.																																				
5	CSIhnFLF	<p>A flag indicating that the buffer is full in FIFO mode. 0: FIFO buffer is not full. 1: FIFO buffer is full.</p> <p>This bit is cleared by CSIhnSTCR0.CSIhnPCT. The FIFO buffer might be filled with unsent data or received data.</p>																																				

**Table 15.18 CSIhnSTR0 Register Contents (3/3)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
4	CSIhnEMF	A flag indicating that the buffer is empty in FIFO mode. 0: FIFO buffer is not empty. 1: FIFO buffer is empty. This bit is set to 1 by CSIhnSTCR0.CSIhnPCT. This bit is set to 1 when CSIhnSTR0.CSIhnSRP[7:0] + CSIhnSTR0.CSIhnSPF[7:0] = 00 <sub>H</sub> . The FIFO buffer might be filled with unsent data or received data.
3	CSIhnDCE	Data Consistency Check Error Flag 0: No data consistency error is detected. 1: Data consistency error is detected. This bit is cleared by writing 1 to CSIhnSTCR0.CSIhnDCEC. When setting to 1 by data consistency error detection and clearing to 0 by CSIhnSTCR0.CSIhnDCEC occur simultaneously, setting to 1 takes precedence over clearing to 0. This bit is initialized when CSIhnCTL0.CSIhnPWR is changed from 0 to 1 or from 1 to 0.
2	Reserved	When read, the value after reset is returned.
1	CSIhnPE	Parity Error Flag 0: No parity error is detected. 1: Parity error is detected. This bit is cleared by writing 1 to CSIhnSTCR0.CSIhnPEC. When setting to 1 due to parity error detection and clearing to 0 by CSIhnSTCR0.CSIhnPEC occur simultaneously, setting to 1 by parity error detection takes precedence over clearing to 0. This bit is initialized when CSIhnCTL0.CSIhnPWR changes from 0 to 1 or from 1 to 0.
0	CSIhnOVE	Overrun Error Flag (Fixed to 0 in dual buffer mode) 0: No overrun error is detected. 1: Overrun error is detected. This bit is cleared by writing 1 to CSIhnSTCR0.CSIhnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIhnSTCR0.CSIhnOVEC occur simultaneously, setting to 1 by overrun error detection takes precedence over clearing to 0. This bit is initialized when CSIhnCTL0.CSIhnPWR changes from 0 to 1 or from 1 to 0.

**Table 15.19 Behaviors in Various Memory Modes**

<b>Bit Name</b>	<b>Bit Position</b>	<b>Direct Access Mode</b>	<b>FIFO Mode</b>	<b>Transmit-Only Buffer Mode</b>	<b>Dual Buffer Mode</b>
CSIHnSRP[7:0]	31 to 24	Fixed to 0	Number of received data	Fixed to 0	Fixed to 0
CSIHnSPF[7:0]	23 to 16	Fixed to 0	Number of untransmitted data packets	Fixed to 0	Fixed to 0
CSIHnTMOE	15	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnTSF	7	0: Idle state 1: Communication is in progress or being prepared			
CSIHnFLF	5	Fixed to 0	0: FIFO is not full 1: FIFO is full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: FIFO is not empty 1: FIFO is empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: No error is detected. 1: An error is detected.			
CSIHnPE	1	0: No error is detected. 1: An error is detected.			
CSIHnOVE	0	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	Fixed to 0

**CAUTION**

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.6 CSIhNSTCR0 — CSIhN Status Clear Register 0

This register clears the status flags of the CSIhNSTR0 status register.

**Access:** This register can be read or written in 16-bit units.  
When read, the value 0000<sub>H</sub> is always returned.

**Address:** <CSIhN\_base> + 0008<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIhN TMOEC	CSIhN OFEC	—	—	—	—	—	CSIhN PCT	—	—	—	—	CSIhN DCEC	—	CSIhN PEC	CSIhN OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R	R	R/W	R	R/W	R/W

Table 15.20 CSIhNSTCR0 Register Contents

Bit Position	Bit Name	Function
15	CSIhNTMOEC	Controls the time-out error flag clear command. 0: No operation. The read value is always 0. 1: Clears the time-out error flag (CSIhNSTR0.CSIhNTMOE).
14	CSIhNOFEC	Controls the overflow error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overflow error flag (CSIhNSTR0.CSIhNFE).
13 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CSIhNPCT	Controls the FIFO pointer clear command. 0: No operation. The read value is always 0. 1: Clears the following FIFO buffer pointers (in FIFO mode, dual buffer mode, and transmit-only buffer mode) and status bits.

FIFO Buffer Pointer	Status Bit
CSIhNMRWP0.CSIhNTRWA[6:0]	CSIhNSTR0.CSIhNSPF[7:0]
CSIhNMRWP0.CSIhNRRA[6:0]	CSIhNSTR0.CSIhNSRP[7:0]
CSIhNMCTL2.CSIhNSOP[6:0]	CSIhNSTR0.CSIhNLF
	CSIhNSTR0.CSIhNTSF

Additionally, the CSIhNSTR0.CSIhEMF bit is set to 1 (FIFO empty) (in FIFO mode only).

7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIhNDCEC	Controls the data consistency error flag clear command. 0: No operation. The read value is always 0. 1: Clears the data consistency error flag (CSIhNSTR0.CSIhDCE).
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIhNPEC	Controls the parity error flag clear command. 0: No operation. The read value is always 0. 1: Clears the parity error flag (CSIhNSTR0.CSIhPE).
0	CSIhNOVEC	Controls the overrun error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overrun error flag (CSIhNSTR0.CSIhOVE).

#### CAUTION

When setting this register, see Table 15.32, Notes on Setting Registers.

### 15.3.7 CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

**Access:** This register can be read or written in 16-bit units.

**Address:** <CSIHn\_base> + 1040<sub>H</sub>

**Value after reset:** 001F<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	CSIHn MMS[1:0]	—	—	—	—	CSIHnTO[4:0]	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

**Table 15.21 CSIHnMCTL0 Register Contents**

Bit Position	Bit Name	Function	
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.	
9 to 8	CSIHnMMS [1:0]	Selects the memory mode.	
	CSIHn MMS1	CSIHn MMS0	Description
	0	0	FIFO mode
	0	1	Dual buffer mode
	1	0	Transmit-only buffer mode
	1	1	Prohibited
After a change of the memory mode, the respective buffer pointers must be cleared by setting the CSIHnSTCRO.CSIHnPCT bit to 1. In direct access mode, the setting of these bits is ignored.			
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.	
4 to 0	CSIHnTO[4:0]	Selects the time-out setting in FIFO mode.	
	CSIHnTO[4:0]	Description	
	00000 <sub>B</sub>	No time-out is detected	
	00001 <sub>B</sub>	Time-out is (1 × 8 × BRG output clocks)	
	00010 <sub>B</sub>	Time-out is (2 × 8 × BRG output clocks)	
	...		
	11111 <sub>B</sub>	Time-out is (31 × 8 × BRG output clocks)	

#### CAUTION

Changing the time-out setting is only permitted when CSIHnCTL0.CSIHnPWR = 0.

Set the CSIHnTO[4:0] bits to 00000<sub>B</sub> in direct access mode, dual buffer mode, or transmit-only buffer mode (except FIFO mode).

For details about time-out detection, see also **Section 15.5.12.3, Time-Out Error.**

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers.**

### 15.3.8 CSIHnMCTL1 — CSIHn Memory Control Register 1

This register selects the conditions to generate the interrupt requests INTCSIHTIC and INTCSIHTIR in FIFO mode.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIHn\_base> + 1000<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—							CSIHnFES[6:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—							CSIHnFFS[6:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						

**Table 15.22 CSIHnMCTL1 Register Contents**

Bit Position	Bit name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CSIHnFES[6:0]	Selects the conditions to generate the INTCSIHTIC interrupt (transmit data empty) in FIFO mode. When the number of unsent data to be transmitted in FIFO (checked by the CSIHnSTRO.CSIHnSPF[7:0] bit) and CSIHnMCTL1.CSIHnFES[6:0] match, the FIFO empty flag (CSIHnSTRO.CSIHnEMF bit) is set to 1, and the INTCSIHTIC interrupt request is generated.
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CSIHnFFS[6:0]	Selects the conditions to generate the INTCSIHTIR interrupt (receive data full) in FIFO mode. When the number of received data in FIFO (checked by the CSIHnSTRO.CSIHnSRP[7:0] bit) and (128 - CSIHnMCTL1.CSIHnFFS[6:0]) match, the INTCSIHTIR interrupt request is generated.

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.9 CSIHnMCTL2 — CSIHn Memory Control Register 2

This register controls the operation of the memory in dual buffer or transmit-only buffer mode and triggers to start communication.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIHn\_base> + 1004<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
CSIHn BTST	—	—	—	—	—	—	—	—	CSIHnND[7:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
—	—	—	—	—	—	—	—	—	CSIHnSOP[6:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
R/W	R	R	R	R	R	R	R	R	R	R/W													

Table 15.23 CSIHnMCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function																																																		
31	CSIHnBTST	Provides a start trigger for buffer transfer. 0: No operation. 1: Issues the start transfer command. The read value is always 0.																																																		
<b>CAUTION</b>																																																				
This bit can only be used in dual buffer mode and transmit-only buffer mode.																																																				
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																																		
23 to 16	CSIHnND[7:0]	Specifies the number of data for each memory mode. The read value indicates the number of remaining communication data.																																																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CSIHn ND[7:0]</th> <th>Dual Buffer Mode</th> <th>Transmit-Only Buffer Mode</th> <th>FIFO Mode</th> <th>Direct Access Mode</th> </tr> </thead> <tbody> <tr> <td>00<sub>H</sub></td> <td>Send 0 data</td> <td>Send 0 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>01<sub>H</sub></td> <td>Send 1 data</td> <td>Send 1 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>3F<sub>H</sub></td> <td>Send 63 data</td> <td>Send 63 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>40<sub>H</sub></td> <td>Send 64 data</td> <td>Send 64 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>...</td> <td>Prohibited</td> <td>...</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>7F<sub>H</sub></td> <td>Prohibited</td> <td>Send 127 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>80<sub>H</sub></td> <td>Prohibited</td> <td>Send 128 data</td> <td>No influence</td> <td>No influence</td> </tr> <tr> <td>Other than the above</td> <td colspan="3" style="text-align: center;">Setting is prohibited.</td><td></td></tr> </tbody> </table>			CSIHn ND[7:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode	00 <sub>H</sub>	Send 0 data	Send 0 data	No influence	No influence	01 <sub>H</sub>	Send 1 data	Send 1 data	No influence	No influence	...	...	...	No influence	No influence	3F <sub>H</sub>	Send 63 data	Send 63 data	No influence	No influence	40 <sub>H</sub>	Send 64 data	Send 64 data	No influence	No influence	...	Prohibited	...	No influence	No influence	7F <sub>H</sub>	Prohibited	Send 127 data	No influence	No influence	80 <sub>H</sub>	Prohibited	Send 128 data	No influence	No influence	Other than the above	Setting is prohibited.			
CSIHn ND[7:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode																																																
00 <sub>H</sub>	Send 0 data	Send 0 data	No influence	No influence																																																
01 <sub>H</sub>	Send 1 data	Send 1 data	No influence	No influence																																																
...	...	...	No influence	No influence																																																
3F <sub>H</sub>	Send 63 data	Send 63 data	No influence	No influence																																																
40 <sub>H</sub>	Send 64 data	Send 64 data	No influence	No influence																																																
...	Prohibited	...	No influence	No influence																																																
7F <sub>H</sub>	Prohibited	Send 127 data	No influence	No influence																																																
80 <sub>H</sub>	Prohibited	Send 128 data	No influence	No influence																																																
Other than the above	Setting is prohibited.																																																			
The values are automatically decremented after data transfer (not decremented in direct access mode).																																																				
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																																		

**Table 15.23 CSIhnMCTL2 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>		
6 to 0	CSIhnSOP[6:0]	Selects the pointer of the data to be sent. If communication is forced to stop by setting CSIhnCTL0.CSIhnPWR to 0 or CSIhnSTCR0.CSIhnPCT to 1, these bits are cleared by hardware. In FIFO mode, these bits indicate the send address.		
CSIhn SOP[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00 <sub>H</sub>	0000 <sub>H</sub>	0000 <sub>H</sub>	0000 <sub>H</sub>	No influence
01 <sub>H</sub>	0004 <sub>H</sub>	0004 <sub>H</sub>	0004 <sub>H</sub>	No influence
...	...	...	...	No influence
3F <sub>H</sub>	00FC <sub>H</sub>	00FC <sub>H</sub>	00FC <sub>H</sub>	No influence
40 <sub>H</sub>	Prohibited	0100 <sub>H</sub>	0100 <sub>H</sub>	No influence
...	Prohibited	...	...	No influence
7F <sub>H</sub>	Prohibited	01FC <sub>H</sub>	01FC <sub>H</sub>	No influence

**CAUTION**

In direct access mode, these bits are not incremented.

**CAUTION**

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.10 CSIhnMRWP0 — CSIhn Memory Read/Write Pointer Register 0

This register sets the pointers for reading from and writing to the dual buffer or transmit-only buffer.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIhn\_base> + 1018<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIhnRRA[6:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIhnTRWA[6:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						

Table 15.24 CSIhnMRWP0 Register Contents (1/2)

Bit Position	Bit Name	Function				
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.				
22 to 16	CSIhnRRA[6:0]	Selects the read pointer of the receive buffer.				
	CSIhnRRA[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode	
	00 <sub>H</sub>	0000 <sub>H</sub>	No influence	0000 <sub>H</sub>	No influence	
	01 <sub>H</sub>	0004 <sub>H</sub>	No influence	0004 <sub>H</sub>	No influence	
	...	...	No influence	...	No influence	
	3F <sub>H</sub>	00FC <sub>H</sub>	No influence	00FC <sub>H</sub>	No influence	
	40 <sub>H</sub>	Prohibited	No influence	0100 <sub>H</sub>	No influence	
	...	Prohibited	No influence	...	No influence	
	7F <sub>H</sub>	Prohibited	No influence	01FC <sub>H</sub>	No influence	
These bits are automatically incremented when received data is read. If an overrun error is generated while reading the CSIhnRX0W or CSIhnRX0H register, the read pointer is not incremented. These bits are cleared when CSIhnSTCR0.CSIhnPCT is set to 1. In direct access mode and transmit-only buffer mode, these bits are not incremented. To perform write access in transmit-only buffer mode, set 0000 <sub>H</sub> to these bits. In FIFO mode, these bits indicate the read address of the received data.						
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.				

**Table 15.24 CSI<sub>n</sub>MRWP0 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>				
6 to 0	CSI <sub>n</sub> TRWA [6:0]	Selects the read/write pointer of the transmit buffer.				
		<b>CSI<sub>n</sub> TRWA[6:0]</b>	<b>Dual Buffer Mode</b>	<b>Transmit-Only Buffer Mode</b>	<b>FIFO Mode</b>	<b>Direct Access Mode</b>
	00 <sub>H</sub>	0000 <sub>H</sub>		0000 <sub>H</sub>	0000 <sub>H</sub>	No influence
	01 <sub>H</sub>	0004 <sub>H</sub>		0004 <sub>H</sub>	0004 <sub>H</sub>	No influence
	...	...	...	...	...	No influence
	3F <sub>H</sub>	00FC <sub>H</sub>		00FC <sub>H</sub>	00FC <sub>H</sub>	No influence
	40 <sub>H</sub>	Prohibited		0100 <sub>H</sub>	0100 <sub>H</sub>	No influence
	...	Prohibited	...	...	...	No influence
	7F <sub>H</sub>	Prohibited	01FC <sub>H</sub>	01FC <sub>H</sub>	01FC <sub>H</sub>	No influence

These bits are automatically incremented when the transmission data is written or read.

These bits are cleared when CSI<sub>n</sub>STCR0.CSI<sub>n</sub>PCT is set to 1.

In direct access mode, these bits are not incremented.

In FIFO mode, these bits indicate the read/write address of transmission data.

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.11 CSIHnCFGx — CSIHn Configuration Register x

These eight registers configure the prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase and data phase, idle enforcement configuration, idle time, hold time, inter-data time, and setup for each chip select signal, CSIHTCSSx.

#### Slave mode

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is effective.

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

In slave mode, set all bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG7 registers to 0.

**Access:** This register can be read or written in 32-bit units.

**Address:** CSIHnCFG0: <CSIHn\_base> + 1044<sub>H</sub>  
 CSIHnCFG1: <CSIHn\_base> + 1048<sub>H</sub>  
 CSIHnCFG2: <CSIHn\_base> + 104C<sub>H</sub>  
 CSIHnCFG3: <CSIHn\_base> + 1050<sub>H</sub>  
 CSIHnCFG4: <CSIHn\_base> + 1054<sub>H</sub>  
 CSIHnCFG5: <CSIHn\_base> + 1058<sub>H</sub>  
 CSIHnCFG6: <CSIHn\_base> + 105C<sub>H</sub>  
 CSIHnCFG7: <CSIHn\_base> + 1060<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnBRSSx [1:0]	CSIHnPSx[1:0]		CSIHnDLSx[3:0]		—	—	—	—	—	CSIHnRCBx	CSIHnDIRx	CSIHnCKPx	CSIHnDAPx		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnIDLx	CSIHnIDX[2:0]		CSIHnHDx[3:0]		CSIHnINx[3:0]		CSIHnSPx[3:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.25 CSIhnCFGx Register Contents (1/4)

Bit Position	Bit Name	Function																				
31, 30	CSIhnBRSSx [1:0]	<p>These bits select the baud rate setting register (CSIhnBRSy).</p> <table border="1"> <thead> <tr> <th>CSIhn BRSSx1</th> <th>CSIhn BRSSx0</th> <th>Baud Rate Setting Register Selection</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>The transfer clock frequency is set according to the CSIhnBRS0 setting.</td> </tr> <tr> <td>0</td> <td>1</td> <td>The transfer clock frequency is set according to the CSIhnBRS1 setting.</td> </tr> <tr> <td>1</td> <td>0</td> <td>The transfer clock frequency is set according to the CSIhnBRS2 setting.</td> </tr> <tr> <td>1</td> <td>1</td> <td>The transfer clock frequency is set according to the CSIhnBRS3 setting.</td> </tr> </tbody> </table> <p>The maximum value for setting the transfer clock frequency, in accordance with the CSIhnCTL2.CSIhnPRS[2:0] setting, must be as follows:  Master mode: PCLK/4  Slave mode: PCLK/6</p>	CSIhn BRSSx1	CSIhn BRSSx0	Baud Rate Setting Register Selection	0	0	The transfer clock frequency is set according to the CSIhnBRS0 setting.	0	1	The transfer clock frequency is set according to the CSIhnBRS1 setting.	1	0	The transfer clock frequency is set according to the CSIhnBRS2 setting.	1	1	The transfer clock frequency is set according to the CSIhnBRS3 setting.					
CSIhn BRSSx1	CSIhn BRSSx0	Baud Rate Setting Register Selection																				
0	0	The transfer clock frequency is set according to the CSIhnBRS0 setting.																				
0	1	The transfer clock frequency is set according to the CSIhnBRS1 setting.																				
1	0	The transfer clock frequency is set according to the CSIhnBRS2 setting.																				
1	1	The transfer clock frequency is set according to the CSIhnBRS3 setting.																				
29, 28	CSIhnPSx[1:0]	<p>Selects the parity for transmitting or receiving chip select signal x.</p> <table border="1"> <thead> <tr> <th>CSIhn PSx1</th> <th>CSIhn PSx0</th> <th>Transmission</th> <th>Reception</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Does not transmit any parity bit.</td> <td>Does not wait for reception of the parity bit.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Adds a parity bit fixed to 0.</td> <td>Waits for reception of the parity bit but does not evaluate it.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Adds the odd parity bit.</td> <td>Waits for the odd parity bit.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Adds the even parity bit.</td> <td>Waits for the even parity bit.</td> </tr> </tbody> </table>	CSIhn PSx1	CSIhn PSx0	Transmission	Reception	0	0	Does not transmit any parity bit.	Does not wait for reception of the parity bit.	0	1	Adds a parity bit fixed to 0.	Waits for reception of the parity bit but does not evaluate it.	1	0	Adds the odd parity bit.	Waits for the odd parity bit.	1	1	Adds the even parity bit.	Waits for the even parity bit.
CSIhn PSx1	CSIhn PSx0	Transmission	Reception																			
0	0	Does not transmit any parity bit.	Does not wait for reception of the parity bit.																			
0	1	Adds a parity bit fixed to 0.	Waits for reception of the parity bit but does not evaluate it.																			
1	0	Adds the odd parity bit.	Waits for the odd parity bit.																			
1	1	Adds the even parity bit.	Waits for the even parity bit.																			
27 to 24	CSIhnDLSx [3:0]	<p>Selects the data length for chip select signal x.</p> <table border="1"> <thead> <tr> <th>CSIhnDLSx[3:0]</th> <th>Data Length</th> </tr> </thead> <tbody> <tr> <td>0000<sub>B</sub></td> <td>16 bits</td> </tr> <tr> <td>0001<sub>B</sub></td> <td>1 bit</td> </tr> <tr> <td>0010<sub>B</sub></td> <td>2 bits</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>1111<sub>B</sub></td> <td>15 bits</td> </tr> </tbody> </table>	CSIhnDLSx[3:0]	Data Length	0000 <sub>B</sub>	16 bits	0001 <sub>B</sub>	1 bit	0010 <sub>B</sub>	2 bits	...	...	1111 <sub>B</sub>	15 bits								
CSIhnDLSx[3:0]	Data Length																					
0000 <sub>B</sub>	16 bits																					
0001 <sub>B</sub>	1 bit																					
0010 <sub>B</sub>	2 bits																					
...	...																					
1111 <sub>B</sub>	15 bits																					
<b>CAUTION</b>																						
<p>When CSIhnTX0W.CSIhnEDL = 1, the setting of this bit has no effect.  When CSIhnTX0W.CSIhnEDL = 0 (the data length is 16 bits), the setting of this bit is valid. Setting "1 bit" is only enabled if the previous transmit data was 16 bits with CSIhnEDL = 1.</p>																						
23 to 20	Reserved	<p>When read, the value after reset is returned. When writing, write the value after reset.</p>																				
19	CSIhnRCBx	<p>Selects the recessive configuration for broadcasting for chip select signal x.  0: Dominant (higher priority)  1: Recessive (lower priority)</p> <p>For details, see <b>Section 15.5.3.1, Configuration Registers</b></p>																				
18	CSIhnDIRx	<p>Selects the serial data direction of chip select signal x.  0: Data is transmitted/received with MSB first.  1: Data is transmitted/received with LSB first.</p> <p>For details, see <b>Section 15.5.9, Serial Data Direction Selection</b>.</p>																				

Table 15.25 CSIhnCFGx Register Contents (2/4)

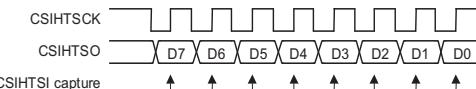
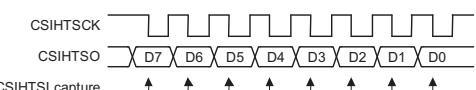
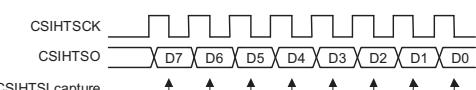
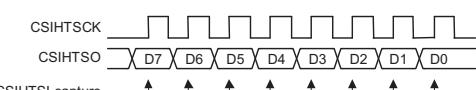
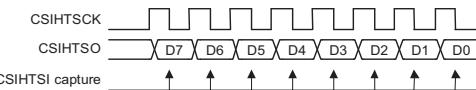
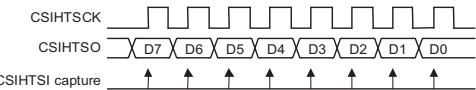
Bit Position	Bit Name	Function
17	CSIhnCKPx	CSIhnCKPx: Clock phase selection bit
16	CSIhnDAPx	CSIhnDAPx: Data phase selection bit • CSIhnCTL1.CSIhnCKR = 0
<b>CSIhn CKPx      CSIhn DAPx      Clock Phase and Data Phase Selection</b>		
0	0	
0	1	
1	0	
1	1	
• CSIhnCTL1.CSIhnCKR = 1		
<b>CSIhn CKPx      CSIhn DAPx      Clock Phase and Data Phase Selection</b>		
0	0	
0	1	
1	—	Setting prohibited
15	CSIhnIDLx	Selects the idle enforcement configuration for chip select signal x 0: If the CSIhnTX0W.CSIhnCSx settings of two consecutive transfers are different, an idle state is inserted between two transfers. If the CSIhnTX0W.CSIhnCSx settings of two consecutive transfers are the same, an idle state is not inserted between two transfers. 1: Regardless of the CSIhnTX0W.CSIhnCSx settings of two consecutive transfers, an idle state are inserted between two transfers. This bit is only available in master mode. For details about the enforced idle state, see <b>Section 15.5.15, Enforced Chip Select Idle Setting.</b>

Table 15.25 CSIhnCFGx Register Contents (3/4)

Bit Position	Bit Name	Function	
14 to 12	CSIhnIDX[2:0]	Selects the idle time for chip select signal x.	
<b>CSIhn IDX[2:0]      Idle time</b>			
	000 <sub>B</sub>	0.5 transmission clock cycle	
	001 <sub>B</sub>	1.0 transmission clock cycle	
	010 <sub>B</sub>	1.5 transmission clock cycles	
	011 <sub>B</sub>	2.5 transmission clock cycles	
	100 <sub>B</sub>	3.5 transmission clock cycles	
	101 <sub>B</sub>	4.5 transmission clock cycles	
	110 <sub>B</sub>	6.5 transmission clock cycles	
	111 <sub>B</sub>	8.5 transmission clock cycles	
These bits are only available in master mode.			
11 to 8	CSIhnHDx[3:0]	Specifies the hold time for chip select signal x in transmission clock cycles.	
<b>CSIhn HDx[3:0]      Hold time when CSIhnCTL1.CSIhnSIT is 0      Hold time when CSIhnCTL1.CSIhnSIT is 1</b>			
	0000 <sub>B</sub>	0.5 transmission clock cycle	1.0 transmission clock cycle
	0001 <sub>B</sub>	1.0 transmission clock cycle	1.5 transmission clock cycles
	0010 <sub>B</sub>	1.5 transmission clock cycles	2.0 transmission clock cycles
	0011 <sub>B</sub>	2.5 transmission clock cycles	3.0 transmission clock cycles
	0100 <sub>B</sub>	3.5 transmission clock cycles	4.0 transmission clock cycles
	0101 <sub>B</sub>	4.5 transmission clock cycles	5.0 transmission clock cycles
	0110 <sub>B</sub>	6.5 transmission clock cycles	7.0 transmission clock cycles
	0111 <sub>B</sub>	8.5 transmission clock cycles	9.0 transmission clock cycles
	1000 <sub>B</sub>	9.5 transmission clock cycles	10.0 transmission clock cycles
	1001 <sub>B</sub>	10.5 transmission clock cycles	11.0 transmission clock cycles
	1010 <sub>B</sub>	11.5 transmission clock cycles	12.0 transmission clock cycles
	1011 <sub>B</sub>	12.5 transmission clock cycles	13.0 transmission clock cycles
	1100 <sub>B</sub>	14.5 transmission clock cycles	15.0 transmission clock cycles
	1101 <sub>B</sub>	16.5 transmission clock cycles	17.0 transmission clock cycles
	1110 <sub>B</sub>	18.5 transmission clock cycles	19.0 transmission clock cycles
	1111 <sub>B</sub>	20.5 transmission clock cycles	21.0 transmission clock cycles
These bits are only available in master mode.			

**Table 15.25 CSIhnCFGx Register Contents (4/4)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>																																																			
7 to 4	CSIhnINx[3:0]	Specifies the inter-data time for chip select signal x in transmission clock cycles.																																																			
		<table border="1"> <thead> <tr> <th><b>CSIhnINx[3:0]</b></th><th><b>Inter-Data Time when CSIhnCTL1.CSIhnSIT is 0</b></th><th><b>Inter-Data Time when CSIhnCTL1.CSIhnSIT is 1</b></th></tr> </thead> <tbody> <tr><td>0000<sub>B</sub></td><td>0.0 transmission clock cycle</td><td>0.5 transmission clock cycle</td></tr> <tr><td>0001<sub>B</sub></td><td>0.5 transmission clock cycle</td><td>1.0 transmission clock cycle</td></tr> <tr><td>0010<sub>B</sub></td><td>1.0 transmission clock cycle</td><td>1.5 transmission clock cycles</td></tr> <tr><td>0011<sub>B</sub></td><td>2.0 transmission clock cycles</td><td>2.5 transmission clock cycles</td></tr> <tr><td>0100<sub>B</sub></td><td>3.0 transmission clock cycles</td><td>3.5 transmission clock cycles</td></tr> <tr><td>0101<sub>B</sub></td><td>4.0 transmission clock cycles</td><td>4.5 transmission clock cycles</td></tr> <tr><td>0110<sub>B</sub></td><td>6.0 transmission clock cycles</td><td>6.5 transmission clock cycles</td></tr> <tr><td>0111<sub>B</sub></td><td>8.0 transmission clock cycles</td><td>8.5 transmission clock cycles</td></tr> <tr><td>1000<sub>B</sub></td><td>9.0 transmission clock cycles</td><td>9.5 transmission clock cycles</td></tr> <tr><td>1001<sub>B</sub></td><td>10.0 transmission clock cycles</td><td>10.5 transmission clock cycles</td></tr> <tr><td>1010<sub>B</sub></td><td>11.0 transmission clock cycles</td><td>11.5 transmission clock cycles</td></tr> <tr><td>1011<sub>B</sub></td><td>12.0 transmission clock cycles</td><td>12.5 transmission clock cycles</td></tr> <tr><td>1100<sub>B</sub></td><td>14.0 transmission clock cycles</td><td>14.5 transmission clock cycles</td></tr> <tr><td>1101<sub>B</sub></td><td>16.0 transmission clock cycles</td><td>16.5 transmission clock cycles</td></tr> <tr><td>1110<sub>B</sub></td><td>18.0 transmission clock cycles</td><td>18.5 transmission clock cycles</td></tr> <tr><td>1111<sub>B</sub></td><td>20.0 transmission clock cycles</td><td>20.5 transmission clock cycles</td></tr> </tbody> </table>	<b>CSIhnINx[3:0]</b>	<b>Inter-Data Time when CSIhnCTL1.CSIhnSIT is 0</b>	<b>Inter-Data Time when CSIhnCTL1.CSIhnSIT is 1</b>	0000 <sub>B</sub>	0.0 transmission clock cycle	0.5 transmission clock cycle	0001 <sub>B</sub>	0.5 transmission clock cycle	1.0 transmission clock cycle	0010 <sub>B</sub>	1.0 transmission clock cycle	1.5 transmission clock cycles	0011 <sub>B</sub>	2.0 transmission clock cycles	2.5 transmission clock cycles	0100 <sub>B</sub>	3.0 transmission clock cycles	3.5 transmission clock cycles	0101 <sub>B</sub>	4.0 transmission clock cycles	4.5 transmission clock cycles	0110 <sub>B</sub>	6.0 transmission clock cycles	6.5 transmission clock cycles	0111 <sub>B</sub>	8.0 transmission clock cycles	8.5 transmission clock cycles	1000 <sub>B</sub>	9.0 transmission clock cycles	9.5 transmission clock cycles	1001 <sub>B</sub>	10.0 transmission clock cycles	10.5 transmission clock cycles	1010 <sub>B</sub>	11.0 transmission clock cycles	11.5 transmission clock cycles	1011 <sub>B</sub>	12.0 transmission clock cycles	12.5 transmission clock cycles	1100 <sub>B</sub>	14.0 transmission clock cycles	14.5 transmission clock cycles	1101 <sub>B</sub>	16.0 transmission clock cycles	16.5 transmission clock cycles	1110 <sub>B</sub>	18.0 transmission clock cycles	18.5 transmission clock cycles	1111 <sub>B</sub>	20.0 transmission clock cycles	20.5 transmission clock cycles
<b>CSIhnINx[3:0]</b>	<b>Inter-Data Time when CSIhnCTL1.CSIhnSIT is 0</b>	<b>Inter-Data Time when CSIhnCTL1.CSIhnSIT is 1</b>																																																			
0000 <sub>B</sub>	0.0 transmission clock cycle	0.5 transmission clock cycle																																																			
0001 <sub>B</sub>	0.5 transmission clock cycle	1.0 transmission clock cycle																																																			
0010 <sub>B</sub>	1.0 transmission clock cycle	1.5 transmission clock cycles																																																			
0011 <sub>B</sub>	2.0 transmission clock cycles	2.5 transmission clock cycles																																																			
0100 <sub>B</sub>	3.0 transmission clock cycles	3.5 transmission clock cycles																																																			
0101 <sub>B</sub>	4.0 transmission clock cycles	4.5 transmission clock cycles																																																			
0110 <sub>B</sub>	6.0 transmission clock cycles	6.5 transmission clock cycles																																																			
0111 <sub>B</sub>	8.0 transmission clock cycles	8.5 transmission clock cycles																																																			
1000 <sub>B</sub>	9.0 transmission clock cycles	9.5 transmission clock cycles																																																			
1001 <sub>B</sub>	10.0 transmission clock cycles	10.5 transmission clock cycles																																																			
1010 <sub>B</sub>	11.0 transmission clock cycles	11.5 transmission clock cycles																																																			
1011 <sub>B</sub>	12.0 transmission clock cycles	12.5 transmission clock cycles																																																			
1100 <sub>B</sub>	14.0 transmission clock cycles	14.5 transmission clock cycles																																																			
1101 <sub>B</sub>	16.0 transmission clock cycles	16.5 transmission clock cycles																																																			
1110 <sub>B</sub>	18.0 transmission clock cycles	18.5 transmission clock cycles																																																			
1111 <sub>B</sub>	20.0 transmission clock cycles	20.5 transmission clock cycles																																																			
		These bits are only available in master mode.																																																			
3 to 0	CSIhnSPx[3:0]	Specifies the setup time for chip select signal x in transmission clock cycles.																																																			
		<table border="1"> <thead> <tr> <th><b>CSIhnSPx[3:0]</b></th><th><b>Setup Time</b></th></tr> </thead> <tbody> <tr><td>0000<sub>B</sub></td><td>0.5 transmission clock cycle</td></tr> <tr><td>0001<sub>B</sub></td><td>1.0 transmission clock cycle</td></tr> <tr><td>0010<sub>B</sub></td><td>1.5 transmission clock cycles</td></tr> <tr><td>0011<sub>B</sub></td><td>2.5 transmission clock cycles</td></tr> <tr><td>0100<sub>B</sub></td><td>3.5 transmission clock cycles</td></tr> <tr><td>0101<sub>B</sub></td><td>4.5 transmission clock cycles</td></tr> <tr><td>0110<sub>B</sub></td><td>6.5 transmission clock cycles</td></tr> <tr><td>0111<sub>B</sub></td><td>8.5 transmission clock cycles</td></tr> <tr><td>1000<sub>B</sub></td><td>9.5 transmission clock cycles</td></tr> <tr><td>1001<sub>B</sub></td><td>10.5 transmission clock cycles</td></tr> <tr><td>1010<sub>B</sub></td><td>11.5 transmission clock cycles</td></tr> <tr><td>1011<sub>B</sub></td><td>12.5 transmission clock cycles</td></tr> <tr><td>1100<sub>B</sub></td><td>14.5 transmission clock cycles</td></tr> <tr><td>1101<sub>B</sub></td><td>16.5 transmission clock cycles</td></tr> <tr><td>1110<sub>B</sub></td><td>18.5 transmission clock cycles</td></tr> <tr><td>1111<sub>B</sub></td><td>20.5 transmission clock cycles</td></tr> </tbody> </table>	<b>CSIhnSPx[3:0]</b>	<b>Setup Time</b>	0000 <sub>B</sub>	0.5 transmission clock cycle	0001 <sub>B</sub>	1.0 transmission clock cycle	0010 <sub>B</sub>	1.5 transmission clock cycles	0011 <sub>B</sub>	2.5 transmission clock cycles	0100 <sub>B</sub>	3.5 transmission clock cycles	0101 <sub>B</sub>	4.5 transmission clock cycles	0110 <sub>B</sub>	6.5 transmission clock cycles	0111 <sub>B</sub>	8.5 transmission clock cycles	1000 <sub>B</sub>	9.5 transmission clock cycles	1001 <sub>B</sub>	10.5 transmission clock cycles	1010 <sub>B</sub>	11.5 transmission clock cycles	1011 <sub>B</sub>	12.5 transmission clock cycles	1100 <sub>B</sub>	14.5 transmission clock cycles	1101 <sub>B</sub>	16.5 transmission clock cycles	1110 <sub>B</sub>	18.5 transmission clock cycles	1111 <sub>B</sub>	20.5 transmission clock cycles																	
<b>CSIhnSPx[3:0]</b>	<b>Setup Time</b>																																																				
0000 <sub>B</sub>	0.5 transmission clock cycle																																																				
0001 <sub>B</sub>	1.0 transmission clock cycle																																																				
0010 <sub>B</sub>	1.5 transmission clock cycles																																																				
0011 <sub>B</sub>	2.5 transmission clock cycles																																																				
0100 <sub>B</sub>	3.5 transmission clock cycles																																																				
0101 <sub>B</sub>	4.5 transmission clock cycles																																																				
0110 <sub>B</sub>	6.5 transmission clock cycles																																																				
0111 <sub>B</sub>	8.5 transmission clock cycles																																																				
1000 <sub>B</sub>	9.5 transmission clock cycles																																																				
1001 <sub>B</sub>	10.5 transmission clock cycles																																																				
1010 <sub>B</sub>	11.5 transmission clock cycles																																																				
1011 <sub>B</sub>	12.5 transmission clock cycles																																																				
1100 <sub>B</sub>	14.5 transmission clock cycles																																																				
1101 <sub>B</sub>	16.5 transmission clock cycles																																																				
1110 <sub>B</sub>	18.5 transmission clock cycles																																																				
1111 <sub>B</sub>	20.5 transmission clock cycles																																																				
		These bits are only available in master mode.																																																			

**CAUTION**

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.12 CSIhnTX0W — CSIhn Transmit Data Register 0 for Word Access

This register stores transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

**Access:** This register can be read or written in 32-bit units.

**Address:** <CSIhn\_base> + 1008<sub>H</sub>

**Value after reset:** X0XX XXXX<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIhn CIRE	CSIhn EOJ	CSIhn EDL	—	—	—	—	—	CSIhnC S7	CSIhnC S6	CSIhnC S5	CSIhnC S4	CSIhnC S3	CSIhnC S2	CSIhnC S1	CSIhnC S0
Value after reset	—	—	—	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIhnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.26 CSIhnTX0W Register Contents (1/2)

Bit Position	Bit name	Function
31	CSIhnCIRE	Enables the communication interrupt request INTCSIHTIC in dual buffer or transmit-only buffer mode, or the job completion interrupt request INTCSIHTIJC in FIFO mode. 0: No interrupt is requested. 1: An interrupt is requested. Generates interrupt INTCSIHTIC or INTCSIHTIJC after transmission. For details, see <b>Section 15.4.3, INTCSIHTIC (Communication Status Interrupt)</b> and <b>Section 15.4.6, INTCSIHTIJC (Job Completion Interrupt)</b> .
	<b>CAUTION</b>	
	This bit is only valid when job mode is enabled (CSIhnCTL1.CSIhnJE = 1).	
30	CSIhnEOJ	Specifies the end of a job. 0: Indicates that it is not end-of-job data. The job continues. 1: Indicates end-of-job data.
	<b>CAUTION</b>	
	This bit is only valid when job mode is enabled (CSIhnCTL1.CSIhnJE = 1). This bit must be set to 0 in slave mode.	
29	CSIhnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Enables the extended data length. The associated data is transmitted as a 16-bit packet. No inter-data time or idle time will be inserted after the data is transmitted. If CSIhnCTL1.CSIhnEDLE = 1 and CSIhnTX0W.CSIhnEDL = 1, the subsequent data must have the same CS selection. If CS is modified for the subsequent data, the correct operation is not assured.
	<b>CAUTION</b>	
	This bit is only available if CSIhnCTL1.CSIhnEDLE = 1.	
28 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

**Table 15.26 CSI<sub>n</sub>TX0W Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit name</b>	<b>Function</b>
23 to 16	CSI <sub>n</sub> CS[7:0]	Activates one or more chip select signals. 0: Activates chip select signals x for the associated transmission. 1: Deactivates chip select signals x for the associated transmission. Setting CSI <sub>n</sub> TX0W.CSI <sub>n</sub> CS[7:0] = FF <sub>H</sub> is prohibited.
<b>CAUTION</b>		
If several chip select signals are enabled for broadcasting, the configuration of one with CSI <sub>n</sub> CFGx.CSI <sub>n</sub> RCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same value. In slave mode, set the CSI <sub>n</sub> CS[7:0] bit to FE <sub>H</sub> .		
15 to 0	CSI <sub>n</sub> TX[15:0]	Stores the transmission data.
<b>CAUTION</b>		
When setting this register, see <b>Table 15.32, Notes on Setting Registers</b> .		

### 15.3.13 CSI<sub>n</sub>TX0H — CSI<sub>n</sub> Transmit Data Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSI<sub>n</sub>TX0W.

The settings specified by the upper 16 bits of CSI<sub>n</sub>TX0W are applied to the transmission. Set transmit data to CSI<sub>n</sub>TX0W before using this register because the value of CSI<sub>n</sub>TX0W is undefined after the reset.

**Access:** This register can be read or written in 16-bit units.

**Address:** <CSI<sub>n</sub>\_base> + 100C<sub>H</sub>

**Value after reset:** Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI <sub>n</sub> TX[15:0]																
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 15.27 CSI<sub>n</sub>TX0H Register Contents**

Bit Position	Bit name	Function
15 to 0	CSI <sub>n</sub> TX[15:0]	Stores the transmission data.
<b>CAUTION</b>		
When setting this register, see <b>Table 15.32, Notes on Setting Registers</b> .		

### 15.3.14 CSIhnRX0W — CSIhn Receive Data Register 0 for Word Access

This register stores the received data.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <CSIhn\_base> + 1010H

**Value after reset:** 0XXX XXXX<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIhnRX[15:0]																
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 15.28** CSIhnRX0W Register Contents

Bit Position	Bit name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CSIhnRPE	Indicates whether a reception data parity error was detected. 0: No parity error was detected on the associated reception data. 1: A parity error was detected on the associated reception data.
24	CSIhnTDCE	Indicates whether a transmission data consistency error was detected. 0: No consistency error was detected on the associated transmission data. 1: A consistency error was detected on the associated transmission data.
23 to 16	CSIhnCSx (x = 7 to 0)	Indicates which chip select signal was activated. 0: Chip select x was activated for the associated reception. 1: Chip select x was deactivated for the associated reception.
15 to 0	CSIhnRX[15:0]	Stores the received data.

#### NOTE

Read this register at least one serial clock cycle before an interrupt is generated.

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.15 CSIhnRX0H — CSIhn Receive Data Register 0 for Half Word Access

This register stores the received data. This register is the same as bits 15 to 0 of register CSIhnRX0W.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <CSIhn\_base> + 1014<sub>H</sub>

**Value after reset:** Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIhnRX[15:0]																
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.29 CSIhnRX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIhnRX[15:0]	Stores the received data.

**CAUTION**

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.16 CSIhnEMU — CSIhn Emulation Register

This register controls operation of SVSTOP.

**Access:** This register can be read or written in 8-bit or 1-bit units.  
Perform write operation when (EPC.SVSTOP = 0).

**Address:** <CSIhn\_base> + 0018<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
CSIhnSVSDIS	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Table 15.30 CSIhnEMU Register Contents

Bit Position	Bit name	Function
7	CSIhn SVSDIS	Selects whether to continue or stop transmit/receive operation during debugging. <ul style="list-style-type: none"> <li>When the EPC.SVSTOP bit is set to 0 Continues transmit/receive operation regardless of the setting of this bit.</li> <li>When the EPC.SVSTOP bit is set to 1 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.</li> </ul>
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

**CAUTION**

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.17 CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3)

This register sets the transfer clock frequency for each chip select signal.

With CSIHnCFG0 to 7.CSIHnBRSSx[1:0] bits, one of the four types of transfer clock frequency settings can be selected for each chip select signal. For details of transfer clock frequency setting, see **Section 15.5.5, Transmission Clock Selection**.

**Access:** This register can be read or written in 16-bit units.

**Address:** CSIHnBRS0: <CSIHn\_base> + 1068<sub>H</sub>  
 CSIHnBRS1: <CSIHn\_base> + 106C<sub>H</sub>  
 CSIHnBRS2: <CSIHn\_base> + 1070<sub>H</sub>  
 CSIHnBRS3: <CSIHn\_base> + 1074<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	—	—	—	—	CSIHnBRS[11:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				

**Table 15.31 CSIHnBRSy Register Contents**

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CSIHnBRS [11:0]	0: BRG stopped 1: PCLK / (2 <sup>a</sup> × 1 × 2) 2: PCLK / (2 <sup>a</sup> × 2 × 2) 3: PCLK / (2 <sup>a</sup> × 3 × 2) 4: PCLK / (2 <sup>a</sup> × 4 × 2) . . . 4095: PCLK / (2 <sup>a</sup> × 4095 × 2)
		α is the value of CSIHnCTL2.CSIHnPMS[2:0].

#### CAUTION

When setting this register, see **Table 15.32, Notes on Setting Registers**.

### 15.3.18 List of Cautions

Table 15.32 Notes on Setting Registers (1/2)

Register Name	Bit Name	Cautions
CSIHnCTL0	CSIHnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIHnCTL0	CSIHnTXE CSIHnRXE	Do not modify any of these bits while CSIHnCTL0.CSIHnPWR = 0. (These bits can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIHnCTL0	CSIHnJOBE	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in slave mode.
CSIHnCTL0	CSIHnMBS	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. (This bit can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only permitted while CSIHnSTR0.CSIHnTSF = 0. Do not change the mode between FIFO mode and direct access mode while CSIHnCTL0.CSIHnPWR = 1. When the CPU-controlled high-priority communication is enabled, the operation is the same as that in direct access mode regardless of the CSIHnMBS bit setting.
CSIHnCTL1	CSIHnCKR	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. When CS is not used, use this bit instead of CSIHnCFGx.CSIHnCKPx and set CSIHnCFGx.CSIHnCKPx to 0. This bit must be used in slave mode.
CSIHnCTL1	CSIHnSLIT CSIHnCSL[7:0] CSIHnEDLE CSIHnDCS CSIHnCSRI CSIHnHSE	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.
CSIHnCTL1	CSIHnPHE CSIHnJE CSIHnLBM	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of this bit is prohibited in slave mode.
CSIHnCTL1	CSIHnSSE	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIHnCTL1	CSIHnSIT	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIHnCTL2	CSIHnPRS[2:0]	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of the maximum transfer clock frequency is as follows. <ul style="list-style-type: none"> <li>• Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)</li> <li>• Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)</li> </ul>
CSIHnSTR0	CSIHnSRP[7:0] CSIHnSPF[7:0] CSIHnFLF CSIHnEMF CSIHnTSF	Writing to these bits is prohibited, and only reading is permitted.
CSIHnSTR0	CSIHnTMOE CSIHnOFE CSIHnDCE CSIHnPE CSIHnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0.
CSIHnSTCR0	CSIHnPCT	If this bit is set to 1 during communication, ongoing communication is aborted.
CSIHnMCTL0	CSIHnMMS[1:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0.
CSIHnMCTL0	CSIHnTO[4:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Set these bits to 0 in master mode. Set these bits to 0 in direct access, dual buffer, and transmit-only buffer mode.
CSIHnMCTL1	CSIHnFES[6:0] CSIHnFFS[6:0]	Writing to these bits while communication is ongoing is permitted.

**Table 15.32 Notes on Setting Registers (2/2)**

<b>Register Name</b>	<b>Bit Name</b>	<b>Cautions</b>
CSIHnMCTL2	CSIHnBTST CSIHnND[7:0] CSIHnSOP[6:0]	Writing to these bits is prohibited when CSIhnCTL0.CSIhnPWR = 0. Writing to these bits is prohibited when CSIhnCTL0.CSIhnTXE = CSIhnCTL0.CSIhnRXE = 0. Writing to these bits is prohibited when CSIhnSTR0.CSIhnTSF = 1. Writing to these bits is prohibited in direct access or FIFO mode.
CSIHnMRWP0	CSIHnRRA[6:0]	Writing to these bits while communication is ongoing is permitted. Writing to these bits is prohibited in direct access or FIFO mode. When writing is required, set “0000 <sub>H</sub> ” to these bits in transmit-only buffer mode.
CSIHnMRWP0	CSIHnTRWA[6:0]	Writing to these bits while communication is ongoing is permitted. Writing to these bits is prohibited in direct access or FIFO mode.
CSIHnCFGx <i>x</i> = 0 to 7	CSIHnBRSSx[1:0] CSIHnRCBx CSIHnIDLx CSIHnIDX[2:0] CSIHnHDx[3:0] CSIHnINx[3:0] CSIHnSPx[3:0]	Modification of these bits is only permitted while CSIhnCTL0.CSIhnPWR = 0. These bits must be set to 0 in slave mode.
CSIHnCFGx <i>x</i> = 0 to 7	CSIHnPtx[1:0] CSIHnDLSx[3:0] CSIHnDIRx CSIHnDAPx	Modification of these bits is only permitted while CSIhnCTL0.CSIhnPWR = 0. In slave mode, the CSIhnCFG0 setting is used for the configuration. Therefore, all the bits in CSIhnCFG1 to CSIhnCFG7 must be set to 0.
CSIHnCFGx <i>x</i> = 0 to 7	CSIHnCKPx	Modification of these bits is only permitted while CSIhnCTL0.CSIhnPWR = 0. Set this bit to 0 as CSIhnCTL1.CSIhnCKR must be used in slave mode. If CS is not used, the CSIhnCTL1.CSIhnCKR bit instead of this bit, and clear this bit to 0.
CSIHnTX0W	CSIHnEOJ CSIHnCIRE	These bits are only valid when CSIhnCTL1.CSIhnJE = 1. While CSIhnCTL1.CSIhnJE = 0, the values of these bits are ignored even if 1 is read. Set these bits to 0 in slave mode.
CSIHnTX0W	CSIHnEDL	This bit is only valid when CSIhnCTL1.CSIhnEDLE = 1. While CSIhnCTL1.CSIhnEDLE = 0, the value of this bit is ignored even if 1 is read.
CSIHnTX0W	CSIHnCS[7:0]	In master mode, setting this bit to “FF <sub>H</sub> ” is prohibited. In slave mode, set this bit to “FE <sub>H</sub> ”.
CSIHnTX0W CSIHnTX0H		Reading these bits while communication is ongoing is prohibited in FIFO mode. While CSIhnCTL0.CSIhnPWR = 0, reading and writing to these bits is prohibited in FIFO mode. While CSIhnCTL0.CSIhnTXE = CSIhnCTL0.CSIhnRXE = 0, writing to these bits are prohibited in direct access mode.
CSIHnRX0W		Writing is prohibited and only reading is permitted while CSIhnCTL0.CSIhnPWR=1. Writing is permitted while CSIhnCTL0.CSIhnPWR=0. These bits are initialized when CSIhnCTL0.CSIhnPWR = 0 → 1 or CSIhnCTL0.CSIhnPWR = 1 → 0. While CSIhnCTL0.CSIhnPWR = 0, reading and writing these bits is prohibited in FIFO mode.
CSIHnRX0H		These bits are initialized when CSIhnCTL0.CSIhnPWR = 0 → 1 or CSIhnCTL0.CSIhnPWR = 1 → 0. While CSIhnCTL0.CSIhnPWR = 0, reading these bits is prohibited in FIFO mode.
CSIHnEMU	CSIHnSVSDIS	Modification of this bit is only permitted while SVSTOP = 0.
CSIHnBRSy <i>y</i> = 0 to 3		Modification of these bits is only permitted while CSIhnCTL0.CSIhnPWR = 0.

## 15.4 Interrupt Sources

CSIH can generate the following interrupt requests:

- INTCSIHTIC (communication status interrupt)
- INTCSIHTIR (reception status interrupt)
- INTCSIHTIRE (communication error interrupt)
- INTCSIHTIJC (job completion interrupt)

### 15.4.1 Overview

The communication error interrupt INTCSIHTIRE is generated when an error is detected. The generation of the other interrupts depends on the memory mode, the job mode, and – in case of the job completion interrupt INTCSIHTIJC –also the operating mode.

The job completion interrupt INTCSIHTIJC is only generated when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). It is not available in slave mode.

The following table gives an overview.

Table 15.33 Interrupt Generation

Memory Mode	Interrupt	Interrupt Source	
		Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	INTCSIHTIC	Tx data empty <sup>*1</sup>	Tx data empty <sup>*1</sup> except job abort <sup>*4</sup>
	INTCSIHTIR	Rx data full <sup>*2</sup> and CSIHnCTL0.CSIHnRXE = 1	Rx data full <sup>*2</sup> and CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC <sup>*3</sup>	Not applicable	CSIHnTX0W.CSIHnCIRE = 1 (except Tx data empty), or job abort <sup>*4</sup>
Transmit-only buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 and (CSIHnCTL0.CSIHnJOBE = 0 or CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	Data received and CSIHnCTL0.CSIHnRXE = 1	Data received and CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC <sup>*3</sup>	Not applicable	Job abort <sup>*4</sup>
Dual buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 and (CSIHnCTL0.CSIHnJOBE = 0 or CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	End of communication and CSIHnCTL0.CSIHnRXE = 1	Data received and CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC <sup>*3</sup>	Not applicable	Job abort <sup>*4</sup>
Direct access	INTCSIHTIC	One data transfer	One data transfer except the state of job abort <sup>*4</sup>
	INTCSIHTIR	Data received and CSIHnCTL0.CSIHnRXE = 1	Data received and CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC <sup>*3</sup>	Not applicable	Job abort <sup>*4</sup>

Note 1. “Tx data empty” refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFES[6:0].

- Note 2. "Rx data full" refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFFS[6:0].
- Note 3. INTCSIHTIJC is not available in slave mode.
- Note 4. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1  
During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

### 15.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half a cycle of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting CSIHnCTL1.CSIHnSIT = 1. (The setting of the CSIGnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (clock phase and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000<sub>B</sub> (data length 8 bits).

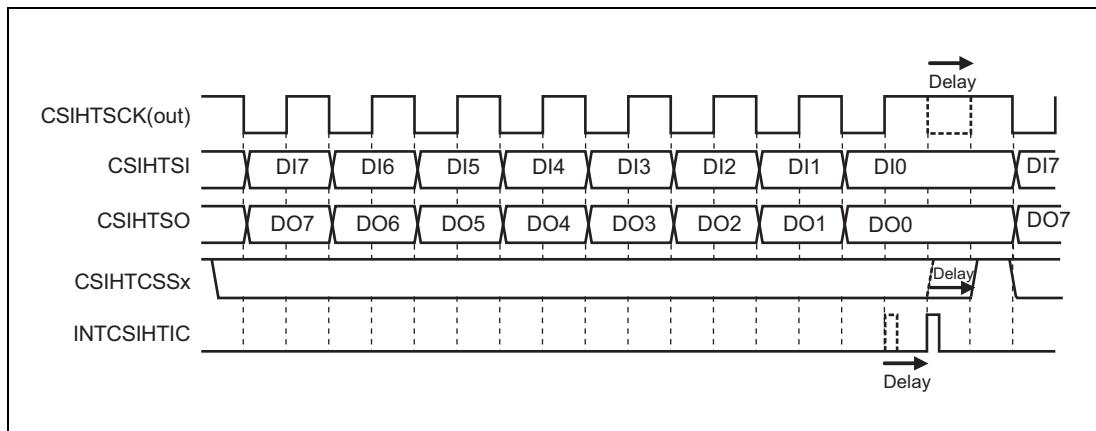


Figure 15.2 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds a half cycle delay to the transmission clock. This also delays the end of the current chip select signal (CSIHTCSSx).

### 15.4.3 INTCSIHTIC (Communication Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions shown in the following table.

Table 15.34 INTCSIHTIC Interrupt Generation

Memory Mode	Interrupt Source	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt is generated just before transmission data in the FIFO runs out, notifying the application that new data should be added. INTCSIHTIC is generated if the number of transmit data remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMCTL1.CSIHnFES[6:0]. However, it is not generated if a job is aborted.	Similar to "when JE is 0", an interrupt is generated when the number of transmit data remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMCTL1.CSIHnFES[6:0]. However, it is not generated if a job is aborted.
Transmit-only buffer, dual buffer	An interrupt is generated at the end of communication. (specified by the CSIHnMTLC2.CSIHnND[7:0] bit)	Generated when data is transmitted while CSIHnTX0W.CSIHnCIRE = 1. Note that if data and job abort*1 are transmitted while CSIHnTX0W.CSIHnCIRE = 1, the INTCSIHTJC interrupt is generated instead of INTCSIHTIC.
Direct access	Generated after every data transfer.	Generated after every data transfer, except when the communication was aborted.

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1.  
During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

#### 15.4.3.1 INTCSIHTIC in Direct Access Mode

The examples below show the INTCSIHTIC behavior in direct access mode.

The examples assume:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000<sub>B</sub>)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

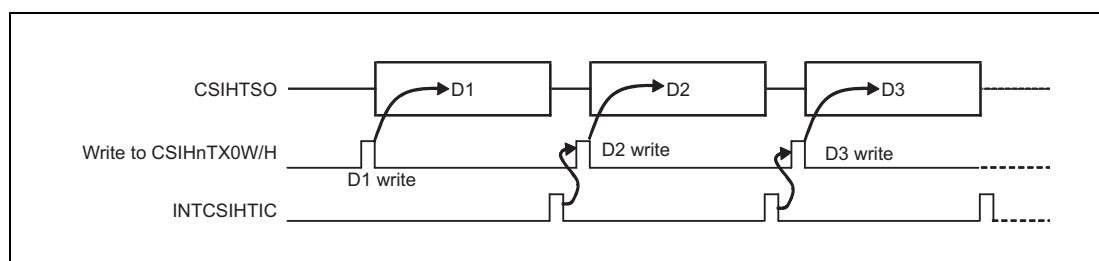
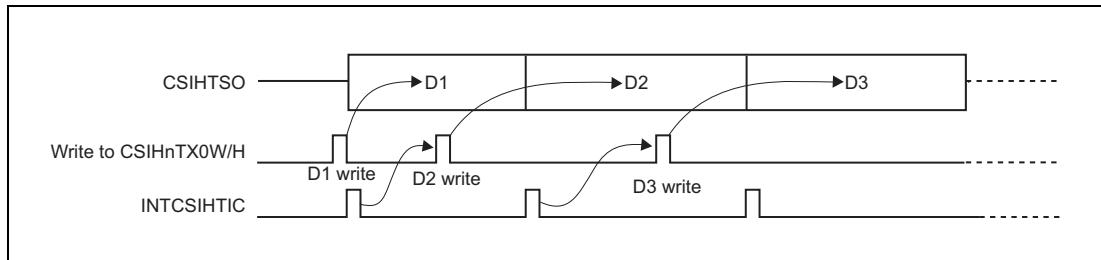


Figure 15.3 Generation of INTCSIHTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTXOW.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then INTCSIHTIC is replaced by the job completion interrupt INTCSIHTJC.

INTCSIHTIC can also be set up to occur as soon as the CSIHnTX0W/H register becomes empty and available for storing the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.



**Figure 15.4 Immediate Generation of INTCSIHTIC (CSIHnCTL1.CSIHnSLIT = 1)**

Thus, the new data can be written in advance.

#### NOTE

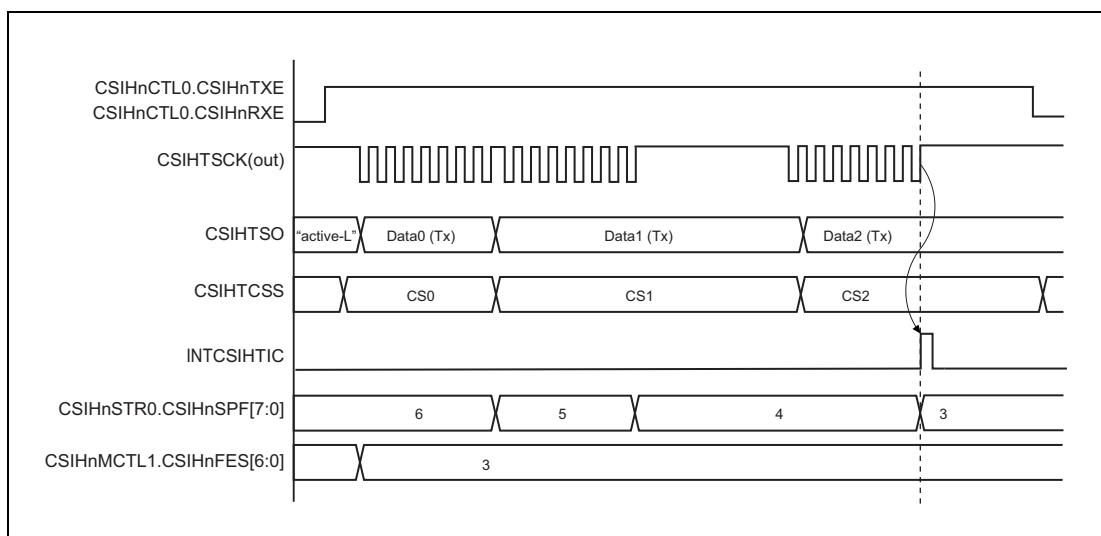
During high priority communication in transmit-only buffer mode, the operation is in the same as that in direct access mode.

#### 15.4.3.2 INTCSIHTIC in FIFO Mode

The example below shows the INTCSIHTIC behavior in FIFO mode.

The example assumes:

- Master mode
- FIFO mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase  
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000<sub>B</sub>)



**Figure 15.5 Generation of INTCSIHTIC in FIFO Memory Mode**

The condition for “FIFO empty” is specified in CSIhnMCTL1.CSIhnFES[6:0]. In the example of the diagram above, the number of unsent data in FIFO is set to 3.

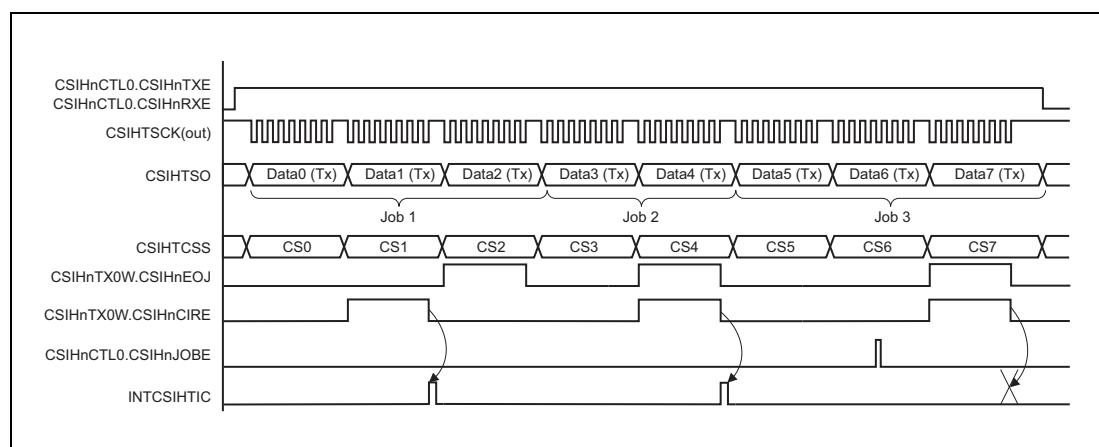
CSIHnSTR0.CSIHnSPF[7:0] indicates the number of unsent data. When both match, the interrupt INTCSIHTIC occurs.

#### 15.4.3.3 INTCSIHTIC in Job Mode

The example below shows the INTCSIHTIC behavior in job mode.

The example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase  
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000<sub>B</sub>)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)



**Figure 15.6 Generation of INTCSIHTIC in Job Mode**

The rules for generating INTCSIHTIC in job mode are shown in the following table.

**Table 15.35 Generation of INTCSIHTIC in Job Mode**

CSIHnTX0W. CSIHnEOJ	CSIHnTX0W. CSIHnCIRE	INTCSIHTIC
0	0	Not generated
0	1	Generated
1	0	Not generated
1	1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt INTCSIHTIJC

#### 15.4.4 INTCSIHTIR (Reception Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions below.

**Table 15.36 INTCSIHTIR Interrupt Generation**

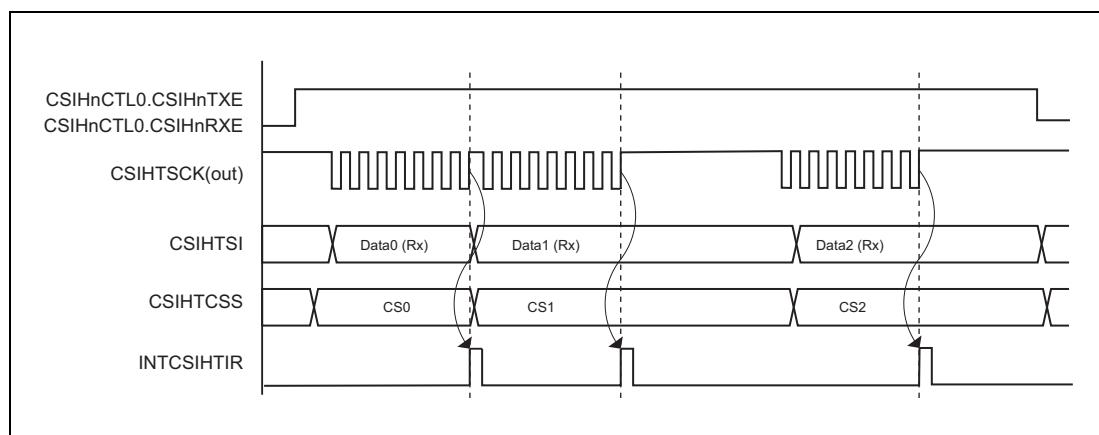
Memory Mode	Interrupt Source	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt occurs when CSIhnCTL0.CSIhnRXE is 1 and the FIFO buffer is almost full with received data, notifying the application that the FIFO must be emptied. INTCSIHTIR is generated, if the number of received data in the FIFO (CSIhnSTR0.CSIhnSRP[7:0]) equals (128 – CSIhnMCTL1.CSIhnFFS[6:0]).	
Dual buffer	An interrupt is generated when the communication has finished (as specified by the CSIhnMCTL2.CSIhnND[7:0] bits) and CSIhnCTL0.CSIhnRXE = 1.	An interrupt is generated after every data transfer.
Transmit-only buffer, Direct access	An interrupt is generated after every data transfer.	

##### 15.4.4.1 INTCSIHTIR in Direct Access Mode

The example below shows the INTCSIHTIR behavior in direct access mode.

The example below assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIhnCTL1.CSIhnSIT = 0)
- Normal clock phase and data phase  
(CSIhnCFGx.CSIhnCKPx = 0, CSIhnCFGx.CSIhnDAPx = 0)
- Data length 8 bits (CSIhnCFGx.CSIhnDLSx[3:0] = 1000<sub>B</sub>)



**Figure 15.7 Generation of INTCSIHTIR in Direct Access Mode**

#### 15.4.4.2 INTCSIHTIR in Dual Buffer Mode

The example below shows the INTCSIHTIR behavior in dual buffer mode.

The example assumes:

- Master mode
- Dual buffer mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Default clock phase and data phase  
(CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000<sub>B</sub>)

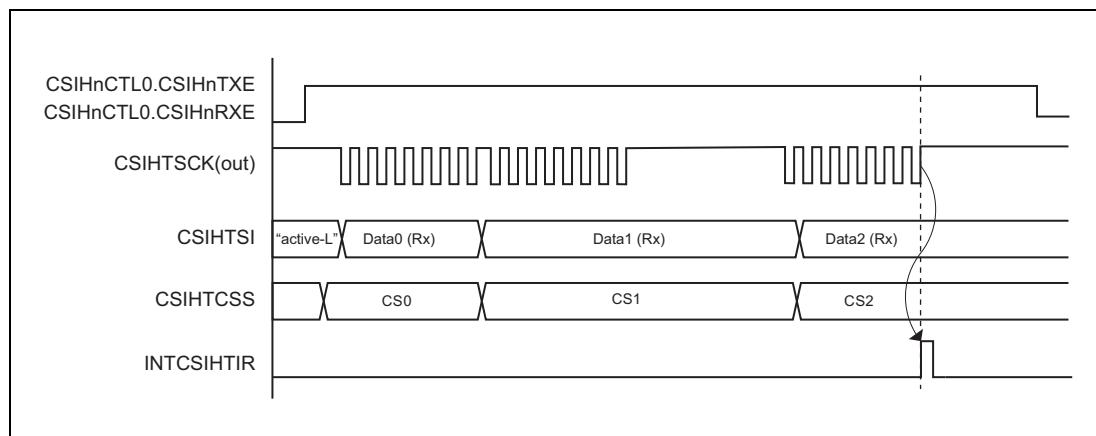


Figure 15.8 Generation of INTCSIHTIR in Dual Buffer Mode

#### 15.4.5 INTCSIHTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about interrupt generation timing, see **Section 15.5.12, Error Detection**.

Table 15.37 Data Error Types (1/2)

Error Type	Communication Status after Error Interrupt	Note
FIFO overflow error	Communication continues even if an interrupt is generated.	The data is not written to the FIFO buffer and the data that overflowed is lost, but communications started before the error continue.
Parity error	Communication continues even if an interrupt is generated.	—
Data consistency error	Communication continues even if an interrupt is generated.	—

Table 15.37 Data Error Types (2/2)

Error Type	Communication Status after Error Interrupt	Note
Time-out error	Communication continues even if an interrupt is generated.	—
Overrun error	<p><b>Condition for errors 1:</b> In FIFO mode, when the number of received data is 0 and CPU reads the CSIHnRX0W/H register, an interrupt is generated and communication continues.</p> <p><b>Condition for errors 2:</b> In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled):</p> <ul style="list-style-type: none"> <li>[1] In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data is retained in the CSIHnRX0W/H register, an interrupt is generated, and communication continues.</li> <li>[2] In FIFO mode, when reception by the FIFO buffer is completed and the buffer is in the full state, an interrupt is generated. Communication continues.</li> </ul>	In slave mode, when CSIHnCTL1.CSIHnHSE = 1 (handshake function enabled), communication is suspended due to handshake, an overrun error is not generated.

The type of error that caused the generation of INTCSIHTIRE is flagged in register CSIHnSTR0.

Additionally a parity error flag and a data consistency error flag are attached to the received data in CSIHnRX0W.

For details about the various error types, see **Section 15.5.12, Error Detection**.

#### 15.4.6 INTCSIHTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs. See **Section 15.5.3.3, Job Concept**. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, INTCSIHTIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the condition shown in the following table.

Table 15.38 INTCSIHTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	Not applicable	Indicates that the communication stopped at the end of a job after a job abort <sup>*1</sup> was triggered If FIFO empty is not detected, INTCSIHTIJC is generated when CSIHnCIRE is 1.
Transmit-only buffer		Indicates that the communication stopped at the end of a job after a job abort <sup>*1</sup> was triggered.
Dual buffer		
Direct access		

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

## 15.5 Operation

### 15.5.1 Operating Modes (Master/Slave)

Whether CSHI operates in the master or slave mode determines the source of the serial clock.

#### 15.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and supplied to the slave(s) by signal CSIHTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to values other than 111<sub>B</sub>. In master mode, the BRG frequency can be specified by setting the CSIHnCTL2.CSIHnPRS[2:0] bits in combination with the CSIHnBRSy.CSIHnBRS[11:0] bits.

##### (1) Chip select signals

In master mode, one or more chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to select one or more slaves. Only the selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see **Section 15.5.3, Chip Selection (CS) Features**.

##### (2) Clock defaults

The default level of CSIHTSCK depends on the clock phase inversion function bit of the CSIHTSCK, and is high when CSIHnCTL1.CSIHnCKR = 0 and is low when CSIHnCTL1.CSIHnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first.

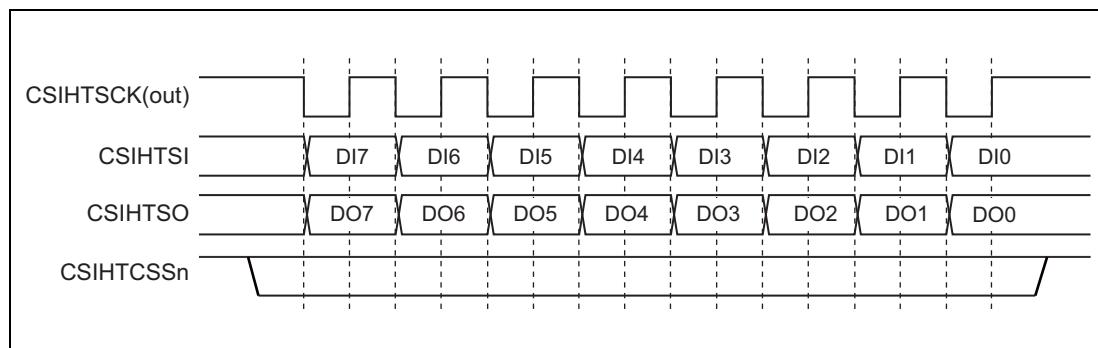


Figure 15.9 Transmission/Reception in Master Mode

### 15.5.1.2 Slave Mode

In slave mode, another device is the communication master and supplies the transmission clock. Normal transmit/receive operation is started as soon as a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bits to 111<sub>B</sub>.

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is enabled (settings of the CSIHnCFG1 - CSIHnCFG7 registers are disabled).

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

#### NOTE

When using slave mode, disable the baud rate generator (BRG) by setting the CSIHnBRSy.CSIHnBRS[11:0] bits to 000<sub>H</sub>. However, if you are using a time-out error, set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000<sub>H</sub>.

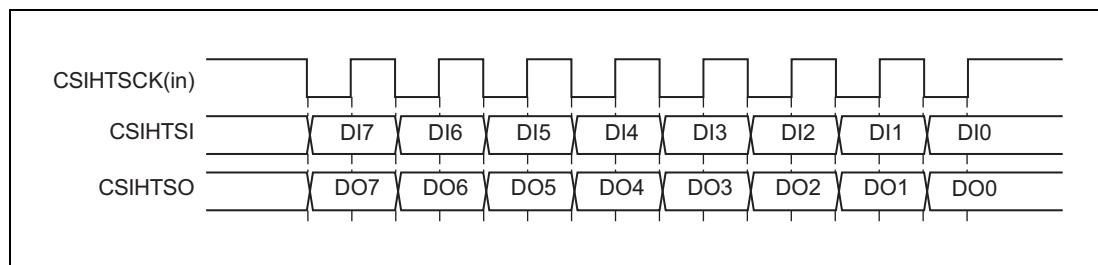


Figure 15.10 Transmission/Reception in Slave Mode

## 15.5.2 Master/Slave Connections

### 15.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

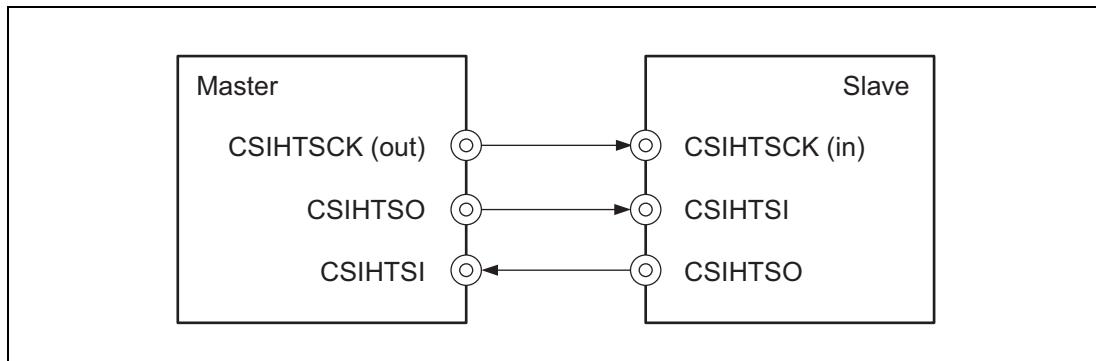


Figure 15.11 Direct Master/Slave Connection

### 15.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master supplies one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input CSIHTSSI of the slave.

The CSIHTSSI signal can be enabled/disabled by using the CSIHnCTL1.CSIHnSSE bit.

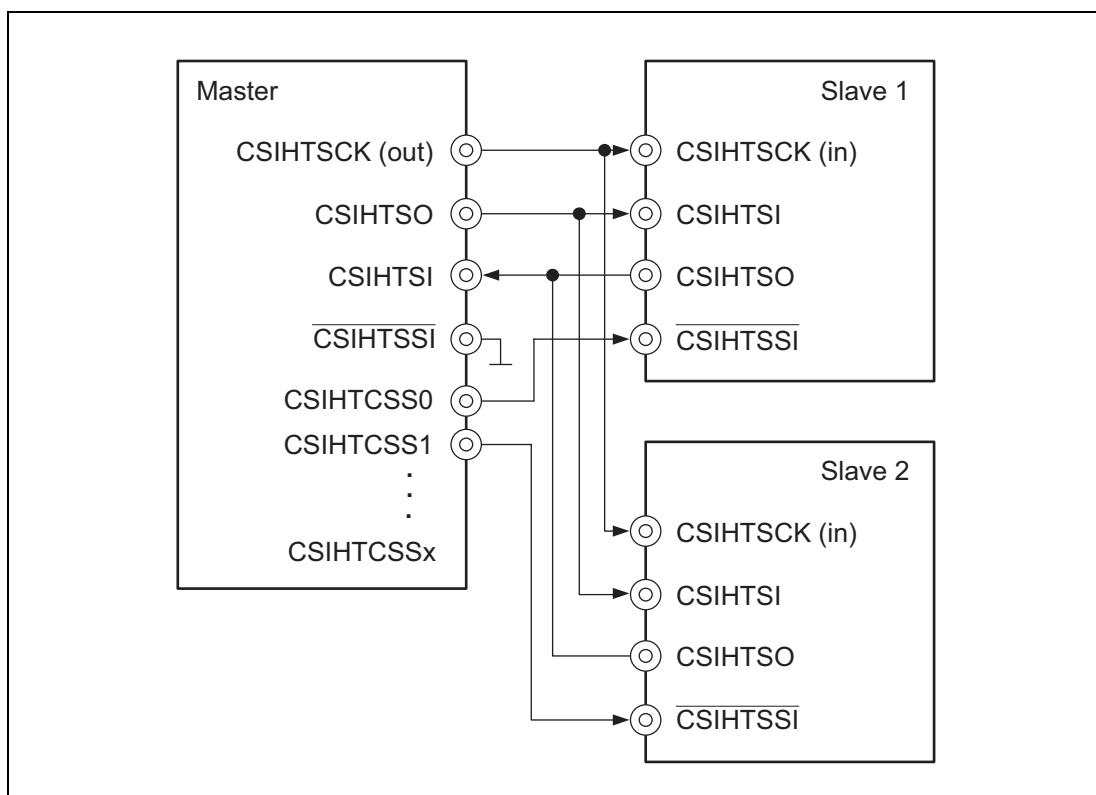


Figure 15.12 Connections between One Master and Multiple Slaves

By default, the chip select level is active low. That means, a slave is selected (enabled) as a CSHI slave when its CSIHTSSI signal is low level. However, to adapt the CS to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set (CSIHnCTL0.CSIHnTXE = 1), the CSIHTSO output of the slaves that are not selected is disabled and set to input mode in order to avoid interference with the output of the selected slave.

### 15.5.3 Chip Selection (CS) Features

The chip select signal, CSIHTCSSx can be used by the master to select one or more slaves for communication.

#### 15.5.3.1 Configuration Registers

The parameters for each chip select signal CSIHTCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received.  
(CSIHnCFGx.CSIHnDLSx[3:0])
- Transfer direction: MSB or LSB first.  
(CSIHnCFGx.CSIHnDIRx)
- Parity usage: Odd, even, 0 parity or none.  
(CSIHnCFGx.CSIHnPSx[1:0])
- Clock phase and data phase.  
(CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx)

Additional parameters for each chip select signal that are only available in master mode are:

- Individual selection of the baud rate generator prescaler for each chip select signal  
(CSIHnCFGx.CSIHnBRSSx[1:0])
- Chip select priority: Categorizes chip select signals into “dominant” and “recessive”. The priority applies if two or more chip selects signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration specified for dominant chip select signals is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

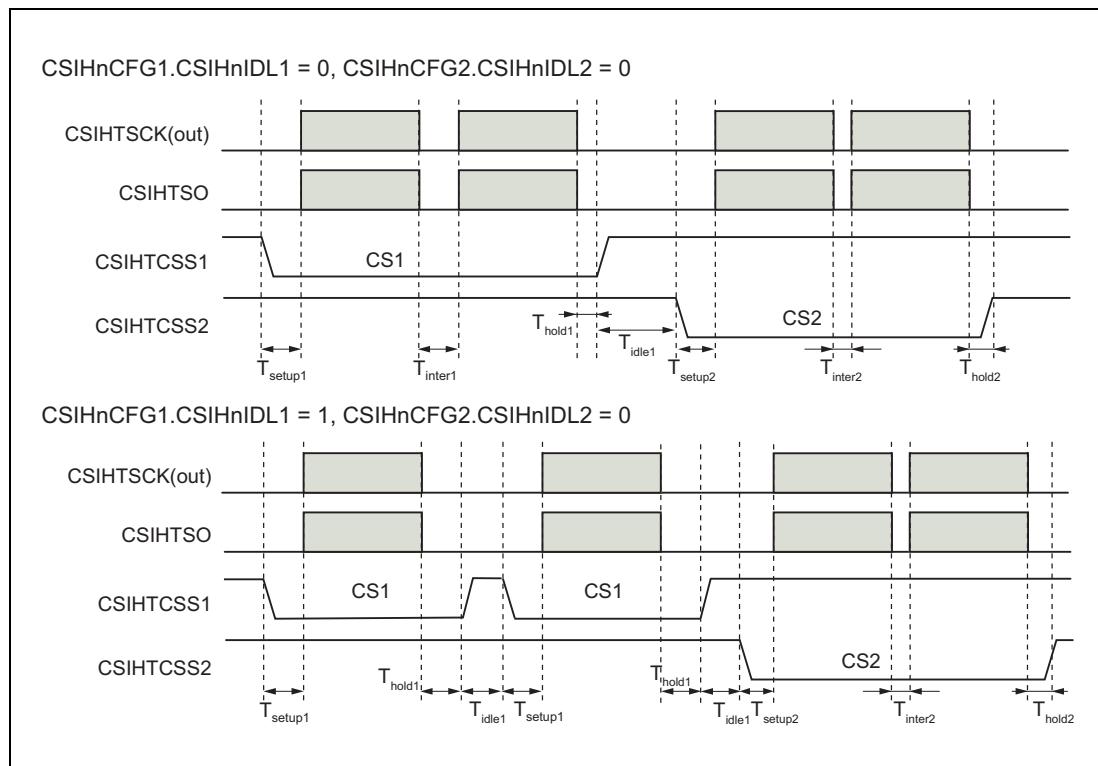
#### CAUTION

**When specifying multiple chip select signals as dominant, be sure to configure the same settings for all dominant signals.**

- Chip select timing:
  - Setup time  $T_{\text{setup}}$ : The time from when the CS signal becomes active to the start of data output.  
(CSIHnCFGx.CSIHnSPx[3:0])
  - Inter-data time  $T_{\text{inter}}$ : The time between one data and the next data while the same CS signal is active.  
(CSIHnCFGx.CSIHnINx[3:0])
  - Hold time  $T_{\text{hold}}$ : The time during which the CS signal remains active until CS is switched.  
(CSIHnCFGx.CSIHnHDx[3:0])
  - Idle time  $T_{\text{idle}}$ : Inactive time after terminating a CS signal or after every data transfer to the same CSx. (CSIHnCFGx.CSIHnIDx[2:0])

The CS timings of the setup time, the inter-data time, the hold time, and the idle time are illustrated in the figure below. When CSIhnCFGx.CSIhnIDLx bit is set to 1, IDLE time is inserted for every transfer regardless of CS signal.

**Figure 15.13** provides an example of when the default active low setting is specified for the CSIHTCSS1 and CSIHTCSS2 signals (CSIhnCTL1.CSIhnCSL1 bit = 0, CSIhnCTL1.CSIhnCSL2 bit = 0). The active level can be specified individually for each CS.



**Figure 15.13 Chip Select Timings**

Note that each CS signal can have a different value for the setup time, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CSIhnTX0W.CSIhnCSx.

CSIhnRX0W.CSIhnCSx in the reception register indicates the chip select signal associated with the received data.

#### CAUTION

**When high priority communication function by CPU control is enabled (CSIhnCTL1.CSIhnPHE = 1), IDLE state is inserted regardless of IDLn bit settings when priority communication mode is changed from low to high and from high to low.**

### 15.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with a single slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority” and the priority of CS1 to “dominant: high priority”. Consequently, the second communication is conducted using the CS1 settings, which are set as dominant.

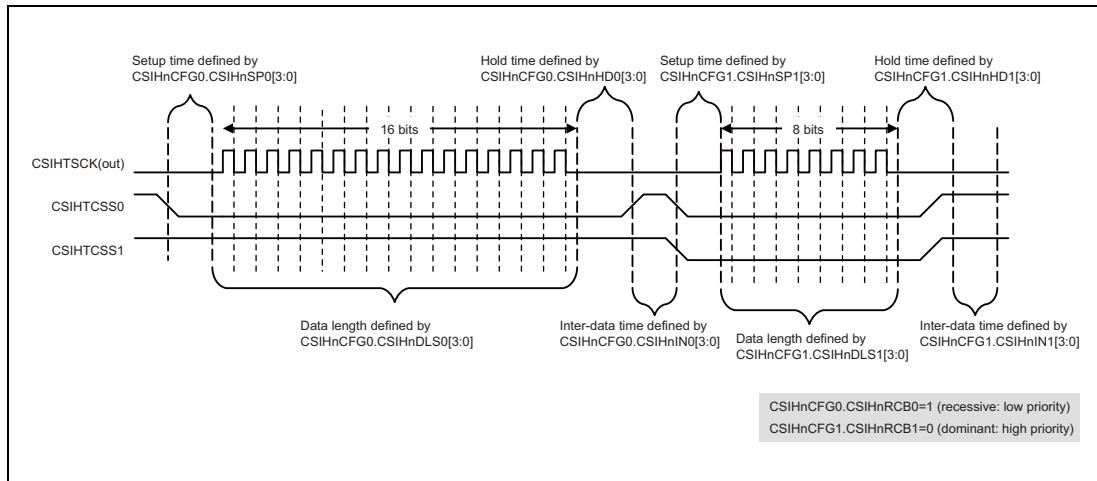


Figure 15.14 Chip Select and RCB Example

### 15.5.3.3 Job Concept

In CSHI, a job consists of the number of data targeted for transfer.

#### Job mode enable

The job mode can only be enabled in master mode. The job mode is enabled or disabled by CSIHnCTL1.CSIHnJE, while the CSHI is disabled by CSIHnCTL0.CSIHnPWR = 0.

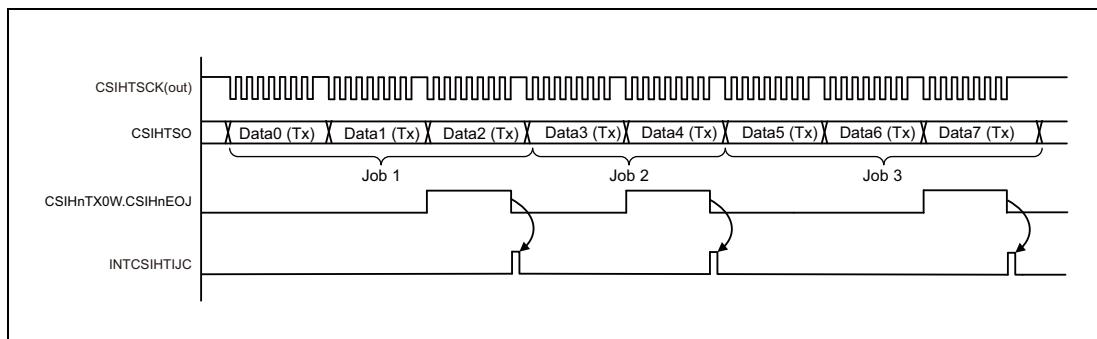


Figure 15.15 Job Examples

A job ends by transmitting data with CSIHnTX0W.CSIHnEOJ = 1.

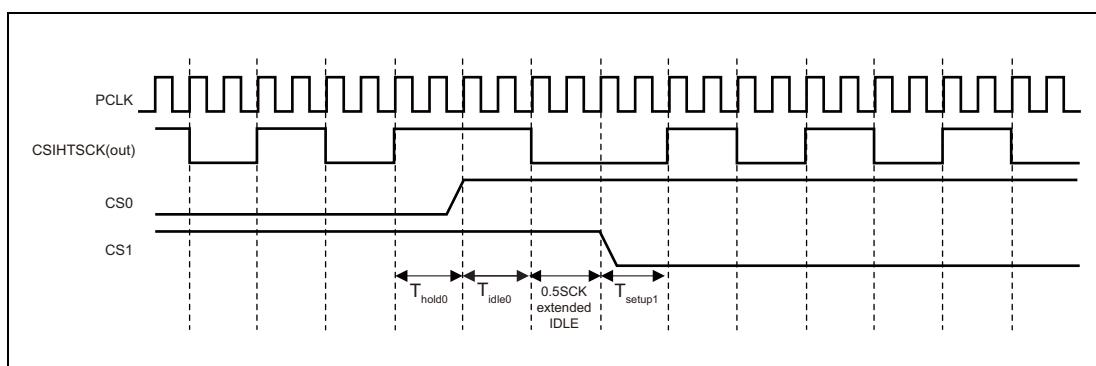
Communication can be specified to stop when a job is finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until the data for which the CSIHnEOJ bit is set is transmitted. After this data is sent, the communication is stopped and the job completion interrupt INTCSIHTIJC is generated.

## 15.5.4 Details of Chip Select Timing

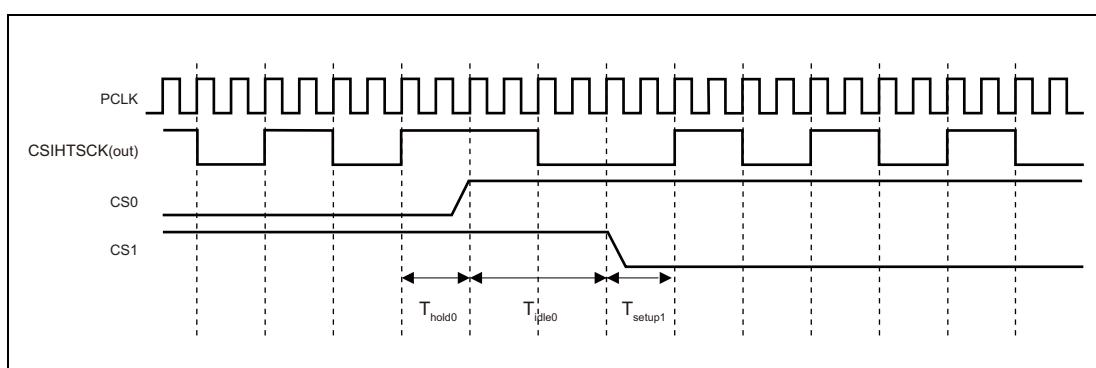
### 15.5.4.1 Changing the Clock Phase

The serial clock level specified by CSIhnCFGx.CSIhnCKPx can be changed while communication is stopped. The minimum value of an idle time is one transmission clock (CSIHTSCK(out)) cycle.

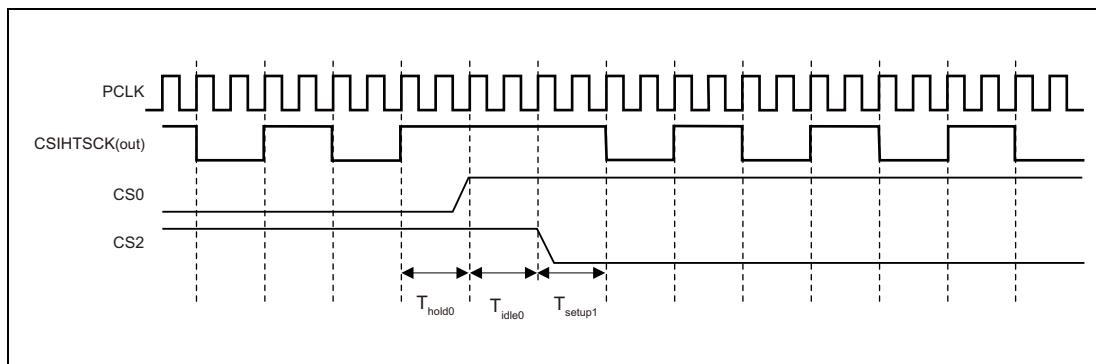
If the idle time is set to 0.5 transmission clock cycles (in CSIhnCFGx.CSIhnIDx[2:0]) and two consecutive data is sent with different CSIhnCFGx.CSIhnCKPx configurations, the idle time is automatically extended to one CSIHTSCK(out) cycle.



**Figure 15.16 Clock Phase Timing with PCLK/4,  $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$ ,  $T_{idle0} = 0.5CSIHTSCK$ , CSIhnCFG0.CSIhnCKP0 = 0 (CSIHTCSS0) → CSIhnCFG1.CSIhnCKP1 = 1 (CSIHTCSS1)**



**Figure 15.17 Clock Phase Timing with PCLK/4,  $T_{hold0} = T_{setup1} = 0.5CSIHTSCK$ ,  $T_{idle0} = 1CSIHTSCK$ , CSIhnCFG0.CSIhnCKP0 = 0 (CSIHTCSS0) → CSIhnCFG1.CSIhnCKP1 = 1 (CSIHTCSS1)**



**Figure 15.18 Clock Phase Timing with PCLK/4,  $T_{hold0} = T_{setup1} = 0.5\text{CSIHTSCK}$ ,  $T_{idle0} = 0.5\text{CSIHTSCK}$ ,  $\text{CSIHnCFG0.CSIHnCKP0} = 0$  ( $\text{CSIHTCSS0} \rightarrow \text{CSIHnCFG2.CSIHnCKP2} = 0$ ) ( $\text{CSIHTCSS2}$ )**

#### 15.5.4.2 Changing the Data Phase

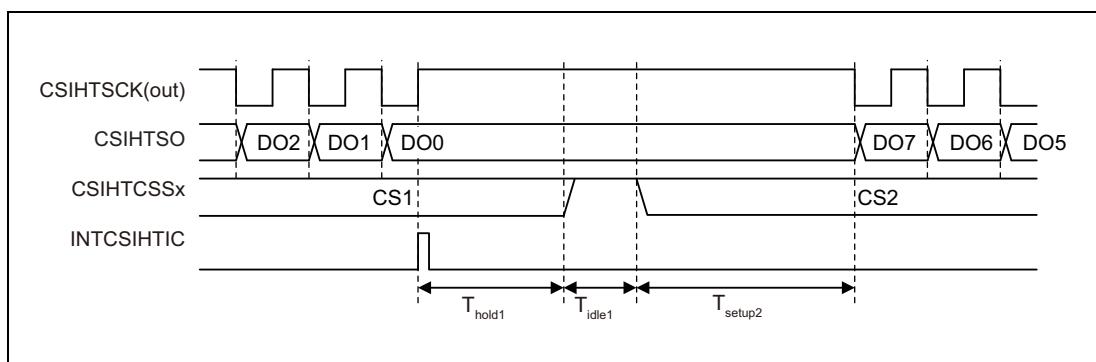
The  $\text{CSIHnCFGx.CSIHnDAPx}$  bit defines the phase of the data bits relative to the clock.

The relation between the setting of the  $\text{CSIHnCFGx.CSIHnDAPx}$  bit and the hold and setup times is as follows:

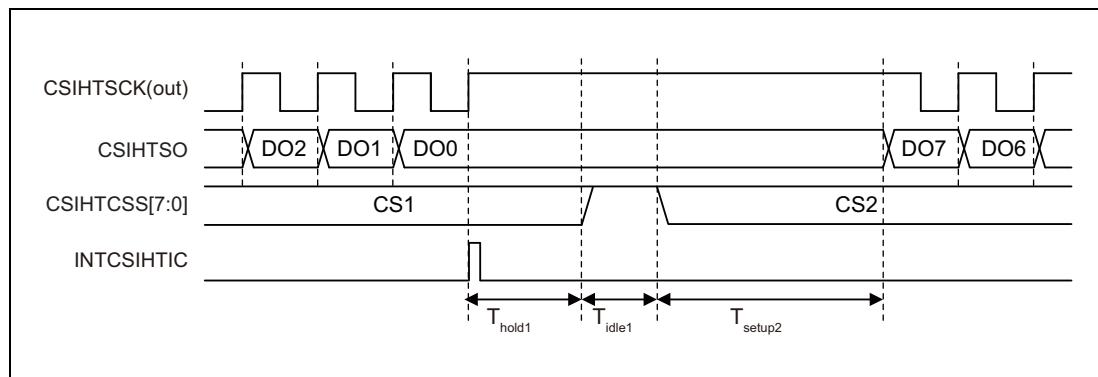
Hold time is the period from the last edge of the serial clock (CSIHTSCK) until the signals on  $\text{CSIHTCSS}[7:0]$  change to the inactive level.

Setup time is the period from when the signals on  $\text{CSIHTCSS}[7:0]$  change to the active level until to when the transmission data (CSIHTSO) is output.

Therefore, there is a gap of 0.5 CSIHTSCK cycles until the edge of the serial clock signal (CSIHTSCK) is output according to the  $\text{CSIHnCFGx.CSIHnDAPx}$  setting.



**Figure 15.19 Data Phase Timing with  $\text{CSIHnCFG1.CSIHnCKP1} = 0$ ,  $\text{CSIHnCFG1.CSIHnDAP1} = 0$  and  $\text{CSIHnCFG2.CSIHnCKP2} = 0$ ,  $\text{CSIHnCFG2.CSIHnDAP2} = 0$**



**Figure 15.20** Data Phase Timing with  
CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and  
CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

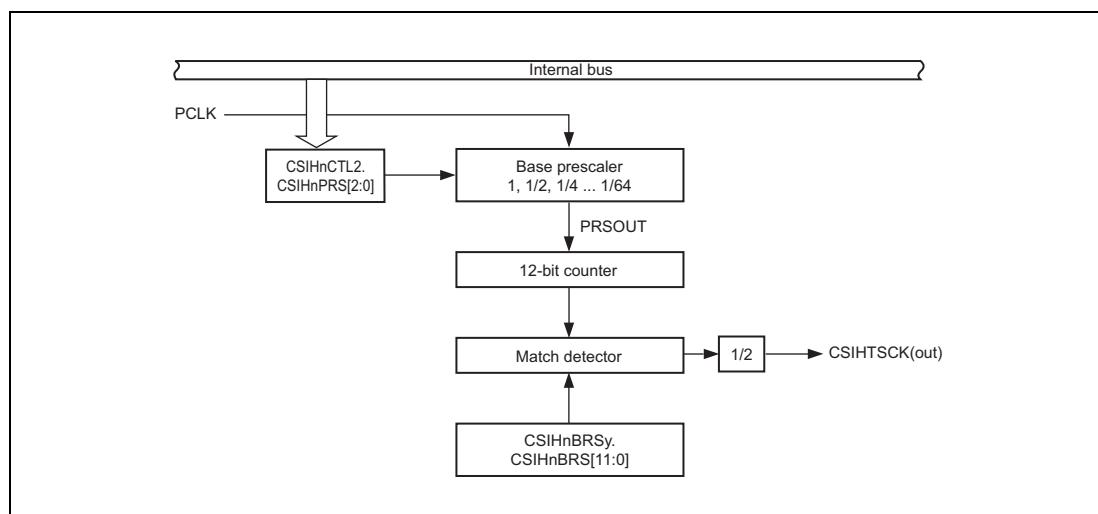
### 15.5.5 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnBRSy.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHnBRSSx[1:0]

The transfer clock frequency of transmission clock CSIHTSCK is determined by the setting of the CSIHnCTL2.CSIHnPRS[2:0] bits and the setting of the CSIHnBRSy.CSIHnBRS[11:0] bits, but any one of CSIHnBRS3 to CSIHnBRS0 can be selected for each chip select signal by using the CSIHnCFGx.CSIHnBRSSx[1:0] bits.

The following figure shows a block diagram of the baud rate generator.



**Figure 15.21** Baud Rate Generator Block Diagram

By setting CSIHnBRSy.CSIHnBRS[11:0] to 000<sub>H</sub>, the baud rate generator is disabled and all CSIHTSCK are stopped.

### Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

$$\text{Transfer clock frequency (CSIHTSCK)} = \text{PCLK} / (\text{division ratio of PCLK}) = \text{PCLK} / (2^{\alpha} \times k \times 2),$$

where:

$$\alpha = \text{CSIHnCTL2.CSIHnPRS}[2:0] = 0 \text{ to } 6$$

$$k = \text{CSIHnBRS0.CSIHnBRS0}[11:0] = 1 \text{ to } 4095$$

(when  $\text{CSIHnCFGx.CSIHnBRSSx}[1:0] = 0$ )

$$\text{CSIHnBRS1.CSIHnBRS1}[11:0] = 1 \text{ to } 4095$$

(when  $\text{CSIHnCFGx.CSIHnBRSSx}[1:0] = 1$ )

$$\text{CSIHnBRS2.CSIHnBRS2}[11:0] = 1 \text{ to } 4095$$

(when  $\text{CSIHnCFGx.CSIHnBRSSx}[1:0] = 2$ )

$$\text{CSIHnBRS3.CSIHnBRS3}[11:0] = 1 \text{ to } 4095$$

(when  $\text{CSIHnCFGx.CSIHnBRSSx}[1:0] = 3$ )

### Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the following:

- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)

### 15.5.6 CSH Buffer Memory

The CSH has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word is comprised of 32-bit data and 7-bit ECC.

The following configurations are available:

Mode	CSIHnCTL0. CSIHnMBS	CSIHnMCTL0. CSIHnMMS[1:0]
FIFO mode	0	00 <sub>B</sub>
Dual buffer mode		01 <sub>B</sub>
Transmit-only buffer mode		10 <sub>B</sub>
Direct access mode	1	X

#### 15.5.6.1 FIFO Mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately, provided the FIFO is not full.

Data to be transmitted is stored in the FIFO memory. Transmission and reception occur simultaneously – one data is received as one data is transmitted. That means, received data overwrites the transmitted data in the FIFO.

The CSH automatically updates the respective FIFO memory pointers when data is written to or read from the FIFO memory, or data is transmitted to or received from the FIFO memory.

Table 15.39 FIFO Mode

Pointer Description	Pointer <sup>*1</sup>	Range
Number of untransmitted words	CSIHnSTR0.CSIHnSPF[7:0]	0 to 128
Number of words received and stored in the FIFO	CSIHnSTR0.CSIHnSRP[7:0]	0 to 128
Address for write/read of transmit data	CSIHnMRWP0.CSIHnTRWA[6:0]	0000 <sub>H</sub> to 01FC <sub>H</sub>
Address for read of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0000 <sub>H</sub> to 01FC <sub>H</sub>
Address to be sent	CSIHnMCTL2.CSIHnSOP[6:0]	0000 <sub>H</sub> to 01FC <sub>H</sub>

Note 1. The values are automatically updated after each read/write or data transmit/receive operation.

The CSH status register contains two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, bit CSIHnSTCR0.CSIHnPCT must be set. By doing this, CSIHnSTR0.CSIHnEMF is not reset, but set.

All FIFO pointers and FIFO flags except CSIHnSTR0.CSIHnEMF are reset and CSIHnSTR0.CSIHnEMF is set.

### 15.5.6.2 Dual Buffer Mode

In this mode, the memory is divided into two parts of equal size; 64 words are allocated to transmission data and 64 words to received data. In dual buffer mode, the respective buffer pointers indicate the values shown in the following table.

**Table 15.40 Dual Buffer Mode**

Pointer Description	Pointer <sup>*1</sup>	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000 <sub>H</sub> to 00FC <sub>H</sub>
Address of data read from receive buffer	CSIHnMRWP0.CSIHnRRA[6:0]	0000 <sub>H</sub> to 00FC <sub>H</sub>
The number of transmit data remaining in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 64
Address to which data is transmitted	CSIHnMCTL2.CSIHnSOP[6:0]	0000 <sub>H</sub> to 00FC <sub>H</sub>

Note 1. Pointers are automatically incremented after each read/write.

### 15.5.6.3 Transmit-Only Buffer Mode

In this mode, the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the respective buffer pointer indicates the values shown in the following table.

**Table 15.41 Transmit-Only Buffer Mode**

Pointer Description	Pointer <sup>*1</sup>	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000 <sub>H</sub> to 01FC <sub>H</sub>
The number of transmit data remained in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 128
Address to which data is transmitted	CSIHnMCTL2.CSIHnSOP[6:0]	0000 <sub>H</sub> to 01FC <sub>H</sub>

Note 1. Pointers are automatically incremented after each read/write.

### 15.5.6.4 Direct Access Mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception register CSIHnRX0W or CSIHnRX0H.

## 15.5.7 Data Transfer Modes

### 15.5.7.1 Transmit-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSHI in transmit-only mode. Start of transmission depends on the memory mode:

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, transmission starts when the CSIHnMCTL2.CSIHnBTST bit is set.

### 15.5.7.2 Receive-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSHI in receive-only mode.

In master mode, start of reception depends on the memory mode:

- In case of FIFO or direct access mode, reception starts when dummy data is written to the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock is received from the master. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

- In case of dual buffer mode or transmit-only buffer mode, reception starts when the CSIHnMCTL2.CSIHnBTST bit is set.

### 15.5.7.3 Transmit/Receive Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSHI in transmit/receive mode.

Start of communication (transmission and reception) depends on the memory mode:

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, communication starts when the CSIHnMCTL2.CSIHnBTST bit is set.

### 15.5.7.4 Summary

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

**Table 15.42 Start of Data Transfer**

Memory and Operating Mode		Transfer Mode	
		Transmit-Only Transmit/Receive	Receive-Only
FIFO, direct access	Master	Writing to the CSIHnTX0W register or the CSIHnTX0H register	Writing to the CSIHnTX0W register or the CSIHnTX0H register
	Slave	Clock reception from master	Incoming clock from the master
Transmit-only buffer, dual buffer	Master	CSIHnMCTL2.CSIHnBTST = 1	CSIHnMCTL2.CSIHnBTST = 1
	Slave	Clock reception from master	Clock reception from master

## 15.5.8 Data Length Selection

### 15.5.8.1 Data Length from 2 to 16 Bits

The length of a data packet is selectable for each chip select signal from 2 to 16 bits using CSIHnCFGx.CSIHnDLSx[3:0]. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000<sub>B</sub>):

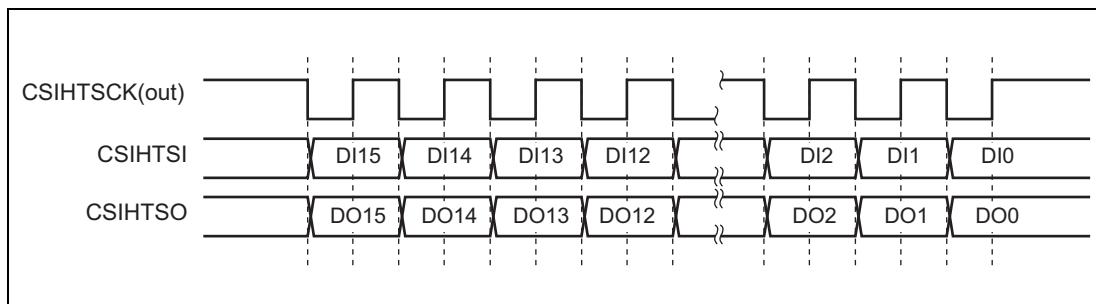


Figure 15.22 16 Bit Data Length, MSB First

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110<sub>B</sub>):

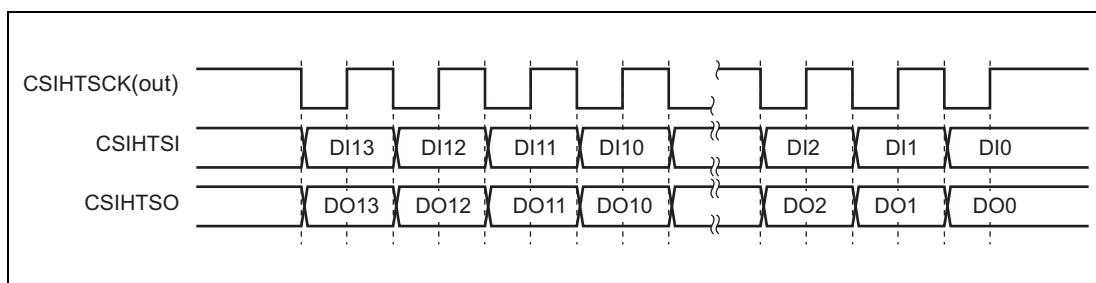


Figure 15.23 14 Bit Data Length, MSB First

### 15.5.8.2 Data Length Greater than 16 Bits

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) function can be used.

EDL function is enabled by setting the CSIHnCTL1.CSIHnEDLE bit to 1.

EDL function works as follows:

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIHnCFGx.CSIHnDLSx[3:0].
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bit setting.
- The transfer is complete after a block with the specified data length (the remainder of data specified with CSIHnTX0W.CSIHnEDL = 0) has been sent.

#### Example

Example for sending 40-bit data ( $123456789A_H$ ) to CS0:

40 bits are split into two 16-bit blocks plus 8 bits.

- Initialize CSIHnCFG0.CSIHnDLS0[3:0] = 8.
- To transmit  $123456789A_H$  with MSB first, write the following sequence to CSIHnTX0W:
  - $20FE\ 1234_H$  (CSIHnTX0W.CSIHnEDL = 1)
  - $20FE\ 5678_H$  (CSIHnTX0W.CSIHnEDL = 1)
  - $00FE\ 009A_H$  (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

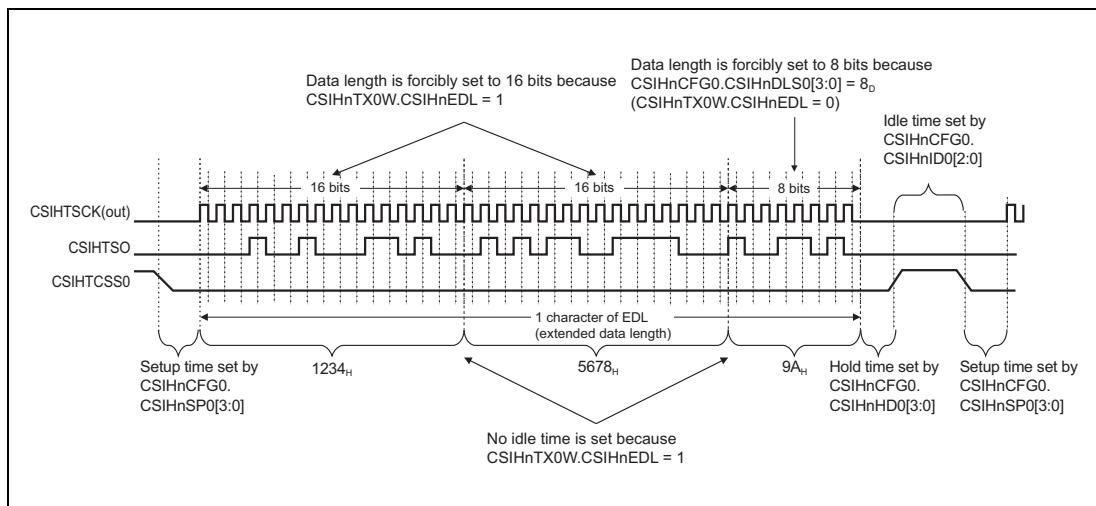


Figure 15.24 EDL Timing Diagram

#### NOTES

1. 1-bit data length is allowed only when using the EDL mode.
2. If parity is enabled, the parity bit is added after the last bit.
3. When data is sent using extended data length (EDL) function, use the same chip select signal.
4. Example for configuring the data direction:
  - Data to be sent:  $123456_H$
  - MSB first:  
Set CSIHnCFGx.CSIHnDIRx = 0  
Write CSIHnTX0W = 20FE 1234H (EDL bit = 1)  
Write CSIHnTX0W = 00FE 0056H (EDL bit = 0)
  - LSB first:  
Set CSIHnCFGx.CSIHnDIRx = 1  
Write CSIHnTX0W = 20FE 3456H (EDL bit = 1)  
Write CSIHnTX0W = 00FE 0012H (EDL bit = 0)
5. Operation is not guaranteed if CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are simultaneously set to "1" while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1.
6. EDL mode cannot be used in receive-only mode of slave mode.  
(CSIHnCTL2.CSIHnPRS[2:0] = 111<sub>B</sub>, CSIHnCTL0.CSIHnTXE = 0,  
CSIHnCTL0.CSIHnRXE = 1)

### 15.5.9 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show communication for a data length of 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000<sub>B</sub>).

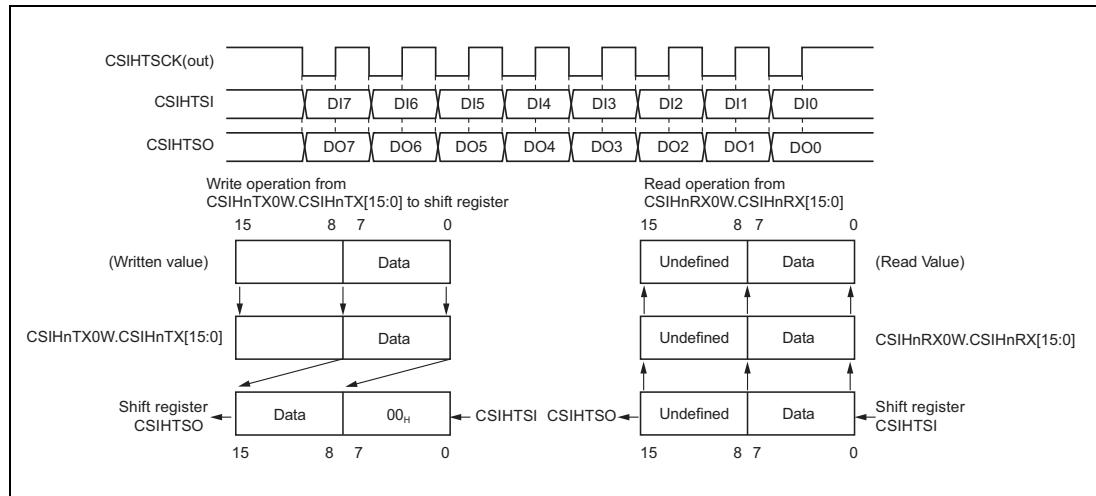


Figure 15.25 Serial Data Direction Select Function - MSB First (CSIHnDIRx = 0)

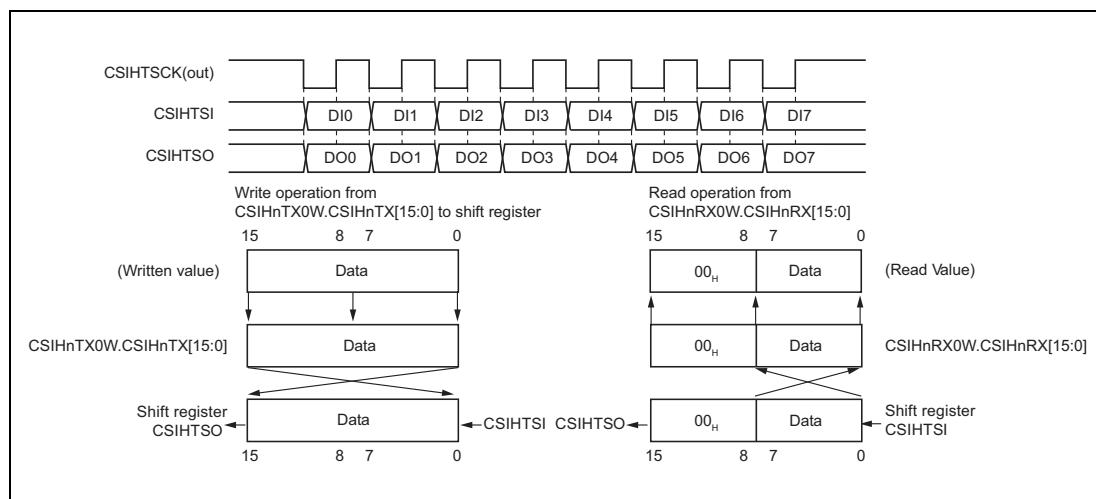


Figure 15.26 Serial Data Direction Select Function - LSB First (CSIHnDIRx = 1)

### 15.5.10 Slave Select (SS) Function

The Slave Select (SS) function enables communication between one master and multiple slaves (SPI communications).

In master mode, the master device outputs the slave select signal (CSIHTCSSx) to a slave.

Communication by a device in slave mode is enabled when the slave input select signal (CSIHTSSI) is at the low level.

See the **Section 15.5.2, Master/Slave Connections**, for examples of connections using the SS function.

#### 15.5.10.1 Communication Timing Using SS Function

The following figure illustrates the communication signal and timings using the SS function.

In slave mode, the data transfer configuration is determined by the CSIInCFG0 register.

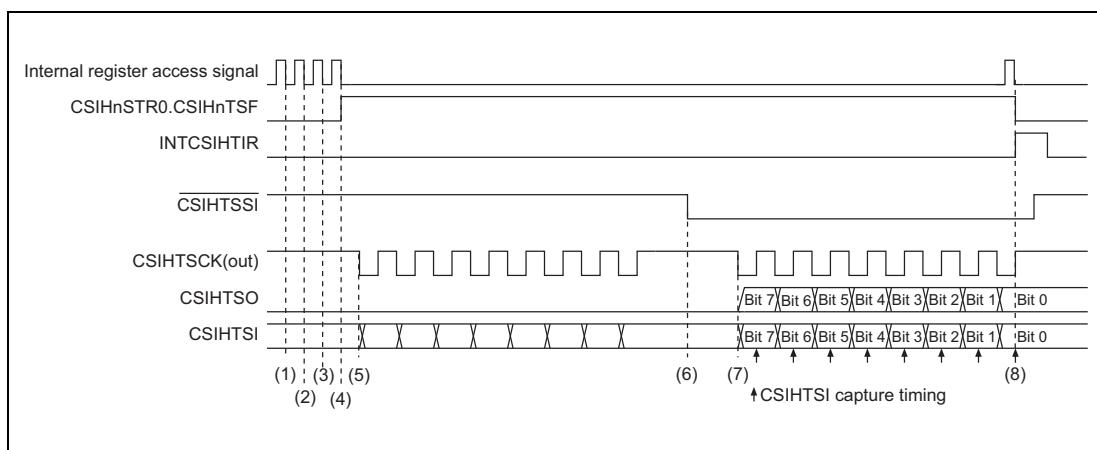


Figure 15.27 Tx/Rx Timing of Communication Using SS Function

- (1) CSIH enters slave mode by setting CSIInCTL2.CSIInPRS[2:0] = 111<sub>B</sub>.  
CSIInCFG0.CSIInCKP0 and CSIInCFG0.CSIInDAP0 are 0.
- (2) The data length is 8 bits (CSIInCFG0.CSIInDLS0[3:0] = 1000<sub>B</sub>). The data direction is MSB first (CSIInCFG0.CSIInDIR0 = 0).
- (3) The transmit/receive mode is set (CSIInCTL0.CSIInTXE = 1, CSIInCTL0.CSIInRXE = 1, and CSIInCTL0.CSIInPWR = 1). Communication start is permitted.
- (4) The transfer status flag CSIInSTR0.CSIInTSF is automatically set when transfer data is written to the CSIInTX0W or CSIInTX0H transmission register during direct access mode or FIFO mode.
- (5) While the signal CSIHTSSI is at high level, transmission/reception is not started, even if an external transmission clock CSIHTSCK is input. Input to CSIHTSI is ignored.
- (6) CSIHTSSI falling to low level indicates that CSIHTSO is enabled and ready for transmission.
- (7) As soon as the external clock signal CSIHTSCK is detected, the slave transmits data to CSIHTSO and simultaneously captures data from CSIHTSI.
- (8) Interrupt INTCSIHTIR indicates that the reception is complete. The CSIInRX0W/H register can be read.

### 15.5.10.2 CSIHTSSO Operation

CSIHnPWR	CSIHnTXE	CSIHnRXE	CSIHnSSE	CSIHTSSO
0	—	—	—	H
1	—	—	0	H
	0		1	H
	1		1	Inverse value of <u>CSIHTSSI</u> level

The CSIHTSSO pin is a signal to control the I/O function of the chip's SO pin when using the SS function.

The CSIHTSO pin is enabled when the CSIHTSSO pin is "High" (the chip's SO pin is being driven).

The CSIHTSO pin is disabled when the CSIHTSSO pin is "Low" (the chip's SO pin is not being driven).

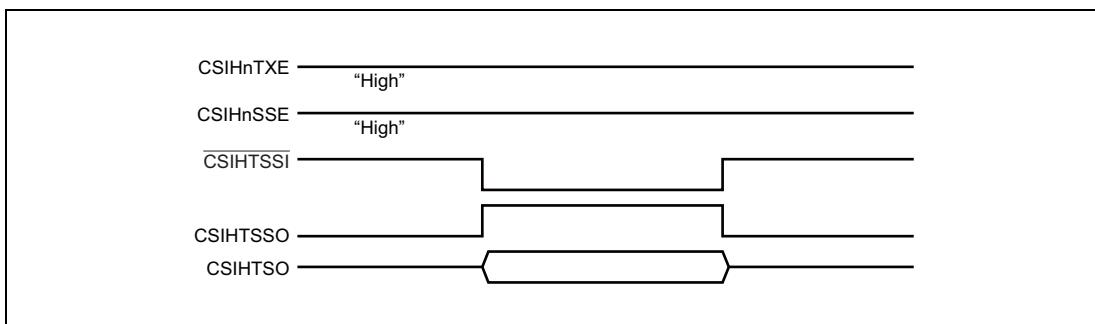


Figure 15.28 Operation of CSIHTSSO

#### CAUTION

If CSIHTSSI pin is changed during communication (CSIHnSTR0.CSIHnTSF = 1), current communication is not guaranteed.

### 15.5.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIHnCTL1.CSIHnHSE bit. For handshake, the signals CSIHTRYI and CSIHTRYO are used.

The busy timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

#### 15.5.11.1 Slave Mode

If CSIHnCTL1.CSIHnHSE = 1, a low-level CSIHTRYO signal is output when the slave becomes busy. This can happen in two cases:

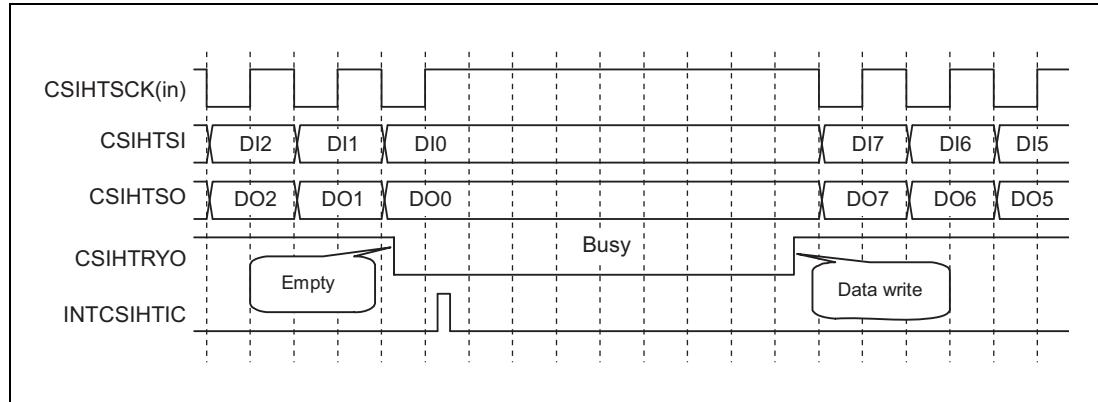
- When the next data to be sent is not ready:

When the slave is in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1) and is in any of the states listed below, the CSIHTRYO outputs the busy state (low level).

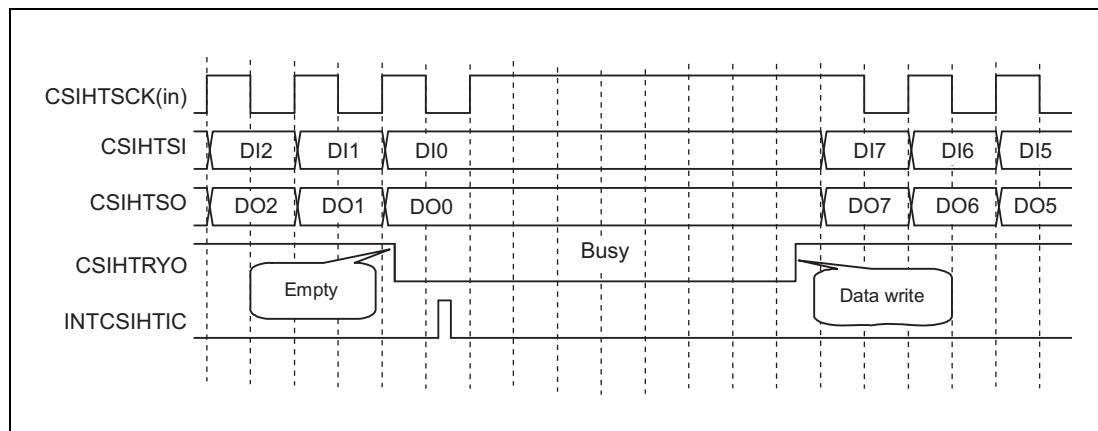
**Table 15.43 Memory Mode and Slave Transfer State**

Memory Mode	Slave Transfer State
Direct access mode	When there is no more data to be sent
FIFO mode	When there is no more data to be sent (CSIHnSTR0.CSIHnEMF = 1)
Dual buffer mode	When CSIHnMCTL2.CSIHnBTST is not set to 1
Transmit-only buffer mode	

The following examples assume an eight-bit data length.



**Figure 15.29 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 0)**



**Figure 15.30 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 1)**

2. When receive register is full:

When slave is configured in receive-only mode or transmit/receive mode

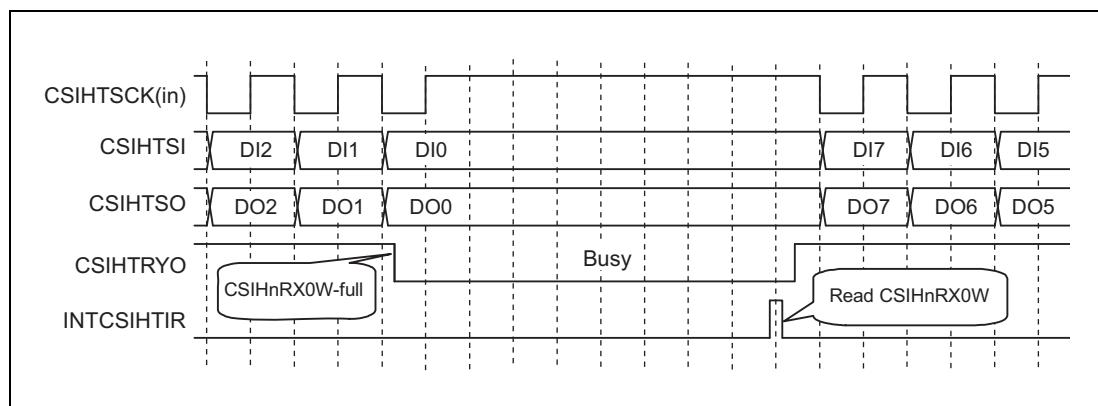
(CSIHnCTL0.CSIHnRXE = 1), and new data cannot be copied from a shift register to CSIHnRX0W/H because the previously received data is still in the CSIHnRX0W/H register (CSIHnRX0W/H is full).

When CSIHnCTL0.CSIHnRXE is 1 and is in any of the following states, CSIHTRYO outputs busy state (low level).

**Table 15.44 Memory Mode and Slave Reception State**

Memory Mode	Slave Reception State
Direct access mode	When CSIHnRX0W or CSIHnRX0H is full
FIFO mode	When received data is remaining in buffer (CSIHnSTR0.CSIHnFLF = 1)
Dual buffer mode	No applicable case
Transmit-only buffer mode	When CSIHnRX0W or CSIHnRX0H is full

The following examples assume an eight-bit data length.



**Figure 15.31 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)**

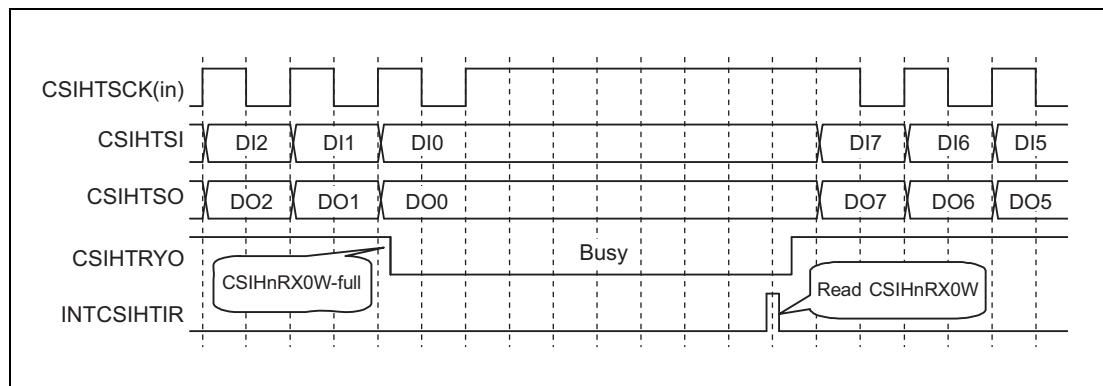


Figure 15.32 Busy Signal from the Slave (Direct Access Mode;  
CSIHnCFGx.CSIHnDAPx = 1)

### 15.5.11.2 Master Mode

When the master detects  $\text{CSIHTRYI} = 0$  while  $\text{CSIHnCTL1.CSIHnHSE} = 1$ , the subsequent transfers are put on hold, the master goes into wait state and suspends the  $\text{CSIHTSCK}$  clock.

The  $\text{CSIHTRYI}$  level is checked at each half clock cycle of  $\text{CSIHTSCK}$ .

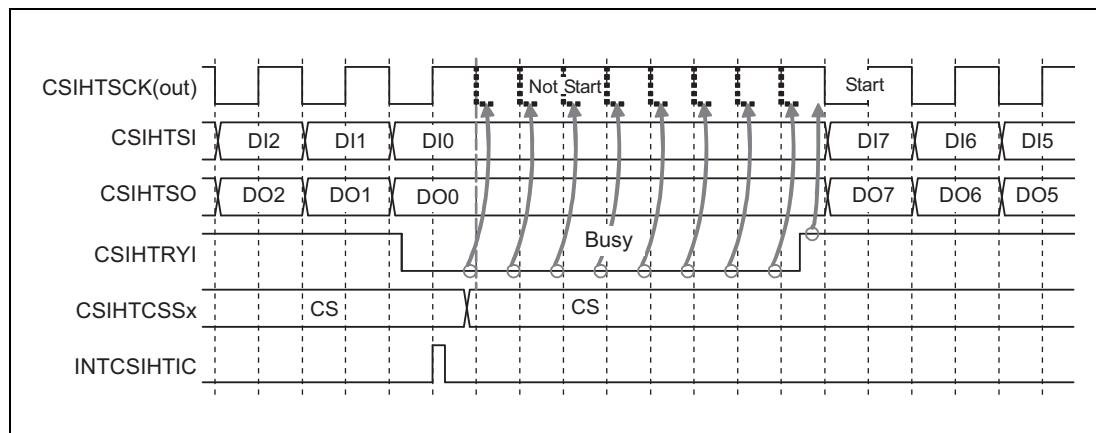


Figure 15.33 Master's Reaction to  $\text{CSIHTRYI}$  ( $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )

The  $\text{CSIHTRYI}$  signal must be pulled down by the slave before the next transfer starts. If this is done while data transfer is in progress, the serial clock from the master is suspended after the transfer is complete.

The master resumes the communication as soon as  $\text{CSIHTRYI}$  becomes high (the slave is “ready”).

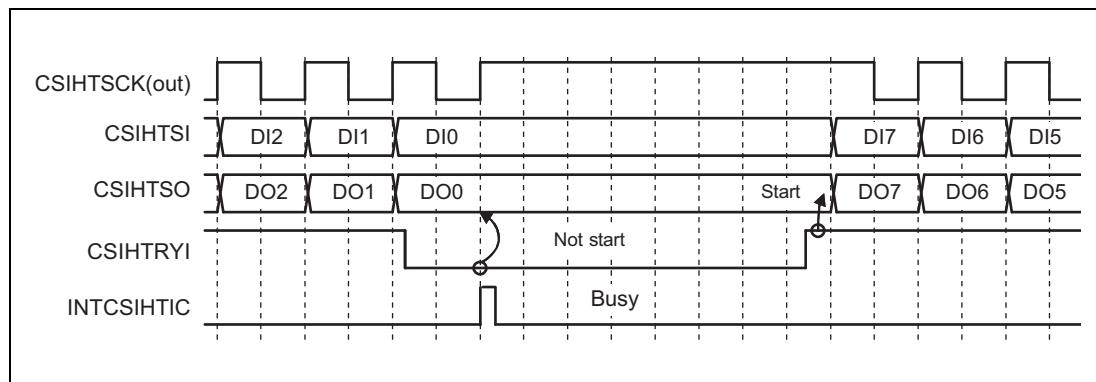


Figure 15.34 Master's Reaction to  $\text{CSIHTRYI}$  ( $\text{CSIHnCFGx.CSIHnDAPx} = 1$ )

#### CAUTIONS

1. If multiple slaves are connected, the master must only detect the  $\text{CSIHTRYI}$  signal of the slave it has selected for communication.
2. Even when the  $\text{CSIHTRYI}$  pin of the master detects a  $\text{CSIHTRYO}$  signal from the slave during data transfer, the communication is not put on hold but continues until the data transfer is completed.

### 15.5.12 Error Detection

CSIH can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Time-out error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency and time-out errors can be enabled/disabled individually.

If any of these errors is detected, the interrupt request INTCSIHTIRE is generated and the corresponding flags are set.

#### 15.5.12.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIHnCTL1.CSIHnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn\_m = 1). It will not be enabled if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIHTSO are read back via the CSIHTDCS signal into the corresponding shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency error and:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnDCE bit is set.

Additionally, CSIHnRX0W.CSIHnTDCE of data that contains the error is set.

The data consistency check function is illustrated in the following block diagram.

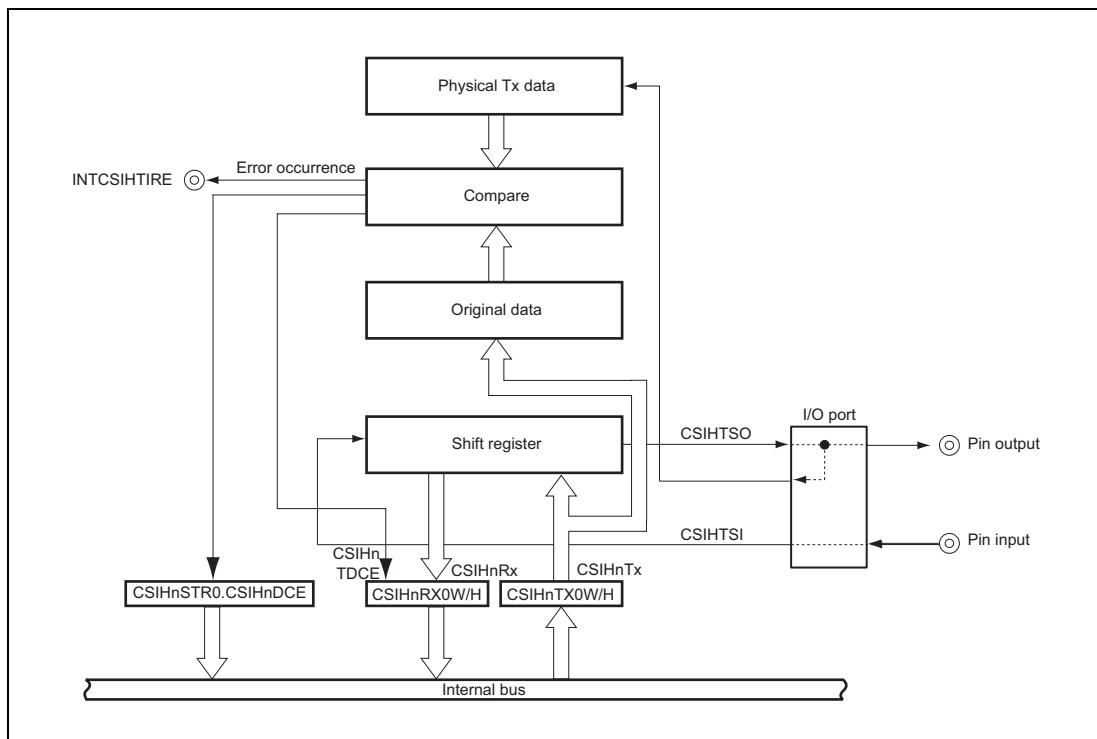


Figure 15.35 Data Consistency Check Functional Block Diagram

### 15.5.12.2 Parity Check

CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIHnCFGx.CSIHnPSx[1:0].

Parity check is enabled if CSIHnCFGx.CSIHnPSx[1] = 1.

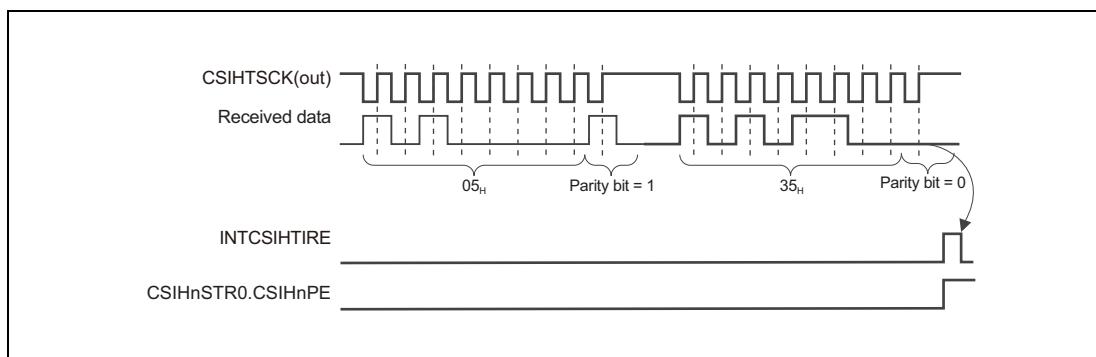
The parity bit is checked after a reception is complete. In case of parity error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnPE bit is set.

Additionally, CSIHnRX0W.CSIHnRPE of data that contains the error is set.

The figure below shows an example.

- Data length is 8 bits.
- The data to be transmitted is  $05_{\text{H}}$  and  $35_{\text{H}}$ .
- Data direction is LSB first.
- Parity type is odd.



**Figure 15.36 Parity Check Example**

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

### 15.5.12.3 Time-Out Error

Time-out errors can be checked only in slave FIFO mode.

This error occurs if neither of the following occurred within a certain period of time:

- Received data in FIFO is read
- FIFO receives data from CSIHTSI

The time-out time is defined in CSIHnMCTL0.CSIHnTO[4:0] in units of “8 x transmission clock CSIHTSCK”. A time-out error occurs when the specified time is exceeded (the time-out time is not detected when CSIHnMCTL0.CSIHnTO[4:0] = 00000<sub>B</sub>).

The dedicated time-out counter is set by the CSIHnCTL2.CSIHnPRTS[2:0] and CSIHnBRSy.CSIHnBRS[11:0] bits.

If the value of the CSIHnBRSy.CSIHnBRS[11:0] bits is left as 000<sub>H</sub>, the dedicated time-out counter does not operate.

The dedicated time-out counter measures the time between the last and the next read operation.

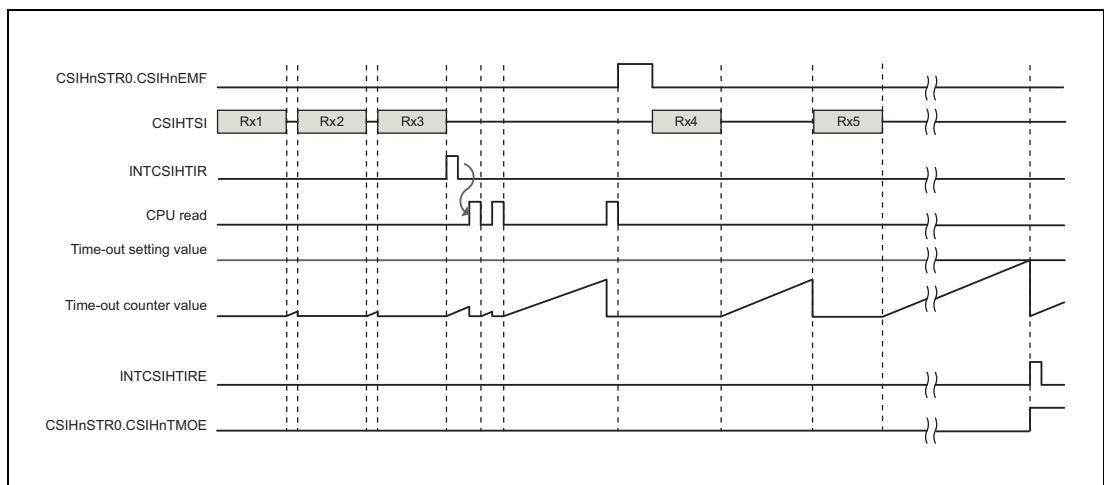


Figure 15.37 Time-Out Check Functional Timing Diagram

The start timing of the time-out counter is as follows:

- When reception is completed
- When data read from the CPU is completed  
(The counter does not start if the buffer is empty.)
- When a time-out error is detected

After a time-out error is detected, if data is still available in FIFO, the time-out counter restarts.

If the value set by bits CSIHnMCTL0.CSIHnTO[4:0] is reached again, the INTCSIHTIRE interrupt is output again.

The timeout counter continues to count until received data is read. To stop the counter, read all received data or set CSIHnSTCR0.CSIHnPCT to 1. Note that the pointer is cleared if you perform the latter.

The counter is reset at the following timing:

- Data is read once.
- A new data item is received.

- A timeout error is detected.
- The CSIInSTCR0.CSIInPCT bit is set to 1.

If a timeout error occurs, the following occur:

- Interrupt INTCSIHTIRE is generated.
- The CSIInSTR0.CSIInTMOE bit is set.

#### 15.5.12.4 Overflow Error

An overflow error can occur in FIFO mode. It occurs when transmission data is written to the CSIInTX0W register while the FIFO buffer is filled with received data.

##### Example

100 data packets have been transmitted. That means, the FIFO contains 100 received data packets. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data packets to the FIFO. However, only 10 received data packets have been read up to now, and 90 are still in the FIFO.

In this case, only 38 buffers are available for new transmission data. When the CPU tries to write the 39th data, an overflow error occurs.

This is illustrated in the following figure.

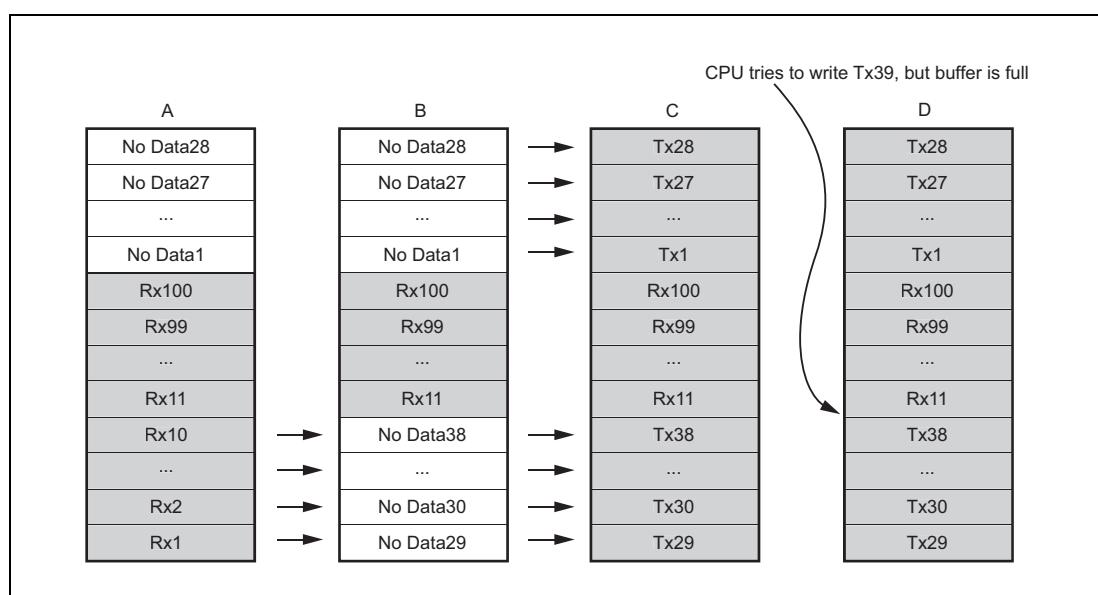
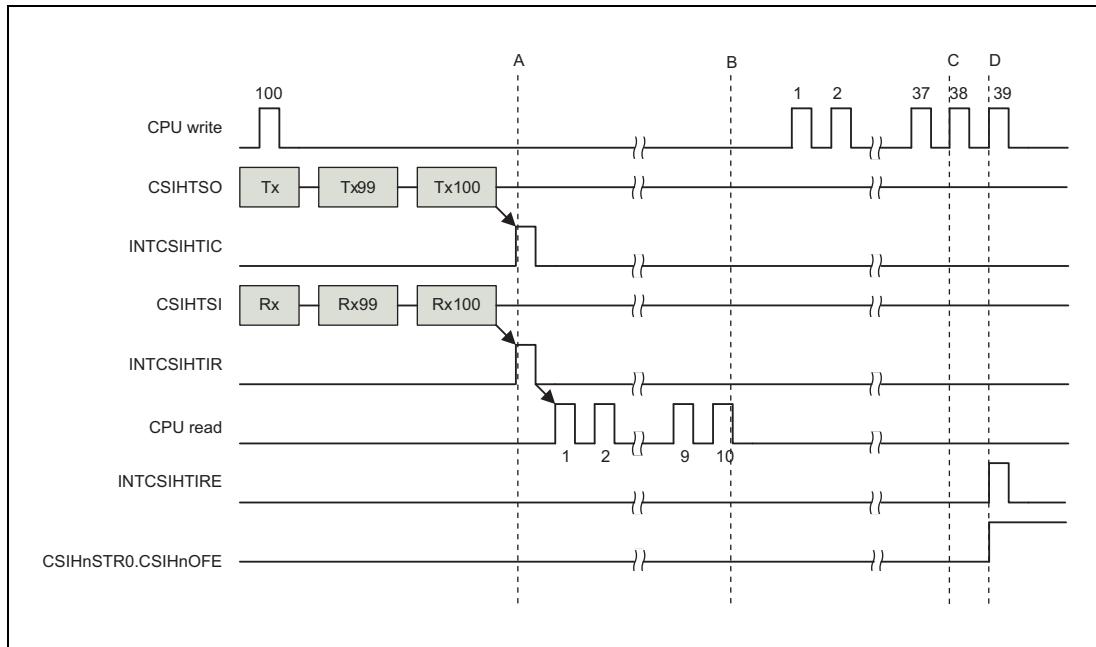


Figure 15.38 FIFO Overview

The 39th and subsequent data packets are discarded. The figure below shows the overflow timing.



**Figure 15.39 FIFO Overflow Timing**

In case of overflow error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOFE bit is set.

### 15.5.12.5 Overrun Error

An overrun error can occur in direct access, transmit-only buffer, and FIFO modes. It does not occur in dual buffer mode. The overrun error does not occur if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

There are two conditions for overrun errors.

#### Condition for errors 1

- In FIFO mode, if the CPU reads the CSIHnRX0W/H register when the number of received data is 0.

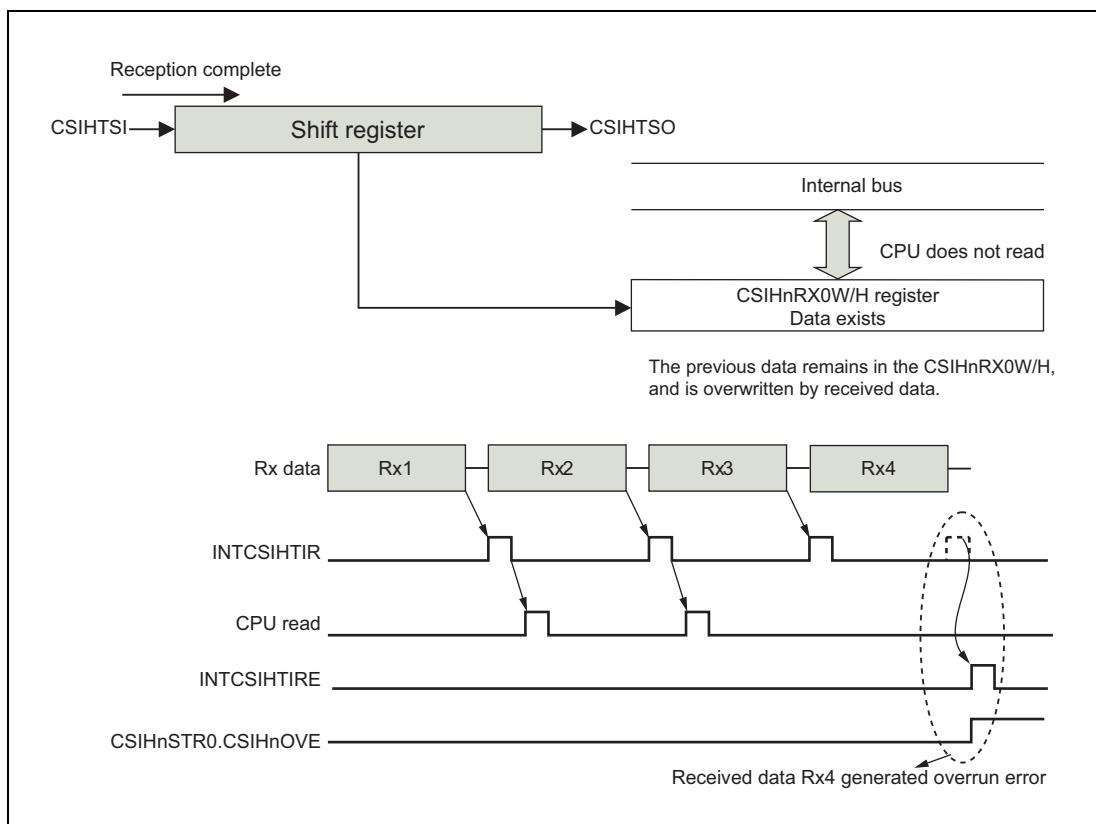
#### Condition for errors 2

- In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled):
  - In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data remains in the CSIHnRX0W/H register.
  - In FIFO mode, when reception is completed while the FIFO buffer is still full of receive data.

#### (1) Direct access/transmit-only buffer

In direct access and transmit-only buffer modes, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSIHnRX0W/H. This happens when CSIHnRX0W/H was not read and therefore contains previously received data.

The following figure illustrates the overrun error detection function.



**Figure 15.40 Overrun Error Detection in Direct Access and Transmit-Only Buffer Modes**

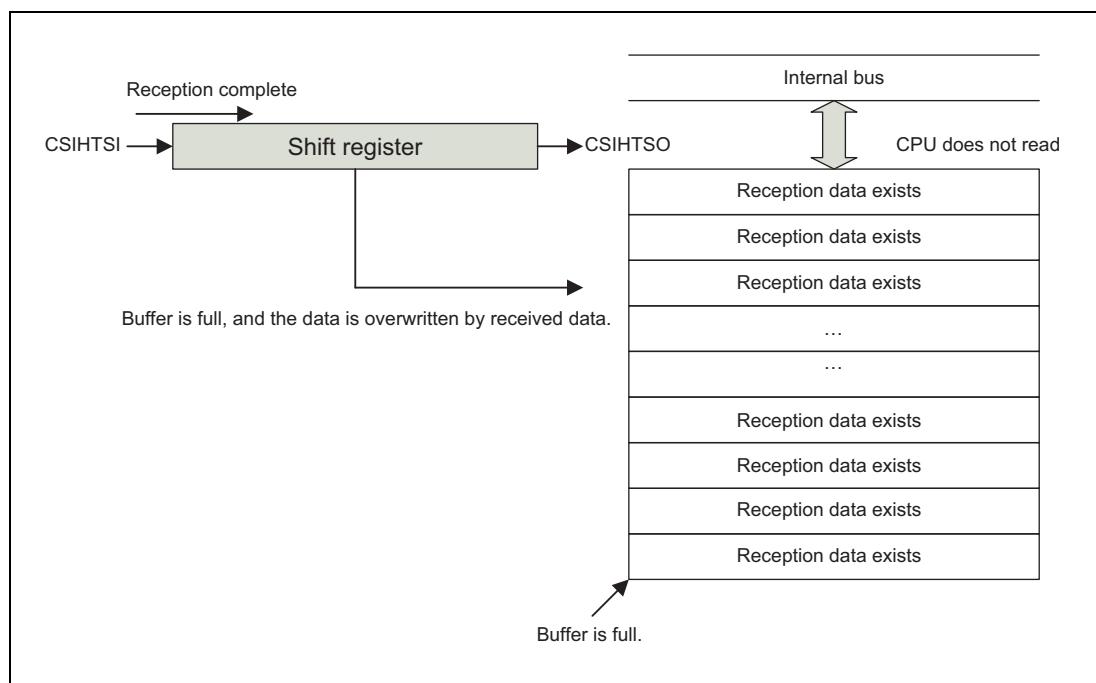
**NOTE**

An overrun error can be avoided in slave mode by using handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

**(2) FIFO mode**

In FIFO mode, this error occurs if:

1. Newly received data cannot be transferred from the shift register to the FIFO because the FIFO is full.

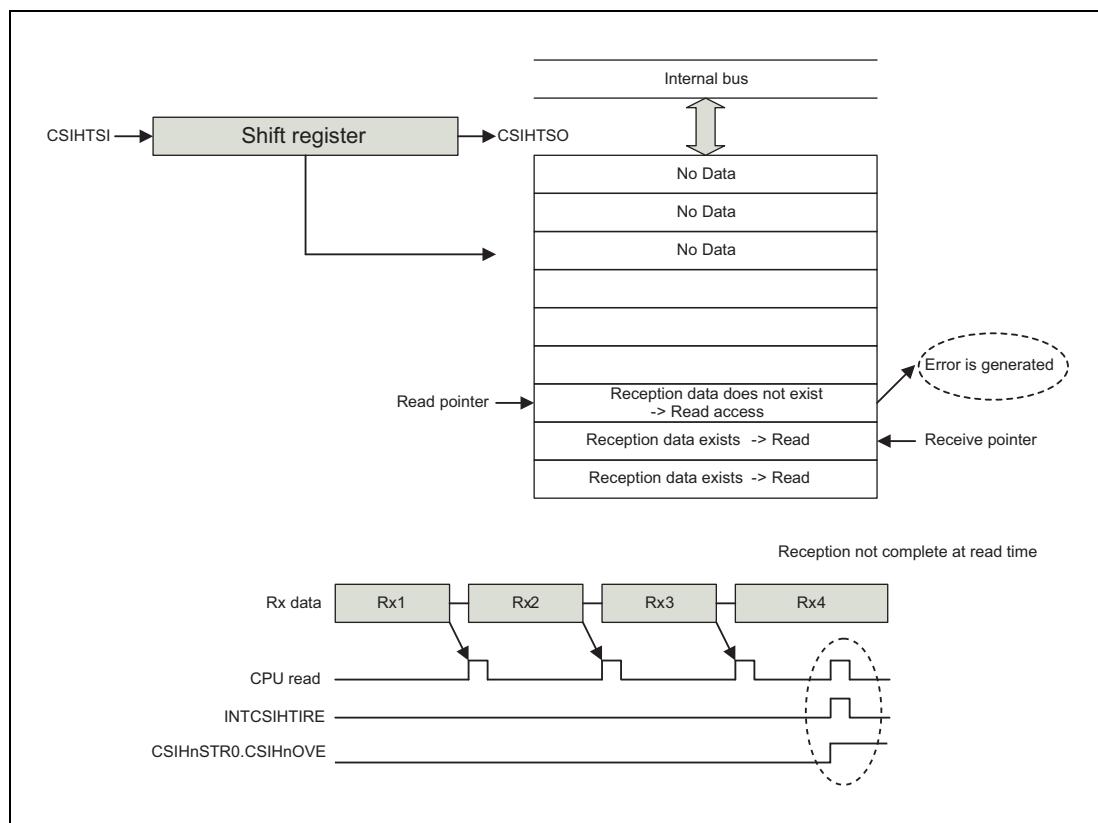


**Figure 15.41 Overrun Error Detection in FIFO Mode (FIFO Full)**

**NOTE**

An overrun error can be avoided in slave mode by using the handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

2. The CPU attempts to read non-existent receive data.



**Figure 15.42 Overrun Error Detection in FIFO Mode (No Data)**

In case of overrun error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOVE bit is set.
- Received data is overwritten and the communication continues.  
(When the CPU tries to read non-existent data, it waits until reception is completed and then resumes reading.)

For details see **Section 15.5.11, Handshake Function**.

### 15.5.13 Loop-Back Mode

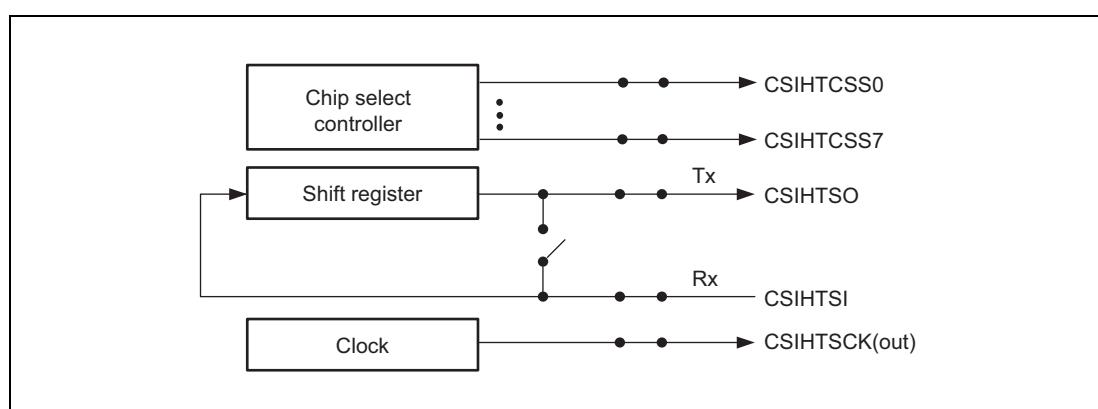
Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIHnCTL1.CSIHnLBM = 1), CSIHTCSSx is fixed to the inactive level (the active level is defined by the CSIHnCTL1.CSIHnCSLx value). The transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHTSCK, CSIHTSO, CSIHTSI, and CSIHTCSSx are disconnected from the ports. In addition, the CSIHTSO output level is fixed to low, and CSIHTSCK is set to reset level (High) regardless of the value of the CSIHnCFGx.CSIHnCKPx. The rest of CSIH works as in normal operation.

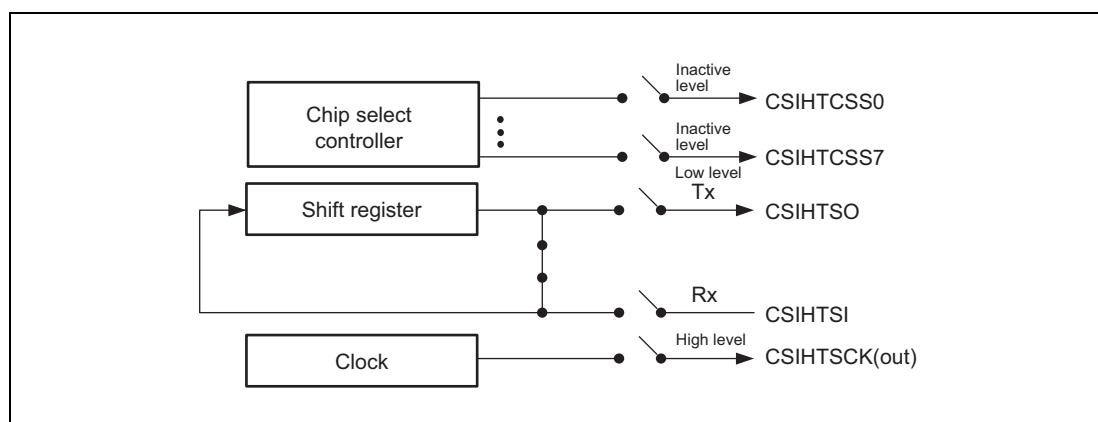
In order to test CSIH, put it in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data. Any connected device remains unaffected by the loop-back test.

**Table 15.45 Pin Output Level in Loop-Back Mode**

Pin Name	Output level
CSIHTSCK(out)	High level
CSIHTCSS[7:0]	Inactive level
CSIHTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIHTRYO	Normal function (Low level)



**Figure 15.43 Normal Operation**

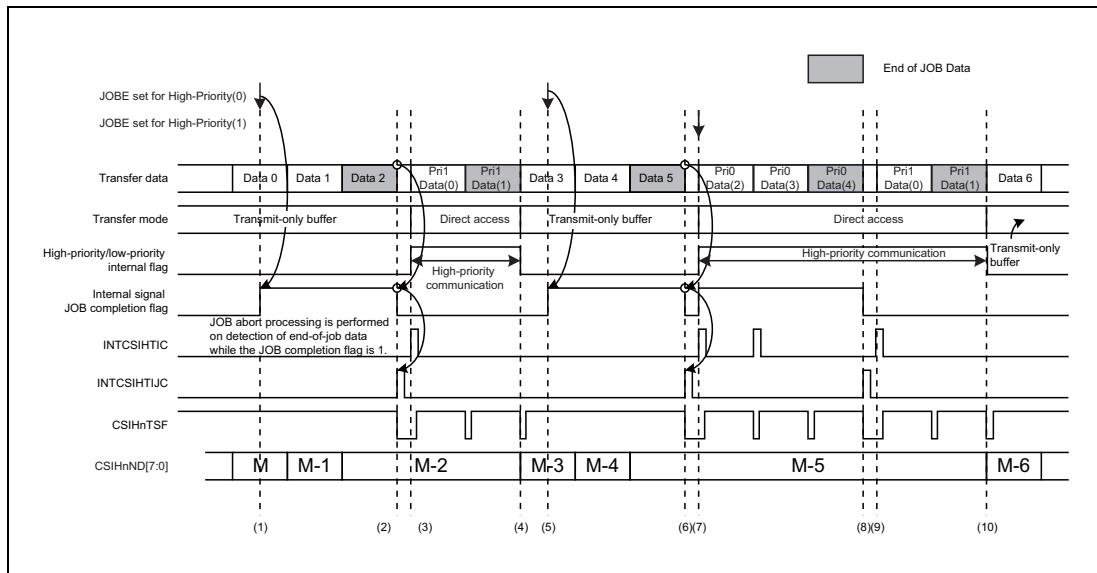


**Figure 15.44 Loop-Back Mode Operation**

### 15.5.14 CPU-Controlled High Priority Communication Function

CSIH has a function to abort low priority communication to perform high priority communication if it receives a high-priority communication request from the CPU while low-priority communication is being used. This function supports transmit-only buffer mode as low priority communication and only direct access mode as high-priority communication. To enable this function, CSIHnCTL1.CSIHnPHE and CSIHnCTL1.CSIHnJE must be set to 1.

The following figure illustrates CPU-controlled high-priority communication.



**Figure 15.45 Example of CPU-Controlled High-Priority Communication**

- (1) By setting CSIHnCTL0.CSIHnJOBE = 1 during low-priority communication, start of high-priority communication following end-of-job data is notified, and the internal signal flag is set.
- (2) When end-of-job data is detected, the current low-priority communication is aborted and the INTCSIHTIJC interrupt occurs. The internal signal job completion flag is cleared due to the communication abort, and memory mode is automatically switched to direct access mode for the subsequent high-priority communication.
- (3) The CPU detects the interrupt and starts communication by writing the first transmission data of high-priority communication to CSIHnTX0W or CSIHnTX0H.
- (4) When end-of-job data is detected, communication is aborted. At this time, because the internal signal end-of-job flag is set to 0, the CSIH determines that the next communication is low-priority, automatically switches memory mode to transmit-only buffer mode, and then resumes the aborted low-priority communication.
- (5) Same as (1) above.
- (6) Same as (2) above.
- (7) The CPU detects an interrupt and starts communication by writing the first transmission data of high-priority communication to CSIHnTX0W or CSIHnTX0H. The CPU sets CSIHnCTL0.CSIHnJOBE = 1 again to notify that the next communication is high-priority.
- (8) When end-of-job data is detected, communication is aborted and the INTCSIHTIJC interrupt is generated. At this time, the CPU determines that the subsequent communication is also high-priority because the internal signal JOB completion flag is 1, and waits for communication to start.

(9) Same as (3) above.

(10) Same as (4) above.

### CAUTION

**Memory mode is switched automatically when communication is changed from low priority to high priority (from transmit-only buffer mode to direct access mode) and from high priority to low priority (from direct access mode to transmit-only buffer mode).**

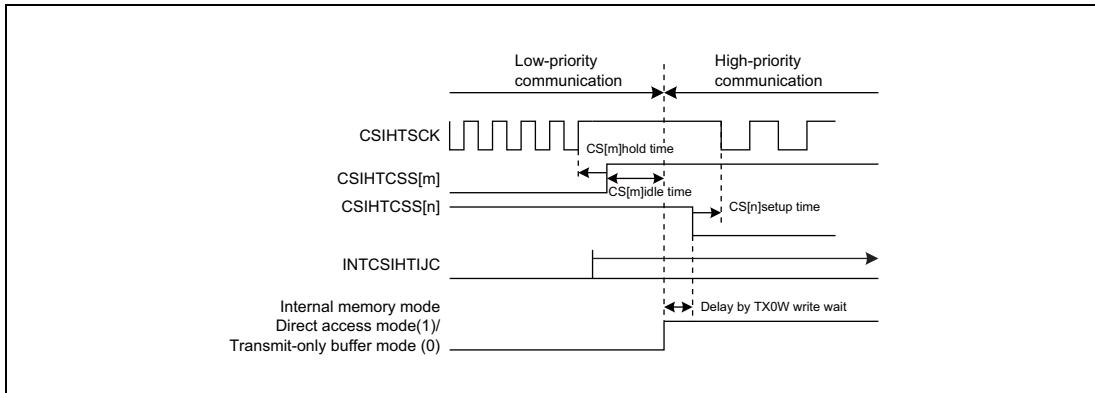


Figure 15.46 Transition from Low-Priority Mode to High-Priority Mode

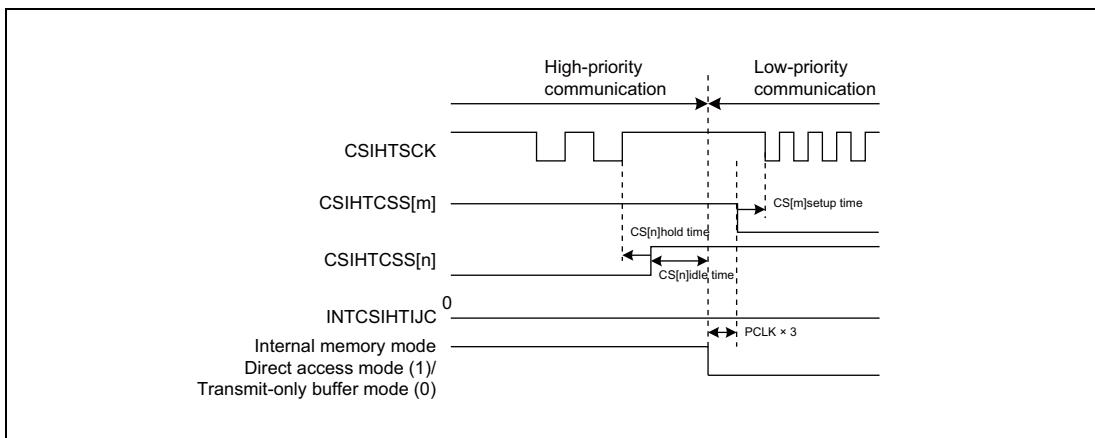


Figure 15.47 Transition from High-Priority Mode to Low-Priority Mode

Do not conduct write operation of communication data or CSIInCTL0.CSIInJOBE bit operation during setting inhibited period to switch low and high priority communication mode correctly.

CSIInTX0W register write inhibited period:

- Period from when CSIInJOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.
- Period from when the last data of high priority communication (End of JOB data) is written to when the CSIInHPST state = 0 is detected.

CSIInJOBE register write inhibited period:

- Period from when CSIInJOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.

During high communication mode period, there is no setting inhibited period for CSIHnJOBE bit. It is possible to set CSIHnJOBE bit before writing communication data. For example, to communicate multiple JOB data in high priority mode, it is possible to set CSIHnJOBE bit before writing the first communication data.

#### **CAUTION**

**When CSIHnJOBE bit is set right before the last communication of high priority communication ends, different operations are required depending on the timing at which setting of CSIHnJOBE bit is internally detected.**

**If the setting of the CSIHnJOBE bit is detected before the transfer of the last bit is completed, high priority communication mode continues.**

**When setting of the CSIHnJOBE bit is detected after the transfer of the last bit is completed, the mode temporarily returns to low priority communication. After detection of End of JOB data in low priority communication, the mode changes back to high priority communication.**

### 15.5.15 Enforced Chip Select Idle Setting

It is possible to insert an idle state between two consecutive data transfers by setting CSIHnCFGx.CSIHnIDLx.

- When CSIHnCFGx.CSIHnIDLx = 0

If the next CSIHTCSSx is the same as the previous one, an idle state is not inserted, but an inter-data time is inserted instead.

If the next CSIHTCSSx is different from the previous one, an idle state is inserted.

- When CSIHnCFGx.CSIHnIDLx = 1

An idle state is always inserted even if a next CSIHTCSSx is not different from the previous one.

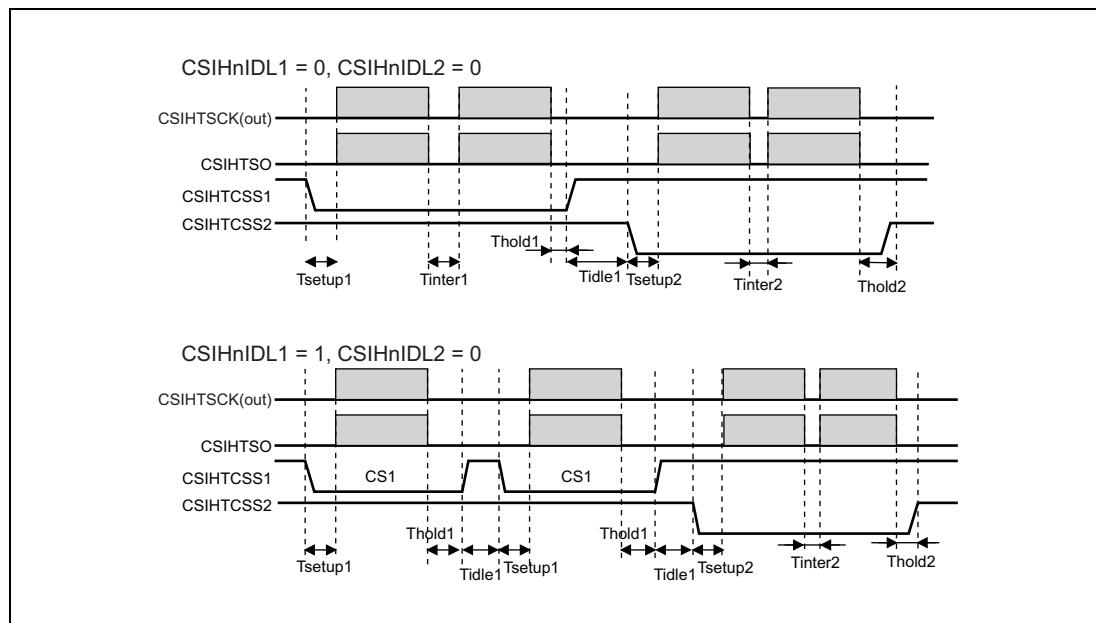


Figure 15.48 Enforced Chip Select Idle Setting Example

#### CAUTION

If the CPU-controlled high priority communication function is enabled (CSIHnCTL1.CSIHnPHE = 1), when the mode is switched from low priority communication to high priority communication or from high priority communication to low priority communication, an IDLE state is inserted regardless of the setting of CSIHnCFGx.CSIHnIDLx bit.

## 15.6 Operating Procedures

The examples and procedures below are described according to the memory mode in the following order:

- Direct access mode
- Transmit-only buffer mode
- Dual buffer mode
- FIFO mode

### 15.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other with job mode enabled.

#### 15.6.1.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below assumes the following conditions:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- Transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is disabled ( $\text{CSIHnCTL1.CSIHnJE} = 0$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- Direct access mode ( $\text{CSIHnCTL0.CSIHnMBS} = 1$ )

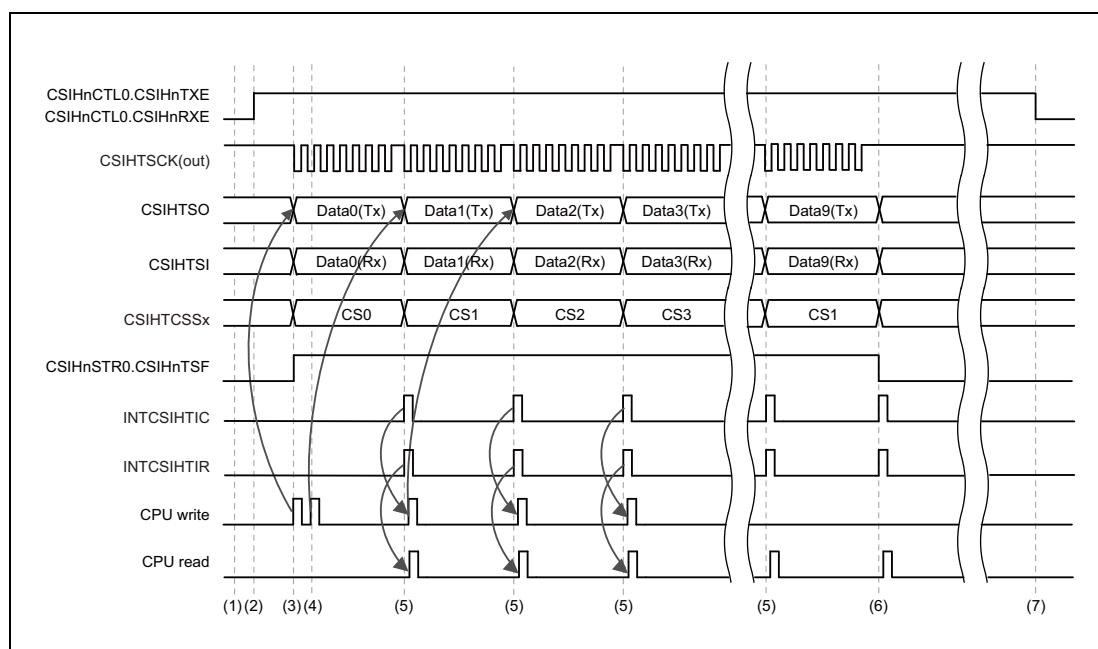


Figure 15.49 Master in Direct Access Mode,  $\text{CSIHnCTL1.CSIHnJE} = 0$

**Procedure:**

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.  
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register, CSIHnTX0W. Within the same write operation, activate CS0. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIHnTX0W. If required, you can change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated:
  - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
  - INTCSIHTIR indicates that the reception register CSIHnRX0W must be read.
6. No more write action is required after completion of data 8. Data 9 (the last data) has been written in advance.  
However, reception register CSIHnRX0W must be read after completion of writing data 8 and 9.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

### 15.6.1.2 Transmission/Reception in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- Transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is enabled ( $\text{CSIHnCTL1.CSIHnJE} = 1$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- Direct access mode ( $\text{CSIHnCTL0.CSIHnMBS} = 1$ )
- Two jobs, each transmitting three data.

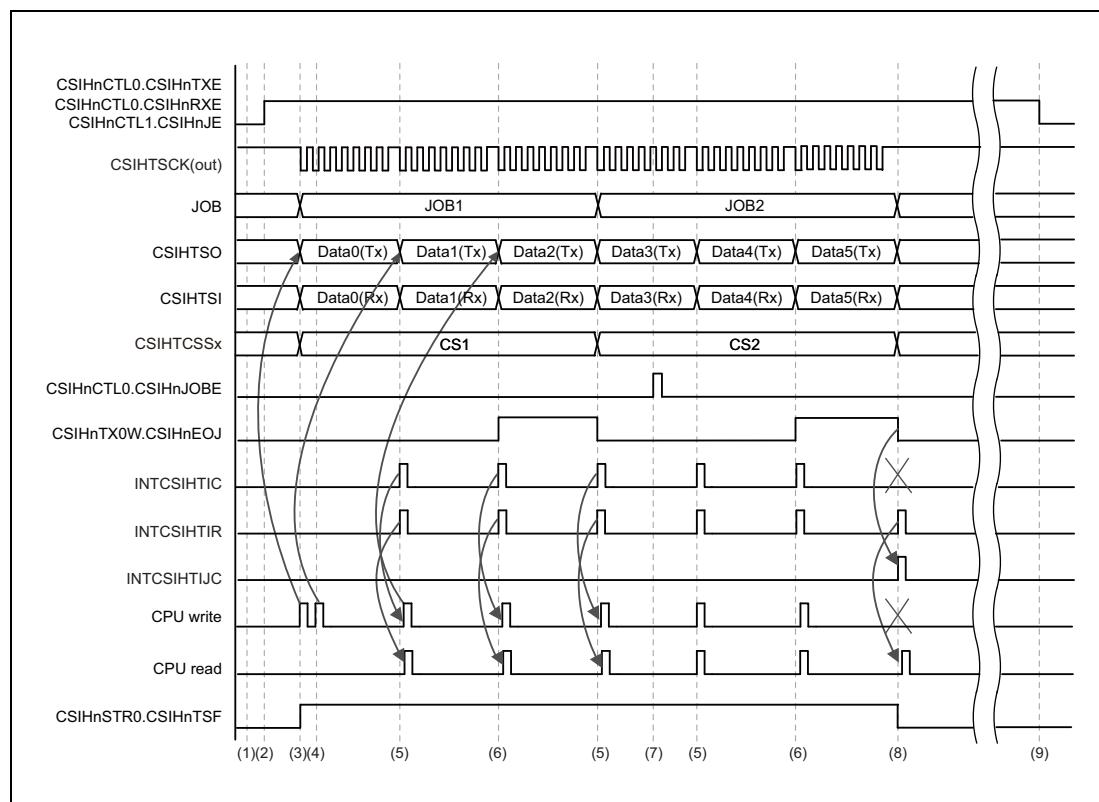


Figure 15.50 Master in Direct Access Mode,  $\text{CSIHnCTL1.CSIHnJE} = 1$

**Procedure:**

1. Configure the communication protocol in the CSIInCFGx register. This example uses chip select signals CS1 and CS2.  
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIInCTL1 and CSIInCTL2.
2. In the CSIInCTL0 register, set CSIInPWR = 1 (enables the clock), CSIInTXE = 1 (permits transmission), CSIInRXE = 1 (permits the reception), and CSIInMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIInTX0W. Transmission starts automatically when the first data becomes available.  
The CSIInSTR0.CSIInTSF flag indicates that communication is in progress.
4. Write the second data to CSIInTX0W. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
  - INTCSIHTIC indicates that the next data can be written to CSIInTX0W.
  - INTCSIHTIR indicates that the reception register, CSIInRX0W must be read.
6. Setting CSIInTX0W.CSIInEOJ = 1 indicates that the last data of the current job is sent. After that, the next job may begin.
7. By setting CSIInCTL0.CSIInJOBE = 1, communication is forced to stop at the end of the current job (JOB2).  
The interrupt request, INTCSIHTIJC indicates that the communication was forcibly stopped at the end of the current job.  
The interrupt request, INTCSIHTIC is not generated. Additionally, the transmission data available in the CSIInTX0W register is not sent.
8. After the forced stop of communication, the interrupt request, INTCSIHTIC is replaced by INTCSIHTIJC. INTCSIHTIR is generated as usual.  
The interrupt request, INTCSIHTIJC indicates that the communication was forcibly stopped at the end of the current job.  
The interrupt request, INTCSIHTIC is not generated. Additionally, the transmission data available in the CSIInTX0W register is not sent.
9. Finally, clear CSIInCTL0.CSIInTXE and CSIInCTL0.CSIInRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIInCTL0.CSIInPWR to 0 to minimize power consumption of the CSIIn.  
To start another transmission without stopping communication, perform steps 3 and later.

## 15.6.2 Procedures in Transmit-Only Buffer Mode

Two examples for a master is provided, one with job mode disabled, and the other one with job mode enabled.

### 15.6.2.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is disabled ( $\text{CSIHnCTL1.CSIHnJE} = 0$ ).
- The number of data is 9 ( $\text{CSIHnMCTL2.CSIHnND[7:0]} = 09_H$ ).
- The transfer start address is  $10_H$  ( $\text{CSIHnMCTL2.CSIHnSOP[6:0]} = 10_H$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- Transmit-only buffer mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS[1:0]} = 10_B$ )

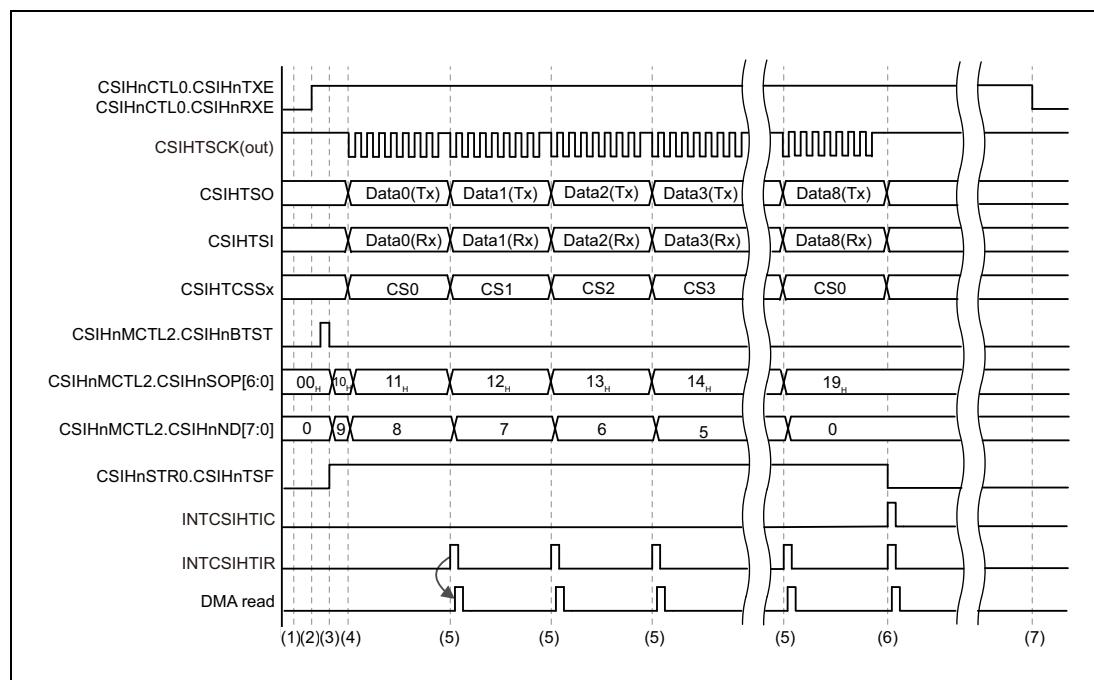


Figure 15.51 Master in Transmit-Only Buffer Mode,  $\text{CSIHnCTL1.CSIHnJE} = 0$

#### NOTE

The procedure of writing the data into the buffer is not described here.

**Procedure:**

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.  
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.  
Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].  
Set CSIHnMCTL0.CSIHnMMS[1:0] = 10<sub>B</sub> (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the receive register CSIHnRX0W must be read.
6. When all transmissions are complete, the interrupt request, INTCSIHTIC is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

### 15.6.2.2 Transmission/Reception in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is enabled ( $\text{CSIHnCTL1.CSIHnJE} = 1$ ).
- The number of data is 8 ( $\text{CSIHnMCTL2.CSIHnND[7:0]} = 08_H$ ).
- The transfer start address is  $10_H$  ( $\text{CSIHnMCTL2.CSIHnSOP[6:0]} = 10_H$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- Transmit-only buffer mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS[1:0]} = 10_B$ )

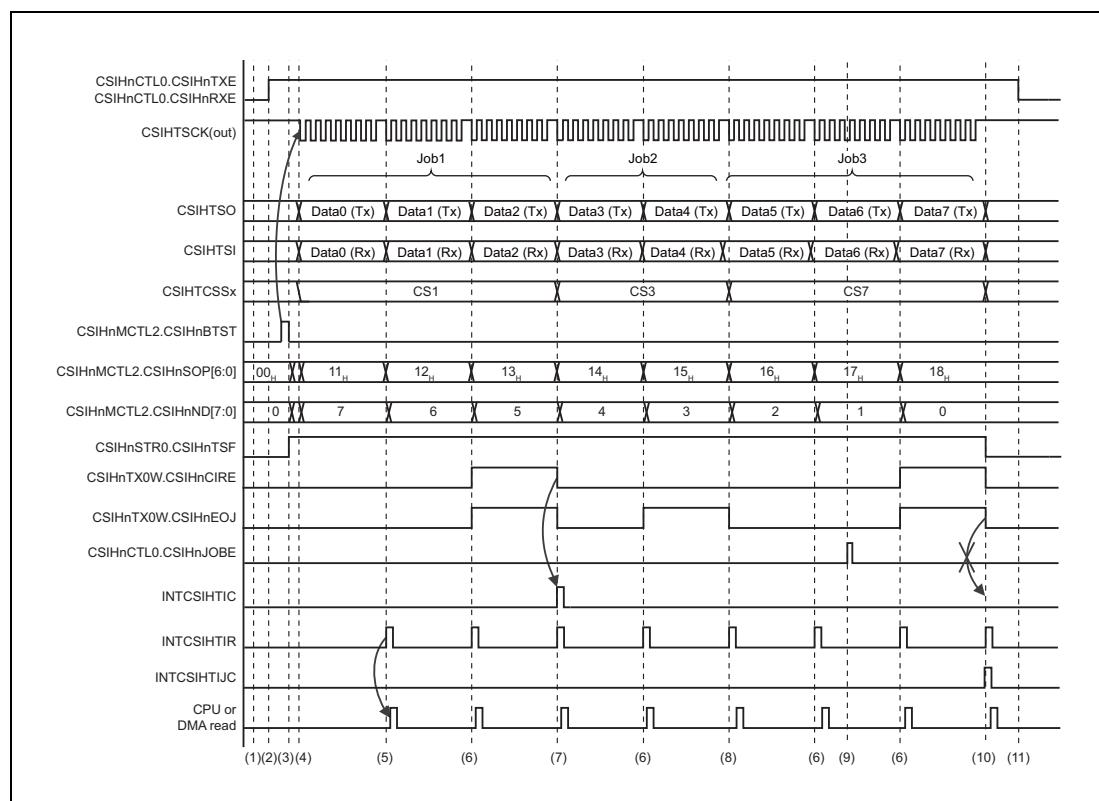


Figure 15.52 Master in Transmit-Only Buffer Mode,  $\text{CSIHnCTL1.CSIHnJE} = 1$

#### NOTE

The process of writing the data into the buffer is not described.

**Procedure:**

1. Configure the communication protocol in the CSIInCFGx register. This example uses chip select signals CS1, CS3, and CS7.  
Specify the transfer mode and job mode by setting the corresponding bits in registers CSIInCTL1 and CSIInCTL2.  
Configure the memory mode in CSIInMCTL0.CSIInMMS[1:0].  
Set CSIInMCTL0.CSIInMMS[1:0] = 10<sub>B</sub> (transmit-only buffer mode).
2. In the CSIInCTL0 register, set CSIInPWR = 1 (enables the clock), CSIInTXE = 1 (permits transmission), and CSIInRXE = 1 (permits reception). The CSIInCTL0.CSIInMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIInMCTL2.CSIInSOP[6:0] and CSIInMCTL2.CSIInND[7:0] bits. Start the buffer transfer by setting CSIInMCTL2.CSIInBTST.
4. Transmission is started. The CSIInMCTL2.CSIInSOP[6:0] bits are automatically incremented and the CSIInMCTL2.CSIInND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the receive register CSIInRX0W must be read.
6. The CSIInTX0W.CSIInEOJ = 1 setting indicates that the last data of the current job is sent.
7. The interrupt request INTCSIHTIC is generated. INTCSIHTIC indicates that the last data of the current job (CSIInTX0W.CSIInEOJ = 1) was sent with CSIInTX0W.CSIInCIRE = 1.
8. The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIInTX0W.CHABnEOJ = 1) was sent with CSIInTX0W.CSIInCIRE = 0.
9. By setting CSIInCTL0.CSIInJOBE = 1, communication is forced to stop at the end of JOB3.
10. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of job3.  
The INTCSIHTIJC interrupt request indicates that the communication was forcibly stopped at the end of the current job.  
The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in the CSIInTX0W register is not sent.
11. Finally, clear CSIInCTL0.CSIInTXE and CSIInCTL0.CSIInRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIInCTL0.CSIInPWR to 0 to minimize power consumption of the CSIIn.

### 15.6.3 Procedures in Dual Buffer Mode

Examples when job mode is enabled in master mode, disabled in master mode, and disabled in slave mode are provided below.

#### 15.6.3.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CHABnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is disabled ( $\text{CSIHnCTL1.CSIHnJE} = 0$ ).
- The number of data is 9 ( $\text{CSIHnMCTL2.CSIHnND[7:0]} = 09_H$ ).
- The transfer start address is  $10_H$  ( $\text{CSIHnMCTL2.CSIHnSOP[6:0]} = 10_H$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- Dual buffer mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS[1:0]} = 01_B$ )

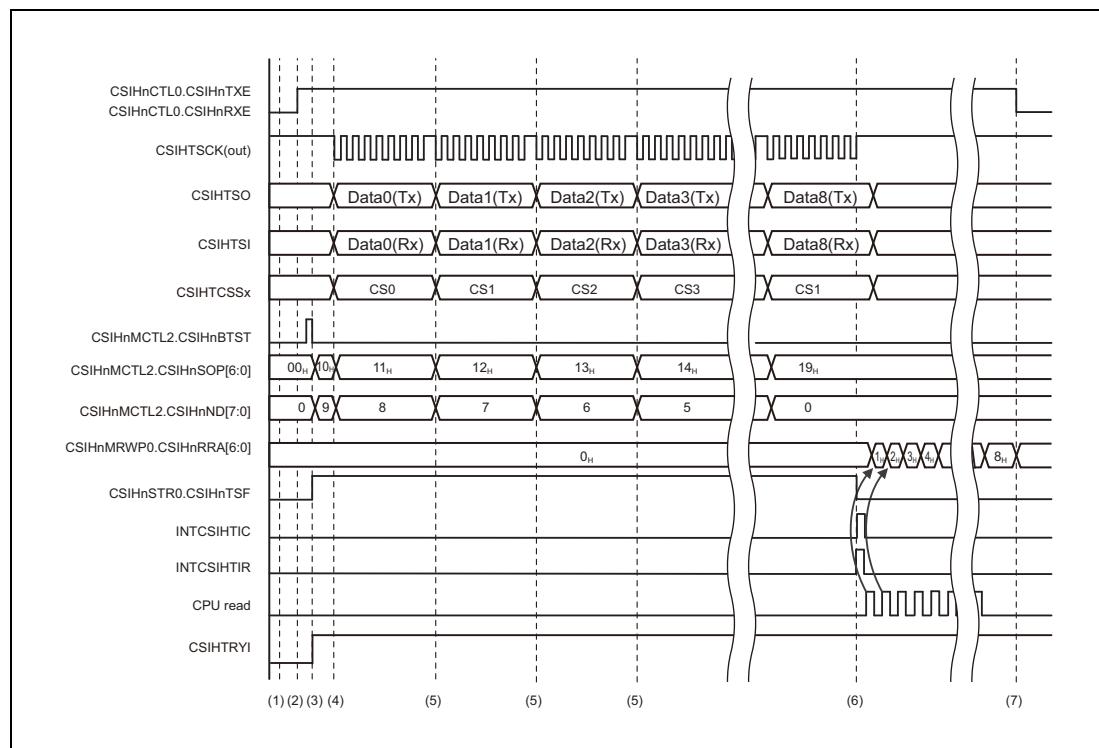


Figure 15.53 Master in Dual Buffer Mode,  $\text{CSIHnCTL1.CSIHnJE} = 0$

#### NOTE

The process of writing the data into the buffer is not described.

**Procedure:**

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.  
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.  
Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0]. Set CSIHnMCTL0.CSIHnMMS[1:0] = 01<sub>B</sub> (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Permit buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.  
The interrupt requests INTCSIHTIC and INTCSIHTIR are not generated.
6. When the last data is transmitted/received, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.  
The CPU starts to read the received data from the receive buffer. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after each data is read.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

### 15.6.3.2 Transmission/Reception in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is enabled ( $\text{CSIHnCTL1.CSIHnJE} = 1$ ).
- The number of data is 8 ( $\text{CSIHnMCTL2.CSIHnND[7:0]} = 08_H$ ).
- The transfer start address is  $00_H$  ( $\text{CSIHnMCTL2.CSIHnSOP[6:0]} = 00_H$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- Dual buffer mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS[1:0]} = 01_B$ )

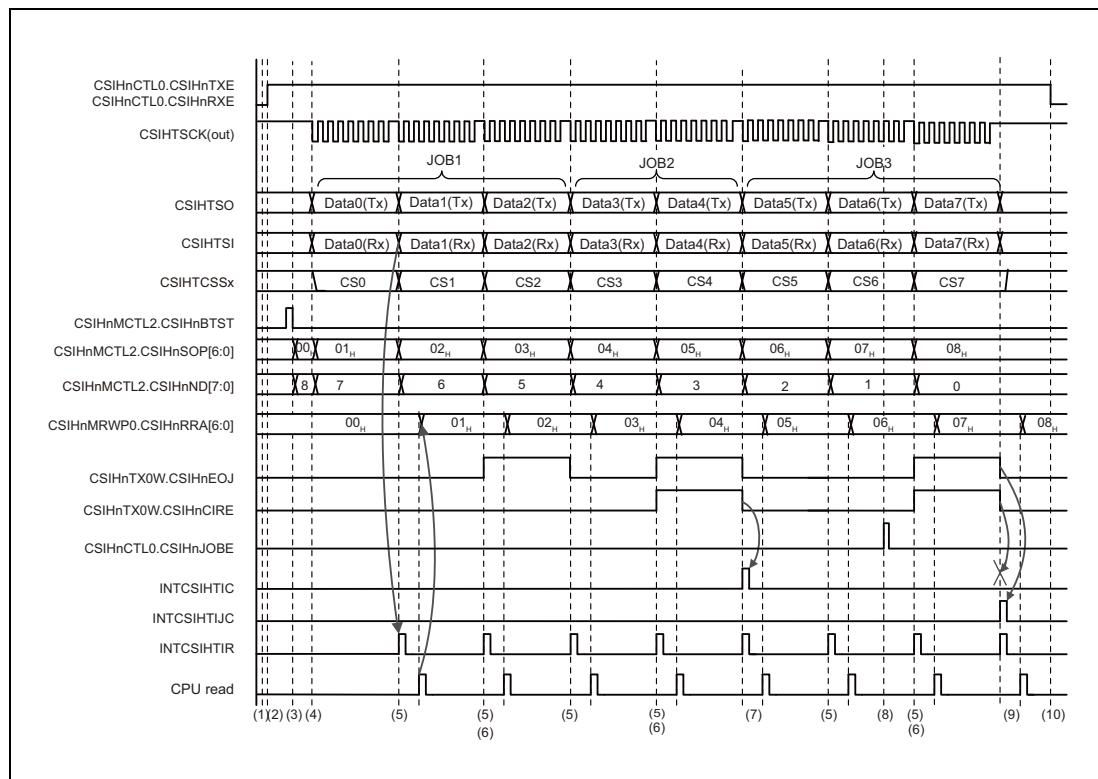


Figure 15.54 Master in Dual Buffer Mode,  $\text{CSIHnCTL1.CSIHnJE} = 1$

#### NOTE

The process of writing the data into the buffer is not described.

**Procedure:**

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS0 to CS7.  
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.  
Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].  
Set CSIHnMCTL0.CSIHnMMS[1:0] = 01<sub>B</sub> (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission. This is repeated until the last data is transmitted/received.
5. The INTCSIHTIR interrupt request is generated everytime data is received.  
The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
6. CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent.
7. The INTCSIHTIC interrupt request is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
9. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of JOB3.  
The INTCSIHTIJC interrupt request indicates that the communication was forcibly stopped at the end of the current job.  
The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in register CSIHnTX0W is not sent.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

### 15.6.3.3 Transmit/Receive in Slave Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCTL1.CSIHnCKR} = 0$ ,  $\text{CSIHnCFG0.CSIHnDAP0} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is disabled ( $\text{CSIHnCTL1.CSIHnJE} = 0$ ).
- The number of data is 9 ( $\text{CSIHnMCTL2.CSIHnND[7:0]} = 09_H$ ).
- The transfer start address is  $10_H$  ( $\text{CSIHnMCTL2.CSIHnSOP[6:0]} = 10_H$ ).
- Dual buffer mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS[1:0]} = 01_B$ )
- Handshake function is enabled ( $\text{CSIHnCTL1.CSIHnHSE} = 1$ )

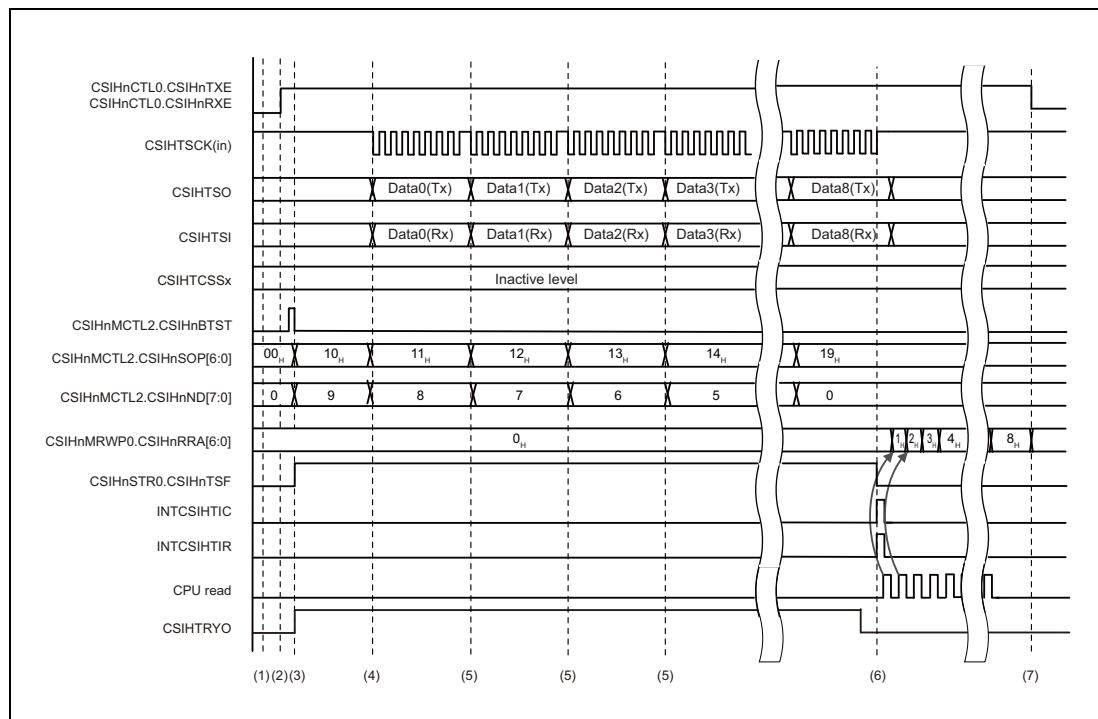


Figure 15.55 Slave in Dual Buffer Mode,  $\text{CSIHnCTL1.CSIHnJE} = 0$

#### NOTE

The process of writing the data into the buffer is not described.

**Procedure:**

1. Configure the communication protocol in register CSIHnCFG0.  
Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.  
Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].  
Set CSIHnMCTL0.CSIHnMMS[1:0] = 01<sub>B</sub> (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Specify the transfer start address by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and the number of data by setting the CSIHnMCTL2.CSIHnND[7:0] bits. Permit the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started when the input clock is received from the master. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.  
The interrupt requests INTCSIHTIC and INTCSIHTIR are not generated because transmission data is sent from the buffer, and received data is stored in the buffer.
6. When the last data is transmitted/received, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.  
The CPU starts to read the received data that is stored in the receive buffer. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after each data is read.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

## 15.6.4 Procedures in FIFO Mode

Two examples for a master is provided, one with job mode disabled, the other one with job mode enabled.

### 15.6.4.1 Transmission/Reception in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx}[3:0] = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ ).
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is disabled ( $\text{CSIHnCTL1.CSIHnJE} = 0$ ).
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- FIFO mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS}[1:0] = 00_B$ )

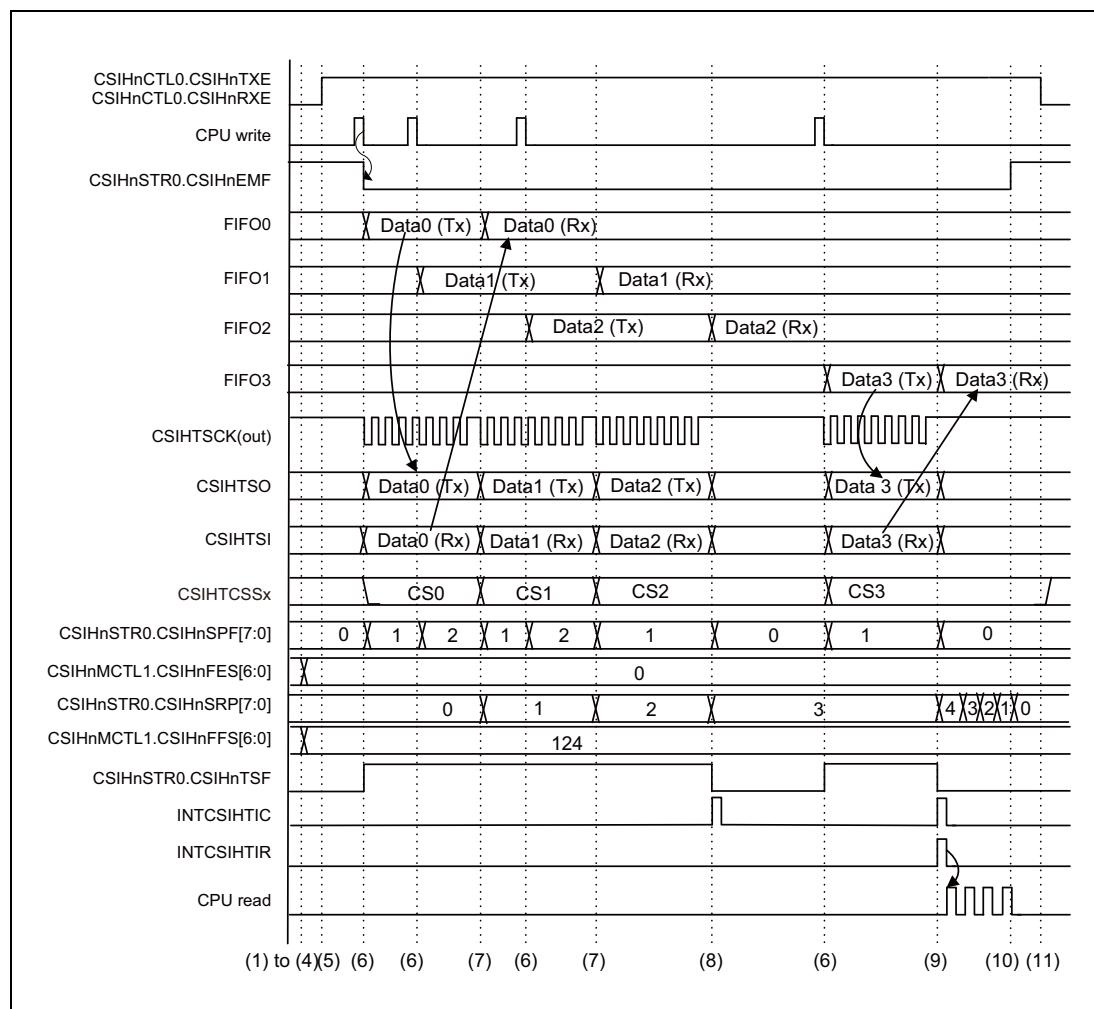


Figure 15.56 Master in FIFO Mode,  $\text{CSIHnCTL1.CSIHnJE} = 0$

**Procedure:**

1. Configure the communication protocol in the CSIhnCFGx register. Specify the job mode disable and master mode by setting the corresponding bits in registers CSIhnCTL1 and CSIhnCTL2. Specify the FIFO mode by setting CSIhnMCTL0.CSIhnMMS[1:0] = 00<sub>B</sub>. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
2. Set CSIhnSTCR0.CSIhnPCT = 1 to clear all buffer pointers.
3. Check that CSIhnSTR0.CSIhnFLF = 0, CSIhnSTR0.CSIhnEMF = 1, and CSIhnSTR0.CSIhnSPF[7:0] = 00<sub>H</sub>.
4. With CSIhnMCTL1.CSIhnFES[6:0], specify the conditions for generating the INTCSIHTIC interrupt output.  
Specify the conditions for generating the INTCSIHTIR interrupt request in the CSIhnMCTL1.CSIhnFFS[6:0] bits.
5. Set CSIhnCTL0.CSIhnPWR = 1 (enables the clock), CSIhnTXE = 1 (permits transmission), and CSIhnRXE = 1 (permits reception). The CSIhnCTL0.CSIhnMBS bit must be cleared.
6. Write the first transmit data to the transmit register CSIhnTX0W. Transmission starts automatically when the first data becomes available.  
Check that CSIhnSTR0.CSIhnEMF = 0.
7. The current transmission is completed. As the CSIhnFES[6:0] bits are not the same as the CSIhnSPF[7:0] bits, the interrupt request INTCSIHTIC is not generated.
8. As the CSIhnFES[6:0] bits are the same as the CSIhnSPF[7:0] bits, the interrupt request INTCSIHTIC is generated.
9. When CSIhnFFS[6:0] = 128 - CSIhnSRP[7:0], the interrupt request INTCSIHTIR is generated. Since CSIhnFES[6:0] = CSIhnSPF[7:0], the interrupt request INTCSIHTIC is generated.  
After the generation of an interrupt, the CPU starts reading received data that is stored in the receive buffer.
10. When the CPU completes reading the received data that is stored in the receive buffer, CSIhnSTR0.CSIhnEMF is set to 1 and the FIFO buffer becomes empty.
11. Finally, clear CSIhnCTL0.CSIhnTXE and CSIhnCTL0.CSIhnRXE to prohibit the transmit/receive operation. In addition, if communication is not performed, set CSIhnCTL0.CSIhnPWR = 0 to minimize the power consumption of CSIhn.

### 15.6.4.2 Transmit/Receive Mode when Job Mode is Enabled in Master Mode

The procedure below is based on the assumption that:

- The transmission data length is 8 bits ( $\text{CSIHnCFGx.CSIHnDLSx[3:0]} = 1000_B$ ).
- The transmission direction is MSB first ( $\text{CSIHnCFGx.CSIHnDIRx} = 0$ )
- Normal clock phase and data phase ( $\text{CSIHnCFGx.CSIHnCKPx} = 0$ ,  $\text{CSIHnCFGx.CSIHnDAPx} = 0$ )
- No interrupt delay ( $\text{CSIHnCTL1.CSIHnSIT} = 0$ )
- Job mode is enabled ( $\text{CSIHnCTL1.CSIHnJE} = 1$ ).
- JOB1 consists of four data, JOB2 consists of three data, and JOB3 consists of five data.
- Normal INTCSIHTIC interrupt timing ( $\text{CSIHnCTL1.CSIHnSLIT} = 0$ )
- FIFO mode ( $\text{CSIHnCTL0.CSIHnMBS} = 0$ ,  $\text{CSIHnMCTL0.CSIHnMMS[1:0]} = 00_B$ )

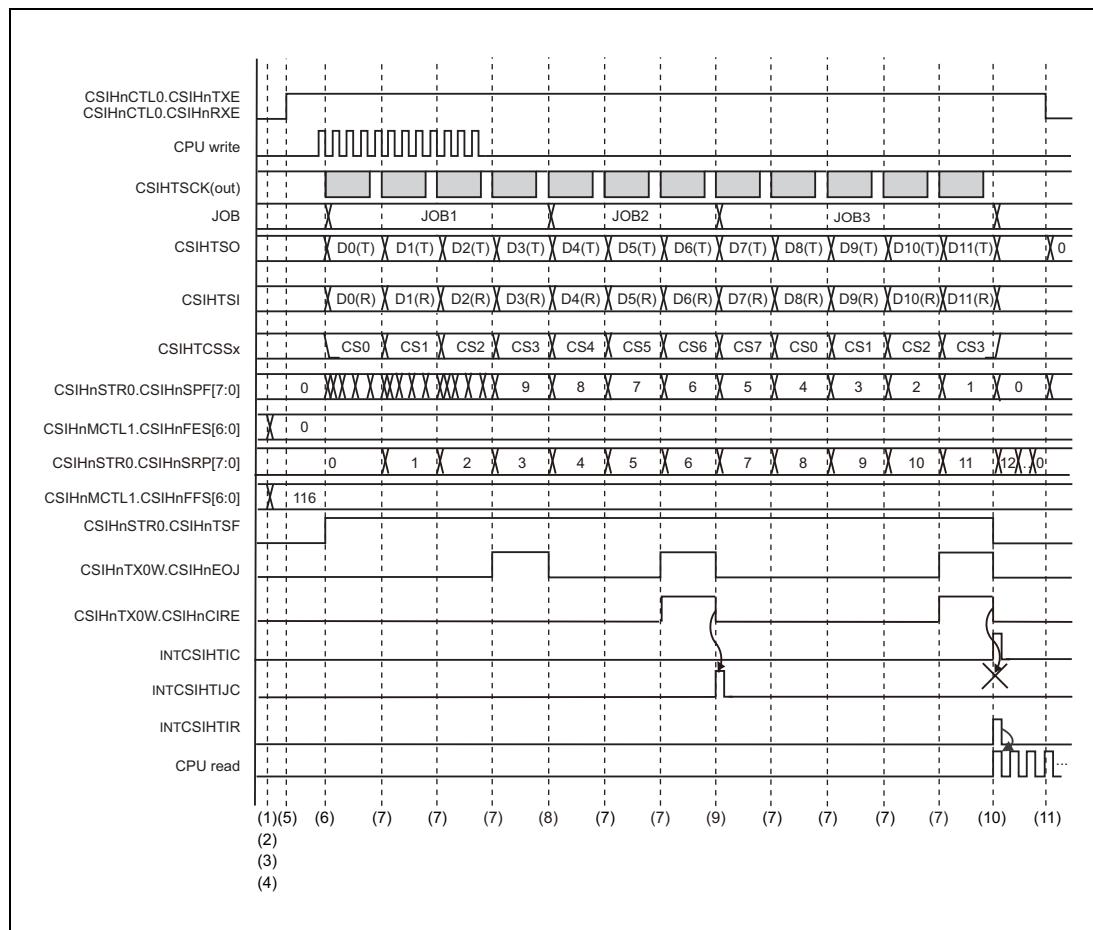


Figure 15.57 Master in FIFO Mode,  $\text{CSIHnCTL1.CSIHnJE} = 1$

**Procedure:**

1. Configure the communication protocol in the CSIHnCFGx register. Set job mode disable and master mode in the corresponding bits of the CSIHnCTL1 and CSIHnCTL2 registers. Set FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] to 00<sub>B</sub>. This example uses chip select signals CS0 to CS7.
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF is set to 0, CSIHnSTR0.CSIHnEMF is set to 1, and CSIHnSTR0.CSIHnSPF[7:0] is set to 00<sub>H</sub>.
4. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for generating the INTCSIHTIC interrupt request. With CSIHnMCTL1.CSIHnFFS[6:0], specify the conditions for generating the INTCSIHTIR interrupt request.
5. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the first data to be sent to the CSIHnTX0W transmission register. Transmission starts automatically when the first data becomes available.  
Make sure CSIHnSTR0.CSIHnEMF is set to 0.
7. The current transmission is completed. Since CSIHnFES[6:0] is not the same as CSIHnSPF[7:0], the interrupt request INTCSIHTIC is not generated.
8. The INTCSIHTIJC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. The INTCSIHTIJC interrupt request is generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
10. The INTCSIHTIC interrupt request is generated because CSIHnFES[6:0] = CSIHnSPF[7:0]. INTCSIHTIC is generated so that INTCSIHTIJC is not generated.  
When CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. After the generation of the INTCSIHTIR interrupt, CPU starts reading the received data stored in received buffer.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

## 15.7 Detection and Correction of Errors in CSI<sub>n</sub> RAM

### 15.7.1 ECC for the CSI<sub>n</sub> RAM

**Table 15.46** gives an outline of the ECC functions for the CSI<sub>n</sub> RAM.

**Table 15.46 List of the ECC Functions for the CSI<sub>n</sub> RAM**

Item	Outline of Functions
ECC error detection/correction	The RAM is checked for ECC errors. The following options are selectable. <ul style="list-style-type: none"> <li>• 2-bit error detection and 1-bit error detection/correction</li> <li>• 2-bit error detection and 1-bit error detection</li> </ul> The ECC error detection/correction can be disabled by using through mode. Error detection/correction is enabled by default.
Error notification	When an ECC 2-bit error is generated, an error notification is generated. <ul style="list-style-type: none"> <li>• Error notification can be enabled or disabled when an ECC 2-bit error is detected.</li> </ul> 2-bit error notification is enabled by default. However, if interrupts are masked by the FEINTFMSK register, interrupt processing is not executed.
Error status	Detection of ECC 2-bit errors and ECC1-bit errors can be monitored. A bit for clearing the error status is provided.

### 15.7.2 Interrupt Requests

**Table 15.47** lists the ECC interrupt requests of CSI<sub>n</sub> RAM.

**Table 15.47 CSI<sub>n</sub> ECC Interrupt Requests (FE-Level Maskable Interrupt)**

Unit interrupt signal	Description	Name	DMA Trigger Number
INTECCDCSIHn	CSI <sub>n</sub> ECC 2-bit error interrupt	INTECCDCSIH0	—
		INTECCDCSIH1	
		INTECCDCSIH2	
		INTECCDCSIH3	

### 15.7.3 ECCCSIHnCTL — CSIHn ECC Control Register

The ECCCSIHnCTL register controls the ECC mode and the status of CSIH.

Bits 7, 5, and 4 should be configured (written) while the CSIHn operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be  $01_B$ .

**Access:** This register can be read or written in 16-bit units.

**Address:**  $FFC7\ 00n0_H$  ( $n = 0$  to  $3$ )

**Value after reset:** Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDI	—	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W*	R/W*	R	R	R	R/W*	R/W*	R	R/W	R	R/W	R/W	R	R	R	R
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—

Note 1. These bits are always read as 0.

Table 15.48 ECCCSIHnCTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	EMCA1	Access Control to ECC Mode Selection Bit
14	EMCA0	These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The values written to these bits are not retained. When these bits are read, the read value is always 0. When these bits are $01_B$ , writing to bit 7 is enabled.
13 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECER2C	2-Bit ECC Error Detection Flag Clear This bit clears 2-bit error detection flags of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-Bit ECC Error Detection Correction Accumulation Flag Clear This bit clears 1-bit error detection/correction flags of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between writing to this bit and setting the ECER1F bit occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC Function through Mode Selection This bit specifies whether to enable or disable the ECC function. Setting this bit to 1 disables the ECC function. When writing to this bit, write $01_B$ to EMCA1 and EMCA0 at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled (ECC function disabled).
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit Error Correction Enable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.

**Table 15.48 ECCCSIHnCTL Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
4	EC2EDIC	<p>2-Bit Error Detection Interrupt Control</p> <p>This bit controls whether to generate an interrupt when 2-bit error is detected.</p> <p>0: When 2-bit error is detected, an INTECCDCSIHn interrupt will not be generated.</p> <p>1: When 2-bit error is detected, an INTECCDCSIHn interrupt will be generated. (the value after reset)</p>
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ECER2F	<p>2-Bit Error Detection Flag</p> <p>This flag indicates whether 2-bit error is detected during read access to the RAM when error detection is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDCSIHn) is output.</p> <p>Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is also cleared when through mode is enabled (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt will not be generated.</p> <p>0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred.</p> <p>This bit is read-only. Writing 0 or 1 does not change internal state.</p>
1	ECER1F	<p>1-Bit Error Detection/Correction Flag</p> <p>This flag indicates whether 1-bit error is detected during read access to the RAM when error detection is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is also cleared when through mode is enabled (ECTHM = 1).</p> <p>0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred.</p> <p>This bit is read-only. Writing 0 or 1 does not change internal state.</p>
0	ECEMF	<p>ECC Error Message Flag</p> <p>This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. Because the initial value of the RAM data is undefined, if the RAM is read before initialization, this bit may be set. This bit is also cleared when through mode is enabled (ECTHM = 1) and there is no 1-bit error in the decode circuit input data.</p> <p>0: The current RAM output data does not have bit errors. 1: The current RAM output data has bit errors.</p>

**CAUTION**

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

### 15.7.4 ECCCSIHnTMC — CSI<sub>n</sub> ECC Test Mode Control Register

The ECCCSIHnTMC register is used to switch to and control the test mode.

This register can be used when CSI<sub>n</sub> is not accessing the RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10<sub>B</sub>.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC7 00n4<sub>H</sub> (n = 0 to 3)

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 15.49 ECCCSIHnTMC Register Contents (1/2)

Bit Position	Bit Name	Function
15	ETMA1	Access Control to ECC Test Mode Bit
14	ETMA0	These bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The values written to these bits are not retained. When these bits are read, the read value is always 0. When these bits are 10 <sub>B</sub> , writing to bit 7 is enabled.
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Enable This bit specifies whether to enable access to the test registers and to the test control bits of this register. When writing to this bit, write 10 <sub>B</sub> to ETMA1 and ETMA0 at the same time. 0: Access to the test registers and test control bits is disabled. 1: Access to the test registers and test control bits is enabled.  Test registers: ECCCSIHnTED, ECCCSIHnTRC, ECCCSIHnSYND, ECCCSIHnHORD, ECCCSIHnECRD, ECCCSIHnERDB Test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Selection This bit selects the targets for reading when the ECCCSIHnTED and ECCCSIHnERDB registers are read. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Read value of the ECCCSIHnTED register will be the write value of the ECCCSIHnTED register. Read value of the ECCCSIHnERDB register will be the write value of the ECCCSIHnERDB register. 1: The read value of the ECCCSIHnTED register will be the RAM data. Read value of the ECCCSIHnERDB register will be the ECC Data to be written to RAM.
3	ECREOS	ECC Redundant Bit Output Data Selection This bit specifies whether to store the ECC data generated for write data or the value of the ECCCSIHnERDB register as ECC data in RAM. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data generated for write data is stored in RAM. 1: The value of ECCCSIHnERDB register is stored in RAM.

**Table 15.49 ECCCSIHnTMC Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
2	ECENS	<p>ECC Encoder Input Selection</p> <p>This bit specifies whether to use the data written to RAM or the value of the ECCCSIHnTED register as the input to the ECC encoder.</p> <p>Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously).</p> <p>This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: ECC data is generated from write data to RAM 1: ECC data is generated from register value of the ECCCSIHnTED.</p>
1	ECDCS	<p>ECC Decoder Input Selection</p> <p>This bit specifies whether to use the RAM data or the value of the ECCCSIHnTED register as the target data for generation of syndrome code and error detection. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Syndrome code generation and error detection are performed from RAM data. 1: Syndrome code generation and error detection are performed from ECCCSIHnTED register value.</p>
0	ECREIS	<p>ECC Redundant Bit Input Data Selection</p> <p>This bit specifies whether to use the ECC data stored in RAM or the value of the ECCCSIHnERDB register as the target ECC data for generation of syndrome code and error detection.</p> <p>Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Syndrome code generation and error detection are performed from ECC data stored in RAM. 1: Syndrome code generation and error detection are performed from ECCCSIHnERDB register value.</p>

### 15.7.5 ECCCSIHnTED — CSIhn ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

The value of this register is used to generate ECC data or syndrome code.

This register can be accessed when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1).

When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000<sub>H</sub> is read.

This register can be used when CSIhn is not accessing the RAM.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFC7 00nC<sub>H</sub> (n = 0 to 3)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECEDB[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ECEDB[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 15.50 ECCCSIHnTED Register Contents**

Bit Position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCCSIHnTMC.ECENS = 1, these bits generate ECC data from value of this register and store the register value in RAM. When ECCCSIHnTMC.ECDCS = 1, these bits generate syndrome code from the value of the register and store the register value in ECC decode syndrome data register (ECCCSIHnSYND). In addition, when ECCCSIHnTMC.ECTRRS = 1, RAM data [31:0] instead of written data is read as the value of this register.

### 15.7.6 ECCCSIHnTRC — CSIhN ECC Redundant Bit Data Control Test Register

This register is a test register for ECC data in ECC test mode and consists of four 8-bit registers, ECCCSIHnSYND, ECCCSIHnHORD, ECCCSIHnECRD, and ECCCSIHnERDB.

This register can be accessed when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1). When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000<sub>H</sub> is read.

This register can be used when CSIhN is not accessing the RAM.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFC7 00n8<sub>H</sub> (n = 0 to 3)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCCSIHnSYND (see <a href="#">Section 15.7.7</a> )								ECCCSIHnHORD (see <a href="#">Section 15.7.8</a> )							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCCSIHnECRD (see <a href="#">Section 15.7.9</a> )								ECCCSIHnERDB (see <a href="#">Section 15.7.10</a> )							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						

### 15.7.7 ECCCSIHnSYND — CSIhN ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrome data in ECC test mode.

Writing to this register is ignored.

This register can only be read when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1).

When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC7 00nB<sub>H</sub> (n = 0 to 3)

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.51 ECCCSIHnSYND Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	These bits store generated syndrome code as needed.

### 15.7.8 ECCCSIHnHORD — CSI<sub>n</sub> ECC 7-Bit Redundant Bit Data Hold Test Register

This register is for storing ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can only be read when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1).

When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC7 00nA<sub>H</sub> (n = 0 to 3)

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.52 ECCCSIHnHORD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	These bits store ECC code for read RAM data as needed. ECC code is also stored when ECCCSIHnTMC.ECTRRS = 1 and if ECCCSIHnTED register is read.

### 15.7.9 ECCCSIHnECRD — CSI<sub>n</sub> ECC Encode Test Register

This register is a read-only register for storing generated ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can only be read when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1). When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC7 00n9<sub>H</sub> (n = 0 to 3)

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 15.53 ECCCSIHnECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECRD[6:0]	These bits can read ECC data generated when data was written to the RAM. When ECCCSIHnTMC.ECENS = 1, these bits indicate the ECC data generated for the data written to the ECCCSIHnTED register.

### 15.7.10 ECCCSIHnERDB — CSI<sub>n</sub> ECC Redundant Bit Input/Output Replacement Buffer Register

In ECC test mode, this register handles ECC data.

The value of this register can be handled as the ECC data generated when data is written to the RAM or as the ECC data which is read when RAM data is read.

This register can be accessed when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1).

When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 00<sub>H</sub> is read.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFC7 00n8<sub>H</sub> (n = 0 to 3)

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W						
Value after reset	0	0	0	0	0	0	0	0

**Table 15.54 ECCCSIHnERDB Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	When ECCCSIHnTMC.ECREOS = 1, the value of this register is stored as the ECC data in the RAM. When ECCCSIHnTEC.ECREIS = 1, the value of this register takes the ECC data read from the RAM. When ECCCSIHnTMC.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instead of written data.

### 15.7.11 SELB\_READTEST — ECCREAD Test Select Register

SELB\_READTEST is used to check read/write access to the CSIhn ECC registers and RS-CANn ECC registers.

Setting the bit corresponding to each function to 1 will enable writing to the read-only bits.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFBC 0600<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	RTCANE7A0	—	—	—	RTCSIHE7A3	RTCSIHE7A2	RTCSIHE7A1	RTCSIHE7A0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

**Table 15.55 SELB\_READTEST Register Contents**

Bit Position	Bit Name	Function
7	RTCANE7A0	RS-CAN ECC <sup>*1</sup> Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RS-CANn ECC read-only bit can be written).
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RTCSIHE7A3	CSIH3 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH3 ECC read-only bit can be written).
2	RTCSIHE7A2	CSIH2 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH2 ECC read-only bit can be written).
1	RTCSIHE7A1	CSIH1 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH1 ECC read-only bit can be written).
0	RTCSIHE7A0	CSIH0 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH0 ECC read-only bit can be written).

Note 1. For details, see **Section 19.11, Detection and Correction of Errors in RS-CAN RAM**.

## Section 16 LIN Master Interface (RLIN2)

This section contains a generic description of the LIN master interface (RLIN2).

The first part in this section describes the features specific to RH850/F1L, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN2.

### 16.1 Features of RH850/F1L RLIN2

#### 16.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN2 units and channels.

**Table 16.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	2	4
Name	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0, 1)	RLIN24n (n = 0, 1) RLIN21n (n = 0, 1)

**Table 16.2 Configurations of RLIN2 Units and Correspondence between the Number of Channels per Unit and the Channel Numbers of the Units**

Unit Name RLIN24n/ RLIN21n	Channels per Unit	Unit Channel Number (i)	Channel Name RLIN2m	RH850/F1L 48 pins (2 ch)	RH850/F1L 64 pins (2 ch)	RH850/F1L 80 pins (2 ch)	RH850/F1L 100 pins (3 ch)	RH850/F1L 144 pins (6 ch)	RH850/F1L 176 pins (10 ch)
RLIN240	4	0	RLIN20	✓	✓	✓	✓	✓	✓
		1	RLIN21	✓	✓	✓	✓	✓	✓
		2	RLIN22				✓	✓	✓
		3	RLIN23					✓	✓
RLIN241	4	0	RLIN24					✓	✓
		1	RLIN25					✓	✓
		2	RLIN26						✓
		3	RLIN27						✓
RLIN210	1	0	RLIN28						✓
RLIN211	1	0	RLIN29						✓

**Table 16.3 Indices**

Index	Description
n	Throughout this section, the individual RLIN2 units are identified by the index "n" (n = 0, 1).
m	Throughout this section, the individual channels are identified by the index "m" (m = 0 to 9).
i	Throughout this section, the individual channels of units that compose RLIN2 are identified by the index "i" (i = 0 to 3).
b	Throughout this section, the individual data buffers implemented in RLIN2 are identified by the index "b" (b = 1 to 8).

For example, RLIN24nGLWBR / RLIN21nGLWBR are the LIN wake-up baud rate select registers, which are the global registers of RLIN2. RLIN24nmLiMD / RLIN21nmLiMD are the LIN mode registers, which are the channel registers.

The following lists the indices corresponding to each product.

**Table 16.4 Indices Correspondence of Each Product**

Indices Correspondence of Each Product			
48 pins, 64 pins, 80 pins	100 pins	144 pins	176 pins
i = 0, 1 (RLIN240)	i = 0 to 2 (RLIN240)	i = 0 to 3 (RLIN240) i = 0, 1 (RLIN241)	i = 0 to 3 (RLIN240, 1) i = 0 (RLIN210, 1)
b = 1 to 8	b = 1 to 8	b = 1 to 8	b = 1 to 8

## 16.1.2 Register Base Address

RLIN2 base addresses are listed in the following table.

RLIN2 register addresses are given as offsets from the base addresses.

**Table 16.5 Register Base Addresses**

Base Address Name	Base Address
<RLIN240_base>	FFCE 0000 <sub>H</sub>
<RLIN241_base>	FFCE 0080 <sub>H</sub>
<RLIN210_base>	FFCE 0100 <sub>H</sub>
<RLIN211_base>	FFCE 0120 <sub>H</sub>

## 16.1.3 Clock Supply

The RLIN2 clock supply is shown in the following table.

**Table 16.6 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
RLIN24n	LIN communication clock sources	CKSCLK_ILIN
	Register access	CKSCLK_ILIN
RLIN21n	LIN communication clock sources	CKSCLK_ILIN
	Register access	CKSCLK_ILIN

## 16.1.4 Interrupt Request

RLIN2 interrupt requests are listed in the following table.

**Table 16.7 Interrupt Requests (1/2)**

Unit Interrupt Signal	Outline	Interrupt Number
<b>RLIN240</b>		
INTRLIN0	RLIN20 interrupt	50
INTRLIN1	RLIN21 interrupt	51
INTRLIN2	RLIN22 interrupt	154
INTRLIN3	RLIN23 interrupt	155

**Table 16.7 Interrupt Requests (2/2)**

Unit	Interrupt Signal	Outline	Interrupt Number
<b>RLIN241</b>			
	INTRLIN4	RLIN24 interrupt	218
	INTRLIN5	RLIN25 interrupt	219
	INTRLIN6	RLIN26 interrupt	267
	INTRLIN7	RLIN27 interrupt	268
<b>RLIN210</b>			
	INTRLIN8	RLIN28 interrupt	277
<b>RLIN211</b>			
	INTRLIN9	RLIN29 interrupt	278

### 16.1.5 Reset Sources

RLIN2 reset sources are listed in the following table. RLIN2 is initialized by these reset sources.

**Table 16.8 Reset Sources**

Unit Name	Reset Source
RLIN24n	All reset sources (ISORES)
RLIN21n	All reset sources (ISORES)

### 16.1.6 External Input/Output Signals

External input/output signals of RLIN2 are listed in the following table.

**Table 16.9 External Input/Output Signals**

Unit	Signal Name	Outline	Alternative Port Pin Signal
<b>RLIN240</b>			
	RLIN2mRX (m = 0 to 3)	RLIN240 receive data input	RLIN2mRX (m = 0 to 3)
	RLIN2mTX (m = 0 to 3)	RLIN240 transmit data output	RLIN2mTX (m = 0 to 3)
<b>RLIN241</b>			
	RLIN2mRX (m = 4 to 7)	RLIN241 receive data input	RLIN2mRX (m = 4 to 7)
	RLIN2mTX (m = 4 to 7)	RLIN241 transmit data output	RLIN2mTX (m = 4 to 7)
<b>RLIN210</b>			
	RLIN2mRX (m = 8)	RLIN210 receive data input	RLIN2mRX (m = 8)
	RLIN2mTX (m = 8)	RLIN210 transmit data output	RLIN2mTX (m = 8)
<b>RLIN211</b>			
	RLIN2mRX (m = 9)	RLIN211 receive data input	RLIN2mRX (m = 9)
	RLIN2mTX (m = 9)	RLIN211 transmit data output	RLIN2mTX (m = 9)

#### CAUTION

When the P0\_0 pin is used as RLIN20RX, the P0\_0 pin outputs a low-level RESETOUT signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

## 16.2 Overview

### 16.2.1 Functional Overview

The LIN Master Interface is a hardware LIN communication controller that complies with LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602 (SEP 2005), and automatically performs frame communication and error determination.

**Table 16.10** shows the LIN Master Interface specifications.

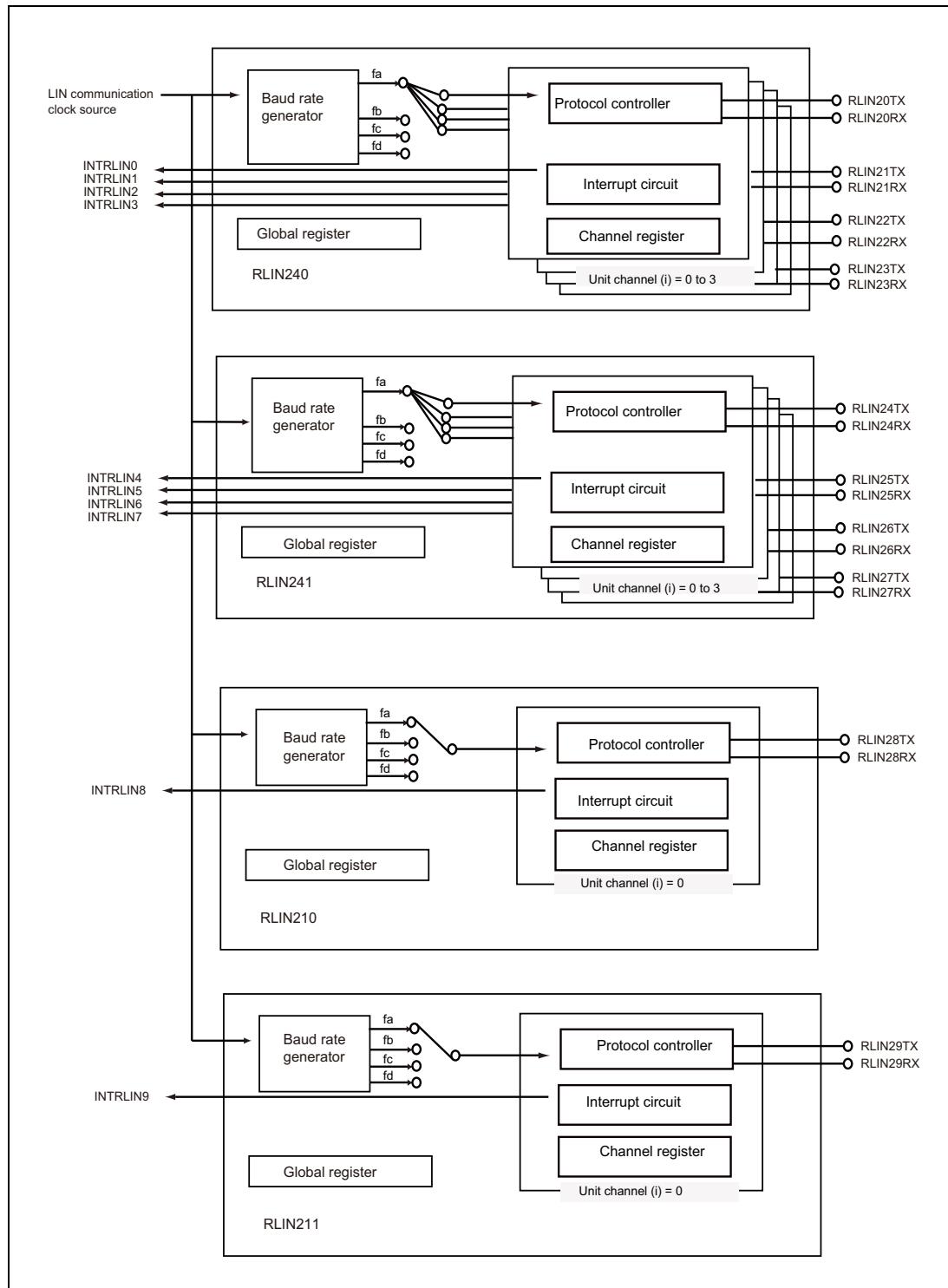
**Table 16.10 LIN Master Interface Specifications**

Item	Specifications	
Channel count	10 channels (In this product, 1-channel version and 4-channel version of RLIN2 is included.)	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602 (SEP 2005)
	Variable frame structure	<ul style="list-style-type: none"> <li>• Transmission break width: 13 to 28 Tbits</li> <li>• Transmission break delimiter width: 1 to 4 Tbits</li> <li>• Inter-byte space (header): 0 to 7 Tbits (space between Sync field and ID field)<sup>*1</sup></li> <li>• Response space: 0 to 7 Tbits<sup>*1</sup></li> <li>• Inter-byte space: 0 to 3 Tbits (space between data bytes in response area)</li> <li>• Transmit wake-up: 1 to 16 Tbits</li> </ul>
	Checksum	<ul style="list-style-type: none"> <li>• Automatic operation for both transmission and reception</li> <li>• Classic or enhanced selectable (for each frame)</li> </ul>
	Response field data byte count	Variable from 0 to 8 bytes
	Frame communication modes	<ul style="list-style-type: none"> <li>• Mode in which header transmission and response transmission/reception are started with a single transmission start request</li> <li>• Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)</li> </ul>
	Wake-up transmission and reception	<p>Available in LIN wake-up mode</p> <ul style="list-style-type: none"> <li>• Wake-up transmission (1 to 16 Tbits)</li> <li>• Wake-up reception</li> </ul> <p>Low-level width of input signals measured</p>
	Status	<ul style="list-style-type: none"> <li>• Successful frame/wake-up transmission</li> <li>• Successful header transmission</li> <li>• Successful frame/wake-up reception<sup>*2</sup></li> <li>• Successful data 1 reception</li> <li>• Error detection</li> <li>• Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)</li> </ul>
	Error status	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Checksum error</li> <li>• Frame timeout error</li> <li>• Physical bus error</li> <li>• Framing error</li> </ul>
	Baud rate selection	Baud rate conforming to the LIN specifications generated using baud rate generator
	Test mode	Self-test mode for user evaluation
	Interrupt function	<ul style="list-style-type: none"> <li>• Successful frame/wake-up transmission</li> <li>• Successful frame/wake-up reception<sup>*2</sup></li> <li>• Error detection</li> </ul>
The logical OR of these three events is the interrupt source (INTRLINm) for each channel.		

- Note 1. Since the same register is used for configuration, the inter-byte space (header) = response space.  
 Note 2. For wake-up reception, the input signal low-level width count is indicated.

### 16.2.2 Block Diagram

**Figure 16.1** shows a block diagram of the LIN master interface.



**Figure 16.1** LIN Master Interface Block Diagram (176 Pins, RLIN2 10 Channels Included)

## 16.3 Registers

The registers of the LIN master interface include global registers and channel registers. As the global registers are allocated for each unit, they can be individually set for each unit. As the channel registers are allocated for each channel, they can individually control each channel.

### 16.3.1 List of Registers

RLIN2 registers are listed in the following table.

For details about <RLIN24n\_base> and <RLIN21n\_base>, see **Section 16.1.2, Register Base Address**.

**Table 16.11 List of Registers (1/2)**

Module	Register	Symbol	Address
<b>Global registers</b>			
RLN24n	LIN wake-up baud rate select register	RLN24nGLWBR	<RLIN24n_base> + 01H
RLN21n		RLN21nGLWBR	<RLIN21n_base> + 01H
RLN24n	LIN baud rate prescaler 0 register	RLN24nGLRP0	<RLIN24n_base> + 02H
RLN21n		RLN21nGLRP0	<RLIN21n_base> + 02H
RLN24n	LIN baud rate prescaler 1 register	RLN24nGLRP1	<RLIN24n_base> + 03H
RLN21n		RLN21nGLRP1	<RLIN21n_base> + 03H
RLN24n	LIN self-test control register	RLN24nGLSTC	<RLIN24n_base> + 04H
RLN21n		RLN21nGLSTC	<RLIN21n_base> + 04H
<b>Channel registers</b>			
RLN24nm	LIN mode register	RLN24nmLiMD	<RLIN24n_base> + 08H + i × 20H
RLN21nm		RLN21nmLiMD	<RLIN21n_base> + 08H + i × 20H
RLN24nm	LIN break field configuration register	RLN24nmLiBFC	<RLIN24n_base> + 09H + i × 20H
RLN21nm		RLN21nmLiBFC	<RLIN21n_base> + 09H + i × 20H
RLN24nm	LIN space configuration register	RLN24nmLiSC	<RLIN24n_base> + 0AH + i × 20H
RLN21nm		RLN21nmLiSC	<RLIN21n_base> + 0AH + i × 20H
RLN24nm	LIN wake-up configuration register	RLN24nmLiWUP	<RLIN24n_base> + 0BH + i × 20H
RLN21nm		RLN21nmLiWUP	<RLIN21n_base> + 0BH + i × 20H
RLN24nm	LIN interrupt enable register	RLN24nmLiIE	<RLIN24n_base> + 0CH + i × 20H
RLN21nm		RLN21nmLiIE	<RLIN21n_base> + 0CH + i × 20H
RLN24nm	LIN error detection enable register	RLN24nmLiEDE	<RLIN24n_base> + 0DH + i × 20H
RLN21nm		RLN21nmLiEDE	<RLIN21n_base> + 0DH + i × 20H
RLN24nm	LIN control register	RLN24nmLiCUC	<RLIN24n_base> + 0EH + i × 20H
RLN21nm		RLN21nmLiCUC	<RLIN21n_base> + 0EH + i × 20H
RLN24nm	LIN transmission control register	RLN24nmLiTRC	<RLIN24n_base> + 10H + i × 20H
RLN21nm		RLN21nmLiTRC	<RLIN21n_base> + 10H + i × 20H
RLN24nm	LIN mode status register	RLN24nmLiMST	<RLIN24n_base> + 11H + i × 20H
RLN21nm		RLN21nmLiMST	<RLIN21n_base> + 11H + i × 20H
RLN24nm	LIN status register	RLN24nmLiST	<RLIN24n_base> + 12H + i × 20H
RLN21nm		RLN21nmLiST	<RLIN21n_base> + 12H + i × 20H
RLN24nm	LIN error status register	RLN24nmLiEST	<RLIN24n_base> + 13H + i × 20H
RLN21nm		RLN21nmLiEST	<RLIN21n_base> + 13H + i × 20H

Table 16.11 List of Registers (2/2)

Module	Register	Symbol	Address
RLN24nm	LIN data field configuration register	RLN24nmLiDFC	<RLIN24n_base> + 14 <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDFC	<RLIN21n_base> + 14 <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN ID buffer register	RLN24nmLiIDB	<RLIN24n_base> + 15 <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiIDB	<RLIN21n_base> + 15 <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN checksum buffer register	RLN24nmLiCBR	<RLIN24n_base> + 16 <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiCBR	<RLIN21n_base> + 16 <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 1 register	RLN24nmLiDBR1	<RLIN24n_base> + 18 <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR1	<RLIN21n_base> + 18 <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 2 register	RLN24nmLiDBR2	<RLIN24n_base> + 19 <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR2	<RLIN21n_base> + 19 <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 3 register	RLN24nmLiDBR3	<RLIN24n_base> + 1A <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR3	<RLIN21n_base> + 1A <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 4 register	RLN24nmLiDBR4	<RLIN24n_base> + 1B <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR4	<RLIN21n_base> + 1B <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 5 register	RLN24nmLiDBR5	<RLIN24n_base> + 1C <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR5	<RLIN21n_base> + 1C <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 6 register	RLN24nmLiDBR6	<RLIN24n_base> + 1D <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR6	<RLIN21n_base> + 1D <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 7 register	RLN24nmLiDBR7	<RLIN24n_base> + 1E <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR7	<RLIN21n_base> + 1E <sub>H</sub> + i × 20 <sub>H</sub>
RLN24nm	LIN data buffer 8 register	RLN24nmLiDBR8	<RLIN24n_base> + 1F <sub>H</sub> + i × 20 <sub>H</sub>
RLN21nm		RLN21nmLiDBR8	<RLIN21n_base> + 1F <sub>H</sub> + i × 20 <sub>H</sub>

**Note:** When writing to a register not used, write the value after reset.

### 16.3.2 Global Registers

#### 16.3.2.1 RLN24nGLWBR / RLN21nGLWBR — LIN Wake-Up Baud Rate Select Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nGLWBR: <RLIN24n\_base> + 01<sub>H</sub>  
RLN21nGLWBR: <RLIN21n\_base> + 01<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 16.12 RLN24nGLWBR / RLN21nGLWBR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified by the LCKS bit setting in the RLN24nmLiMD / RLN21nmLiMD registers is used (when LIN 1.3 is used). 1: In LIN wake-up mode, the clock fa is used regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD registers (when LIN 2.X is used).

Set the RLN24nGLWBR / RLN21nGLWBR register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0).

#### LWBR0 Bit (Wake-Up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR register to 0. This allows the 2.5-Tbit or longer low-level width of the input signal to be measured.

When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. With this setting, fa is selected as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register (the LCKS bit is not changed) and the 2.5-Tbit or longer low-level width of the input signal can be measured.

Setting the baud rate to 19200 bps while fa is selected allows the 130 µs or longer low-level width of the input signal to be detected during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register.

### 16.3.2.2 RLN24nGLBRP0 / RLN21nGLBRP0 — LIN Baud Rate Prescaler 0 Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nGLBRP0: <RLIN24n\_base> + 02H  
RLN21nGLBRP0: <RLIN21n\_base> + 02H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
LBRP0[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 16.13 RLN24nGLBRP0 / RLN21nGLBRP0 Register Contents**

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the LIN communication clock source by N+1. Setting range: 00H to FFH

Set the RLN24nGLBRP0 / RLN21nGLBRP0 register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0).

The value set in this register is used to control the frequency of baud rate clock source fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the LIN communication clock source by N+1.

### 16.3.2.3 RLN24nGLBRP1 / RLN21nGLBRP1 — LIN Baud Rate Prescaler 1 Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nGLBRP1: <RLIN24n\_base> + 03H  
RLN21nGLBRP1: <RLIN21n\_base> + 03H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
LBRP1[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 16.14 RLN24nGLBRP1 / RLN21nGLBRP1 Register Contents**

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the LIN communication clock source by M+1. Setting range: 00H to FFH

Set the RLN24nGLBRP1 / RLN21nGLBRP1 register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0).

The value set in this register is used to control the frequency of baud rate clock source  $f_d$ .

Assuming that the value set in this register is M, baud rate prescaler 1 divides the LIN communication clock source by M+1.

### 16.3.2.4 RLN24nGLSTC / RLN21nGLSTC — LIN Self-Test Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nGLSTC: <RLIN24n\_base> + 04<sub>H</sub>  
RLN21nGLSTC: <RLIN21n\_base> + 04<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.15 RLN24nGLSTC / RLN21nGLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing A7 <sub>H</sub> , 58 <sub>H</sub> , and 01 <sub>H</sub> successively to the RLN24nGLSTC / RLN21nGLSTC register places the module in LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN24nGLSTC / RLN21nGLSTC register cancels protection of LIN self-test mode.

Set the RLN24nGLSTC / RLN21nGLSTC register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0).

Writing A7<sub>H</sub>, 58<sub>H</sub>, and 01<sub>H</sub> successively to the RLN24nGLSTC / RLN21nGLSTC register places the module into LIN self-test mode.

When successive writing is completed and the module is placed in LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For details about transitioning to LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000000<sub>B</sub>”, and bit 7 returns an undefined value.

#### LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.

For details about exiting LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN24nGLSTC / RLN21nGLSTC register if it is not a part of successive writing of A7<sub>H</sub>, 58<sub>H</sub>, and 01<sub>H</sub>.

### 16.3.3 Channel Registers

#### 16.3.3.1 RLN24nmLiMD / RLN21nmLiMD — LIN Mode Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiMD: <RLIN24n\_base> + 08<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiMD: <RLIN21n\_base> + 08<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
—	—	—	—	LCKS[1:0]	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

R/W	0	0	R	R	R/W	0	R/W	R
-----	---	---	---	---	-----	---	-----	---

Table 16.16 RLN24nmLiMD / RLN21nmLiMD Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN24nmLiMD / RLN21nmLiMD register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (LIN reset mode).

#### LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.

With 00<sub>B</sub> set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).

With 01<sub>B</sub> set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).

With 10<sub>B</sub> set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).

With 11<sub>B</sub> set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR is 1 (when LIN 2.x is used) and the RLN24nmLiMST / RLN21nmLiMST register is 01<sub>H</sub> (LIN wake-up mode), regardless of the setting of the LCKS bit, fa is input to the protocol controller (LCKS bit is not changed).

### 16.3.3.2 RLN24nmLiBFC / RLN21nmLiBFC — LIN Break Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiBFC: <RLIN24n\_base> + 09<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiBFC: <RLIN21n\_base> + 09<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.17 RLN24nmLiBFC / RLN21nmLiBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits ... 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN24nmLiBFC / RLN21nmLiBFC register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

#### BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header.  
1 Tbit to 4 Tbits can be specified.

#### BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header.  
13 Tbits to 28 Tbits can be specified.

### 16.3.3.3 RLN24nmLiSC / RLN21nmLiSC — LIN Space Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiSC: <RLIN24n\_base> + 0A<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiSC: <RLIN21n\_base> + 0A<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]	—	IBHS[2:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 16.18 RLN24nmLiSC / RLN21nmLiSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN24nmLiSC / RLN21nmLiSC register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

#### IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.

0 Tbit to 3 Tbits can be specified.

These bits are enabled only during response transmission; they are disabled during response reception.

#### IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.

0 Tbit to 7 Tbits can be specified.

The response space setting is enabled only during response transmission; the setting is disabled during response reception.

The inter-byte space (header) value is equal to the response space value.

### 16.3.3.4 RLN24nmLiWUP / RLN21nmLiWUP — LIN Wake-Up Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiWUP: <RLIN24n\_base> + 0B<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiWUP: <RLIN21n\_base> + 0B<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
				WUTL[3:0]	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 16.19 RLN24nmLiWUP / RLN21nmLiWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN24nmLiWUP / RLN21nmLiWUP register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (LIN reset mode).

#### WUTL[3:0] Bits (Wake-Up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.

1 Tbit to 16 Tbits can be specified.

When the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR is 1 (when LIN 2.x is used), regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register, f<sub>A</sub> is selected as the LIN system clock (f<sub>LIN</sub>) (LCKS bit is not changed).

### 16.3.3.5 RLN24nmLiIE / RLN21nmLiIE — LIN Interrupt Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiIE: <RLIN24n\_base> + 0C<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiIE: <RLIN21n\_base> + 0C<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 16.20 RLN24nmLiIE / RLN21nmLiIE Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN24nmLiIE / RLN21nmLiIE register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (LIN reset mode).

#### ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables interrupt request upon detection of an error.

With 0 set, the interrupt request is not generated when the ERR flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

With 1 set, the interrupt request is generated when the ERR flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

Errors that constitute interrupt sources are bit errors, physical bus errors, frame timeout errors, framing errors, and checksum errors.

Detection of a bit error, physical bus error, frame timeout error, and framing error can be enabled or disabled using the RLN24nmLiEDE / RLN21nmLiEDE register.

#### FRCIE Bit (Successful Frame/Wake-Up Reception Interrupt Request Enable)

The FRCIE bit enables or disables interrupt request upon successful reception of a frame or a wake-up signal (counting of low level width of the input signal).

With 0 set, the interrupt request is not generated when the FRC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

With 1 set, the interrupt request is generated when the FRC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

### FTCIE Bit (Successful Frame/Wake-Up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables interrupt request upon successful transmission of a frame or a wake-up signal.

With 0 set, the interrupt request is not generated when the FTC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

With 1 set, the interrupt request is generated when the FTC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

### 16.3.3.6 RLN24nmLiEDE / RLN21nmLiEDE — LIN Error Detection Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiEDE: <RLIN24n\_base> + 0D<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiEDE: <RLIN21n\_base> + 0D<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0

R/W	0	R	R	R	R/W	R/W	R/W	R/W
-----	---	---	---	---	-----	-----	-----	-----

Table 16.21 RLN24nmLiEDE / RLN21nmLiEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Frame Timeout Error Detection Enable 0: Disables frame timeout error detection. 1: Enables frame timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN24nmLiEDE / RLN21nmLiEDE register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0<sub>B</sub> (LIN reset mode).

### FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is indicated in the FER flag in the RLN24nmLiEST / RLN21nmLiEST register.

For details on the framing error, see **Section 16.14, Error Status**.

**FTERE Bit (Frame Timeout Error Detection Enable)**

The FTERE bit enables or disables detection of the frame timeout error.

With 0 set, the frame timeout error is not detected.

With 1 set, the frame timeout error is detected.

When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN24nmLiEST / RLN21nmLiEST register.

For details on the frame timeout error, see **Section 16.14, Error Status**.

**PBERE Bit (Physical Bus Error Detection Enable)**

The PBERE bit enables or disables detection of the physical bus error.

With 0 set, the physical bus error is not detected.

With 1 set, the physical bus error is detected.

When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN24nmLiEST / RLN21nmLiEST register.

For details on the physical bus error, see **Section 16.14, Error Status**.

**BERE Bit (Bit Error Detection Enable)**

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is indicated in the BER flag in the RLN24nmLiEST / RLN21nmLiEST register.

For details on the bit error, see **Section 16.14, Error Status**.

### 16.3.3.7 RLN24nmLiCUC / RLN21nmLiCUC — LIN Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiCUC: <RLIN24n\_base> + 0E<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiCUC: <RLIN21n\_base> + 0E<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 16.22 RLN24nmLiCUC / RLN21nmLiCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: Transition to LIN reset mode. 1: Exit LIN reset mode.

Set the RLN24nmLiCUC / RLN21nmLiCUC register to 01<sub>H</sub> to transition to LIN wake-up mode or to 03<sub>H</sub> to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN24nmLiCUC / RLN21nmLiCUC register to 03<sub>H</sub> after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN24nmLiMST / RLN21nmLiMST register before writing another value.

#### OM1 Bit (LIN Mode Select)

The OM1 bit selects the operating mode (LIN wake-up mode or LIN operation mode) that is entered after exiting LIN reset mode.

Setting this bit to 0 selects LIN wake-up mode.

Setting this bit to 1 selects LIN operation mode.

This bit is enabled only when the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 1. Writing a value to this bit is disabled while the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 1.

#### OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.

Setting this bit to 0 causes RLIN2 to enter LIN reset mode.

Setting this bit to 1 causes RLIN2 to exit LIN reset mode.

### 16.3.3.8 RLN24nmLiTRC / RLN21nmLiTRC — LIN Transmission Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiTRC: <RLIN24n\_base> + 10<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiTRC: <RLIN21n\_base> + 10<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 16.23 RLN24nmLiTRC / RLN21nmLiTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission Start 0: Response transmission is stopped in frame separate mode. 1: Response transmission is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission/Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

#### RTS Bit (Response Transmission Start)

Set the RTS bit (response transmit start bit) to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame transmission and transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02<sub>H</sub> to the RLN24nmLiTRC / RLN21nmLiTRC register using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit is 0 (LIN reset mode). When the OMM1 bit is 0 (Lin wake-up mode), do not write 1.

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is stopped).

#### FTS Bit (Frame Transmission/Wake-Up Transmission/Reception Start)

Set the FTS bit to 1 to start frame/wake-up transmission.

Also set this bit to 1 to allow wake-up reception (counting of the low level width of the input signal).

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit is 0 (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

### 16.3.3.9 RLN24nmLiMST / RLN21nmLiMST — LIN Mode Status Register

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** RLN24nmLiMST: <RLIN24n\_base> + 11<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiMST: <RLIN21n\_base> + 11<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.24 RLN24nmLiMST / RLN21nmLiMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

#### OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

When the OMM0 bit is 0<sub>B</sub> (LIN reset mode), the value of this bit is disabled.

#### OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

### 16.3.3.10 RLN24nmLiST / RLN21nmLiST — LIN Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiST: <RLIN24n\_base> + 12<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiST: <RLIN21n\_base> + 12<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 16.25 RLN24nmLiST / RLN21nmLiST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN24nmLiST / RLN21nmLiST register is automatically cleared to 00<sub>H</sub> upon transition to LIN reset mode and start of the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00<sub>H</sub>.

Writing to this register is prohibited while the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 1 (frame transmission or wake-up transmission/reception is started)

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value from before 1 was written.

“1” is set upon completion of header transmission, but an interrupt request is not generated.

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

### D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value from before 1 is written.

“1” is set upon completion of Data 1 reception, but an interrupt request is not generated.

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

### ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when the value of any of the flags of the RLN24nmLiEST / RLN21nmLiEST registers is 1). Here, an interrupt request is generated if the ERRIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST / RLN21nmLiEST register while in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

### FRC Flag (Successful Frame/Wake-Up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value from before 1 is written.

The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request is generated if the FRCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt is enabled).

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

### FTC Flag (Successful Frame/Wake-Up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value from before 1 is written.

The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request is generated if the FTCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt is enabled).

To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

### 16.3.3.11 RLN24nmLiEST / RLN21nmLiEST — LIN Error Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiEST: <RLIN24n\_base> + 13<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiEST: <RLIN21n\_base> + 13<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 16.26 RLN24nmLiEST / RLN21nmLiEST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Frame Timeout Error Flag 0: Frame timeout error has not been detected. 1: Frame timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN24nmLiEST / RLN21nmLiEST register is automatically cleared to 00<sub>H</sub> upon transition to LIN reset mode and start of the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00<sub>H</sub>.

When the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value from before 1 is written.

The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

### **FER Flag (Framing Error Flag)**

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value from before 1 is written.

The FER flag is set to 1 upon framing error detection when the FERE bit of the RLN24nmLiEST / RLN21nmLiEST register is 1 (framing error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

### **FTER Flag (Frame Timeout Error Flag)**

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value from before 1 is written.

The FTER flag is set to 1 upon frame timeout error detection when the FTERE bit of the RLN24nmLiEDE / RLN21nmLiEDE register is 1 (frame timeout error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

### **PBER Flag (Physical Bus Error Flag)**

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value from before 1 is written.

The PBER flag is set to 1 upon physical bus error detection when the PBERE bit of the RLN24nmLiEDE / RLN21nmLiEDE register is 1 (physical bus error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

### **BER Flag (Bit Error Flag)**

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value from before 1 is written.

The BER flag is set to 1 upon bit error detection when the BERE bit of the RLN24nmLiEDE / RLN21nmLiEDE register is 1 (bit error detection enabled). To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

### 16.3.3.12 RLN24nmLiDFC / RLN21nmLiDFC — LIN Data Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiDFC: <RLIN24n\_base> + 14<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiDFC: <RLIN21n\_base> + 14<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	FSM	CSM	RFT			RFDL[3:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.27 RLN24nmLiDFC / RLN21nmLiDFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 bytes (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

Set the RLN24nmLiDFC / RLN21nmLiDFC register when the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

#### FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response transmission mode.

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 1), response is transmitted/received without the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register being set.

With 1 set, frame separate mode is selected. When the RTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is set to 1 during header transmission, response transmission is executed after header transmission has ended.

For response reception (the RFT bit is 0), set the FSM bit to 0.

When transitioning to LIN self-test mode, set this bit to 0 before transition.

For details on frame separate mode, see **Section 16.11.1, Transmission of LIN Frames**.

**CSM Bit (Checksum Select)**

The CSM bit sets checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When frame timeout error detection is enabled (the FTERE bit in the RLN24nmLiEDE / RLN21nmLiEDE register is 1), the timeout time depends on the setting of this bit. For details, see [Section 16.14, Error Status](#).

**RFT Bit (Response Field Communication Direction Select)**

The RFT bit sets the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low level width of the input signal is counted).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

**RFDL[3:0] Bits (Response Field Length Select)**

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

### 16.3.3.13 RLN24nmLiIDB / RLN21nmLiIDB — LIN ID Buffer Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiIDB: <RLIN24n\_base> + 15<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiIDB: <RLIN21n\_base> + 15<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** Undefined

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0				ID[5:0]		
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 16.28 RLN24nmLiIDB / RLN21nmLiIDB Register Contents**

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bit (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bit (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN24nmLiIDB / RLN21nmLiIDB register when the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted prior to communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.

#### IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 sets P0 and IDP1 sets P1.

Since parity is not automatically calculated, set the calculation value. Note that even if the specified calculation result is incorrect, it is transmitted as is.

#### ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

### 16.3.3.14 RLN24nmLiCBR / RLN21nmLiCBR — LIN Checksum Buffer Register

**Access:** This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

**Address:** RLN24nmLiCBR: <RLIN24n\_base> + 16<sub>H</sub> + i × 20<sub>H</sub>  
RLN21nmLiCBR: <RLIN21n\_base> + 16<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** Undefined

Bit	7	6	5	4	3	2	1	0
CKSM[7:0]								
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.29 RLN24nmLiCBR / RLN21nmLiCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value transmitted or received.

while in LIN operation mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register is 1 (transmission):  
The value transmitted can be read from the register. Read the value after transmission is completed.  
Writing to this register is invalid.
- When the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register is 0 (reception):  
The value received can be read from the register. Read the value after reception is completed.  
Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register is 1 (transmission):  
After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.  
Writing to this register is invalid.
- When the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register is 0 (reception):  
Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.

Set the RLN24nmLiCBR / RLN21nmLiCBR register when the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

### 16.3.3.15 RLN24nmLiDBRb / RLN21nmLiDBRb — LIN Data Buffer b Register

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN24nmLiDBR1: <RLIN24n\_base> + 18<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR1: <RLIN21n\_base> + 18<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR2: <RLIN24n\_base> + 19<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR2: <RLIN21n\_base> + 19<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR3: <RLIN24n\_base> + 1A<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR3: <RLIN21n\_base> + 1A<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR4: <RLIN24n\_base> + 1B<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR4: <RLIN21n\_base> + 1B<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR5: <RLIN24n\_base> + 1C<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR5: <RLIN21n\_base> + 1C<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR6: <RLIN24n\_base> + 1D<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR6: <RLIN21n\_base> + 1D<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR7: <RLIN24n\_base> + 1E<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR7: <RLIN21n\_base> + 1E<sub>H</sub> + i × 20<sub>H</sub>  
 RLN24nmLiDBR8: <RLIN24n\_base> + 1F<sub>H</sub> + i × 20<sub>H</sub>, RLN21nmLiDBR8: <RLIN21n\_base> + 1F<sub>H</sub> + i × 20<sub>H</sub>

**Value after reset:** Undefined

Bit	7	6	5	4	3	2	1	0
LDB[7:0]								
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 16.30 RLN24nmLiDBRb / RLN21nmLiDBRb Register Contents**

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or allows the received data to be read. Setting range: 00 <sub>H</sub> to FF <sub>H</sub>

- For response transmission:

The LiDBRb registers specify the data to be transmitted in the response field.

Configure these registers when:

- The RFT bit in RLN24nmLiDFC / RLN21nmLiDFC register is 1 (transmission).
- The FSM bit in RLN24nmLiDFC / RLN21nmLiDFC register is 0 (not frame separate mode).
- The FTS bit in RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

or

- The RFT bit in RLN24nmLiDFC / RLN21nmLiDFC register is 1 (transmission).
- The FSM bit in RLN24nmLiDFC / RLN21nmLiDFC register is 1 (frame separate mode).
- The RTS bit in RLN24nmLiTRC / RLN21nmLiTRC register is 0 (response transmission is stopped).

- For response reception:

The LiDBRb registers hold the data received in the response field.

The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.

Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted prior to communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.

## 16.4 Interrupt Sources

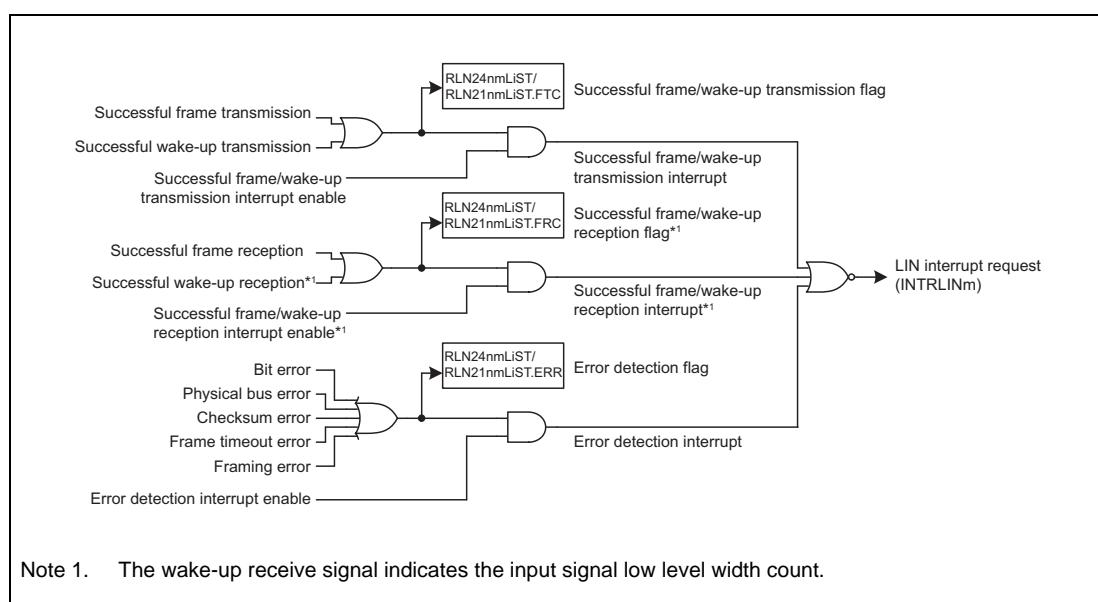
The LIN interrupts are interrupt requests generated by the LIN master interface.

There are three interrupt sources for each channel; successful frame/wake-up transmission, successful frame/wakeup reception, and error detection.

Interrupt requests from these three sources are ORed to generate one interrupt request “LIN interrupt”.

The respective interrupt request is output when the corresponding flag in the RLN24nmLiST / RLN21nmLiST register is set to 1 while the corresponding bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt enabled). However, if an interrupt is requested when the corresponding flag in the RLN24nmLiST / RLN21nmLiST register has been set to 1, it is ignored. Therefore, clear the corresponding flag to 0 to enable the interrupt again.

**Figure 16.2** shows a block diagram of the LIN interrupt.



**Figure 16.2 LIN Interrupt Block Diagram**

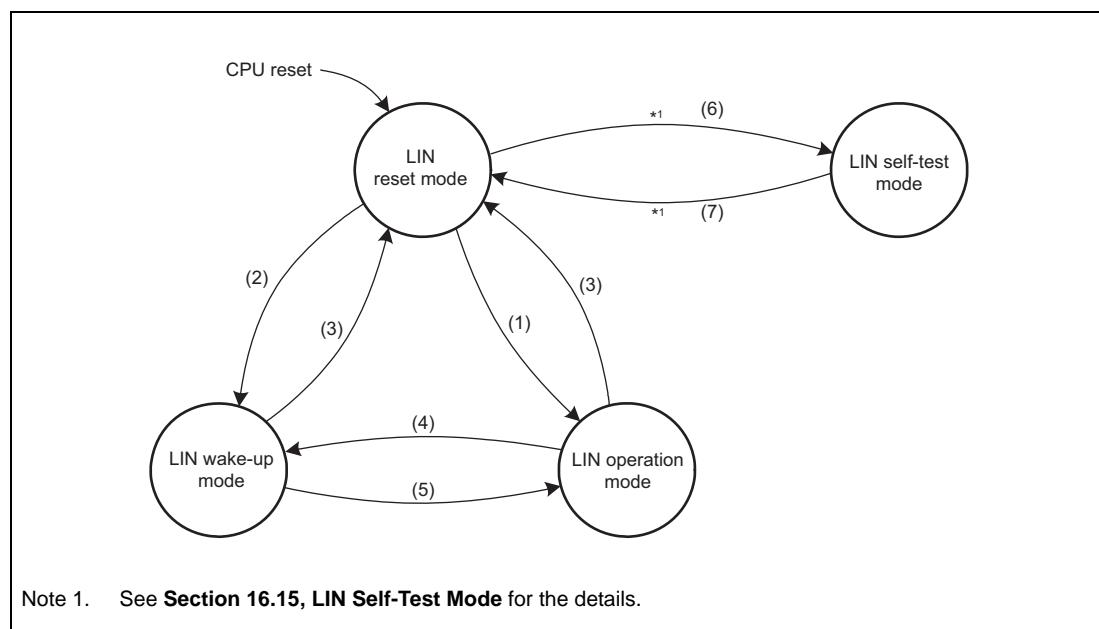
## 16.5 Modes

The LIN master interface provides the following four modes:

- LIN reset mode
- LIN operation mode
- LIN wake-up mode
- LIN self-test mode

The mode transitions except LIN self-test mode is controlled independently for respective channels.

**Figure 16.3** shows mode transitions. **Table 16.31** describes mode transition conditions. **Table 16.32** lists operations available in each mode.



**Figure 16.3 Mode Transitions**

**Table 16.31 Transition Condition of Each Mode**

Mode Transition		Transition Condition
(1)	LIN reset mode	→ LIN operation mode RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 11 <sub>B</sub>
(2)	LIN reset mode	→ LIN wake-up mode RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 01 <sub>B</sub>
(3)	LIN wake-up mode LIN operation mode	→ LIN reset mode RLN24nmLiCUC / RLN21nmLiCUC.OM0 = 0 <sub>B</sub>
(4)	LIN operation mode	→ LIN wake-up mode RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 01 <sub>B</sub>
(5)	LIN wake-up mode	→ LIN operation mode RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 11 <sub>B</sub>
(6)	LIN reset mode	→ LIN self-test mode See <b>Section 16.15, LIN Self-Test Mode.</b>
(7)	LIN self-test mode	→ LIN reset mode See <b>Section 16.15, LIN Self-Test Mode.</b>

**Table 16.32 Operations Available in Each Mode**

LIN Operation Mode	LIN Wake-Up Mode	LIN Self-Test Mode
Header transmission	Wake-up transmission	self-test
Response transmission	Wake-up reception	
Response reception	Error detection	
Error detection		

Whether a transition has been made to LIN reset mode, LIN operation mode, or LIN wake-up mode can be verified by reading the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register.

For a description of LIN self-test mode, see **Section 16.15, LIN Self-Test Mode.**

## 16.6 LIN Reset Mode

Setting the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The transition to LIN reset mode can be verified by determining that the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication stops.

From LIN reset mode, transitions to LIN operation mode, LIN wake-up mode, and LIN self-test mode can be made.

When the mode transitions to LIN reset mode, the following registers are initialized to their reset values and retain their initial values while in LIN reset mode:

- RLN24nmLiTRC / RLN21nmLiTRC register
- RLN24nmLiST / RLN21nmLiST register
- RLN24nmLiEST / RLN21nmLiEST register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN24nGLWBR / RLN21nGLWBR register
- RLN24nGLRP0 / RLN21nGLRP0 register
- RLN24nGLRP1 / RLN21nGLRP1 register
- RLN24nmLiMD / RLN21nmLiMD register
- RLN24nmLiBFC / RLN21nmLiBFC register
- RLN24nmLiSC / RLN21nmLiSC register
- RLN24nmLiWUP / RLN21nmLiWUP register
- RLN24nmLiIE / RLN21nmLiIE register
- RLN24nmLiEDE / RLN21nmLiEDE register
- RLN24nmLiDFC / RLN21nmLiDFC register
- RLN24nmLiIDB / RLN21nmLiIDB register
- RLN24nmLiCBR / RLN21nmLiCBR register
- RLN24nmLiDBRb / RLN21nmLiDBRb register

## 16.7 LIN Operation Mode

while in LIN operation mode, frame processing (header transmission, response transmission, response reception, and error detection) is performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register to  $11_B$  changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register to  $11_B$ . Communication settings should be performed after the RLN24nmLiMST / RLN21nmLiMST register has become  $11_B$ .

## 16.8 LIN Wake-Up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) is performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register to  $01_B$  changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register to  $01_B$ . Communication settings should be performed after the RLN24nmLiMST / RLN21nmLiMST register has become  $01_B$ .

## 16.9 Header Transmission/Response Transmission/Response Reception

### 16.9.1 Header Transmission

**Figure 16.4** shows the operation of the LIN master interface in header transmission. **Table 16.33** shows processing in header transmission.

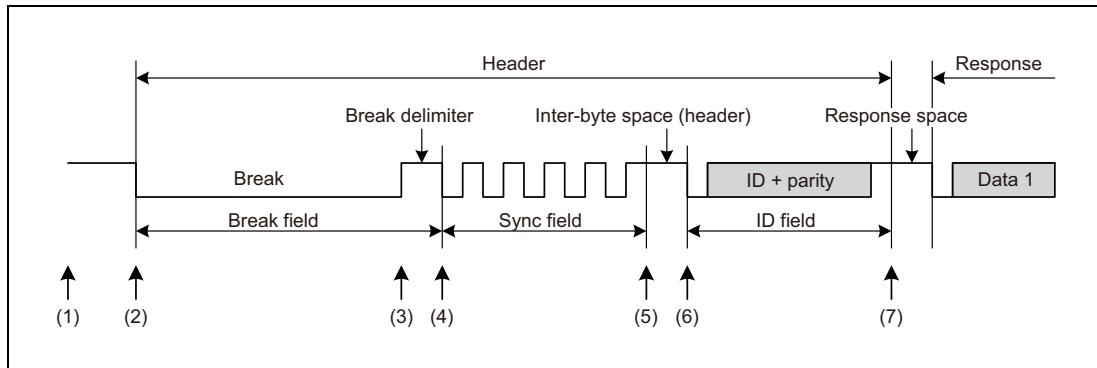


Figure 16.4 Operation in Header Transmission

Table 16.33 Processing in Header Transmission

Software Processing	LIN Master Interface Processing
(1) <ul style="list-style-type: none"> <li>• Sets a baud rate</li> <li>• Enables interrupts</li> <li>• Enables error detection</li> <li>• Sets frame configuration parameters</li> <li>• Transitions to LIN operation mode</li> <li>• Sets information on the frame to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data)</li> </ul>	Waits for the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to be set by software (idle).
(2) Sets the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3) Waits for an interrupt request	Transmits a break delimiter.
(4)	Transmits a sync field (55H).
(5)	Transmits an inter-byte space (header).
(6)	Transmits an ID field.
(7)	Sets the successful header transmission flag.

#### NOTE

For information about error detection, see **Section 16.14, Error Status**.

### 16.9.2 Response Transmission

**Figure 16.5** shows the operation of the LIN master interface in response transmission. **Table 16.34** shows the processing in response transmission.

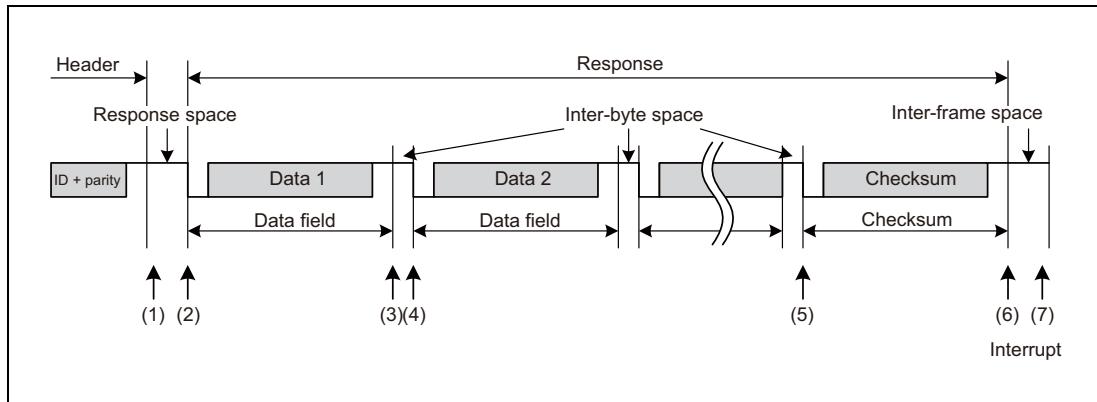


Figure 16.5 Operation in Response Transmission

Table 16.34 Processing in Response Transmission

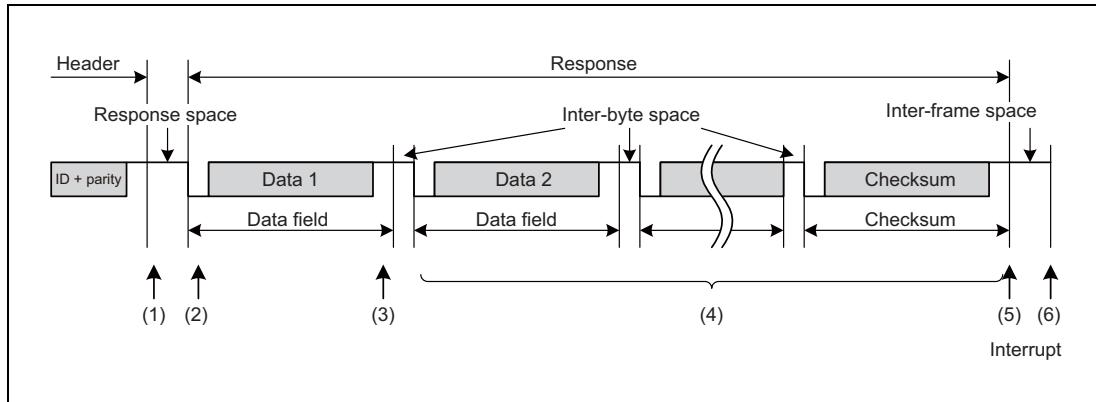
Software Processing	LIN Master Interface Processing
(1) (When in frame separate mode) <ul style="list-style-type: none"> <li>Sets the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (response transmission started)</li> </ul> (When not in frame separate mode) <ul style="list-style-type: none"> <li>Waits for an interrupt request</li> </ul>	(When in frame separate mode) <ul style="list-style-type: none"> <li>Waits for the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to be set to 1 by software. (During this time, "1" is output.)</li> <li>When the bit is set to 1, sends a response space.</li> </ul> (When not in frame separate mode) <ul style="list-style-type: none"> <li>Sends a response space.</li> </ul>
(2) Waits for an interrupt request	Transmits data 1.
(3)	Transmits an inter-byte space.
(4)	<ul style="list-style-type: none"> <li>Transmits data 2.</li> <li>Transmits an inter-byte space</li> <li>Transmits data 3.</li> <li>Transmits an inter-byte space</li> </ul>
	(Repeats the transmission of data and inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC / RLN21nmLiDFC register.) <ul style="list-style-type: none"> <li>:</li> <li>:</li> </ul>
(5)	Transmits the checksum.
(6)	<ul style="list-style-type: none"> <li>Sets the successful frame/wake-up transmission flag.</li> <li>Sets the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped)</li> </ul> (When in frame separate mode) <ul style="list-style-type: none"> <li>Sets the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 0 (response transmission stopped).</li> </ul>
(7) • Processing after communication Checks the RLN24nmLiST / RLN21nmLiST register, and clears flags.	Idle

**NOTE**

For information about error detection, see **Section 16.14, Error Status.**

### 16.9.3 Response Reception

**Figure 16.6** shows the operation of the LIN master interface in response reception. **Table 16.35** shows the processing in response reception.



**Figure 16.6 Operation in Response Reception**

**Table 16.35 Processing in Response Reception**

Software Processing	LIN Master Interface Processing
(1) Waits for an interrupt request (no processing).	Waits for detection of a start bit.
(2) Waits for an interrupt request.	Receives data 1 when the start bit is detected.
(3)	Sets the successful data 1 reception flag.
(4)	<ul style="list-style-type: none"> <li>• Receives data 2 when the start bit is detected.</li> <li>• Receives data 3 when the start bit is detected.</li> <li>(Repeats the reception of data as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC / RLN21nmLiDFC register.)</li> </ul>
(5)	<ul style="list-style-type: none"> <li>• Receives the checksum when the start bit is detected.</li> <li>• Evaluates the checksum.</li> <li>• Sets the successful frame/wake-up reception flag.</li> <li>• Sets the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped).</li> </ul>
(6)	<ul style="list-style-type: none"> <li>• Processing after communication</li> <li>Reads the received data.</li> <li>Checks the RLN24nmLiST / RLN21nmLiST register, and clears flags.</li> </ul>

**NOTE**

For information about error detection, see **Section 16.14, Error Status.**

## 16.10 Data Transmission/Reception

### 16.10.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag of the RLN24nmLiEST / RLN21nmLiEST register (see **Section 16.14, Error Status**).

In the LIN master interface, because data is generated every 1 Tbit = 16fLIN, the sampling point for received data is the 13th clock cycle (81.25% position).

**Figure 16.7** shows an example of data transmission timing.

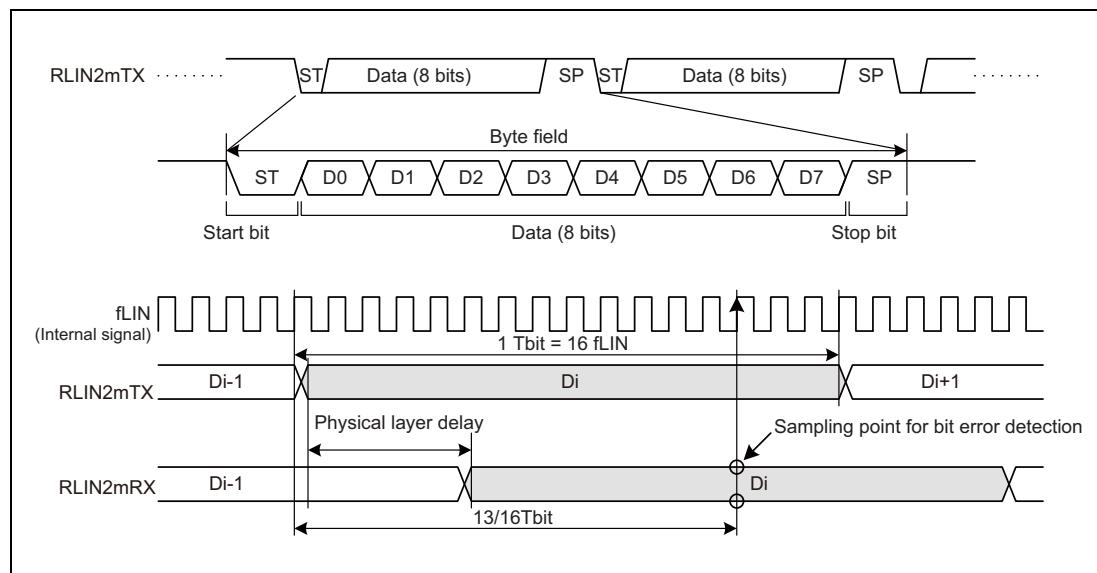


Figure 16.7 Example of Data Transmission Timing

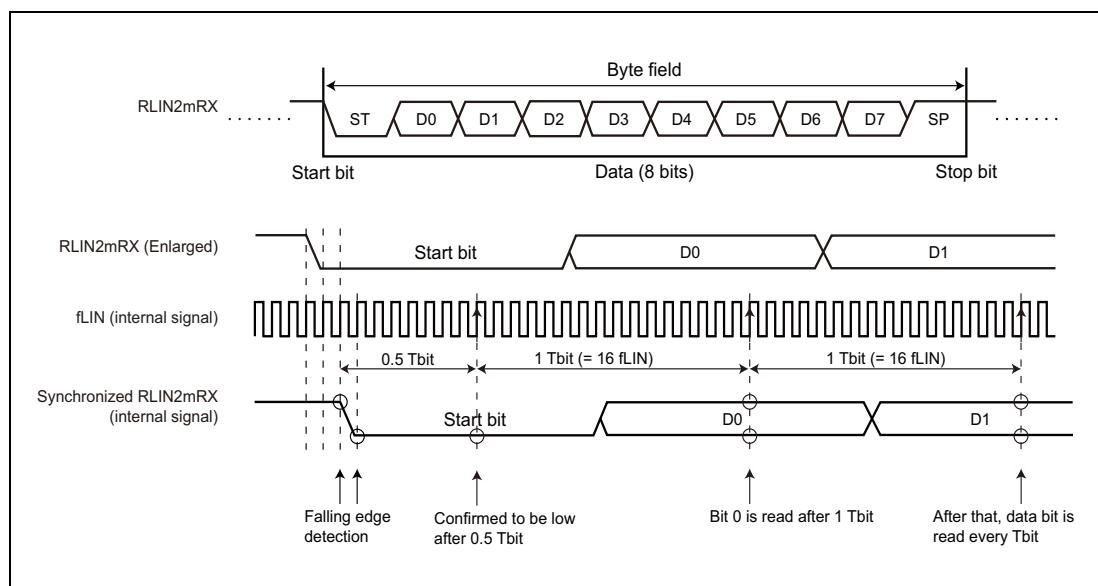
### 16.10.2 Data Reception

Data reception is performed by using the synchronized RLIN2mRX signal (an internal signal) that is the input from the RLIN2mRX pin synchronized with the LIN system clock (fLIN).

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN2mRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN2mRX signal is low level. The falling edge is not recognized as a start bit if the RLIN2mRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

The bit sampling period after detection of the start bit is one Tbit.

**Figure 16.8** shows an example of data reception timing.



**Figure 16.8 Example of Data Reception Timing**

## 16.11 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN master interface sends or receives data continuously.

### 16.11.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8 / RLN21nmLiDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-byte transmission, the contents stored in registers RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR4 / RLN21nmLiDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN24nmLiDBR5 / RLN21nmLiDBR5 to RLN24nmLiDBR8 / RLN21nmLiDBR8 are not transmitted. The transmitted checksum data is stored in the RLN24nmLiCBR / RLN21nmLiCBR register.

**Figure 16.9** shows the LIN transmission processing and the corresponding buffers.

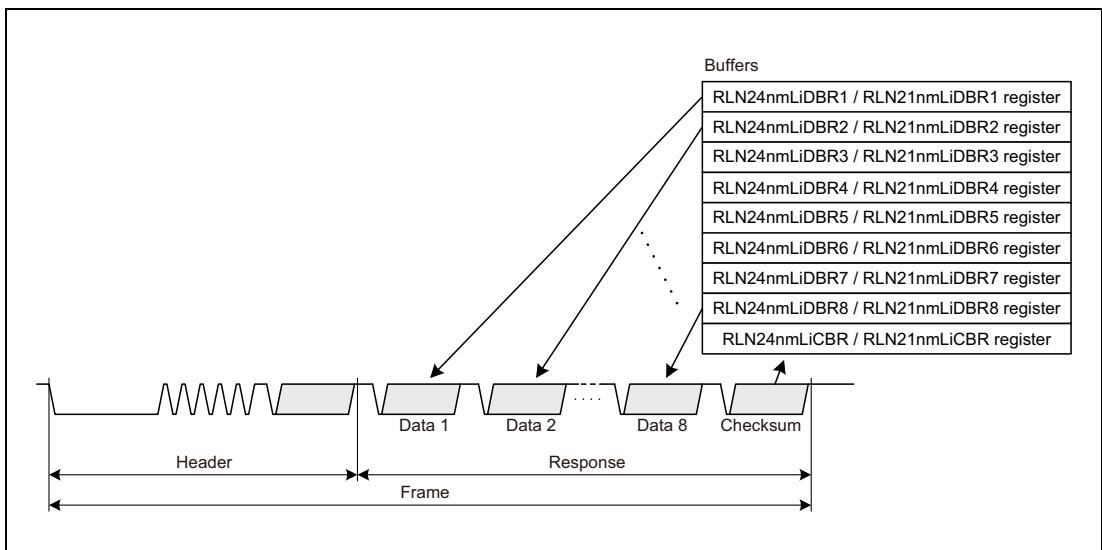


Figure 16.9 LIN Transmission Processing and Corresponding Buffers

#### (1) Frame Separate Mode

Setting the FSM bit in the RLN24nmLiDFC / RLN21nmLiDFC register to 1 sets the frame separate mode.

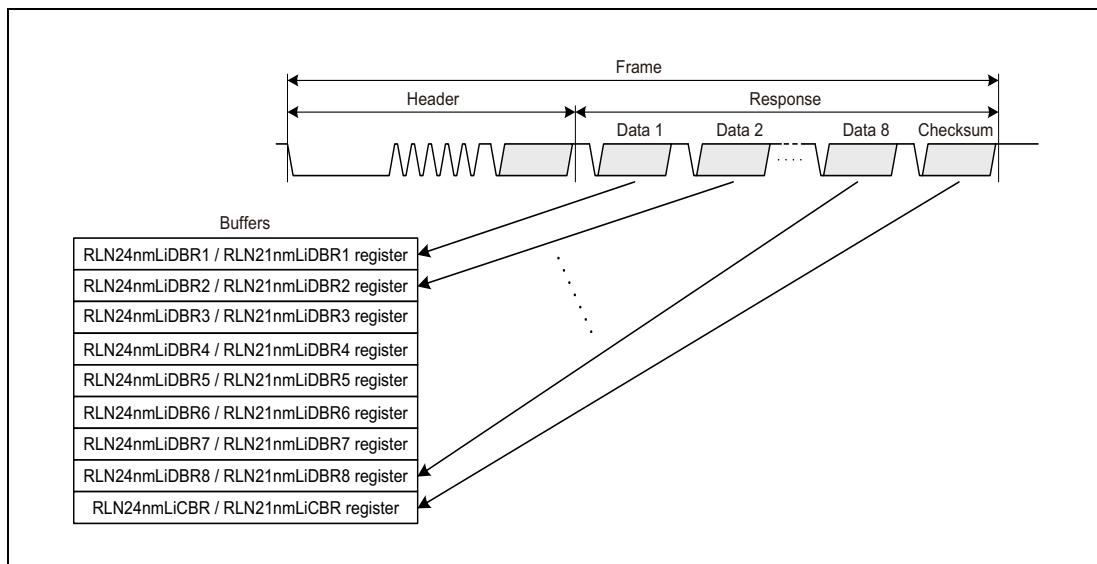
In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1 (successful header transmission).

### 16.11.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8 / RLN21nmLiDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR4 / RLN21nmLiDBR4, respectively; no data is stored in registers RLN24nmLiDBR5 / RLN21nmLiDBR5 to RLN24nmLiDBR8 / RLN21nmLiDBR8. The received checksum data is stored in the RLN24nmLiCBR / RLN21nmLiCBR register.

**Figure 16.10** shows the LIN reception processing and the corresponding buffers.



**Figure 16.10 LIN Reception Processing and Corresponding Buffers**

#### (1) Reception of Data 1

When the reception of the first byte of data is finished, the D1RC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1 (successful data 1 reception).

## 16.12 Wake-Up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

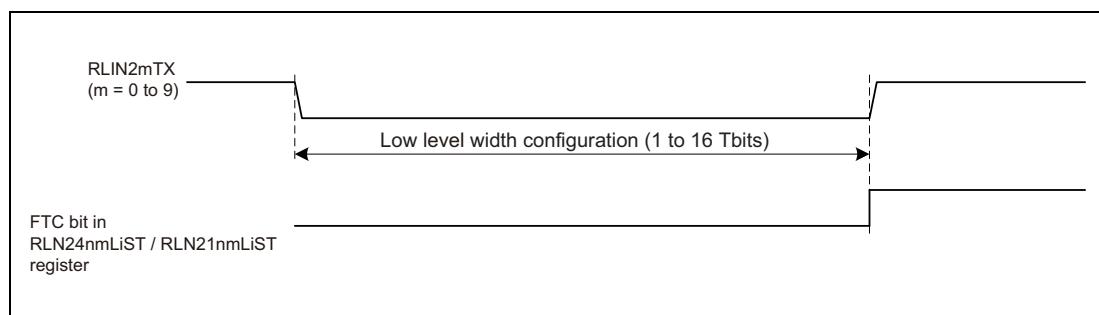
### 16.12.1 Wake-Up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register to 1 (transmission) and the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low level width of the wake-up signal is set using the WUTL[3:0] bits in the RLN24nmLiWUP / RLN21nmLiWUP register. However, if the value of the LWBR0 bit of the RLN24nGLWBR / RLN21nGLWBR register is 1 (when LIN 2.x is used), the LIN system clock (fLIN) has the low-level width of  $f_a$  (JW) regardless of the setting of the LCKS bit of the RLN24nmLiMD / RLN21nmLiMD register. By setting the baud rate to 19200 bps when  $f_a$  is selected and setting the WUTL[3:0] bits of the RLN24nmLiWUP / RLN21nmLiWUP register to 0100<sub>B</sub> (5 Tbits), 260  $\mu$ s low level width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN24nmLiMD / RLN21nmLiMD register.

If a wake-up low level is output without any error, the FTC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1 (successful frame or wake-up transmission); when the FTCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (successful frame/wake-up transmission interrupt enabled), an interrupt request is generated.

If an error is detected, wake-up transmission is aborted and the error flag for the error detected (the PBER flag or BER flag in the RLN24nmLiEST / RLN21nmLiEST register) is set to 1 (physical bus error detection / bit error detection).

**Figure 16.11** shows the wake-up transmission timing.



**Figure 16.11** Wake-Up Transmission Timing

### 16.12.2 Wake-Up Reception

The detection of a wake-up signal involves the use of an input signal low level width count function.

The input signal low level width count function measures the low level width of the input signal to the RLIN2mRX pin, using the same sampling point as data reception. This allows an input signal of fLIN with a low-level width of 2.5-Tbit or longer to be measured.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR register to 0. When LIN Specification Package Revision 2.x is used, set LWBR0 bit to 1.

When LWBR0 bit is set to 1, regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register, fa is selected as the LIN system clock (fLIN) (the LCKS bit is not changed).

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 µs or longer to be detected during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register.

When using this function, in LIN wake-up mode set the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register to 0 (reception), and the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1 (successful frame or wake-up reception) and if the FRCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (successful frame or wake-up reception interrupt enabled), an interrupt request is generated.

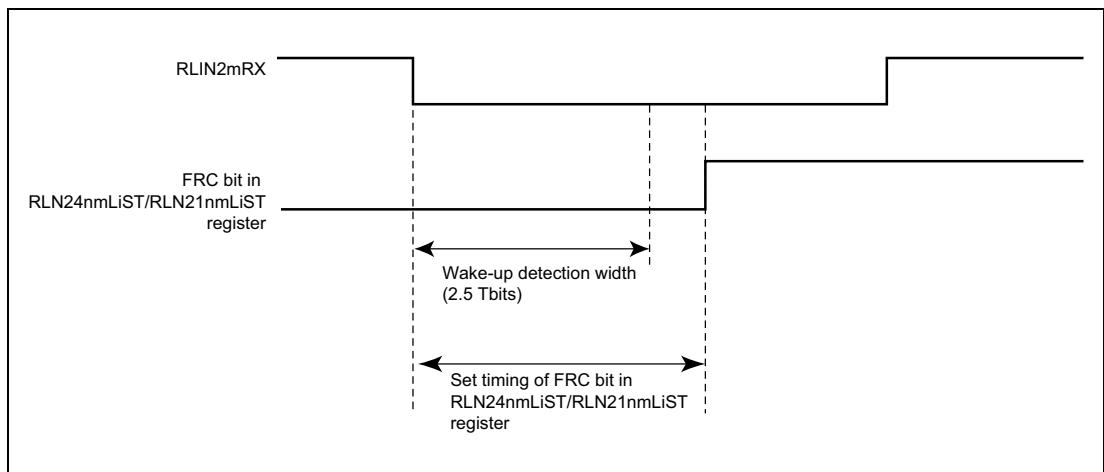


Figure 16.12 Input Signal Low Level Count Function

### 16.12.3 Wake-Up Collision

If the master node and the slave node transmit wake-up signals simultaneously, a collision will occur on the LIN bus; however the a collision of wake-up signals is not detected in the LIN master interface.

## 16.13 Status

During LIN mode operation, the LIN master interface can detect seven types of statuses.

Three of these statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, can generate interrupt requests.

**Table 16.36** shows the types of statuses.

**Table 16.36 Types of Statuses**

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to not-LIN-reset-mode, if the LIN master interface actually exits LIN reset mode.	After the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to LIN reset mode, if the LIN master interface enters LIN reset mode.	All modes	OMM0 bit in the RLN24nmLiMST/ RLN21nmLiMST register	—
Operation mode	After the OM1 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to LIN operation mode, if the LIN master interface actually enters LIN operation mode.	After the OM1 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to LIN wake-up mode, if the LIN master interface enters LIN wake-up mode.	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	OMM1 bit in the RLN24nmLiMST/ RLN21nmLiMST register	—
Successful frame/wake-up transmission	When a frame (header transmission + response transmission) or a wake-up signal is transmitted successfully.	<ul style="list-style-type: none"> <li>• When the next communication is started</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	FTC flag in the RLN24nmLiST/ RLN21nmLiST register	✓
Successful frame/wake-up reception	When a frame (header transmission + response reception) or a wake-up signal is received successfully.	<ul style="list-style-type: none"> <li>• When the next communication is started</li> <li>• When cleared by software<sup>*1</sup></li> <li>• After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	FRC flag in the RLN24nmLiST/ RLN21nmLiST register	✓
Error detection	If any of the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST / RLN21nmLiEST register is set to 1 (error detected).	<ul style="list-style-type: none"> <li>• When the next communication is started</li> <li>• When cleared by software<sup>*1</sup></li> <li>• After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	ERR flag in the RLN24nmLiST/ RLN21nmLiST register	✓
Successful data 1 reception	When the RFT bit in the RLN24nmLiDFC/RLN21nmLiDFC register is 0 (reception) and the first byte of the response field is received successfully. <sup>*2</sup>	<ul style="list-style-type: none"> <li>• When the next communication is started</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	LIN operation mode	D1RC flag in the RLN24nmLiST/ RLN21nmLiST register	—
Successful header transmission	When a header field is transmitted successfully.	<ul style="list-style-type: none"> <li>• When the next communication is started</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	LIN operation mode	HTRC flag in the RLN24nmLiST/ RLN21nmLiST register	—

Note 1. while in LIN operation mode, the ERR flag in the RLN24nmLiST / RLN21nmLiST register is cleared to 0 by writing 0 to the CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN24nmLiEST / RLN21nmLiEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN24nmLiDFC / RLN21nmLiDFC register are 0000<sub>B</sub> (0 bytes + checksum).

## 16.14 Error Status

### 16.14.1 Types of Error Statuses

The LIN master interface can detect five types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN24nmLiEST / RLN21nmLiEST register.

All error statuses represent interrupt sources.

**Table 16.37** shows the types of error statuses.

**Table 16.37 Types of Error Statuses**

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Selection of Detection Enable/Disable	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match <sup>*1</sup>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	Aborted	√	BER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Physical bus error	<ul style="list-style-type: none"> <li>• LIN bus is detected to be high level when sending a break</li> <li>• LIN bus is detected to be low level when sending a break delimiter</li> <li>• LIN bus is detected to be high level when sending a wake-up</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	Aborted	√	PBER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Frame timeout error	A frame transmission/reception does not complete within a given time <sup>*2</sup>	LIN operation mode	Aborted	√	FTER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Framing error	In response field reception, the stop bit of each data byte is low level	LIN operation mode	Aborted	√	FER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Checksum error	In response field reception, checksum test results in an error	LIN operation mode	—	✗	CSER flag in the RLN24nmLiEST/ RLN21nmLiEST register

Note 1. If a bit error is detected, the process is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted after the bit that caused the error is transmitted.

Note 2. The timeout time depends on the response field data length (the RFDL[3:0] bits in the RLN24nmLiDFC / RLN21nmLiDFC register) and the checksum selection (the CSM bit in the RLN24nmLiDFC / RLN21nmLiDFC register), and this can be calculated according to the following formula:

When classic checksum is selected (when the CSM bit in the RLN24nmLiDFC / RLN21nmLiDFC is 0):

Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

When enhanced checksum is selected (when the CSM bit in the RLN24nmLiDFC / RLN21nmLiDFC is 1):

Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is longer than the TFRAME\_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME\_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.

The error status is cleared when the next communication is started, when it is cleared by software, or at a transition to LIN reset mode.

### 16.14.2 Target Time Domain for Error Detection

Figure 16.13 shows the time domain in which the LIN master interface monitors for error detection.

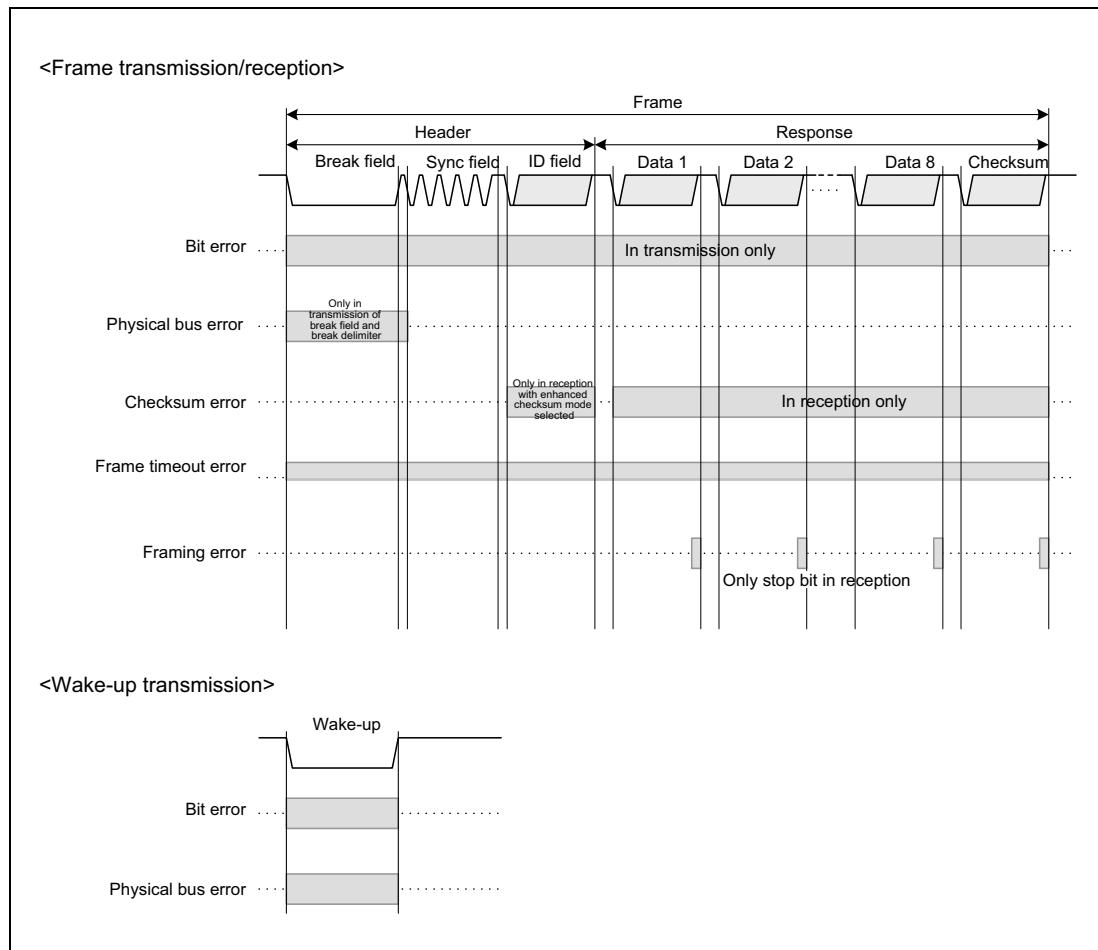


Figure 16.13 Target Time Domain for Error Detection

## 16.15 LIN Self-Test Mode

The LIN master interface provides LIN self-test mode. When the LIN master interface enters LIN self-test mode, RLIN2mTX and RLIN2mRX are disconnected from the external pins, and are internally connected in the LIN master interface. Thus, the frame transmitted from RLIN2mTX is looped back to RLIN2mRX.

Two types of self-test can be performed:

- LIN self-test mode (transmission): header transmission and response transmission
- LIN self-test mode (reception): header transmission and response reception

In LIN self-test mode, the operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator. Regardless of the setting of the baud rate related registers, the baud rate operates at the LIN communication clock source/16 [bps].

In LIN self-test mode, the following functions are not supported:

- LIN wake-up mode
- Frame separate mode

Do not use these functions.

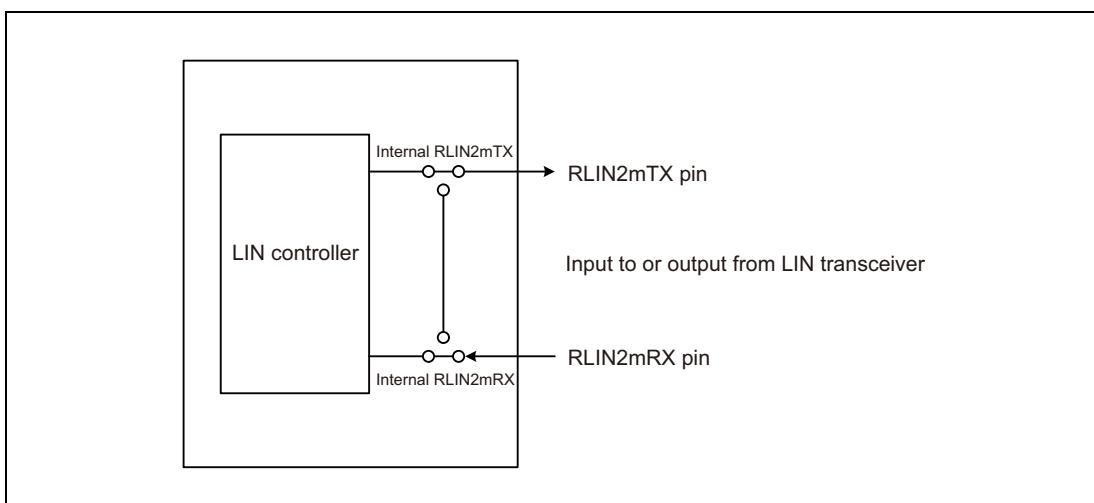
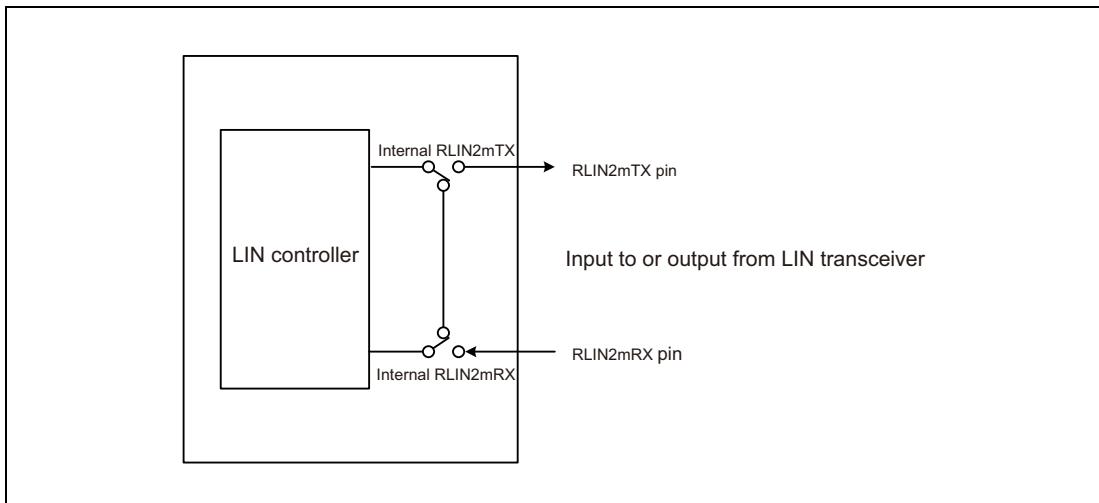


Figure 16.14 Connection in LIN Reset Mode, LIN Wake-Up Mode, and LIN Operation Mode



**Figure 16.15 Connection in LIN Self-Test Mode**

### 16.15.1 Transition to LIN Self-Test Mode

Writing to the RLN24nGLSTC / RLN21nGLSTC register enables LIN self-test mode.

The LSTM bit in the RLN24nGLSTC / RLN21nGLSTC register being set to 1 indicates that the mode has transitioned to LIN self-test mode.

A specific sequence is required to transition to LIN self-test mode. In this sequence, information must be written three times consecutively to the LIN self-test control register, as follows:

- Switch all channels of the unit to LIN reset mode.  
Set the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to 0 (LIN reset mode).  
Read the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register and confirm that it is 0 (LIN reset mode).
- 1st write: RLN24nGLSTC / RLN21nGLSTC register =  $1010\ 0111_B$  ( $A7_H$ )
- 2nd write: RLN24nGLSTC / RLN21nGLSTC register =  $0101\ 1000_B$  ( $58_H$ )
- 3rd write: RLN24nGLSTC / RLN21nGLSTC register =  $0000\ 0001_B$  ( $01_H$ )
- Confirm that all channels have transitioned to LIN self-test mode  
Read the LSTM bit in the RLN24nGLSTC / RLN21nGLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write ( $A7_H$ ) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the 1st write step. In addition, if a write to another LIN-related register in the same unit is performed during transition to LIN self-test mode (three consecutive write operations to the RLN24nGLSTC / RLN21nGLSTC register), the transition is also canceled.

### 16.15.2 Transmission in LIN Self-Test Mode

To execute a self-test on transmission, perform the procedure below:

- Set the baud rate related registers.

RLN24nGLBRP0 / RLN21nGLBRP0 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN24nGLBRP1 / RLN21nGLBRP1 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN24nmLiMD / RLN21nmLiMD register = 0000 xx00<sub>B</sub><sup>\*1</sup>

- Set interrupt enable register and error enable related registers.

RLN24nmLiIE / RLN21nmLiIE register = 0000 0xxx<sub>B</sub><sup>\*2</sup>

RLN24nmLiEDE / RLN21nmLiEDE register = 0000 xxxx<sub>B</sub>

- Set the break field and space related registers.

RLN24nmLiBFC / RLN21nmLiBFC register = 00xx xxxx<sub>B</sub>

RLN24nmLiSC / RLN21nmLiSC register = 00xx 0xxx<sub>B</sub>

- Exit the LIN reset mode.

Write 11<sub>B</sub> to the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are 11<sub>B</sub>.

- Set the transmit frame related registers.

RLN24nmLiDFC / RLN21nmLiDFC register = 00x1 xxxx<sub>B</sub>

RLN24nmLiIDB / RLN21nmLiIDB register = xxxx xxxx<sub>B</sub>

RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8 / RLN21nmLiDBR8 registers = xxxx xxxx<sub>B</sub>

- Start header transmission → response transmission

Set the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).

The LIN self-test mode (transmission) is executed, interrupts are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN master interface.

To suspend the LIN self-test mode (transmission) while it is running, set the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to 0 (LIN reset mode), which causes a transition to the LIN reset mode.

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB / RLN21nmLiIDB, RLN24nmLiDBRb / RLN21nmLiDBRb, and

RLN24nmLiCBR / RLN21nmLiCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is cleared.

- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is cleared.

#### NOTE

x: Don't care

**Note 1.** The settings of the following registers are not reflected in the operation of the LIN self-test mode:

the RLN24nGLBRP0 / RLN21nGLBRP0 register, the RLN24nGLBRP1 / RLN21nGLBRP1 register, and the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register. Therefore, setting these registers is not necessary.

**Note 2.** If necessary, set the related registers described in **Section 6, Exceptions/Interrupts**.

### 16.15.3 Reception in LIN Self-Test Mode

To execute a self-test on reception, perform the procedure below:

- Set the baud rate related registers.

RLN24nGLRP0 / RLN21nGLRP0 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>  
 RLN24nGLRP1 / RLN21nGLRP1 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>  
 RLN24nmLiMD / RLN21nmLiMD register = 0000 xx00<sub>B</sub><sup>\*1</sup>

- Set the interrupt enable and error enable related registers.

RLN24nmLiIE / RLN21nmLiIE register = 0000 0xxx<sub>B</sub><sup>\*2</sup>  
 RLN24nmLiEDE / RLN21nmLiEDE register = 0000 x0xx<sub>B</sub>

- Set the break field and space related registers.

RLN24nmLiBFC / RLN21nmLiBFC register = 00xx xxxx<sub>B</sub>  
 RLN24nmLiSC / RLN21nmLiSC register = 00xx 0xxx<sub>B</sub><sup>\*1</sup>

- Exit the LIN reset mode.

Write 11<sub>B</sub> to the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are 11<sub>B</sub>.

- Set the receive frame related registers.

RLN24nmLiDFC / RLN21nmLiDFC register = 00x0 xxxx<sub>B</sub>  
 RLN24nmLiIDB / RLN21nmLiIDB register = xxxx xxxx<sub>B</sub>  
 RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8 / RLN21nmLiDBR8 registers = xxxx xxxx<sub>B</sub>

RLN24nmLiCBR / RLN21nmLiCBR register = xxxx xxxx<sub>B</sub>

Since the checksum value to be transmitted is not automatically calculated, perform the calculation and specify the calculated value in the RLN24nmLiCBR / RLN21nmLiCBR register. By specifying an incorrect checksum, the checksum error can be tested.

- Start header transmission → response reception

Set the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).

The LIN self-test mode (reception) is executed, interrupts are generated, and status and error status are also updated.

To suspend the LIN self-test mode (reception) while it is running, set the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to 0 (LIN reset mode), which causes a transition to the LIN reset mode.

- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB / RLN21nmLiIDB, RLN24nmLiDBRb / RLN21nmLiDBRb, and RLN24nmLiCBR / RLN21nmLiCBR registers (the data is reversed before being stored because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is cleared.

**NOTE**

x: Don't care

- Note 1.** The settings of the following registers are not reflected to the operation of the LIN self-test mode:  
the RLN24nGLRP0 / RLN21nGLRP0 register, the RLN24nGLRP1 / RLN21nGLRP1 register, the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register, and the IBS bit and IBHS bit (response space only) in the RLN24nmLiSC / RLN21nmLiSC register. Therefore, setting these registers is not necessary.

- Note 2.** If necessary, set the related registers described in **Section 6, Exceptions/Interrupts**.

#### 16.15.4 Exiting LIN Self-Test Mode

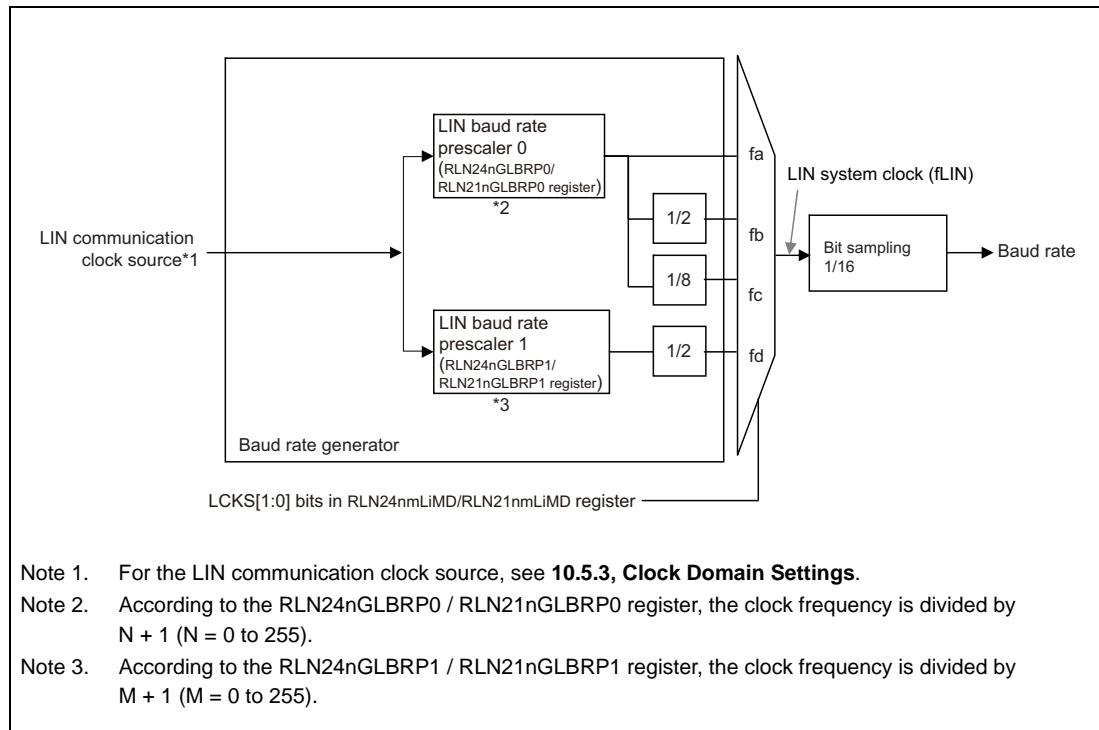
To exit LIN self-test mode, perform the procedure below:

- Switch all channels of the unit to LIN reset mode.  
Write 0 to the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to make a transition to LIN reset mode. However, if the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are not  $11_B$  in any channels of the unit after the transition to LIN self-test mode, write  $11_B$  to the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register in any one channel. Check that the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are set to  $11_B$ , and then make a transition to LIN reset mode.
- Verify that the LIN master interface has exited LIN self-test mode.  
Read the LSTM bit in the RLN24nGLSTC / RLN21nGLSTC register and confirm that it is not 0 (not in LIN self-test mode)
- Verify the transition to LIN reset mode.  
Read the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register and confirm that it is 0 (LIN reset mode).

## 16.16 Baud Rate Generator

The LIN system clock ( $f_{LIN}$ ) is obtained by dividing the LIN communication clock source frequency by the baud rate generator, and the baud rate is obtained by dividing that clock by 16. The inverse of this baud rate is called the bit time ( $T_{bit}$ ).

**Figure 16.16** shows a block diagram of baud rate generation.



**Figure 16.16 Block Diagram of Baud Rate Generation**

Set the LIN communication clock source in a range from 4 MHz to 40 MHz.

By setting the RLN24nGLBRP0 / RLN21nGLBRP0 register so that  $fa$  is 307200 Hz ( $= 19200 \times 16$ ), the resulting system clock frequencies are  $fa = 19200 \times 16$ ,  $fb = 9600 \times 16$ , and  $fc = 2400 \times 16$ . These system clock frequencies are divided by 16 in the bit timing generator, enabling baud rates of 1920 bps, 9600 bps, and 2400 bps to be generated. Also, by setting the RLN24nGLBRP1 / RLN21nGLBRP1 register so that  $fd$  is 166672 Hz ( $= 10417 \times 16$ ), the resulting system clock frequency is  $fd = 10417 \times 16$ . This system clock frequency is divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The formula for calculating baud rate is shown below.

*Baud rate:*

$$\begin{aligned}
 &= \{\text{Frequency of LIN communication clock source}\} \div ((\text{RLN24nGLBRP0 or RLN21nGLBRP0}) + 1) \div 16 \text{ [bps]} \text{ (When } fa \text{ is selected)} \\
 &= \{\text{Frequency of LIN communication clock source}\} \div ((\text{RLN24nGLBRP0 or RLN21nGLBRP0}) + 1) \div 2 \div 16 \text{ [bps]} \text{ (When } fb \text{ is selected)} \\
 &= \{\text{Frequency of LIN communication clock source}\} \div ((\text{RLN24nGLBRP0 or RLN21nGLBRP0}) + 1) \div 8 \div 16 \text{ [bps]} \text{ (When } fc \text{ is selected)} \\
 &= \{\text{Frequency of LIN communication clock source}\} \div ((\text{RLN24nGLBRP1 or RLN21nGLBRP1}) + 1) \div 2 \div 16 \text{ [bps]} \text{ (When } fd \text{ is selected)}
 \end{aligned}$$

## Section 17 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

### 17.1 Features of RH850/F1L RLIN3

#### 17.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

Each RLIN3 unit has a single channel interface. "Number of channels" therefore has the same meaning as "number of units" in this section.

**Table 17.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of units	1	2	3	4	6	6
Name	RLIN3n (n = 0)	RLIN3n (n = 0, 1)	RLIN3n (n = 0 to 2)	RLIN3n (n = 0 to 3)	RLIN3n (n = 0 to 5)	RLIN3n (n = 0 to 5)

**Table 17.2 Unit Configurations and Channels**

Unit Name (Channel Name) RLIN3n	Channels per Unit	RH850/F1L 48 pins (1 ch)	RH850/F1L 64 pins (2 ch)	RH850/F1L 80 pins (3 ch)	RH850/F1L 100 pins (4 ch)	RH850/F1L 144 pins (6 ch)	RH850/F1L 176 pins (6 ch)
RLIN30	1	✓	✓	✓	✓	✓	✓
RLIN31	1		✓	✓	✓	✓	✓
RLIN32	1			✓	✓	✓	✓
RLIN33	1				✓	✓	✓
RLIN34	1					✓	✓
RLIN35	1					✓	✓

**Note:** The channel names are same as those of the corresponding units.

**Table 17.3 Indices**

Index	Description
n	Throughout this section, the individual RLIN3 units are identified by the index "n" (n = 0 to 5); for example, RLIN3nLCUC is the LIN control register.
b	Throughout this section, the individual transmit/ received data buffers of RLIN3n are identified by the index "b" (b = 1 to 8); for example, RLIN3nLDBRb is the data buffer register.

The following lists the index value corresponding to each product.

**Table 17.4 Index Correspondence of Each Product**

Index Corresponding to Product
All products
b = 1 to 8

### 17.1.2 Register Base Address

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses.

**Table 17.5 Register Base Addresses**

Base Address Name	Base Address
<RLIN30_base>	FFCF 0000 <sub>H</sub>
<RLIN31_base>	FFCF 0040 <sub>H</sub>
<RLIN32_base>	FFCF 0080 <sub>H</sub>
<RLIN33_base>	FFCF 00C0 <sub>H</sub>
<RLIN34_base>	FFCF 0100 <sub>H</sub>
<RLIN35_base>	FFCF 0140 <sub>H</sub>

### 17.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

**Table 17.6 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
RLIN3n	LIN communication clock sources	CKSCLK_ILIN <sup>*1,*2</sup>
	Register access clock	CKSCLK_ILIN <sup>*1,*2</sup>

Note 1. The clock domain CKSCLK\_ILIN divided clock can be supplied only to RLIN30 channel.

Note 2. Set the LIN communication clock source in the range of 4 MHz to 40 MHz.

### 17.1.4 Interrupt Request

RLIN3 interrupt requests are listed in the following table.

**Table 17.7 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>RLIN30</b>			
INTRLIN3n (n = 0)	RLIN30 interrupt	25	—
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	26	10 (channels 0 to 7)
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	27	11 (channels 0 to 7)
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	28	—
<b>RLIN31</b>			
INTRLIN3n (n = 1)	RLIN31 interrupt	112	—
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	113	22 (channels 8 to 15)
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	114	23 (channels 8 to 15)
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	115	—
<b>RLIN32</b>			
INTRLIN3n (n = 2)	RLIN32 interrupt	156	—
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	157	44 (channels 0 to 7)
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	158	45 (channels 0 to 7)
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	159	—
<b>RLIN33</b>			
INTRLIN3n (n = 3)	RLIN33 interrupt	220	—
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	221	47 (channels 8 to 15)
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	222	48 (channels 8 to 15)
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	223	—
<b>RLIN34</b>			
INTRLIN3n (n = 4)	RLIN34 interrupt	224	—
INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	225	50 (channels 0 to 7)
INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	226	51 (channels 0 to 7)
INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	227	—
<b>RLIN35</b>			
INTRLIN3n (n = 5)	RLIN35 interrupt	228	—
INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	229	49 (channels 8 to 15)
INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	230	50 (channels 8 to 15)
INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	231	—

### 17.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

**Table 17.8 Reset Sources**

Unit Name	Reset Source
RLIN3n	All reset sources (ISORES)

### 17.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

**Table 17.9 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal
<b>RLIN30</b>		
RLIN3nRX (n = 0)	RLIN30 receive data input	RLIN30RX
RLIN3nTX (n = 0)	RLIN30 transmit data output	RLIN30TX
<b>RLIN31</b>		
RLIN3nRX (n = 1)	RLIN31 receive data input	RLIN31RX
RLIN3nTX (n = 1)	RLIN31 transmit data output	RLIN31TX
<b>RLIN32</b>		
RLIN3nRX (n = 2)	RLIN32 receive data input	RLIN32RX
RLIN3nTX (n = 2)	RLIN32 transmit data output	RLIN32TX
<b>RLIN33</b>		
RLIN3nRX (n = 3)	RLIN33 receive data input	RLIN33RX
RLIN3nTX (n = 3)	RLIN33 transmit data output	RLIN33TX
<b>RLIN34</b>		
RLIN3nRX (n = 4)	RLIN34 receive data input	RLIN34RX
RLIN3nTX (n = 4)	RLIN34 transmit data output	RLIN34TX
<b>RLIN35</b>		
RLIN3nRX (n = 5)	RLIN35 receive data input	RLIN35RX
RLIN3nTX (n = 5)	RLIN35 transmit data output	RLIN35TX

## 17.2 Overview

### 17.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master, LIN slave, or UART.

#### LIN master

- LIN reset mode
- LIN mode (LIN master mode)
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

#### LIN slave

- LIN reset mode
- LIN mode (LIN slave mode [auto baud rate] or LIN slave mode [fixed baud rate])
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

#### UART

- LIN reset mode
- UART mode

**Table 17.10** shows the LIN/UART interface specifications.

**Table 17.10 LIN/UART Interface Specifications (1/3)**

Item	Specifications					
	Channel count Up to 6 channels					
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602				
	Variable frame structure	<table> <tr> <td>Master</td> <td> <ul style="list-style-type: none"> <li>Break transmission width: 13 to 28 Tbits</li> <li>Break delimiter transmission width: 1 to 4 Tbits</li> <li>Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)<sup>1</sup></li> <li>Transmission response space width: 0 to 7 Tbits<sup>1</sup></li> <li>Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area)</li> <li>Transmission wake-up width: 1 to 16 Tbits</li> </ul> </td> </tr> <tr> <td>Slave</td> <td> <ul style="list-style-type: none"> <li>Break reception width : 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate]</li> <li>Transmission response space width: 0 to 7 Tbits</li> <li>Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area)</li> <li>Transmission wake-up width: 1 to 16 Tbits</li> </ul> </td> </tr> </table>	Master	<ul style="list-style-type: none"> <li>Break transmission width: 13 to 28 Tbits</li> <li>Break delimiter transmission width: 1 to 4 Tbits</li> <li>Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)<sup>1</sup></li> <li>Transmission response space width: 0 to 7 Tbits<sup>1</sup></li> <li>Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area)</li> <li>Transmission wake-up width: 1 to 16 Tbits</li> </ul>	Slave	<ul style="list-style-type: none"> <li>Break reception width : 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate]</li> <li>Transmission response space width: 0 to 7 Tbits</li> <li>Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area)</li> <li>Transmission wake-up width: 1 to 16 Tbits</li> </ul>
Master	<ul style="list-style-type: none"> <li>Break transmission width: 13 to 28 Tbits</li> <li>Break delimiter transmission width: 1 to 4 Tbits</li> <li>Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)<sup>1</sup></li> <li>Transmission response space width: 0 to 7 Tbits<sup>1</sup></li> <li>Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area)</li> <li>Transmission wake-up width: 1 to 16 Tbits</li> </ul>					
Slave	<ul style="list-style-type: none"> <li>Break reception width : 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate]</li> <li>Transmission response space width: 0 to 7 Tbits</li> <li>Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area)</li> <li>Transmission wake-up width: 1 to 16 Tbits</li> </ul>					
Checksum	<ul style="list-style-type: none"> <li>Automatic operation for both transmission and reception</li> <li>Classic or enhanced selectable (for each frame)</li> </ul>					
Response field data byte count	<table> <tr> <td>Variable from 0 to 8 bytes</td> <td></td> </tr> <tr> <td>Multi-byte (9 or more bytes) response transmission and reception also possible</td> <td></td> </tr> </table>		Variable from 0 to 8 bytes		Multi-byte (9 or more bytes) response transmission and reception also possible	
Variable from 0 to 8 bytes						
Multi-byte (9 or more bytes) response transmission and reception also possible						
Frame communication modes	Master	<ul style="list-style-type: none"> <li>Mode in which header transmission and response transmission/reception are started with a single transmission start request</li> <li>Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)</li> </ul>				
	Slave	<ul style="list-style-type: none"> <li>Mode in which header is automatically received with fixed baud rate</li> <li>Mode in which header is automatically received with the baud rate set according to the sync field measurement result of the sync field and break field detected</li> </ul>				
Wake-up transmission and reception	<table> <tr> <td>LIN wake-up mode provided</td> <td></td> </tr> <tr> <td> <ul style="list-style-type: none"> <li>Wake-up transmission (1 to 16 Tbits)</li> <li>Wake-up reception</li> </ul> </td> <td>Low-level width of input signals measured</td> </tr> </table>		LIN wake-up mode provided		<ul style="list-style-type: none"> <li>Wake-up transmission (1 to 16 Tbits)</li> <li>Wake-up reception</li> </ul>	Low-level width of input signals measured
LIN wake-up mode provided						
<ul style="list-style-type: none"> <li>Wake-up transmission (1 to 16 Tbits)</li> <li>Wake-up reception</li> </ul>	Low-level width of input signals measured					
Status	Master	<ul style="list-style-type: none"> <li>Successful frame/wake-up transmission</li> <li>Successful header transmission</li> <li>Successful frame/wake-up reception<sup>2</sup></li> <li>Successful data 1 reception</li> <li>Error detection</li> <li>Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)</li> </ul>				
	Slave	<ul style="list-style-type: none"> <li>Successful response/wake-up transmission</li> <li>Successful response/wake-up reception<sup>2</sup></li> <li>Successful header reception</li> <li>Successful data 1 reception</li> <li>Error detection</li> <li>Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)</li> </ul>				

**Table 17.10 LIN/UART Interface Specifications (2/3)**

<b>Item</b>	<b>Specifications</b>				
LIN communication function	Master	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Checksum error</li> <li>• Frame timeout error/response timeout error</li> <li>• Physical bus error</li> <li>• Framing error</li> <li>• Response preparation error</li> </ul>			
	Slave	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Checksum error</li> <li>• Frame timeout error/response timeout error</li> <li>• Sync field error</li> <li>• ID parity error</li> <li>• Framing error</li> <li>• Response preparation error</li> </ul>			
Baud rate selection	Baud rates conforming to the LIN specifications generated using baud rate generator				
Test mode	Self-test mode for user evaluation				
Interrupt function	Master	<ul style="list-style-type: none"> <li>• Successful header/frame/wake-up transmission</li> <li>• Successful frame/wake-up reception*<sup>2</sup></li> <li>• Error detection</li> </ul>			
	Slave	<ul style="list-style-type: none"> <li>• Successful response/wake-up transmission</li> <li>• Successful Header/response/wake-up reception*<sup>2</sup></li> <li>• Error detection</li> </ul>			
UART communication function	Data buffer	<ul style="list-style-type: none"> <li>• Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported)</li> <li>• UART buffer (exclusively for transmission; variable data length from 1 to 9. Character length of 7 and 8 bits supported)</li> <li>• Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported)</li> </ul>			
	Data format	<p>Character length: 7 or 8 bits Length of 9 bits supported by using the expansion bit.</p>			
Transmission stop bit: 1 or 2 bits					
Parity function: odd, even, 0, or none					
LSB- or MSB-first transfer selectable					
Reverse input/output of transmission/reception data possible					
Status	<ul style="list-style-type: none"> <li>• Transmission status</li> <li>• Reception status</li> <li>• Successful UART buffer transmission</li> <li>• Error detection</li> <li>• Expansion bit detection</li> <li>• ID match</li> <li>• Reset mode status</li> </ul>				
Error status	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Framing error</li> <li>• Parity error</li> <li>• Overrun error</li> </ul>				
Baud rate selection	With the built-in baud rate generator, any baud rate can be set.				
When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register.					

**Table 17.10 LIN/UART Interface Specifications (3/3)**

Item	Specifications	
UART communication function	Reception of the stop bit is guaranteed. (Start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).	
Interrupt function	• Transmission start/complete • Reception complete • Status/error detection	

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

## 17.2.2 Block Diagram

Figure 17.1 shows a block diagram of the LIN/UART interface.

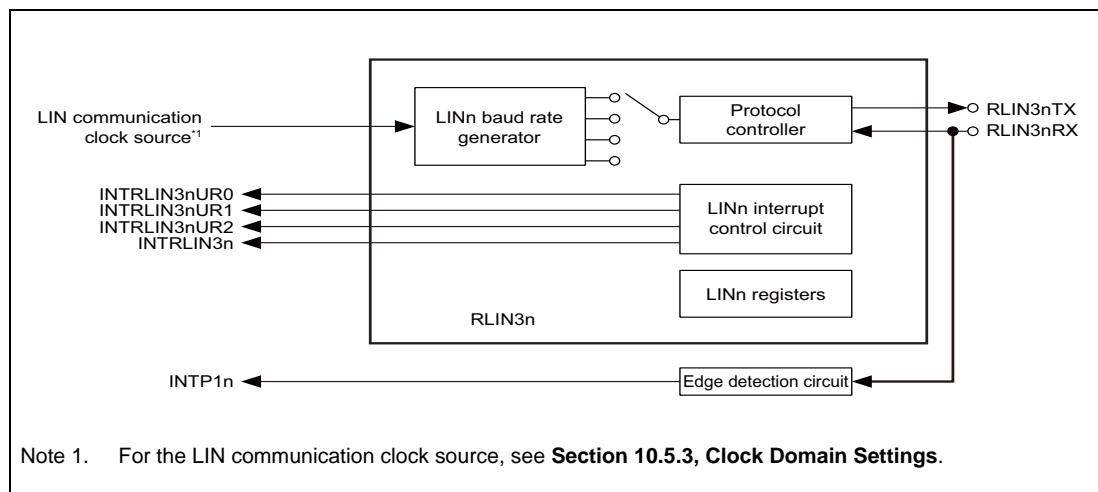


Figure 17.1 LIN/UART Interface Block Diagram

## 17.2.3 Terms used in block diagram

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock
- LINn registers: LIN/UART interface registers
- LINn interrupt control circuit: Controls interrupt requests generated by the LIN/UART interface

## 17.3 Registers

### 17.3.1 List of Registers

RLIN3 registers are listed in the following table.

For details about <RLIN3n\_base>, see **Section 17.1.2, Register Base Address**.

**Table 17.11 List of Registers (1/2)**

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART
RLN3n	LIN wake-up baud rate select register	RLN3nLWBR	<RLIN3n_base> + 01H	✓	✓	✓
RLN3n	LIN / UART baud rate prescaler 01 register	RLN3nLBRP01	<RLIN3n_base> + 02H	—	✓	✓
RLN3n	LIN / UART baud rate prescaler 0 register	RLN3nLBRP0	<RLIN3n_base> + 02H	✓	✓	✓
RLN3n	LIN / UART baud rate prescaler 1 register	RLN3nLBRP1	<RLIN3n_base> + 03H	✓	✓	✓
RLN3n	LIN self-test control register	RLN3nLSTC	<RLIN3n_base> + 04H	✓	✓	—
RLN3n	LIN / UART mode register	RLN3nLMD	<RLIN3n_base> + 08H	✓	✓	✓
RLN3n	LIN break field configuration register/UART configuration register	RLN3nLBFC	<RLIN3n_base> + 09H	✓	✓	✓
RLN3n	LIN / UART space configuration register	RLN3nLSC	<RLIN3n_base> + 0AH	✓	✓	✓
RLN3n	LIN wake-up configuration register	RLN3nLWUP	<RLIN3n_base> + 0BH	✓	✓	—
RLN3n	LIN interrupt enable register	RLN3nLIE	<RLIN3n_base> + 0CH	✓	✓	—
RLN3n	LIN / UART error detection enable register	RLN3nLEDE	<RLIN3n_base> + 0DH	✓	✓	✓
RLN3n	LIN / UART control register	RLN3nLCUC	<RLIN3n_base> + 0EH	✓	✓	✓
RLN3n	LIN / UART transmission control register	RLN3nLTRC	<RLIN3n_base> + 10H	✓	✓	✓
RLN3n	LIN / UART mode status register	RLN3nLMST	<RLIN3n_base> + 11H	✓	✓	✓
RLN3n	LIN / UART status register	RLN3nLST	<RLIN3n_base> + 12H	✓	✓	✓
RLN3n	LIN / UART error status register	RLN3nLEST	<RLIN3n_base> + 13H	✓	✓	✓
RLN3n	LIN data field configuration register	RLN3nLDFC	<RLIN3n_base> + 14H	✓	✓	✓
RLN3n	LIN / UART ID buffer register	RLN3nLIDB	<RLIN3n_base> + 15H	✓	✓	✓
RLN3n	LIN checksum buffer register	RLN3nLCBR	<RLIN3n_base> + 16H	✓	✓	—
RLN3n	UART data buffer 0 register	RLN3nLUDB0	<RLIN3n_base> + 17H	—	—	✓
RLN3n	LIN / UART data buffer 1 register	RLN3nLDBR1	<RLIN3n_base> + 18H	✓	✓	✓
RLN3n	LIN / UART data buffer 2 register	RLN3nLDBR2	<RLIN3n_base> + 19H	✓	✓	✓
RLN3n	LIN / UART data buffer 3 register	RLN3nLDBR3	<RLIN3n_base> + 1AH	✓	✓	✓
RLN3n	LIN / UART data buffer 4 register	RLN3nLDBR4	<RLIN3n_base> + 1BH	✓	✓	✓
RLN3n	LIN / UART data buffer 5 register	RLN3nLDBR5	<RLIN3n_base> + 1CH	✓	✓	✓
RLN3n	LIN / UART data buffer 6 register	RLN3nLDBR6	<RLIN3n_base> + 1DH	✓	✓	✓
RLN3n	LIN / UART data buffer 7 register	RLN3nLDBR7	<RLIN3n_base> + 1EH	✓	✓	✓
RLN3n	LIN / UART data buffer 8 register	RLN3nLDBR8	<RLIN3n_base> + 1FH	✓	✓	✓
RLN3n	UART operation enable register	RLN3nLUOER	<RLIN3n_base> + 20H	—	—	✓
RLN3n	UART option register 1	RLN3nLUOR1	<RLIN3n_base> + 21H	—	—	✓
RLN3n	UART transmission data register	RLN3nLUTDR	<RLIN3n_base> + 24H	—	—	✓
RLN3n	UART transmission data register L	RLN3nLUTDRL	<RLIN3n_base> + 24H	—	—	✓
RLN3n	UART transmission data register H	RLN3nLUTDRH	<RLIN3n_base> + 25H	—	—	✓
RLN3n	UART reception data register	RLN3nLURDR	<RLIN3n_base> + 26H	—	—	✓
RLN3n	UART reception data register L	RLN3nLURDRL	<RLIN3n_base> + 26H	—	—	✓

Table 17.11 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART
RLN3n	UART reception data register H	RLN3nLURDRH	<RLIN3n_base> + 27H	—	—	✓
RLN3n	UART wait transmission data register	RLN3nLUWTDR	<RLIN3n_base> + 28H	—	—	✓
RLN3n	UART wait transmission data register L	RLN3nLUWTDRL	<RLIN3n_base> + 28H	—	—	✓
RLN3n	UART wait transmission data register H	RLN3nLUWTDRH	<RLIN3n_base> + 29H	—	—	✓

**Note:** ✓: Used, —: Not used

When writing to an unused register, write the value after reset.

### 17.3.2 LIN Master Related Registers

#### 17.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 01H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
			NSPB[3:0]		LPRS[2:0]			LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.12 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS bit of the RLN3nLMD register is used. (LIN1.3) 1: In LIN wake-up mode, the clock fa is used regardless of the setting in the LCKS bit of the RLN3nLMD register. (LIN2.x)

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

#### NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In LIN master mode (LIN/UART mode select bits in LIN mode register = 00B), set these bits to 0000B or 1111B (16 sampling).

#### LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.

The LIN communication clock source is divided by this prescaler.

#### LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the

RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 us or longer to be detected in the LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

### 17.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 02H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
LBRP0[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.13 RLN3nLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00H to FFH

Configure the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by N + 1.

### 17.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 03H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
LBRP1[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.14 RLN3nLBRP1 Register Contents**

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00H to FFH

Configure the RLN3nLBRP1 register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source  $f_d$ .

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M+1.

### 17.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 04H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.15 RLN3nLSTC Register Contents**

Bit Position	Bit Name	Function
7 to 0	—	Writing A7H, 58H, and 01H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Configure the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

Writing A7H, 58H, and 01H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the mode is changed to LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000000B”, and bit 7 returns an undefined value.

#### LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.

For exiting LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of A7H, 58H, and 01H.

### 17.3.2.5 RLIN3nLMD — LIN Mode Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base>+ 08H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.16 RLIN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3n interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 0: LIN master mode

Configure the RLIN3nLMD register when the OMM0 bit in the RLIN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

#### LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.

With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.

With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.

For each interrupt source, see **Section 17.4, Interrupt Sources**.

#### LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.

With 00<sub>B</sub> set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).

With 01<sub>B</sub> set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).

With 10<sub>B</sub> set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).

With 11<sub>B</sub> set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When  $1_B$  is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), and the RLN3nLMST register is  $01_H$  (LIN wake-up mode), the protocol controller is supplied with  $f_a$  regardless of the setting of this bit (the LCKS bit is not changed).

#### **LMD[1:0] Bits (LIN/UART Mode Select)**

The LMD bits select the LIN/UART interface mode.

To use the LIN/UART interface as an LIN master, set these bits to  $00_B$ .

### 17.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 09H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]			BLT[3:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.17 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

Some combinations of the specified values result in a frame length exceeding the timeout time. Set the appropriate values in this register.

#### BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header.  
1 Tbit to 4 Tbits can be set.

#### BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header.  
13 Tbits to 28 Tbits can be set.

### 17.3.2.7 RLN3nLSC — LIN Space Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base>+ 0A<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—		IBHS[2:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 17.18 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Configure the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Specify the appropriate values in this register.

#### IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.

0 Tbit to 3 Tbits can be set.

These bits are enabled only during response transmission; these are disabled during response reception.

#### IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.

0 Tbit to 7 Tbits can be set.

The response space setting is enabled only during response transmission; setting is disabled during response reception.

The inter-byte space (header) is equal to the response space.

### 17.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base>+ 0B<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
WUTL[3:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 17.19 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b <sub>7</sub> b <sub>4</sub> 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.

1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

### 17.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

**Table 17.20 RLN3nLIE Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Request Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Configure the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### SHIE Bit (Successful Header Transmission Interrupt Request Enable)

The SHIE bit enables or disables interrupt request upon successful transmission of a header. With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1. With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

#### ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error. With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1. With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1. Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors. Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

#### FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (input signal low-level width count).

With 0 set, the interrupt request for RLIN3n successful reception is not generated when the FRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

#### **FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Request Enable)**

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.

With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

### 17.3.2.10 RLN3nLEDE —LIN Error Detection Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0DH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

**Table 17.21 RLN3nLEDE Register Contents**

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable *1 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable *1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1.

Configure the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

#### LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.

With 0 set, the timeout function applies to frame timeout.

With 1 set, the timeout function applies to response timeout.

For details on the timeout error, see **Section 17.7.7, Error Statuses**.

#### FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

Set this bit to 1. The detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 17.7.7, Error Statuses**.

#### FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is reflected in the FTER flag of the RLN3nLEST register. With the LTES bit, either the frame timeout error or response timeout error can be selected. Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received. For details on the timeout error, see **Section 17.7.7, Error Statuses**.

#### **PBERE Bit (Physical Bus Error Detection Enable)**

The PBERE bit enables or disables detection of the physical bus error. With 0 set, the physical bus error is not detected. With 1 set, the physical bus error is detected. When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register. For details on the physical bus error, see **Section 17.7.7, Error Statuses**.

#### **BERE Bit (Bit Error Detection Enable)**

The BERE bit enables or disables detection of the bit error. With 0 set, the bit error is not detected. With 1 set, the bit error is detected. Set this bit to 1. The detection result is indicated in the BER flag in the RLN3nLEST register. For details on the bit error, see **Section 17.7.7, Error Statuses**.

### 17.3.2.11 RLN3nLCUC — LIN Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0E<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 17.22 RLN3nLCUC Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode 1: LIN operation mode
0	OM0	LIN Reset 0: LIN reset mode 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01H to transition to LIN wake-up mode or to 03H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03<sub>H</sub> after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

#### OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.

With 0 set, LIN wake-up mode.

With 1 set, LIN operation mode.

This bit is enabled only when the OMM0 bit in the RLN3nLMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN3nLTCR register is 1.

#### OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.

With 0 set, LIN reset mode.

With 1 set, LIN reset mode is canceled.

### 17.3.2.12 RLN3nLTRC — LIN Transmission Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 10H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 17.23 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission/Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

#### RTS Bit (Response Transmission/Reception Start)

In frame separate mode, set the RTS bit to 1 after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02H to the RLN3nLTRC register using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of data group communication or transition to LIN reset mode.

#### FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission and reception.

Also set this bit to 1 to allow wake-up transmission and reception (input signal low-level width count).

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

### 17.3.2.13 RLN3nLMST — LIN Mode Status Register

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RLIN3n\_base> + 11<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 17.24 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

#### OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

#### OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

### 17.3.2.14 RLN3nLST — LIN Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 12H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

**Table 17.25 RLN3nLST Register Contents**

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### **HTRC Flag (Successful Header Transmission Flag)**

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value it had before 1 is written.

The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for RLN3n transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

#### **D1RC Flag (Successful Data 1 Reception Flag)**

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value it had before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. To clear the bit to 0 before the next communication is started (when the FTS bit of the

(when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

### **ERR Flag (Error Detection Flag)**

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register while in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

### **FRC Flag (Successful Frame/Wake-up Reception Flag)**

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value it had before 1 is written.

The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLN3n successful reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode. When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

### **FTC Flag (Successful Frame/Wake-up Transmission Flag)**

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value it had before 1 is written.

The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode. When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

### 17.3.2.15 RLN3nLEST — LIN Error Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 13H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 17.26 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value it had before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

### **CSER Flag (Checksum Error Flag)**

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value it had before 1 is written.

The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

### **FER Flag (Framing Error Flag)**

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the value of the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

### **FTER Flag (Timeout Error Flag)**

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

### **PBER Flag (Physical Bus Error Flag)**

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

### **BER Flag (Bit Error Flag)**

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode or LIN wake-up mode.

### 17.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 14H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT			RFDL[3:0]	
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W								

**Table 17.27 RLN3nLDFC Register Contents**

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

#### LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.

With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.

With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

#### FSM Bit (Frame Separate Mode Select)

The FSM bit selects the response communication mode.

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN3nLTRC register.

With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.

When transitioning to LIN self-test mode, set this bit to 0 before transition.

For details on frame separate mode, see **Section 17.7.4.1, Transmission of LIN Frames**.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

### **CSM Bit (Checksum Select)**

The CSM bit selects the checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When the timeout error detection is enabled (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details, see **Section 17.7.7, Error Statuses**.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.

During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

### **RFT Bit (Response Field Communication Direction Select)**

The RFT bit sets the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

### **RFDL[3:0] Bits (Response Field Length Select)**

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).

To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).

When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).

Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

### 17.3.2.17 RLN3nLIDB — LIN ID Buffer Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 15H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0			ID[5:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.28 RLN3nLIDB Register Contents**

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

In LIN self-test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

#### IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

#### ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

### 17.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

**Access:** This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 16<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
CKSM[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.29 RLN3nLCBR Register Contents**

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Stores the checksum value transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):  
The value transmitted can be read from the register. Read the value after transmission is completed.  
Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):  
The value received can be read from the register. Read the value after reception is completed.  
Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):  
After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):  
Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

### 17.3.2.19 RLIN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

**Access:** This register can be read or written in 8-bit units.

**Address:** RLIN3nLDBR1: <RLIN3n\_base> + 18<sub>H</sub>  
 RLIN3nLDBR2: <RLIN3n\_base> + 19<sub>H</sub>  
 RLIN3nLDBR3: <RLIN3n\_base> + 1A<sub>H</sub>  
 RLIN3nLDBR4: <RLIN3n\_base> + 1B<sub>H</sub>  
 RLIN3nLDBR5: <RLIN3n\_base> + 1C<sub>H</sub>  
 RLIN3nLDBR6: <RLIN3n\_base> + 1D<sub>H</sub>  
 RLIN3nLDBR7: <RLIN3n\_base> + 1E<sub>H</sub>  
 RLIN3nLDBR8: <RLIN3n\_base> + 1F<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
LDB[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.30 RLIN3nLDBRb Register Contents**

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or reads the received data. Setting range: 00 <sub>H</sub> to FF <sub>H</sub>

- For response transmission:

These registers set the data to be transmitted in the response field.

Use these registers with the following settings.

- RFT in RLIN3nLDFC register is 1 (transmission)
- FSM in RLIN3nLDFC register is 0 (not frame separate mode)
- FTS in RLIN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped)

or

- RFT in RLIN3nLDFC register is 1 (transmission)
- FSM in RLIN3nLDFC register is 1 (frame separate mode)
- RTS in RLIN3nLTRC register is 0 (response transmission/reception is stopped)

- For response reception:

These registers store the data received in the response field.

The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.

Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)

- For transmission of response data of 9 bytes or more:

Use these registers with the following settings.

- RFT in RLIN3nLDFC register is 1 (transmission)
- FSM in RLIN3nLDFC register is 1 (frame separate mode)
- RTS in RLIN3nLTRC register is 0 (response transmission/reception is stopped)

- For reception of response data of 9 bytes or more:

Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

### 17.3.3 LIN Slave Related Registers

#### 17.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 01H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R							

Table 17.31 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 0 1 1: 4 samplings 0 1 1 1: 8 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

#### NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).

When the frame communication is performed in LIN slave mode (fixed baud rate) (LMD[1:0] bits in the RLN3nLMD register = 11B), set these bits to “0000B” or “1111B” (16 samplings).

When the frame communication is performed in LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10B), set these bits to “0011B” (4 samplings) or “0111B” (8 samplings).

#### LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler. The LIN communication clock source is divided by this prescaler.

In LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10B), set these bits so that the prescaler clock becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz <sup>*1</sup>
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

**Note 1.** Use the clock with NSPB bits set to “0011<sub>B</sub>” (4 samplings).

### 17.3.3.2 RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register

**Access:** RLN3nLBRP01 can be read or written in 16-bit units.  
 RLN3nLBRP0 can be read or written in 8-bit units.  
 RLN3nLBRP1 can be read or written in 8-bit units.

**Address:** RLN3nLBRP01: <RLIN3n\_base> + 02<sub>H</sub>  
 RLN3nLBRP0: <RLIN3n\_base> + 02<sub>H</sub>  
 RLN3nLBRP1: <RLIN3n\_base> + 03<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRP[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.32 RLN3nLBRP01 Register Contents**

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000 <sub>H</sub> to FFFF <sub>H</sub>

Configure the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units using registers RLN3nLBRP0 and RLN3nLBRP1.

#### NOTE

In LIN slave mode [auto baud rate], the system automatically sets the result of baud rate correction to the RLN3nLBRP01 register on successful reception of the sync field.

### 17.3.3.3 RLN3nLSTC — LIN Self-Test Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 04H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.33 RLN3nLSTC Register Contents**

Bit Position	Bit Name	Function
7 to 0	-	Writing A7H, 58H, and 01H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Configure the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

Writing A7H, 58H, and 01H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the module is placed in LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

When read, bits 6 to 1 return “000000B”, and bit 7 returns an undefined value.

#### LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.

For exiting LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of A7H, 58H, and 01H.

### 17.3.3.4 RLIN3nLMD — LIN Mode Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 08H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 17.34 RLIN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3 interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are used.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 1 0: LIN Slave mode (auto baud rate) 1 1: LIN Slave mode (fixed baud rate)

Configure the RLIN3nLMD register when the OMM0 bit in the RLIN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.  
With 0 set, the noise filter is enabled when receiving data.  
With 1 set, the noise filter is disabled when receiving data.

#### LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.  
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.  
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are generated from the LIN/UART interface.  
For each interrupt source, see **Section 17.4, Interrupt Sources**.

#### LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.  
To use this module as an LIN slave, set these bits to “10<sub>B</sub>” (auto baud rate) or “11<sub>B</sub>” (fixed baud rate).

### 17.3.3.5 RLN3nLBFC — LIN Break Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 09H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LBLT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 17.35 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LBLT	Reception Break (Low-Level) Detection Width Setting 0: A break (low-level) is detected in 9.5 or 10 Tbits 1: A break (low-level) is detected in 10.5 or 11 Tbits

Configure the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### LBLT Bit (Reception Break (Low-Level) Detection Width Setting)

- When RLN3nLMD.LMD is “10<sub>B</sub>” (LIN slave mode (auto baud rate))
  - 0: Low-level width of 10 Tbits or longer is detected.
  - 1: Low-level width of 11 Tbits or longer is detected.
- When RLN3nLMD.LMD is “11<sub>B</sub>” (LIN slave mode (fixed baud rate))
  - 0: Low-level width of 9.5 Tbits or longer is detected.
  - 1: Low-level width of 10.5 Tbits or longer is detected.

### 17.3.3.6 RLN3nLSC — LIN Space Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0A<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]	—	—	IBHS[2:0]	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 17.36 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Response Space Setting b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Configure the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

This register is enabled only during response transmission, and disabled during response reception.

Some combinations of the specified values result in a frame or response length exceeding the timeout time. Specify the appropriate values in this register.

#### IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the inter-byte space width of the response transmission.

0 Tbit to 3 Tbits can be set.

#### IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the transmission width of the response space.

0 Tbit to 7 Tbits can be set.

### 17.3.3.7 RLN3nLWUP — LIN Wake-Up Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0B<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
WUTL[3:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 17.37 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low-level width of the wake-up frame transmission.  
 1 Tbit to 16 Tbits can be set.

### 17.3.3.8 RLN3nLIE — LIN Interrupt Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 17.38 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Reception Interrupt Request Enable 0: Disables successful header reception interrupt request. 1: Enables successful header reception interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Response/Wake-up Reception Interrupt Request Enable 0: Disables successful Response/wake-up reception interrupt request. 1: Enables successful Response/wake-up reception interrupt request.
0	FTCIE	Successful Response/Wake-up Transmission Interrupt Request Enable 0: Disables successful Response/wake-up transmission interrupt request. 1: Enables successful Response/wake-up transmission interrupt request.

Configure the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### SHIE Bit (Successful Header Reception Interrupt Request Enable)

The SHIE bit enables or disables an interrupt request upon successful reception of a header.

With 0 set, the interrupt request for RLIN3n successful reception is not generated when the HTRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n successful reception is generated when the HTRC flag in the RLN3nLST register is set to 1.

#### ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.

With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.

Error types that are interrupt sources are the bit error, frame/response timeout error, framing error, sync filed error, ID parity error, checksum error, and response preparation error.

Detection of the bit error, frame/response timeout error, sync filed error, ID parity error, and framing error can be enabled or disabled using the RLN3nLEDE register.

**FRCIE Bit (Successful Response/Wake-up Reception Interrupt Request Enable)**

The FRCIE bit enables or disables an interrupt request upon successful reception of a response or a wake-up frame (input signal low-level width count).

With 0 set, the interrupt request for RLIN3n successful reception is not generated when the FRC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n successful reception is generated when the FRC flag in the RLN3nLST register is set to 1.

**FTCIE Bit (Successful Response/Wake-up Transmission Interrupt Request Enable)**

The FTCIE bit enables or disables an interrupt request upon successful transmission of a response or a wake-up frame.

With 0 set, the interrupt request for RLIN3n successful transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.

With 1 set, the interrupt request for RLIN3n successful transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

### 17.3.3.9 RLN3nLEDE — LIN Error Detection Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0D<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	LTES	IPERE	—	SFERE	FERE	TERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

Table 17.39 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6	IPERE	ID Parity Error Detection Enable 0: Disables ID Parity error detection. 1: Enables ID Parity error detection.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SFERE	Sync Field Error Detection Enable 0: Disables Sync Field error detection. 1: Enables Sync Field error detection.
3	FERE	Framing Error Detection Enable *1 0: Disables framing error detection. 1: Enables framing error detection.
2	TERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable *1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1

Configure the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.

With 0 set, the timeout function applies to frame timeout.

With 1 set, the timeout function applies to response timeout.

For details on the timeout error, see **Section 17.7.7, Error Statuses**.

#### IPERE Bit (ID Parity Error Detection Enable)

This bit enables or disables detection of the ID parity error.

With 0 set, the ID parity error is not detected.

With 1 set, the ID parity error is detected.

When this bit is set to 1, the detection result is reflected in the IPER flag of the RLN3nLEST register.

For details on the ID parity error, see **Section 17.7.7, Error Statuses**.

### SFERE Bit (Sync Field Error Detection Enable)

This bit enables or disables detection of the sync field error.

With 0 set, the sync field error is not detected.

With 1 set, the sync field error is detected.

Regardless of the setting of this bit, when a sync field error is detected, this module waits for the next header.

When this bit is set to 1, the detection result is reflected in the SFER flag of the RLN3nLEST register.

For details on the sync filed error, see **Section 17.7.7, Error Statuses**.

### FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

Set this bit to 1. The detection result is indicated in the FER flag in the RLN3nLEST register.

For details on the framing error, see **Section 17.7.7, Error Statuses**.

### TERE Bit (Timeout Error Detection Enable)

The TERE bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is reflected in the TER flag of the RLN3nLEST register.

With the LTES bit, either the frame timeout error or response timeout error can be selected.

The timeout error should not be used in LIN slave mode [auto baud rate] (when the LMD[1:0] bits in the RLN3nLMD register are “10<sub>B</sub>”).

Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.

For details on the timeout error, see **Section 17.7.7, Error Statuses**.

### BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

Set this bit to 1. The detection result is indicated in the BER flag in the RLN3nLEST register.

For details on the bit error, see **Section 17.7.7, Error Statuses**.

### 17.3.3.10 RLN3nLCUC — LIN Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0E<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 17.40 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode 1: LIN operation mode
0	OM0	LIN Reset 0: LIN reset mode 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01H to transition to LIN wake-up mode or to 03H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

#### OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific LIN operation mode (LIN wake-up mode or LIN operation mode) that is entered after exiting LIN reset mode

With 0 set, the LIN/UART interface enters LIN wake-up mode.

With 1 set, the LIN/UART interface enters LIN operation mode.

This bit is enabled only when the OMM0 bit in the RLN3nLMST register is 1.

Writing a value to this bit is disabled while the FTS bit in the RLN3nLTCR register is 1.

#### OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.

With 0 set, the LIN/UART interface enters LIN reset mode.

With 1 set, the LIN/UART interface exits LIN reset mode.

### 17.3.3.11 RLN3nLTRC — LIN Transmission Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 10H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	LNRR	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

**Table 17.41 RLN3nLTRC Register Contents**

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LNRR	No LIN Response Request 0: Response for the reception ID 1: No response for the reception ID
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped. 1: Response transmission/reception is started.
0	FTS	LIN Communication Start 0: Header reception/wake-up transmission/reception is stopped. 1: Header reception/wake-up transmission/reception is started.

#### LNRR Bit (No LIN Response Request)

Set this bit to 1 if no response is to be transmitted/received after receiving the header and checking the received ID.

Once set, this bit is automatically cleared to 0 upon detection of new sync field or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 04H using the store instruction.

Do not set this bit and the RTS bit to 1 simultaneously.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, use this bit only after the completion of the header. (Do not use this bit for the second or later data group.)

#### RTS Bit (Response Transmission/Reception Start)

Set this bit to 1 to start response transmission or reception after receiving the header and checking the received ID.

Once set, this bit is automatically cleared to 0 upon completion of response transmission or reception (including error detection) or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02H to the RLN3nLTCR register using the store instruction.

Do not set this bit and the LNRR bit to 1 simultaneously

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group transmission/reception or transition to LIN reset mode.

#### **FTS Bit (LIN Communication Start)**

Set this bit to 1 to start header reception or wake-up transmission/reception.

Also set this bit to 1 to allow wake-up reception (counting of the low level width of the input signal). Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is  $0_B$  (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

### **17.3.3.12 RLN3nLMST — LIN Mode Status Register**

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RLIN3n\_base> +11<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 17.42 RLN3nLMST Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

#### **OMM1 Bit (LIN Mode Status Monitor)**

The OMM1 bit indicates the current operating mode.

#### **OMM0 Bit (LIN Reset Status Monitor)**

The OMM0 bit indicates the current operating mode.

### 17.3.3.13 RLN3nLST — LIN Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 12H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

**Table 17.43 RLN3nLST Register Contents**

Bit Position	Bit Name	Function
7	HTRC	Successful Header Reception Flag 0: Header reception has not been completed. 1: Header reception has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Response/Wake-up Reception Flag 0: Response or wake-up reception has not been completed. 1: Response or wake-up reception has been completed.
0	FTC	Successful Response/Wake-up Transmission Flag 0: Response or wake-up transmission has not been completed. 1: Response or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00H upon transition to LIN reset mode.

In LIN reset mode, writing a value to this register is disabled. In LIN reset mode, the register retains 00H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### **HTRC Flag (Successful Header Reception Flag)**

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value before 1 is written. The HTRC flag is set to 1 upon completion of header reception. Here, an interrupt request for RLIN3n successful reception is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if header reception is completed while this bit is 1, an interrupt is not generated. To clear this bit to 0, write 0 to the bit.

To detect a new header in the response field upon completion of header reception, clear this bit after it is set to 1.

#### **D1RC Flag (Successful Data 1 Reception Flag)**

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value it had before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not

generated. Write 0 to clear this bit.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

### **ERR Flag (Error Detection Flag)**

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLIN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if an error is detected while this bit is 1, an interrupt is not generated. To clear the bit to 0, write 0 to the RPER, IPER, CSER, SFER, FER, TER, and BER flags in the RLN3nLEST register. This clears the ERR flag to 0.

### **FRC Flag (Successful Response/Wake-up Reception Flag)**

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value it had before 1 is written.

The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt request for RLIN3n successful reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response reception or wake-up reception is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

### **FTC Flag (Successful Response/Wake-up Transmission Flag)**

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value it had before 1 is written.

The FTC flag is set to 1 upon completion of response or wake-up transmission. Here, an interrupt request for RLIN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response transmission or wake-up transmission is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

### 17.3.3.14 RLN3nLEST — LIN Error Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 13H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	RPER	IPER	CSER	SFER	FER	TER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 17.44 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	IPER	ID Parity Error Flag 0: ID parity error has not been detected. 1: ID parity error has been detected.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	SFER	Sync Field Error Flag 0: Sync field error has not been detected. 1: Sync field error has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	TER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value it had before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. Write 0 to clear this bit.

#### IPER Flag (ID Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written. When the IPERE bit of the RLN3nLEDE register is 1 (ID parity error detection enabled), this bit is set to 1 upon ID parity error detection. Write 0 to clear this bit.

**CSER Flag (Checksum Error Flag)**

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value it had before 1 is written.

The CSER flag is set to 1 upon checksum error detection. Write 0 to clear this bit.

**SFER Flag (Sync Field Error Flag)**

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written. When the SFERE bit of the RLN3nLEDE register is 1 (sync field error detection enabled), this bit is set to 1 upon sync field error detection. Write 0 to clear this bit.

**FER Flag (Framing Error Flag)**

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. Write 0 to clear this bit.

**TER Flag (Timeout Error Flag)**

Only 0 can be written to the TER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the TERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), this flag is set to 1 upon frame timeout error or response timeout error detection. Write 0 to clear this bit.

**BER Flag (Bit Error Flag)**

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.

When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. Write 0 to clear this bit.

### 17.3.3.15 RLN3nLDFC — LIN Data Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 14H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	LSS	—	LCS	RCDS			RFDL[3:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.45 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Data transmission/reception continues without waiting for reception of the next header.)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LCS	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RCDS	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

#### LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received. With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one. With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

When multi-byte response transmission/reception function is not used, set it to “0”.

Set this bit when the RTS bit is 0 (response transmission/reception stopped).

#### LCS Bit (Checksum Select)

The LCS bit selects the checksum mode.

With 0 set, classic checksum mode is selected.

With 1 set, enhanced checksum mode is selected.

When the timeout error detection is enabled (the TERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 17.7.7, Error Statuses.**

When the length of the response field data is 0 bytes (the RFDL bit is 0), do not set this bit to “1” (enhanced).

When response data of 9 bytes or more is to be transmitted or received, do not change the LCS bit setting after the first data group through the last data group.

During transmission or reception of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

Set this bit when the RTS bit is 0 (response transmission/reception stopped).

### **RCDS Bit (Response Field Communication Direction Select)**

This bit selects the direction of the response field/wake-up signal communication.

With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).

With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

Set this bit when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stopped) in LIN operation mode or when the FTS bit is 0 (header reception or wake-up transmission/reception stopped) in LIN wake-up mode.

When response data of 9 bytes or more is to be transmitted or received, do not change this bit setting after the first data group through the last data group.

### **RFDL[3:0] Bits (Response Field Length Select)**

The RFDL bits set the length of the response field data.

The data length can be 0 to 8 bytes excluding the checksum size.

Set these bits when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stopped).

When response data of 9 bytes or more is to be transmitted or received, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit in the RLN3nLDTC register is 1) include the checksum.

### 17.3.3.16 RLN3nLIDB — LIN ID Buffer Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 15H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0			ID[5:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.46 RLN3nLIDB Register Contents**

Bit Position	Bit Name	Function
7, 6	IDP[1:0]	Parity Setting Stores the parity bits (P0 and P1) received in the ID field.
5 to 0	ID[5:0]	ID Setting Stores the 6-bit ID value received in the ID field.

The value in the RLN3nLIDB register is enabled after the completion of header reception. In LIN mode (LIN operation mode, LIN wake-up mode), writing to this register is disabled.

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

#### IDP[1:0] Bits (Parity Setting)

The IDP bits store the parity bits (P0 and P1) received in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

When the IPERE bit in the RLN3nLEDE register is 1 (ID parity detection enabled), the received value and the value calculated internally are compared. If they do not match, IPER (ID parity error flag) is set.

#### ID[5:0] Bits (ID Setting)

The ID bits store the 6-bit ID value received in the ID field of the LIN frame.

### 17.3.3.17 RLN3nLCBR — LIN Checksum Buffer Register

**Access:** This register is a read-only register that can be read in 8-bit units. However, in LIN self-test mode, this register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 16H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
CKSM[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.47 RLN3nLCBR Register Contents**

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Stores the checksum value transmitted or received.

In LIN operation mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):  
The value transmitted can be read from the register. Read the value after transmission is completed.  
Writing to this register is invalid.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):  
The value received can be read from the register. Read the value after reception is completed.  
Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):  
After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):  
Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is stopped).

### 17.3.3.18 RLIN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

**Access:** This register can be read or written in 8-bit units.

**Address:** RLIN3nLDBR1: <RLIN3n\_base> + 18<sub>H</sub>  
 RLIN3nLDBR2: <RLIN3n\_base> + 19<sub>H</sub>  
 RLIN3nLDBR3: <RLIN3n\_base> + 1A<sub>H</sub>  
 RLIN3nLDBR4: <RLIN3n\_base> + 1B<sub>H</sub>  
 RLIN3nLDBR5: <RLIN3n\_base> + 1C<sub>H</sub>  
 RLIN3nLDBR6: <RLIN3n\_base> + 1D<sub>H</sub>  
 RLIN3nLDBR7: <RLIN3n\_base> + 1E<sub>H</sub>  
 RLIN3nLDBR8: <RLIN3n\_base> + 1F<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
LDB[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.48 RLIN3nLDBRb Register Contents**

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or holds the received data. Setting range: 00 <sub>H</sub> to FF <sub>H</sub>

- For response transmission:  
The RLIN3nLDBRb registers set the data to be transmitted in the response field.  
These registers should be set when the RTS bit in the RLIN3nLTRC register is 0 (response transmission/reception stop).
- For response reception:  
The RLIN3nLDBRb registers store the data received in the response field.  
The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.  
Do not read these registers when the RTS bit is 1 (response transmission/reception is started)

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

### 17.3.4 UART Related Registers

#### 17.3.4.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 01H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]					LPRS[2:0]		—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 17.49 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 1 0 1: 6 samplings 0 1 1 0: 7 samplings 0 1 1 1: 8 samplings 1 0 0 0: 9 samplings 1 0 0 1: 10 samplings 1 0 1 0: 11 samplings 1 0 1 1: 12 samplings 1 1 0 0: 13 samplings 1 1 0 1: 14 samplings 1 1 1 0: 15 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

#### NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate). In UART mode, it is possible to set the NSPB bits from 6 samplings to 16 samplings.

#### LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler. The LIN communication clock source is divided by this prescaler.

### 17.3.4.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

**Access:** RLN3nLBRP01 register can be read or written in 16-bit units.  
 RLN3nLBRP0 register can be read or written in 8-bit units.  
 RLN3nLBRP1 register can be read or written in 8-bit units.

**Address:** RLN3nLBRP01: <RLIN3n\_base> + 02<sub>H</sub>  
 RLN3nLBRP0: <RLIN3n\_base> + 02<sub>H</sub>  
 RLN3nLBRP1: <RLIN3n\_base> + 03<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRP[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.50 RLN3nLBRP01 Register Contents**

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000 <sub>H</sub> to FFFF <sub>H</sub>

Configure the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units using the registers RLN3nLBRP0 and RLN3nLBRP1.

### 17.3.4.3 RLN3nLMD — UART Mode Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 08H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	—	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R/W	R/W

**Table 17.51 RLN3nLMD Register Contents**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	UART Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select $b_1\ b_0$ 0 1: UART mode

Configure the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

#### LRDNFS Bit (UART Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.

With 0 set, the noise filter is enabled when receiving data.

With 1 set, the noise filter is disabled when receiving data.

#### LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.

To use the LIN/UART interface as an UART, set these bits to 01<sub>B</sub>.

#### 17.3.4.4 RLN3nLBFC — UART Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 09H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	UTPS	URPS		UPS[1:0]	USBLS	UBOS	UBLS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.52 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmission data normal output 1: Transmission data with inverted output
5	URPS	UART Input Polarity Switch 0: Reception data normal output 1: Reception data with inverted output
4, 3	UPS[1:0]	UART Parity Select 00: Parity disabled 01: Even parity 10: 0 Parity 11: Odd parity
2	USBLS	UART Stop Bit length Select 0: Stop bit: 1 bit 1: Stop bit: 2 bits
1	UBOS	UART Transfer Format Order Select 0: LSB First 1: MSB First
0	UBLS	UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication

Configure the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

##### UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.

With 0 set, transmit data is output without inversion.

With 1 set, inverted transmit data is output.

The setting of this bit is valid in all the bits of the UART frame.

In half-duplex communication, this setting should match the setting of URPS bit.

##### URPS Bit (UART Input Polarity Switch)

This bit specifies the input polarity for UART communication.

With 0 set, received data is input without inversion.

With 1 set, received data is input with inversion.

The setting of this bit is valid in all the bits of the UART frame.

In half-duplex communication, this setting should match the setting of UTPS bit.

When setting this bit to “1” and expansion bit reception ((with expansion bit comparison) or (with data

comparison)) is performed, set the inverse of the expected value to the UEBDL bit in the RLN3nLUOR1 register and RLN3nLIDB register to enable comparison of the inverted values of the received values.

### **UPS[1:0] Bits (UART Parity Select)**

Sets the UART parity.

- When these bits are set to “00<sub>B</sub>”, data is communicated without the parity.

[Transmission]

A parity bit is not added to transmit data.

[Reception]

Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to “01<sub>B</sub>”, data is communicated with the even parity.

[Transmission]

If the number of 1s in transmit data is odd, “1” is added to the parity bit. If the number of 1s in transmit data is even, “0” is added to the parity bit.

[Reception]

If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to “10<sub>B</sub>”, data is communicated with 0 parity.

[Transmission]

Regardless of the number of 1s in transmit data, “0” is added to the parity bit.

[Reception]

The value of the parity bit is not evaluated. Therefore, no parity error occurs.

- When these bits are set to “11<sub>B</sub>”, data is communicated with the odd parity.

[Transmission]

If the number of 1s in transmit data is odd, “0” is added to the parity bit. If the number of 1s in transmit data is even, “1” is added to the parity bit.

[Reception]

If the number of 1s in receive data including the parity bit is even, a parity error occurs.

### **USBLS Bit (UART Stop Bit Length Select)**

Sets the stop bit length of data for UART communication.

With 0 set, stop bit length of 1 bit is selected.

With 1 set, stop bit length of 2 bits is selected.

### **UBOS Bit (UART Transfer Format Select)**

Sets the bit order of data for UART communication.

With 0 set, LSB first is selected.

With 1 set, MSB first is selected.

### UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.

With 0 set, the character length is 8 bits.

With 1 set, the character length is 7 bits.

When the character length of one frame is 9 bits (the UEBE bit in the RLN3nLUOR1 register is 1), the setting of this bit is ignored.

#### 17.3.4.5 RLN3nLSC — UART Space Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0A<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 17.53 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Configure the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

### IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the space between the UART frames when transmitting data from the UART buffer. 0 to 3 Tbits can be set.

When data is transferred from the UART transmission data register (RLN3nLUTDR) and the UART wait transmission data register (RLN3nLUWTDR), the setting of these bits is ignored. Set these bits to “00<sub>B</sub>”.

### 17.3.4.6 RLN3nLEDE —UART Error Detection Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0DH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	OERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

**Table 17.54 RLN3nLEDE Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	OERE	Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Configure the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0B (LIN reset mode).

#### FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.

With 0 set, the framing error is not detected.

With 1 set, the framing error is detected.

When this bit is set to 1, the detection result is reflected in the FER flag of the RLN3nLEST register.

For details on the framing error, see **Section 17.8.5, Error Statuses**.

#### OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.

With 0 set, the overrun error is not detected.

With 1 set, the overrun error is detected.

When this bit is set to 1, the detection result is reflected in the OER flag of the RLN3nLEST register.

For details on the overrun error, see **Section 17.8.5, Error Statuses**.

#### BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.

With 0 set, the bit error is not detected.

With 1 set, the bit error is detected.

When this bit is set to 1, the detection result is reflected in the BER flag of the RLN3nLEST register.

In full-duplex communication, do not set this bit to “1”.

Do not set this register when the NSPB bits in the RLN3nLWBR register are 0101B (6 samplings) and the LRDNFS bit in the RLN3nLMD register is 0 (noise filtering is enabled).

For details on the bit error, see **Section 17.8.5, Error Statuses**.

#### 17.3.4.7 RLN3nLCUC — UART Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 0E<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

**Table 17.55 RLN3nLCUC Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode 1: LIN reset mode is canceled.

After a value is written to this register, confirm that the value written is reflected in the RLN3nLMST register before writing another value.

##### OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.

With 0 set, reset mode.

With 1 set, reset mode is canceled.

### 17.3.4.8 RLN3nLTRC — UART Transmission Control Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 10H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R

Table 17.56 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

#### RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.

Only 1 can be written to this bit; 0 cannot be written.

Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enable) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).

Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDTC register. This bit is also automatically cleared to 0 upon transition to reset mode.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0B (LIN reset mode).

When writing 1 to this bit while the UTSW bit in the RLN3nLRFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the reception of stop bit is completed), write only during the reception of stop bit.

### 17.3.4.9 RLN3nLMST — UART Mode Status Register

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RLIN3n\_base> + 11<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 17.57 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

#### OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

### 17.3.4.10 RLN3nLST — UART Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 12H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	URS	UTS	ERR	—	—	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 17.58 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	URS	Reception Status Flag 0: Reception is stopped. 1: Reception is started.
4	UTS	Transmission Status Flag 0: Transmission is stopped. 1: Transmission is started.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FTC	Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed.

The RLN3nLST register is automatically cleared to “00H” upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains “00H”. To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

#### URS Flag (Reception Status Flag)

At the start of the reception, this flag is set to 1.

The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.

The reception is ended under the following conditions.

- Sampling point of the first bit of the stop bits

#### UTS Flag (Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.

The transmission is started under the following conditions.

- When transmission data is specified in the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the completion of transmission.

The transmission is ended under the following conditions.

- When transmission of data specified in the RLN3nLUTDR or RLN3nLUWTDR register is completed and next data is not specified
- When transmission from UART buffer is completed (when the RTS bit in the RLN3nLTRC register is cleared to 0)

### **ERR Flag (Error Detection Flag)**

This flag is set to 1 upon detection of an error, detection of an expansion bit, or when ID's match (when at least one of the flags of the RLN3nLEST register is 1). Here, an interrupt request for RLIN3n status is generated. However, if an error or expansion bit is detected or ID's match while this bit is 1, an interrupt is not generated. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register.

### **FTC Flag (Successful UART Buffer Transmission Flag)**

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that it was before 1 was written.

Regardless of errors, this bit is set to 1 upon completion of transmission of the number of data specified by the MDL bit in the RLN3nLDTC register from the UART buffer. Here, an interrupt request for RLIN3n transmission is generated. Write 0 to clear this flag.

### 17.3.4.11 RLN3nLEST — UART Error Status Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 13H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	UPER	IDMT	EXBT	FER	OER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 17.59 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UPER	Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected.
5	IDMT	ID Match Flag 0: The received data does not match the ID value. 1: The received data matches the ID value.
4	EXBT	Expansion Bit Detection Flag 0: Expansion bit has not been detected. 1: Expansion bit has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	OER	Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00H upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00H is retained. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

#### UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

This flag is set to 1 upon parity error detection. Write 0 to clear this flag.

#### IDMT Flag (ID Match Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.

The IDMT flag is set to 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)
- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)
- The UEBDCE bit in the RLN3nLUOR1 register is 1 (expansion bit/data comparison enabled)

- The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register match.
- The 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register match.

Write 0 to clear this flag.

#### **EXBT Flag (Expansion Bit Detection Flag)**

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written. When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches with the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1. Write 0 to clear this flag.

#### **FER Flag (Framing Error Flag)**

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written. The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled). Write 0 to clear this flag.

#### **OER Flag (Overrun Error Flag)**

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written. The OER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). Write 0 to clear this flag.

#### **BER Flag (Bit Error Flag)**

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.

The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).

Write 0 to clear this flag.

### 17.3.4.12 RLN3nLDFC — UART Data Field Configuration Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 14H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	UTSW	—			MDL[3:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 17.60 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	UTSW	Transmission Start Wait 0: When UART buffer transmission is requested, transmission is started immediately. 1: When UART buffer transmission is requested, transmission is not started until reception of the stop bit is completed.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MDL[3:0]	UART Buffer Data Length Select b3    b0 0 0 0 0: 9 data 0 0 0 1: 1 data 0 0 1 0: 2 data 0 0 1 1: 3 data 0 1 0 0: 4 data 0 1 0 1: 5 data 0 1 1 0: 6 data 0 1 1 1: 7 data 1 0 0 0: 8 data 1 0 0 1: 9 data  Settings other than the above are prohibited.

#### UTSW Bit (Transmission Start Wait)

This bit controls the transmission start timing of UART buffer.

With 0 set, transmission is started as soon as the start of UART buffer transmission is requested.

With 1 set, transmission is started after the completion of the stop bit reception.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

This bit is enabled when the RTS bit in the RLN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).

Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

#### MDL[3:0] Bits (UART Buffer Data Length Select)

These bits specify the data length of the UART buffer.

Writing a value to these bits is disabled when the RTS bit in the RLN3nLTRC register is 1 (UART buffer transmission started).

### 17.3.4.13 RLN3nLIDB — UART ID Buffer Register

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RLIN3n_base> + 15H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.61 RLN3nLIDB Register Contents**

Bit Position	Bit Name	Function
7 to 0	ID[7:0]	Specifies the ID value that is referred in expansion bit data comparison

#### ID Bit (ID Bit)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (data comparison after expansion bit is detected), set the value to be compared with the received data. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (reception is stopped).

### 17.3.4.14 RLN3nLUDB0 — UART Data Buffer 0 Register

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RLIN3n_base> + 17H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	UDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.62 RLN3nLUDB0 Register Contents**

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Sets the data to be transmitted. Setting range: 00H to FFH

If the data length selection corresponds to 9 data bytes (RLN3nLDFC.MDL bit is “0H” or “9H”) for multi-byte UART transmission, then the first data value for UART communication is present in this buffer.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

**Table 17.63, Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format**, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 17.8.1.2, UART Buffer Transmission, (1) UART Buffer Transmission**.

**Table 17.63 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format**

	RLN3nLUDB0							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

**17.3.4.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)**

**Access:** This register can be read or written in 8-bit units.

**Address:** RLN3nLDBR1: <RLIN3n\_base> +18<sub>H</sub>  
 RLN3nLDBR2: <RLIN3n\_base> +19<sub>H</sub>  
 RLN3nLDBR3: <RLIN3n\_base> +1A<sub>H</sub>  
 RLN3nLDBR4: <RLIN3n\_base> +1B<sub>H</sub>  
 RLN3nLDBR5: <RLIN3n\_base> +1C<sub>H</sub>  
 RLN3nLDBR6: <RLIN3n\_base> +1D<sub>H</sub>  
 RLN3nLDBR7: <RLIN3n\_base> +1E<sub>H</sub>  
 RLN3nLDBR8: <RLIN3n\_base> +1F<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
LDB[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17.64 RLN3nLDBRb Register Contents**

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Specifies the data to be transmitted. Setting range: 00 <sub>H</sub> to FF <sub>H</sub>

These registers specify the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

**Table 17.65, Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format**, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 17.8.1.2, UART Buffer Transmission, (1) UART Buffer Transmission**.

**Table 17.65 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format**

	RLN3nLDBRb							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

### 17.3.4.16 RLN3nLUOER — UART Operation Enable Register

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 20H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UROE	UTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 17.66 RLN3nLUOER Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	UROE	Reception Enable 0: Disables reception. 1: Enables reception.
0	UTOE	Transmission Enable 0: Disables transmission. 1: Enables transmission.

The RLN3nLUOER register is automatically cleared to 00H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to.

In LIN reset mode, the register retains 00H.

#### UROE Bit (Reception Enable)

The UROE bit enables or disables reception.

With 0 set, reception is disabled.

With 1 set, reception is enabled.

Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. Note that transmission is also aborted.

Do not set this bit to 1 when data transmission from the UART buffer is in progress.

#### UTOE Bit (Transmission Enable)

The UTOE bit enables or disables transmission.

With 0 set, transmission is disabled.

With 1 set, transmission is enabled.

Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode.

Note that reception is also aborted.

### 17.3.4.17 RLN3nLUOR1 — UART Option Register 1

**Access:** This register can be read or written in 8-bit units.

**Address:** <RLIN3n\_base> + 21H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 17.67 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UECD	Expansion Bit Comparison Disable 0: Enables expansion bit comparison. 1: Disables expansion bit comparison.
3	UTIGTS	Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.
2	UEBDCE	Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected.
1	UEBDL	Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.
0	UEBE	Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation.

#### UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).

With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.

With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0B (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

Do not set this bit to 1 when the UEBDCE bit is 1 (expansion bit/data comparison enable).

#### UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit selects the generation timing of the transmission interrupt.

With 0 set, the transmission interrupt is generated at the start of transmission.

With 1 set, the transmission interrupt is generated at the completion of transmission.

When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDFC register).

When transmission from the UART buffer is performed with 1 set, the transmission interrupt is

generated only at the completion of the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDFC register).

#### **UEBDCE Bit (Expansion Bit Data Comparison Enable)**

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.

With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.

With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).

Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).

Do not set this bit to 1 when the UART buffer is used.

#### **UEBDL Bit (Expansion Bit Detection Level Select)**

The UEBDL bit selects the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (expansion bit comparison enabled).

With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.

With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

#### **UEBE Bit (Expansion Bit Enable Bit)**

The UEBE bit enables or disables expansion bit operation.

With 0 set, expansion bit operation is disabled.

With 1 set, expansion bit operation is enabled.

Set this bit when the OMM0 bit of the RLN3nLMST register is 0<sub>B</sub> (LIN reset mode).

Do not set this bit to 1 when the UART buffer is used.

### 17.3.4.18 RLN3nLUTDR — UART Transmission Data Register

**Access:** RLN3nLUTDR register can be read or written in 16-bit units.  
 RLN3nLUTDRL register can be read or written in 8-bit units.  
 RLN3nLUTDRH register can be read or written in 8-bit units.

**Address:** RLN3nLUTDR: <RLIN3n\_base> + 24<sub>H</sub>  
 RLN3nLUTDRL: <RLIN3n\_base> + 24<sub>H</sub>  
 RLN3nLUTDRH: <RLIN3n\_base> + 25<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.68 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UTD[8:0]	Specifies the data to be transmitted. Setting range: 000 <sub>H</sub> to 1FF <sub>H</sub>

The RLN3nLUTDR register specifies the data to be transmitted from the transmission data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 17.69 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

	RLN3nLUTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

### 17.3.4.19 RLN3nLURDR — UART Reception Data Register

**Access:** RLN3nLURDR register is a read-only register that can be read in 16-bit units.  
 RLN3nLURDRL register is a read-only register that can be read in 8-bit units.  
 RLN3nLURDRH register is a read-only register that can be read in 8-bit units.

**Address:** RLN3nLURDR: <RLIN3n\_base> + 26<sub>H</sub>  
 RLN3nLURDRL: <RLIN3n\_base> + 26<sub>H</sub>  
 RLN3nLURDRH: <RLIN3n\_base> + 27<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
	—	—	—	—	—	—	—	URD [8:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R						

Table 17.70 RLN3nLURDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned.
8 to 0	URD [8:0]	Stores the received data Setting range: 000 <sub>H</sub> to 1FF <sub>H</sub>

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the received data is stored in this register and can be read out.

This register is updated upon reception of stop bit in the received data.

This register is also updated even when an error is caused by the parity or stop bit.

However, the value of this register is not updated if an overrun error occurs when the OERE bit of the RLN3nLEDE register is 1 (overrun detection enabled). The value of this register is updated even if an overrun error occurs when the OERE bit is 0 (overrun detection disabled).

Read this register upon occurrence of a reception error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8-bit units.

However, when expansion bit is used (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 17.71 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

	RLN3nLURDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

### 17.3.4.20 RLN3nLUWTDR — UART Wait Transmission Data Register

**Access:** RLN3nLUWTDR register can be read or written in 16-bit units.  
 RLN3nLUWTDRL register can be read or written in 8-bit units.  
 RLN3nLUWTDRH register can be read or written in 8-bit units.

**Address:** RLN3nLUWTDR: <RLIN3n\_base> + 28<sub>H</sub>  
 RLN3nLUWTDRL: <RLIN3n\_base> + 28<sub>H</sub>  
 RLN3nLUWTDRH: <RLIN3n\_base> + 29<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
	—	—	—	—	—	—	—	UWTD[8:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							

Table 17.72 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UWTD[8:0]	Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000 <sub>H</sub> to 1FF <sub>H</sub>

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

Also, write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the value of the RLN3nLUTDR register is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 17.73 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

	RLN3nLUWTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

## 17.4 Interrupt Sources

The LIN/UART interface generates four types of interrupt requests.

- RLIN3n transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt
- RLIN3n interrupt

Setting the LIOS bit in the RLN3nLMD register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the RLIN3n interrupt.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt request.

**Table 17.74** lists the sources for each interrupt.

**Table 17.74 Interrupt Sources**

		LIOS bit in RLN3nLMD register is 0	LIOS bit in RLN3nLMD register is 1 <sup>1</sup>		
		RLIN3n Interrupt	RLIN3n Transmission Interrupt	RLIN3n Successful Reception Interrupt	RLIN3n Status Interrupt
LIN mode	LIN master mode	<ul style="list-style-type: none"> <li>• Successful frame transmission</li> <li>• Successful frame reception</li> <li>• Successful wake-up transmission</li> <li>• Successful wake-up reception</li> <li>• Successful header transmission</li> <li>• Bit error</li> <li>• Physical bus error</li> <li>• Frame/response timeout error</li> <li>• Framing error</li> <li>• Checksum error</li> <li>• Response preparation error</li> </ul>	<ul style="list-style-type: none"> <li>• Successful frame transmission</li> <li>• Successful wake-up transmission</li> <li>• Successful header transmission</li> </ul>	<ul style="list-style-type: none"> <li>• Successful wake-up reception</li> <li>• Successful wake-up transmission</li> </ul>	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Physical bus error</li> <li>• Frame/response timeout error</li> <li>• Framing error</li> <li>• Checksum error</li> <li>• Response preparation error</li> </ul>
	LIN slave mode	<ul style="list-style-type: none"> <li>• Successful response transmission</li> <li>• Successful response reception</li> <li>• Successful wake-up transmission</li> <li>• Successful wake-up reception</li> <li>• Successful header reception</li> <li>• Bit error</li> <li>• Frame/response timeout error</li> <li>• Framing error</li> <li>• Sync field error</li> <li>• Checksum error</li> <li>• ID parity error</li> <li>• Response preparation error</li> </ul>	<ul style="list-style-type: none"> <li>• Successful response transmission</li> <li>• Successful wake-up transmission</li> </ul>	<ul style="list-style-type: none"> <li>• Successful response reception</li> <li>• Successful wake-up reception</li> <li>• Successful header reception</li> </ul>	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Frame/response timeout error</li> <li>• Framing error</li> <li>• Sync field error</li> <li>• Checksum error</li> <li>• ID parity error</li> <li>• Response preparation error</li> </ul>
UART mode		—	<ul style="list-style-type: none"> <li>• Transmission start/ successful transmission</li> </ul>	<ul style="list-style-type: none"> <li>• Successful reception</li> <li>• Expansion bit mismatch</li> </ul>	<ul style="list-style-type: none"> <li>• Bit error</li> <li>• Overrun error</li> <li>• Framing error</li> <li>• Expansion bit match</li> <li>• ID match</li> <li>• Parity error</li> </ul>

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

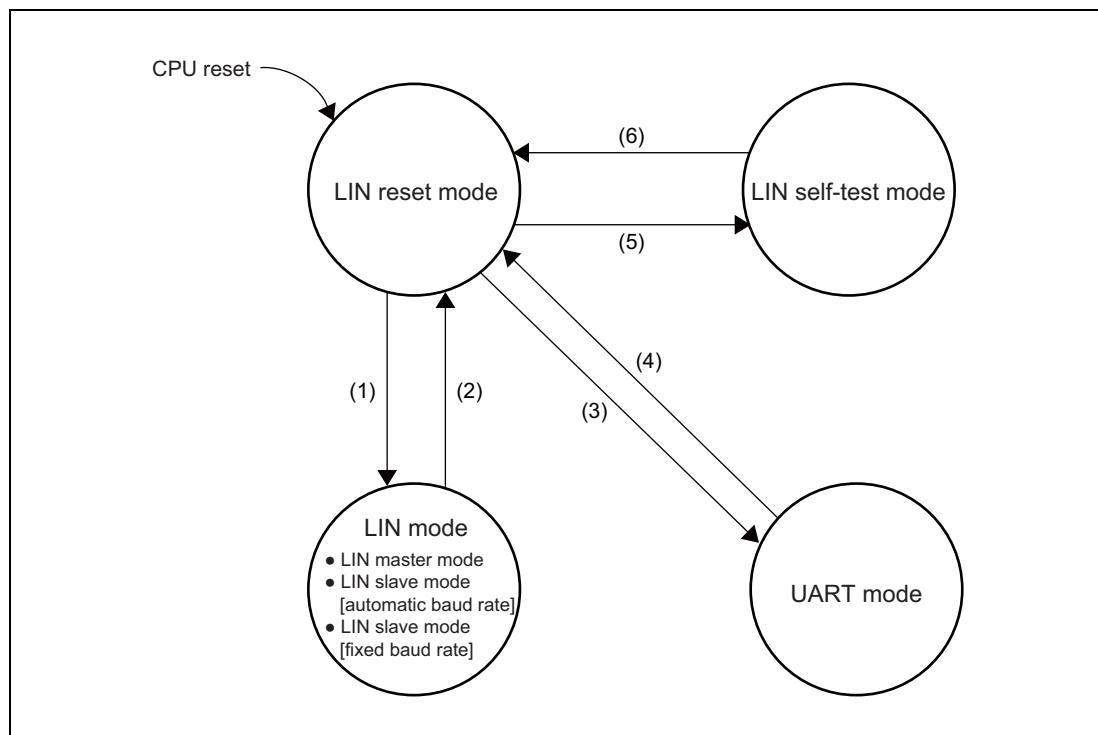
In LIN mode, each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is set to 1.

## 17.5 Modes

The LIN/UART interface provides the following four modes, depending on the specific function to be performed:

- LIN reset mode
- LIN mode
  - LIN master mode
  - LIN slave mode [auto baud rate]
  - LIN slave mode [fixed baud rate]
- UART mode
- LIN self-test mode

**Figure 17.2** shows mode transitions. **Table 17.79** describes mode transition conditions. **Table 17.79** lists operations available in each mode.



**Figure 17.2 Mode Transitions**

**Table 17.75 Transition Condition of Each Mode**

Mode Transition		Transition Condition
(1) LIN reset mode	→ LIN mode	<ul style="list-style-type: none"> <li>• LIN master mode</li> <li>• LIN slave mode [auto baud rate]</li> <li>• LIN slave mode [fixed baud rate]</li> </ul> LMD bits in RLN3nLMD register = $00_B$ and OM1 and OM0 bits in RLN3nLCUC register = $01_B$ or $11_B$ <ul style="list-style-type: none"> <li>• LMD bits in RLN3nLMD register = <math>11_B</math> and OM1 and OM0 bits in RLN3nLCUC register = <math>01_B</math> or <math>11_B</math></li> <li>• LMD bits in RLN3nLMD register = <math>10_B</math> and OM1 and OM0 bits of RLN3nLCUC register = <math>01_B</math> or <math>11_B</math></li> </ul>
(2) LIN mode	→ LIN reset mode	OM0 bit in RLN3nLCUC register = $0_B$
(3) LIN reset mode	→ UART mode	LMD bits in RLN3nLMD register = $01_B$ and OM0 bit in RLN3nLCUC register = $1_B$
(4) UART mode	→ LIN reset mode	OM0 bit in RLN3nLCUC register = $0_B$
(5) LIN reset mode	→ LIN self-test mode	See <b>Section 17.9, LIN Self-Test Mode.</b>
(6) LIN self-test mode	→ LIN reset mode	See <b>Section 17.9, LIN Self-Test Mode.</b>

**Table 17.76 Operations Available in Each Mode**

LIN Mode		UART Mode	LIN Self-Test Mode
LIN Master Mode	LIN Slave Mode [auto baud rate] LIN Slave Mode [fixed baud rate]		
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	Header reception Response transmission Response reception Wake-up transmission Wake-up reception Error detection	UART transmission UART reception Error detection	Self-test

Whether mode has transitioned to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register and the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode.**

## 17.6 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The transition to LIN reset mode can be verified by checking that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode transitions to LIN reset mode, the following registers are initialized to their reset values and they retain their initial values while in LIN reset mode:

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register ( $b = 1$  to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTDR register

## 17.7 LIN Mode

LIN mode can operate in the following submodes: LIN master mode, LIN slave mode [auto baud rate], and LIN slave mode [fixed baud rate].

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to  $00_B$  (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either  $01_B$  or  $11_B$  causes the transition to LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either  $01_B$  to  $11_B$ .

In LIN slave mode [auto baud rate] and LIN slave mode [fixed baud rate], header reception, response transmission, response reception, wake-up transmission, wake-up reception, and error detection can be performed.

The LIN slave mode [auto baud rate] allows automatic detection of the break field and the sync field, and sets a baud rate based on the results of measurement of a sync field. The baud rate can be set to 1 kbps to 20 kbps.

Set the LPRS[2:0] bits in the RLN3nLWBR register so that the prescaler clock (the clock obtained by dividing the frequency of the LIN communication clock source by the prescaler) is configured as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz <sup>*1</sup>
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

**Note 1.** Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to “ $0011_B$ ” (4 samplings).

LIN slave mode [fixed baud rate] allows automatic detection of the break field, the sync field, and the ID field at a baud rate that is preset by the baud rate generator.

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to  $10_B$  (LIN slave mode [auto baud rate]) and setting the OM1 and OM0 bits in the RLN3nLCUC register to  $01_B$  or  $11_B$  causes the transition to LIN slave mode [auto baud rate]; and setting the LMD bits in the RLN3nLMD register to  $11_B$  (LIN slave mode [fixed baud rate]), and setting the OM1 and OM0 bits in the RLN3nLCUC register to  $01_B$  or  $11_B$  causes the transition to LIN slave mode [fixed baud rate], turning the OMM1 and OMM0 bits in the RLN3nLMST register to  $01_B$  or  $11_B$ .

When transitioning from one submode to another submode within LIN mode, transition to LIN test mode first and change the LMD bits in the RLN3nLMD register.

The LIN mode provides the following two operating modes:

- LIN operation mode
- LIN wake-up mode

**Figure 17.3** shows the transition of operating modes. **Table 17.77** describes the transition conditions of operating modes.

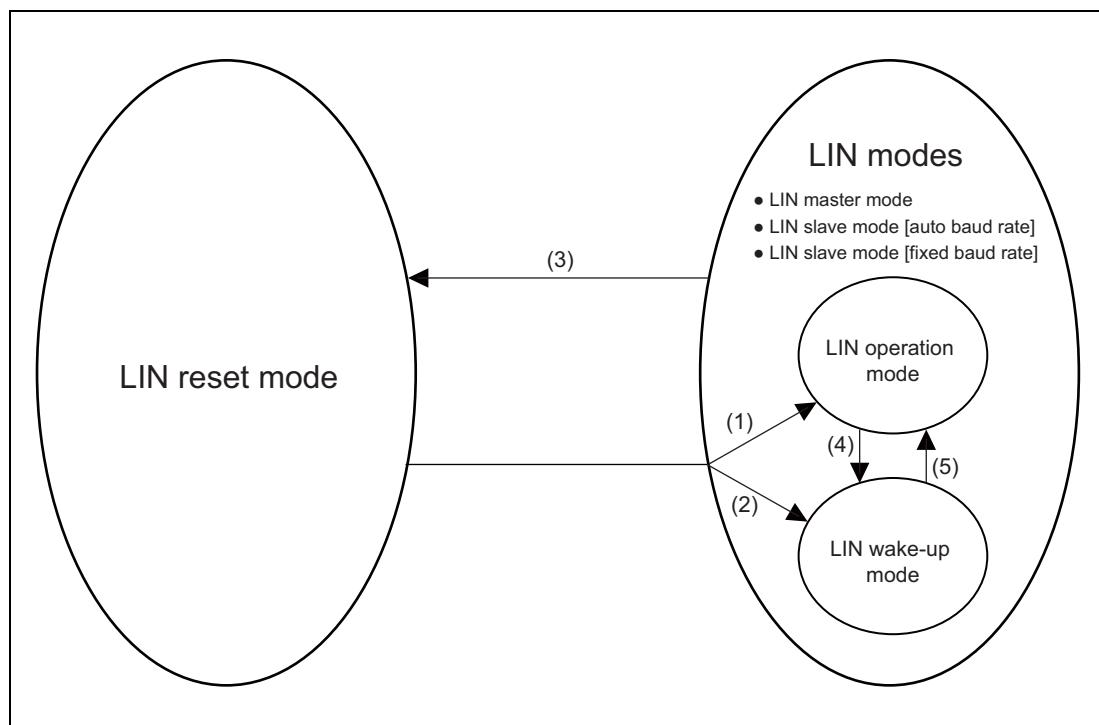


Figure 17.3 Transition of Operating Modes

Table 17.77 Transition Conditions for Operating Modes

Operation Mode Transition		Transition Condition
(1)	LIN reset mode → LIN mode • LIN operation mode	LMD bits in RLN3nLMD register = $00_B$ or $10_B$ or $11_B$ and OM1 and OM0 bits in RLN3nLCUC register = $11_B$
(2)	LIN reset mode → LIN mode • LIN wake-up mode	LMD bits in RLN3nLMD register = $00_B$ or $10_B$ or $11_B$ and OM1 and OM0 bits in RLN3nLCUC register = $01_B$
(3)	LIN mode → LIN reset mode • LIN operation mode • LIN wake-up mode	OM0 bit in RLN3nLCUC register = $0_B$
(4) *1	LIN mode → LIN mode • LIN operation mode • LIN wake-up mode	OM1 and OM0 bits in RLN3nLCUC register = $01_B$
(5) *1	LIN mode → LIN mode • LIN wake-up mode • LIN operation mode	OM1 and OM0 bits in RLN3nLCUC register = $11_B$

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is in progress (when the FTS bit in the RLN3nLTRC register is 1).

### (1) LIN Operation Mode

While in LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to  $11_B$  switches the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to  $11_B$ . Communication settings should be configured after the OMM1 and OMM0 bits have become  $11_B$ .

### (2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to  $01_B$  switches the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to  $01_B$ . Communication settings should be configured after the OMM1 and OMM0 bits have become  $01_B$ .

## 17.7.1 LIN Master Mode

### 17.7.1.1 Header Transmission

**Figure 17.4** shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 17.78** provides processing in header transmission.

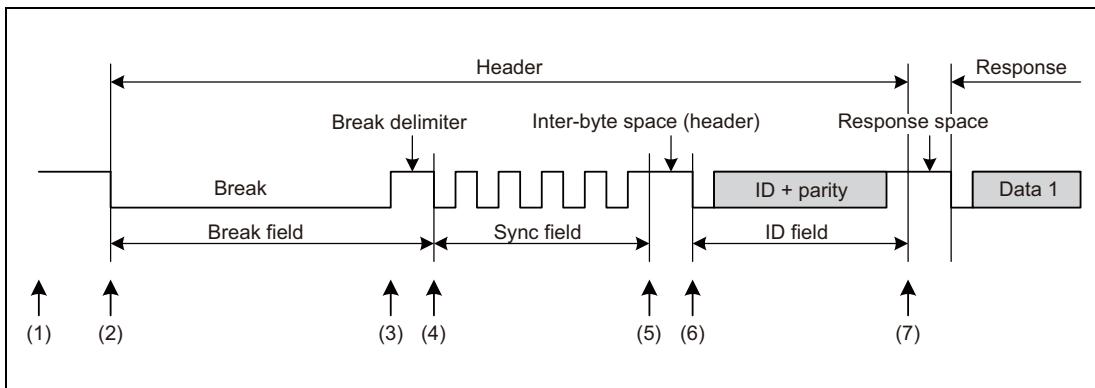


Figure 17.4 Operation in Header Transmission

Table 17.78 Processing in Header Transmission

Software Processing	LIN/UART Interface Processing
(1) • Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Transitions to LIN master mode: LIN operation mode • Configures frame information to be transmitted (ID, parity, data length, response direction, Checksum method, and transmission data)	Waits for the FTS bit in the RLN3nLTRC register to be set by software (idle)
(2) Sets the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3) Waits for an interrupt request	Transmits a break delimiter.
(4)	Transmits a sync field ( $55_{\text{H}}$ ).
(5)	Transmits an inter-byte space (header).
(6)	Transmits an ID field.
(7)	Sets the successful header transmission flag.

#### NOTE

For information about error detection conditions, see **Section 17.7.7, Error Statuses**.

### 17.7.1.2 Response Transmission

**Figure 17.5** shows the operation of the LIN/UART interface (LIN master mode) in response transmission. **Table 17.79** provides processing in response transmission.

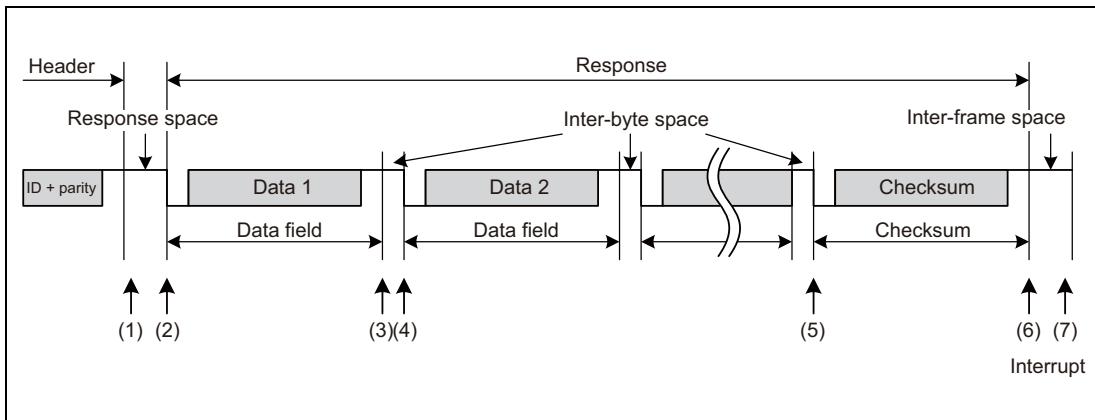


Figure 17.5 Operation in Response Transmission

Table 17.79 Processing in Response Transmission

Software Processing	LIN/UART Interface Processing
(1) [When in frame separate mode] <ul style="list-style-type: none"> <li>Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started)</li> </ul> [When not in frame separate mode] <ul style="list-style-type: none"> <li>Waits for an interrupt request</li> </ul>	[When in frame separate mode] <ul style="list-style-type: none"> <li>Waits for the RTS bit in the RLN3nLTRC register to be set to 1 by software.</li> <li>When the bit is set to 1, transmits a response space.</li> </ul> [When not in frame separate mode] <ul style="list-style-type: none"> <li>Transmits a response space.</li> </ul>
(2) Waits for an interrupt request	Transmits data 1.
(3)	Transmits an inter-byte space.
(4)	<ul style="list-style-type: none"> <li>Transmits data 2.</li> <li>Transmits an inter-byte space</li> <li>Transmits data 3.</li> <li>Transmits an inter-byte space</li> </ul> <p>(Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register).</p> <p>:</p> <p>:</p>
(5)	Transmits the checksum.
(6)	<ul style="list-style-type: none"> <li>Sets the successful frame/wake-up transmission flag.</li> <li>Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped)</li> </ul> [When in frame separate mode] <ul style="list-style-type: none"> <li>Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception is stopped).</li> </ul>
(7) <ul style="list-style-type: none"> <li>Processing after communication</li> <li>Checks the RLN3nLST register, and clears flags.</li> </ul>	Idle

#### NOTE

For information about error detection conditions, see **Section 17.7.7, Error Statuses**.

### 17.7.1.3 Response Reception

**Figure 17.6** shows the operation of the LIN/UART interface (LIN master mode) in response reception. **Table 17.80** provides processing in response reception.

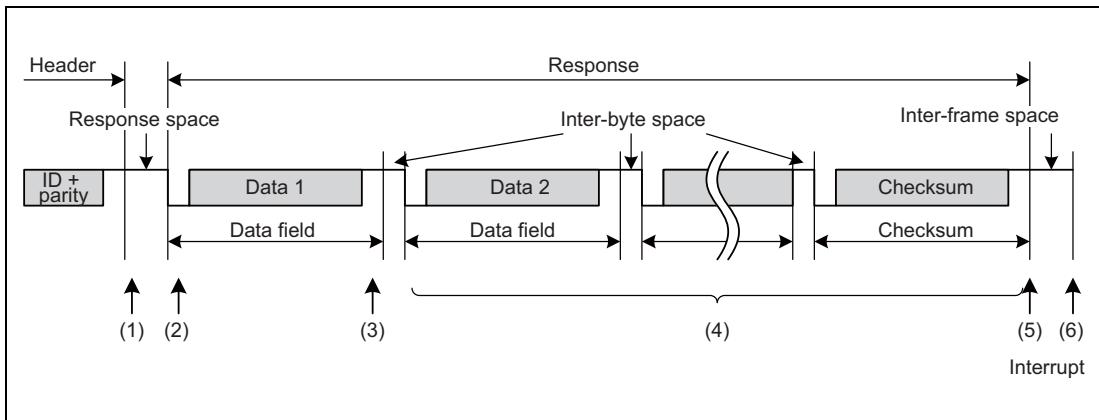


Figure 17.6 Operation in Response Reception

Table 17.80 Processing in Response Reception

Software Processing	LIN/UART Interface Processing
(1) Waits for an interrupt request (no processing)	Waits for detection of a start bit.
(2)	Receives data 1 when the start bit is detected.
(3)	Sets the successful data 1 reception flag.
(4)	<ul style="list-style-type: none"> <li>Receives data 2 when the start bit is detected.</li> <li>Receives data 3 when the start bit is detected.</li> </ul> <p>Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register).</p> <p style="text-align: center;">⋮</p> <ul style="list-style-type: none"> <li>Receives the checksum when the start bit is detected.</li> </ul>
(5)	<ul style="list-style-type: none"> <li>Determines the checksum.</li> <li>Sets the successful frame/wake-up reception flag.</li> <li>Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped).</li> </ul>
(6)	<ul style="list-style-type: none"> <li>Processing after communication</li> <li>Reads the received data.</li> <li>Checks the RLN3nLST register, and clears flags.</li> </ul>

#### NOTE

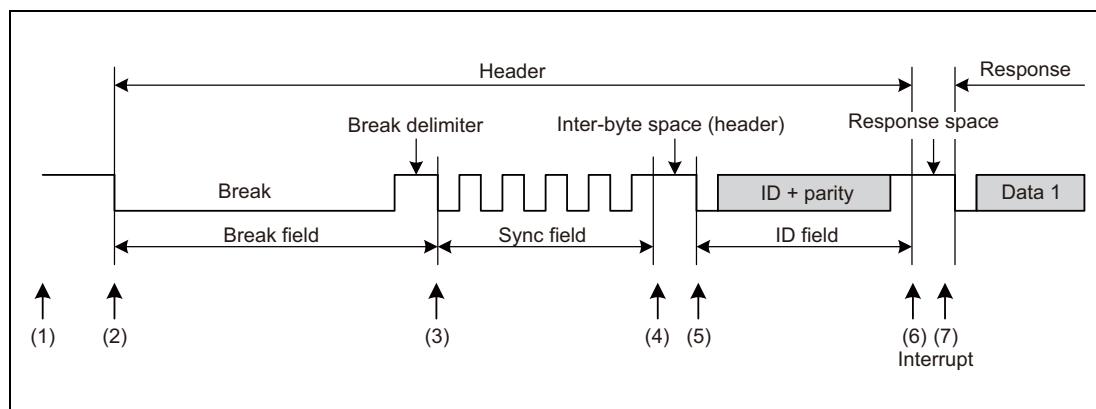
For information about error detection, see **Section 17.7.7, Error Statuses**.

## 17.7.2 LIN Slave Mode

### 17.7.2.1 Header Reception

**Figure 17.7** shows the operation of the LIN/UART interface (LIN slave mode) in header reception.

**Table 17.81** provides processing in header reception.



**Figure 17.7** Operation in Header Reception

**Table 17.81** Processing in Header Reception

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> <li>• Sets a baud rate</li> <li>• Sets noise filter ON/OFF</li> <li>• Enables interrupt</li> <li>• Enables error detection</li> <li>• Sets frame configuration parameters</li> <li>• Transitions to the LIN slave mode: LIN operation mode</li> <li>• Sets the FTS bit in the RLN3nLTCR register to 1 (header reception or wake-up transmission/reception started)</li> </ul>	Waits for the FTS bit in the RLN3nLTCR register to be set by software.
(2) Waits for an interrupt request.	Waits for detection of break field
(3)	Detects a break field (LIN slave mode [fixed baud rate]). For details about the break field detection timing in the case of LIN slave mode [auto baud rate], see [Auto Baud Rate Correction Function].)
(4)	<ul style="list-style-type: none"> <li>• Detects a sync field (<math>55_{H}</math>)</li> <li>• Sets the baud rate generator (in the case of LIN slave mode [auto baud rate])</li> <li>• Clears the no-response request bit (LNRR bit).</li> </ul>
(5)	<ul style="list-style-type: none"> <li>• Receives an ID field.</li> <li>• Checks the ID parity bit</li> </ul>
(6)	Sets the header reception complete flag.
(7) <ul style="list-style-type: none"> <li>• Checks the RLN3nLST register, and clears flags.</li> <li>• Checks the RLN3nLIDB register, and prepares a response.</li> </ul>	<ul style="list-style-type: none"> <li>• Completes a header reception process.</li> <li>• Waits for a response request.</li> </ul>

**NOTE**

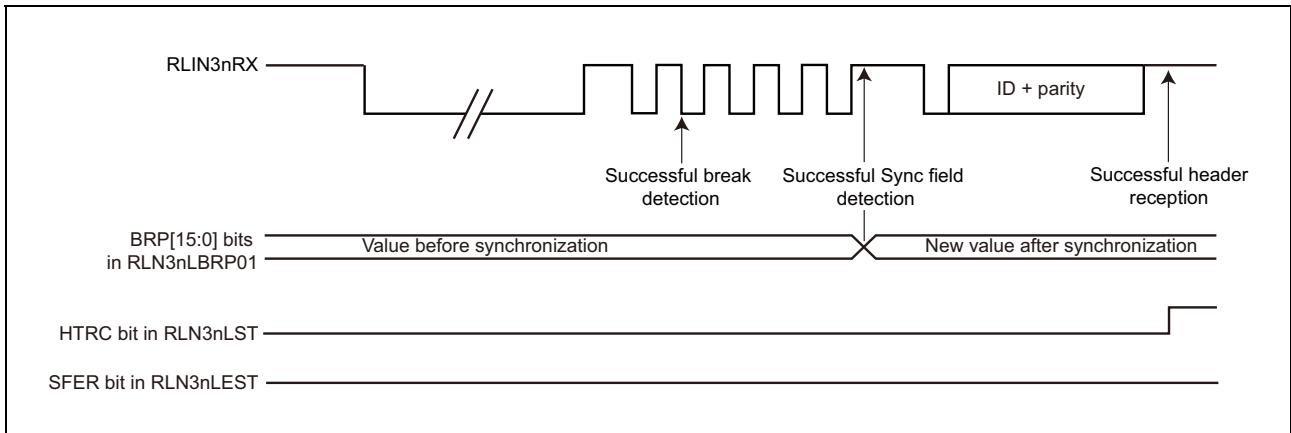
The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the subsequent Sync field and ID field) continues regardless of whether an error occurred. For information about error detection conditions, see **Section 17.7.7, Error Statuses**.

**[Auto Baud Rate Correction Function]**

In LIN slave mode [auto baud rate], the system always measures the low-level widths that are received. If the first “Low level” width is 10 times (if the BLT bit of the RLN3nLBFC register is “0”) or 11 times (if the BLT bit of the RLN3nLBFC register is “1”) or greater than the bit width calculated from the average of the starting 2 bits (the period of the consecutive falling edges from the beginning of the sync field) of the sync field, the system concludes that the detection of break field was successful and verifies that the data in the sync field is  $55_{\text{H}}$ . If the data in the sync field is indeed  $55_{\text{H}}$  and the system determines that sync field reception was successful, the system automatically sets the baud rate correction result to the RLN3nLBRP01 register.

If data is received up to the ID field without error, a successful header reception interrupt is generated at the stop bit position.

On the other hand, if the data in the sync field is not  $55_{\text{H}}$  and the system determines that sync field reception failed, the system sets the sync field error flag and an error interrupt is generated. In that case, baud rate correction is not performed and the LIN/UART interface waits for the detection of the next break field (low level).



**Figure 17.8 Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)**

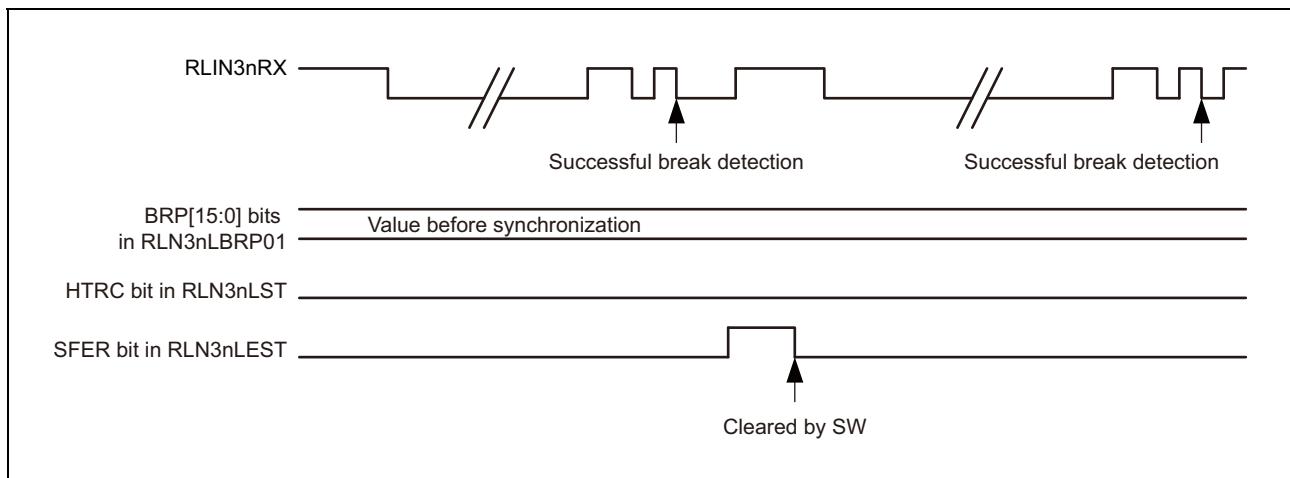


Figure 17.9 Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)

### 17.7.2.2 Response Transmission

**Figure 17.10** shows the operation of the LIN/UART interface (LIN slave mode) in response transmission. **Table 17.82** provides processing in response transmission.

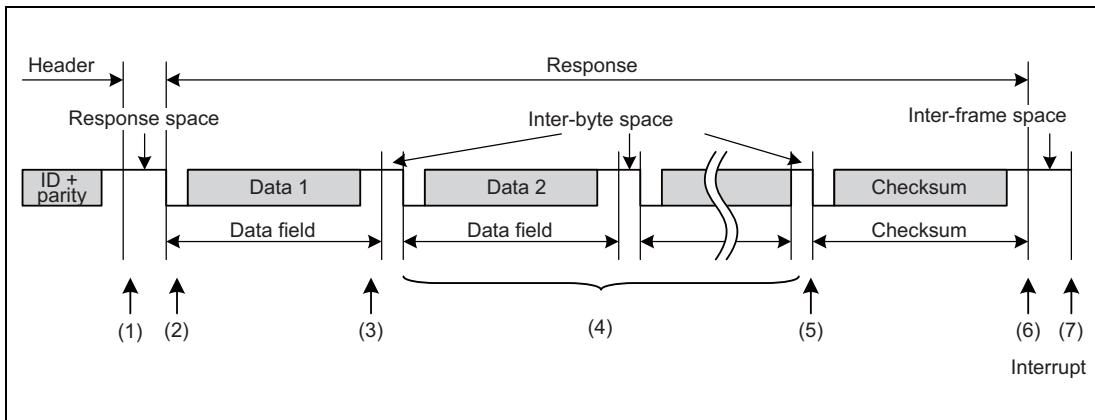


Figure 17.10 Operation in Response Transmission

Table 17.82 Processing in Response Transmission

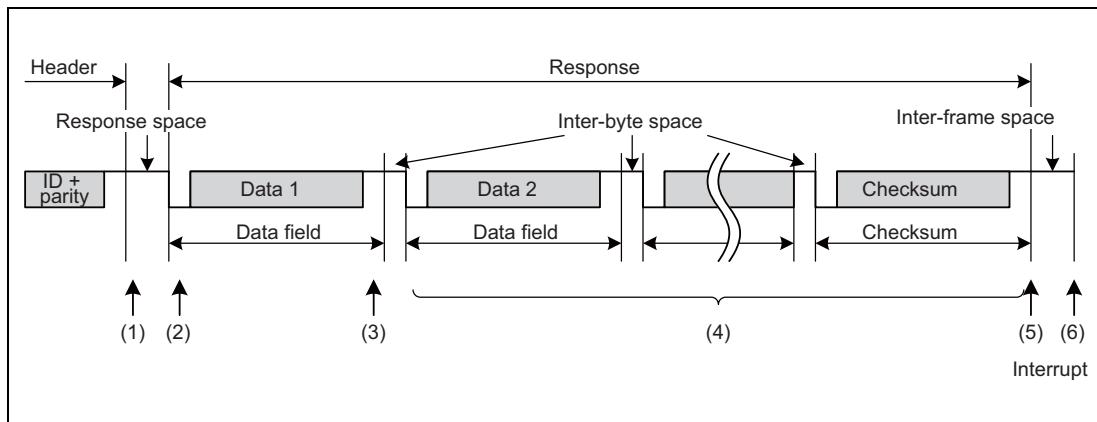
Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> <li>Configures the RLN3nLDTC register.</li> <li>Configures the RLN3nLDBRb registers.(b = 1 to 8)</li> <li>Sets the RTS bit in the RLN3nLTTC register to 1 (response transmission/reception started)</li> </ul>	<ul style="list-style-type: none"> <li>Waits for the RTS or LNRR bit in the RLN3nLTTC register to be set by software</li> <li>Transmits the response space after the RTS bit of the RLN3nLTTC register is set to 1</li> </ul>
(2) Waits for an interrupt request.	Transmits data 1.
(3)	Transmits an inter-byte space.
(4)	<ul style="list-style-type: none"> <li>Transmits data 2.</li> <li>Transmits an inter-byte space</li> <li>Transmits data 3.</li> <li>Transmits an inter-byte space</li> </ul> <p>(Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDTC register).</p> <p>:</p> <p>:</p>
(5)	Transmits the checksum.
(6)	<ul style="list-style-type: none"> <li>Sets the successful response/wake-up transmission flag.</li> <li>Sets the FTS bit in the RLN3nLTTC register to 0 (frame transmission or wake-up transmission/reception stopped)</li> <li>Sets the RTS bit in the RLN3nLTTC register to 0 (response transmission/reception stopped).</li> </ul>
(7) <ul style="list-style-type: none"> <li>Processing after communication</li> <li>Checks the RLN3nLST register, and clears flags.</li> </ul>	<ul style="list-style-type: none"> <li>Completes the response transmission process.</li> <li>Waits for a new break.</li> </ul>

**NOTE**

- For information about error detection, see **Section 17.7.7, Error Statuses**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the subsequent Sync field and ID field) continues regardless of whether an error occurred.

**17.7.2.3 Response Reception**

**Figure 17.11** shows the operation of the LIN/UART interface (LIN slave mode) in response reception. **Table 17.83** provides processing in response reception.



**Figure 17.11 Operation in Response Reception**

**Table 17.83 Processing in Response Reception**

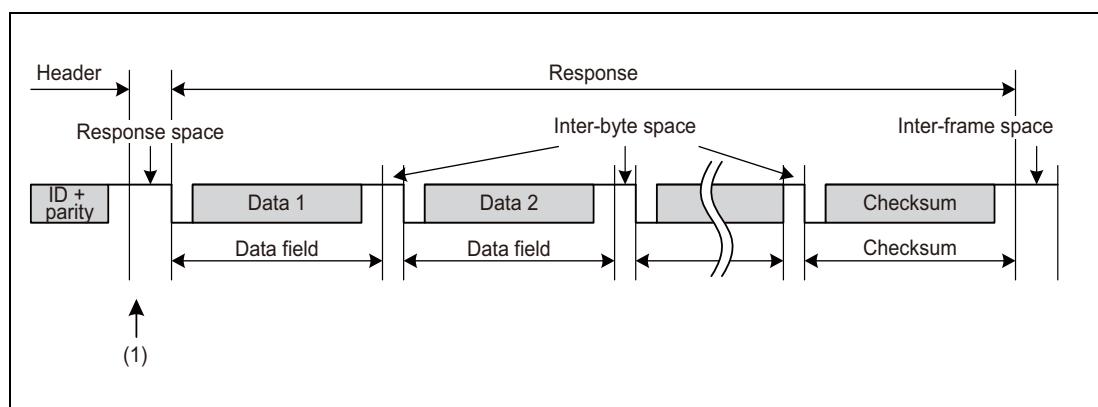
Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> <li>Sets the RLN3nLDFC register.</li> <li>Sets the response transmission/reception start bit (RTS bit) to 1.</li> </ul>	<ul style="list-style-type: none"> <li>Waits for the RTS (response transmission/reception start) or LNRR (no-response request) bit to be set by software.</li> <li>Waits for detection of the start bit.</li> </ul>
(2) Waits for an interrupt request.	Receives data 1 when the start bit is detected.
(3)	Sets the successful data 1 reception flag.
(4)	<ul style="list-style-type: none"> <li>Receives data 2 when the start bit is detected.</li> <li>Receives data 3 when the start bit is detected.</li> </ul>
	Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)	<ul style="list-style-type: none"> <li>Receives the checksum when the start bit is detected.</li> <li>Determines the checksum.</li> <li>Sets the successful response/wake-up reception flag or error flag.</li> <li>Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).</li> </ul>
(6) <ul style="list-style-type: none"> <li>Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.</li> </ul>	<ul style="list-style-type: none"> <li>Completes the response process.</li> <li>Waits for a new break.</li> </ul>

**NOTE**

- For information about error detection conditions, see **Section 17.7.7, Error Statuses**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the subsequent Sync field and ID field) continues regardless of whether an error occurred.

**17.7.2.4 No-response Request**

**Figure 17.12** shows the operation of the LIN/UART interface (LIN slave mode) when no response is requested. **Table 17.84** shows the processing that occurs when no response is requested.



**Figure 17.12 Operation when No Response is Requested**

**Table 17.84 Processing when No Response is Requested**

Software Processing	LIN/UART Interface Processing
(1) <ul style="list-style-type: none"> <li>• Sets the no-response request bit (LNRR bit) to 1.</li> </ul>	<ul style="list-style-type: none"> <li>• Waits for the LNRR (no-response request) bit to be set by software</li> <li>• Completes the frame reception process</li> <li>• Waits for a new break</li> </ul>

### 17.7.3 Data Transmission/Reception

#### 17.7.3.1 Data Transmission

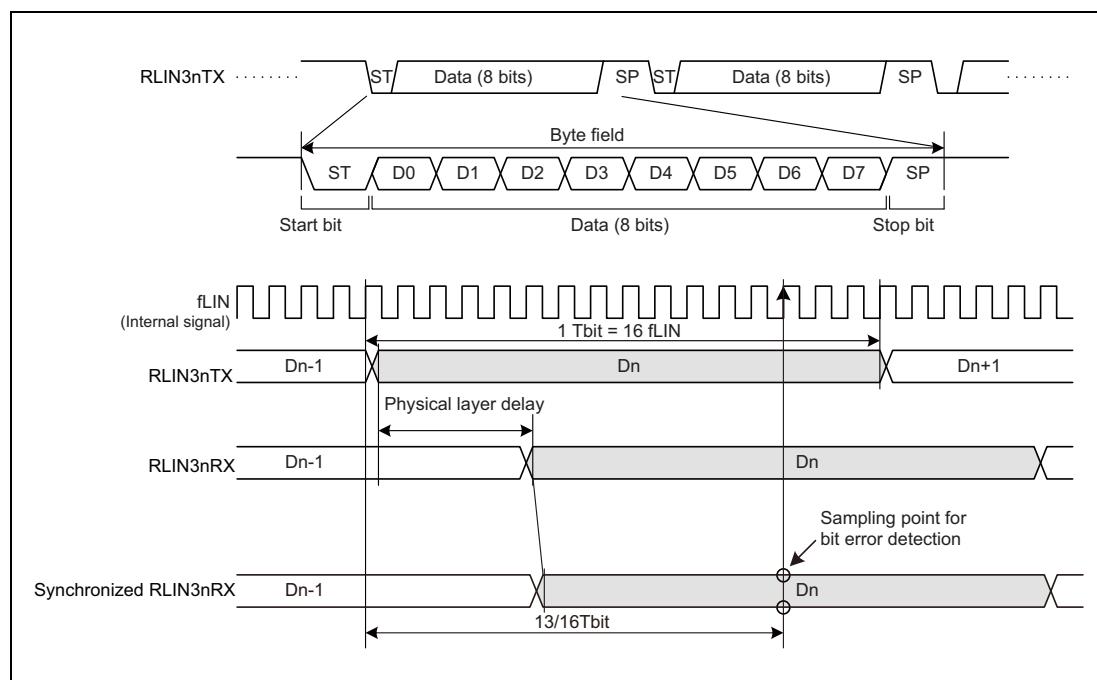
One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag of the RLIN3nLEST register (see **Section 17.7.7, Error Statuses**).

In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be 16fLIN, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).

In LIN slave mode [auto baud rate], if 1 Tbit is generated to be 4fLIN, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be 8fLIN, the sampling point for received data is at the 7th clock cycle (87.5% position).

**Figure 17.13** shows an example of data transmission timing.



**Figure 17.13 Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])**

### 17.7.3.2 Data Reception

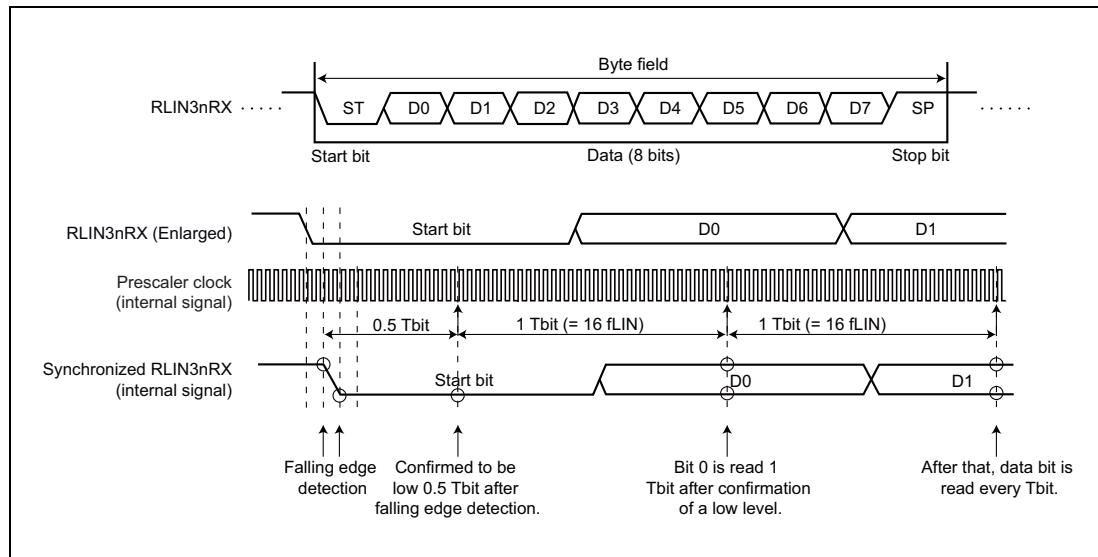
Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function with respect to reception data. If the LRDNFS bit in the RLIN3nLMD register is 0, the LIN/UART interface uses a noise filter, and the value determined by a 3-sampling majority rule on prescaler clocks is used as the sampling value. If the LRDNFS bit in the RLIN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and the value of the synchronized RLIN3nRX value at the sampling position is used as the sampling value.

**Figure 17.14** shows an example of data reception timing.



**Figure 17.14 Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])**

## 17.7.4 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface transmits or receives data continuously.

### 17.7.4.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data regions 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data regions 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

**Figure 17.15** shows the LIN transmission processing and the corresponding buffers.

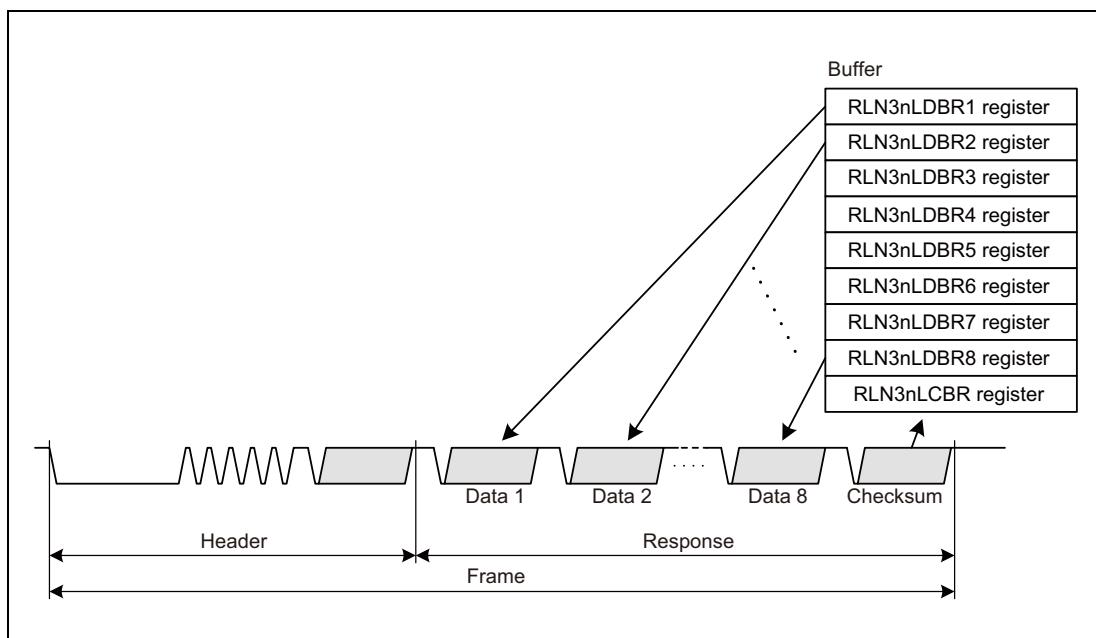


Figure 17.15 LIN Transmission Processing and Corresponding Buffers

#### [Frame Separate Mode]

Setting the FSM bit in the RLN3nLDTC register to 1 turns on the frame separate mode.

In frame separate mode, the header and response are separately transmitted when prompted by respective transmission start requests.

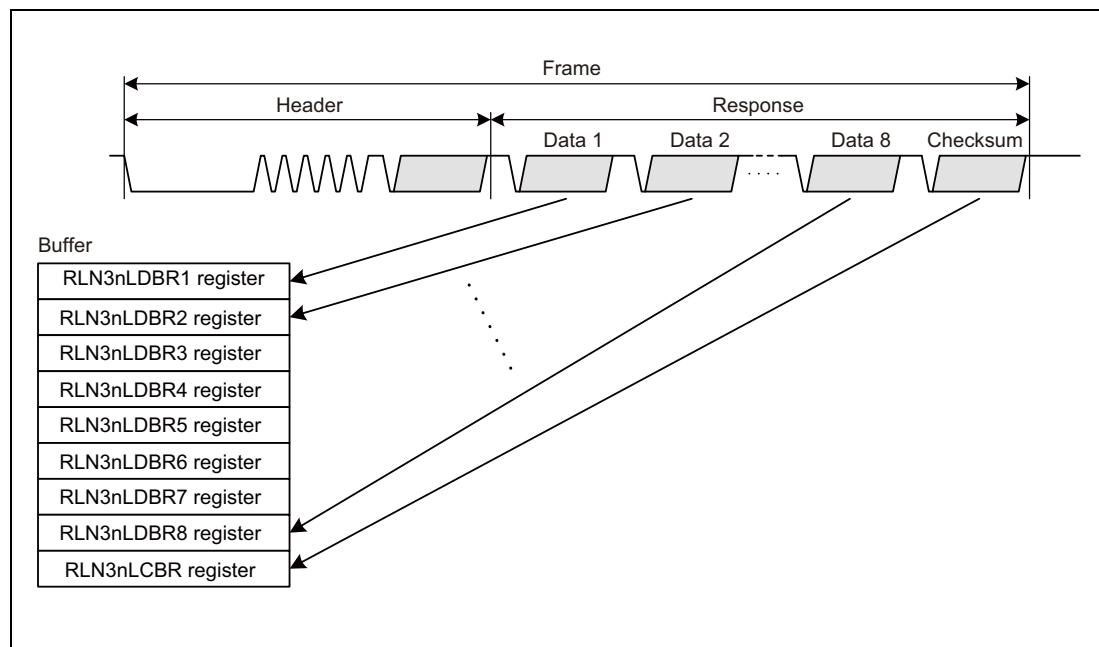
When the transmission of the header is completed, the HTRC flag in the RLN3nLST register is set to 1 (successful header transmission).

Use frame separate mode when transmitting or receiving response data of 9 bytes or greater in LIN master mode.

#### 17.7.4.2 Reception of LIN Frames

For an 8-byte reception, the contents of data regions 1 to 8 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data regions 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

**Figure 17.16** depicts the LIN reception processing and the corresponding buffers.



**Figure 17.16 LIN Reception Processing and Corresponding Buffers**

##### [Reception of Data 1]

When the reception of the first byte of data is completed, the D1RC flag in the RLN3nLST register is set to 1 (successful data 1 reception).

#### 17.7.4.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses of 10 bytes or greater can also be transmitted and received.

In such case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is greater than 8 bytes, the LSS bit in RLN3nLDTC register should be set to 1 (indicating that the next data group to be transmitted or received is not the final data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the final data group. If it is the final data group, the LSS bit in the RLN3nLDTC register should be set to 0 (indicating that the next data group to be transmitted or received is the final data group) before transmitting or receiving the data group, and a checksum should be appended to the final data group.

By changing the RFDL bit setting in RLN3nLDTC register when the RTS bit in RLN3nLTCR register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDTC register to 1 (frame separate mode).

##### NOTE

In LIN slave mode, the LIN/UART interface can detect a new break field during the transmission or reception of a response.

## 17.7.5 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

### 17.7.5.1 Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (LIN master mode: response transmission), or setting the RCDS bit in the RLN3nLDFC register to 1 (LIN slave mode: response transmission), and then the FTS bit in the RLN3nLTRC register to 1 (frame transmission, header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low level width of the wake-up signal is set using the WUTL[3:0] bits in the RLN3nLWUP register.

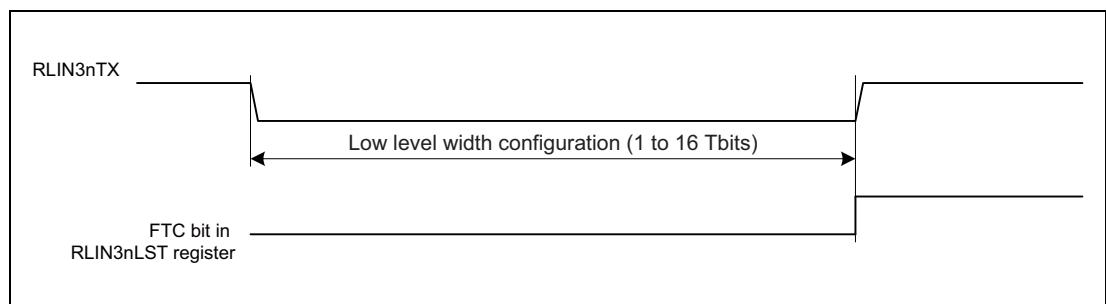
However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x), the LIN system clock ( $f_{LIN}$ ) becomes low level width at  $f_a$  regardless of the setting of the LCKS bit of the RLN3nLMD register. By setting the WUTL[3:0] bits of the RLN3nLWUP register to  $0100_B$  (5 Tbits), 260  $\mu s$  low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low-level width is output without any bit error, the FTC flag in the RLN3nLST register is set to 1 (successful frame response or wake-up transmission) and when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wake-up transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is aborted and the BER flag of the RLN3nLEST register is set to 1 (bit error detected).

When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

**Figure 17.17** shows the wake-up transmission timing.



**Figure 17.17 Wake-up Transmission Timing**

### 17.7.5.2 Wake-up Reception

To detect a wake-up signal, use the input signal low-level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation is executed without changing the baud rate generator setting at a transition between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects the LIN system clock (fLIN) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200bps while fa is selected, an input signal with a low-level width of 130 us or longer regardless of the setting of the LCKS bit in the RLN3nLMD register.

To used the wake-up reception function, in LIN wake-up mode, set the RFT bit in the RLN3nLDTC register to 0 (LIN master mode: response reception) or the RCDS bit in the RLN3nLDTC register to 0 (LIN slave mode: response reception), and then set the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register is set to 1 (successful frame response/wake-up reception) and if the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for RLIN3n successful reception is generated.

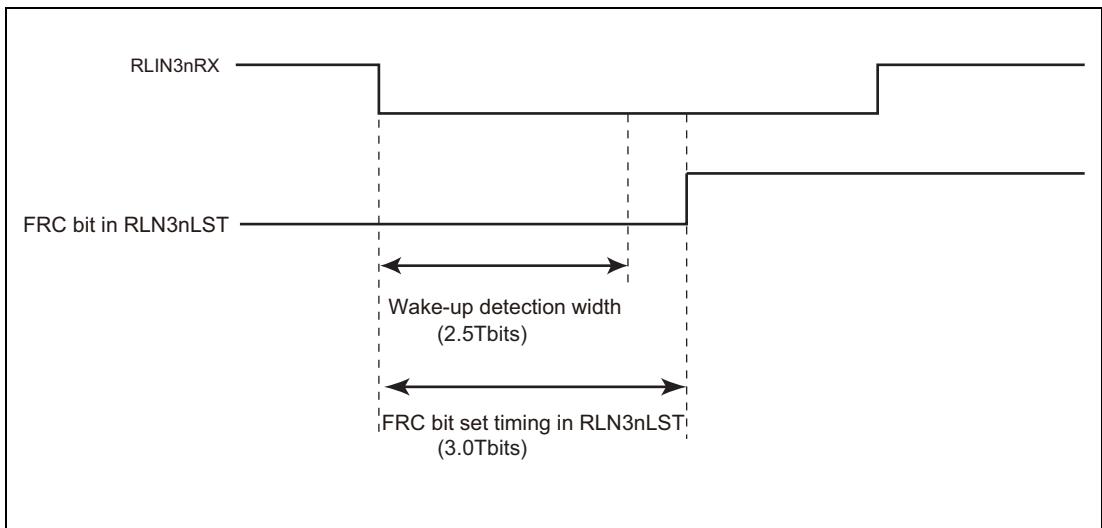


Figure 17.18 Input Signal Low level Count Function

### 17.7.5.3 Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, however the collision of wakeup signals is not detected in the LIN/UART interface.

## 17.7.6 Statuses

During LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, successful header transmission/reception, can generate interrupt requests.

**Table 17.85** shows the types of statuses available in LIN master mode. **Table 17.86** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

**Table 17.85 Types of Statuses in LIN Master Mode**

Status	Status Set Condition	Status Clear Condition	Operating Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if the LIN/UART interface actually exits LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if the LIN/UART interface actually enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if the LIN/UART interface actually enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if the LIN/UART interface actually enters LIN wake-up mode.	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	OMM1 bit in RLN3nLMST register	—
Successful frame/wake-up transmission	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	<ul style="list-style-type: none"> <li>• When the next communication is started (When the FTS bit in the RLN3nLTCR register is set)</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	FTC flag in RLN3nLST register	√
Successful frame/wake-up reception	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	<ul style="list-style-type: none"> <li>• When the next communication is started (When the FTS bit in the RLN3nLTCR register is set)</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	<ul style="list-style-type: none"> <li>• When the next communication is started (When the FTS bit in the RLN3nLTCR register is set)</li> <li>• When cleared by software*1</li> <li>• After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	ERR flag in RLN3nLST register	√
Successful data 1 reception	When the RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received successfully.*2	<ul style="list-style-type: none"> <li>• When the next communication is started (When the FTS bit in the RLN3nLTCR register is set)</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	LIN operation mode	D1RC flag in RLN3nLST register	—
Successful header transmission	When a header field is transmitted successfully.	<ul style="list-style-type: none"> <li>• When the next communication is started (When the FTS bit in the RLN3nLTCR register is set)</li> <li>• When cleared by software</li> <li>• After transition to LIN reset mode</li> </ul>	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000<sub>B</sub> (0-byte + checksum).

Table 17.86 Types of Statuses in LIN Slave Mode

Status	Status Set Condition	Status Clear Condition	Operating Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if the LIN/UART interface actually exits LIN reset mode.	After the OM0 bit of the RLN3nLCUC register is set to LIN reset mode, if the LIN/UART interface actually enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMS T register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if the LIN/UART interface actually enters LIN operation mode.	<ul style="list-style-type: none"> <li>After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if the LIN/UART interface actually enters LIN wake-up mode.</li> </ul>	<ul style="list-style-type: none"> <li>LIN operation mode</li> <li>LIN wake-up mode</li> </ul>	OMM1 bit in RLN3nLMS T register	—
Successful response/wake-up transmission	When a response field, a wake-up signal, or a data group is transmitted successfully.	<ul style="list-style-type: none"> <li>When cleared by software</li> <li>After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>LIN operation mode</li> <li>LIN wake-up mode</li> </ul>	FTC flag in RLN3nLST register	✓
Successful response/wake-up reception	When a response field, a wake-up signal, or a data group is received successfully.	<ul style="list-style-type: none"> <li>When cleared by software</li> <li>After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>LIN operation mode</li> <li>LIN wake-up mode</li> </ul>	FRC flag in RLN3nLST register	✓
Error detection	If any of the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, and BER flag of the RLN3nLEST register is set to 1 (error detected).	<ul style="list-style-type: none"> <li>When cleared by software*1</li> <li>After transition to LIN reset mode</li> </ul>	<ul style="list-style-type: none"> <li>LIN operation mode</li> <li>LIN wake-up mode</li> </ul>	ERR flag in RLN3nLST register	✓
Successful data 1 reception	When the RCDS bit in the RLN3nLDTC register is 0 (reception) and the first byte of the response field or the first byte for each data group is received successfully.*2	<ul style="list-style-type: none"> <li>When cleared by software</li> <li>After transition to LIN reset mode</li> </ul>	LIN operation mode	D1RC flag in RLN3nLST register	—
Successful header reception	When a header field is received successfully.	<ul style="list-style-type: none"> <li>When cleared by software</li> <li>After transition to LIN reset mode</li> </ul>	LIN operation mode	HTRC flag in RLN3nLST register	✓

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, or BER flag of the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDTC register are 0000<sub>B</sub> (0-byte + checksum).

## 17.7.7 Error Statuses

### 17.7.7.1 LIN Master Mode

#### (1) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

All error statuses represent interrupt events.

**Table 17.87** shows the types of error statuses.

**Table 17.87 Types of Error Statuses in LIN Master Mode**

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1*2	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	Aborted	Enabled	BER flag in RLN3nLEST register
Physical bus error	<ul style="list-style-type: none"> <li>• LIN bus is detected to be high level when transmitting a break</li> <li>• LIN bus is detected to be low level when transmitting a break delimiter</li> <li>• LIN bus is detected to be high level when transmitting a wake-up</li> </ul>	<ul style="list-style-type: none"> <li>• LIN operation mode</li> <li>• LIN wake-up mode</li> </ul>	Aborted	Enabled	PBER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception does not terminate within a given time*3	LIN operation mode	Aborted	Enabled	FTER flag in RLN3nLEST register
Framing error	In response field reception, the stop bit of each data byte is low level	LIN operation mode	Aborted	Enabled	FER flag in RLN3nLEST register
Checksum error	In response field reception, checksum test results in an error	LIN operation mode	—	Disabled	CSER flag in RLN3nLEST register
Response preparation error	One of the following conditions occurs in frame separate mode during a multi-byte response reception: <ul style="list-style-type: none"> <li>• The first reception data byte is received after completion of header transmission but before a response transmission/reception request is specified</li> <li>• The first reception data byte is received after the completion of previous data group reception but before a transmission/reception request for the next data group is specified.</li> </ul>	LIN operation mode	Aborted	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately after the bit that caused the error is transmitted. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted after the bit that caused the error is transmitted.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDTC register) and the checksum selection (the CSM bit in the RLN3nLDTC register), and can be calculated using the following formula.

When the FSM bit in the RLN3nLDTC register is set to 1 (frame separate mode), the timeout time is that of the 8 data bytes until the RTS bit of the RLN3nLTC register is set. Once the RTS bit is set, the timeout time is changed to the time based on the response field data length (the RFDL[3:0] bits in the RLN3nLDTC register).

**[Frame timeout]**

When classic checksum is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time =  $49 + (\text{number of data bytes} + 1) \times 14$  [Tbit]

When enhanced checksum is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time =  $48 + (\text{number of data bytes} + 1) \times 14$  [Tbit]

The aforementioned timeout time is longer than the TFRAME\_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME\_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.

**[Response timeout]**

Timeout time =  $(\text{number of data bytes} + 1) \times 14$  [Tbit]

When an error is detected, time-out error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTCR register is set), when it is cleared by software, or at a transition to LIN reset mode.

## (2) Target Time Domain for LIN Error Detection

**Figure 17.19** shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

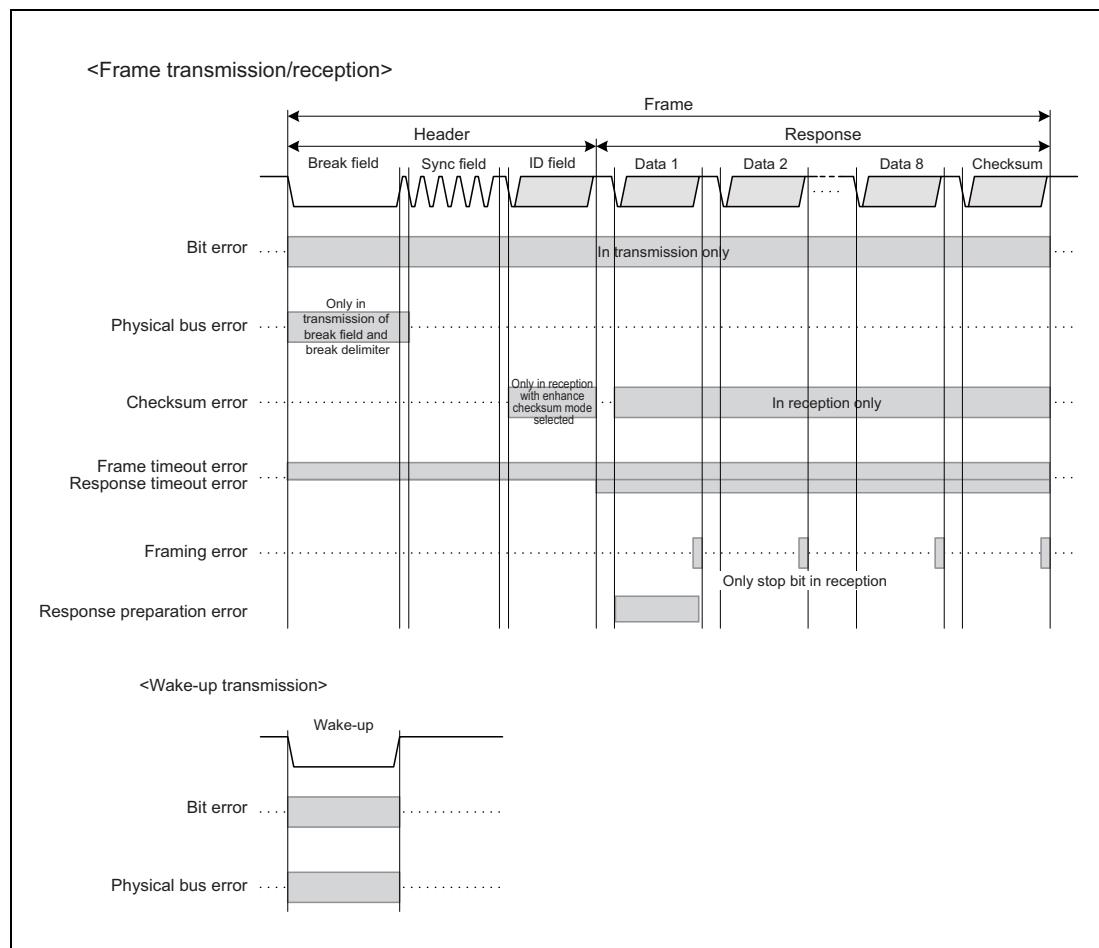


Figure 17.19 Target Time Domain for LIN Error Detection (LIN Master Mode)

### 17.7.7.2 LIN Slave Mode

#### (1) Types of Error Statuses

The LIN/UART interface can detect seven types of error statuses in LIN slave mode [auto baud rate] or in LIN slave mode [fixed baud rate]. These error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

**Table 17.88** shows the types of error statuses.

**Table 17.88 Types of Error Statuses in LIN Slave Mode**

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match* <sup>1</sup> * <sup>2</sup>	<ul style="list-style-type: none"> <li>LIN operation mode</li> <li>LIN wake-up mode</li> </ul>	Aborted	Enabled	BER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception does not terminate within a given time* <sup>3</sup>	LIN operation mode	Aborted	Enabled	TER flag in RLN3nLEST register
Framing error	In frame reception, the stop bit of each data byte is low level	LIN operation mode	Aborted	Enabled	FER flag in RLN3nLEST register
Sync field error	If the width of the break low level is greater than the width set by the LBFT bit in the RLN3nLBFC register and the sync field is not 55H	LIN operation mode	Aborted	Enabled* <sup>4</sup>	SFER flag in RLN3nLEST register
Checksum error	In response field reception, the checksum test results in an error	LIN operation mode	— <sup>5</sup>	Disabled	CSER flag in RLN3nLEST register
ID parity error	If the received ID parity bit does not match the value that is automatically calculated by the LIN/UART interface	LIN operation mode	Aborted	Enabled	IPER flag in RLN3nLEST register
Response preparation error	<ul style="list-style-type: none"> <li>After the reception of a header, if the response is not prepared before the first reception data byte is received</li> <li>In a multi-byte response reception, if the preparation for the reception of next data group does not complete before the first reception data byte for the next data group is received</li> </ul>	LIN operation mode	Aborted	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately after the bit that caused the error is transmitted. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted after the bit that caused the error is transmitted.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL[3:0] bits in the RLN3nLDTC register) and the checksum selection (the LCS bit in the RLN3nLDTC register), and this can be calculated according to the following formula. The timeout time is that of 8 data bytes until the RTS bit or the LNRR bit of the RLN3nLTC register is set. When the RTS bit is set, the timeout time is changed to the time based on the response field data length (RFDL[3:0] bit of the RLN3nLDTC register). When the LNRR bit is set, the timeout function stops.

#### [Frame timeout]

When classic checksum is selected (when the CSM bit in RLN3nLDTC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

When enhanced checksum is selected (when the CSM bit in RLN3nLDTC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is longer than the TFRAME\_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME\_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.

#### [Response timeout]

Timeout time = (number of data bytes + 1) × 14 [Tbit]

When an error is detected, time-out error detection function stops.

- Note 4. Only reflection of the result to the SFER flag can be enabled/disabled. Error detection cannot be enabled/disabled.
- Note 5. Checksum determination is performed upon completion of response frame reception. In case of an error, the successful reception flag is not set to 1.

The error status is cleared by software or at a transition to LIN reset mode.

## (2) Target Time Domain for LIN Error Detection

**Figure 17.20** shows the time domain in which the LIN/UART interface in slave mode performs monitoring for error detection.

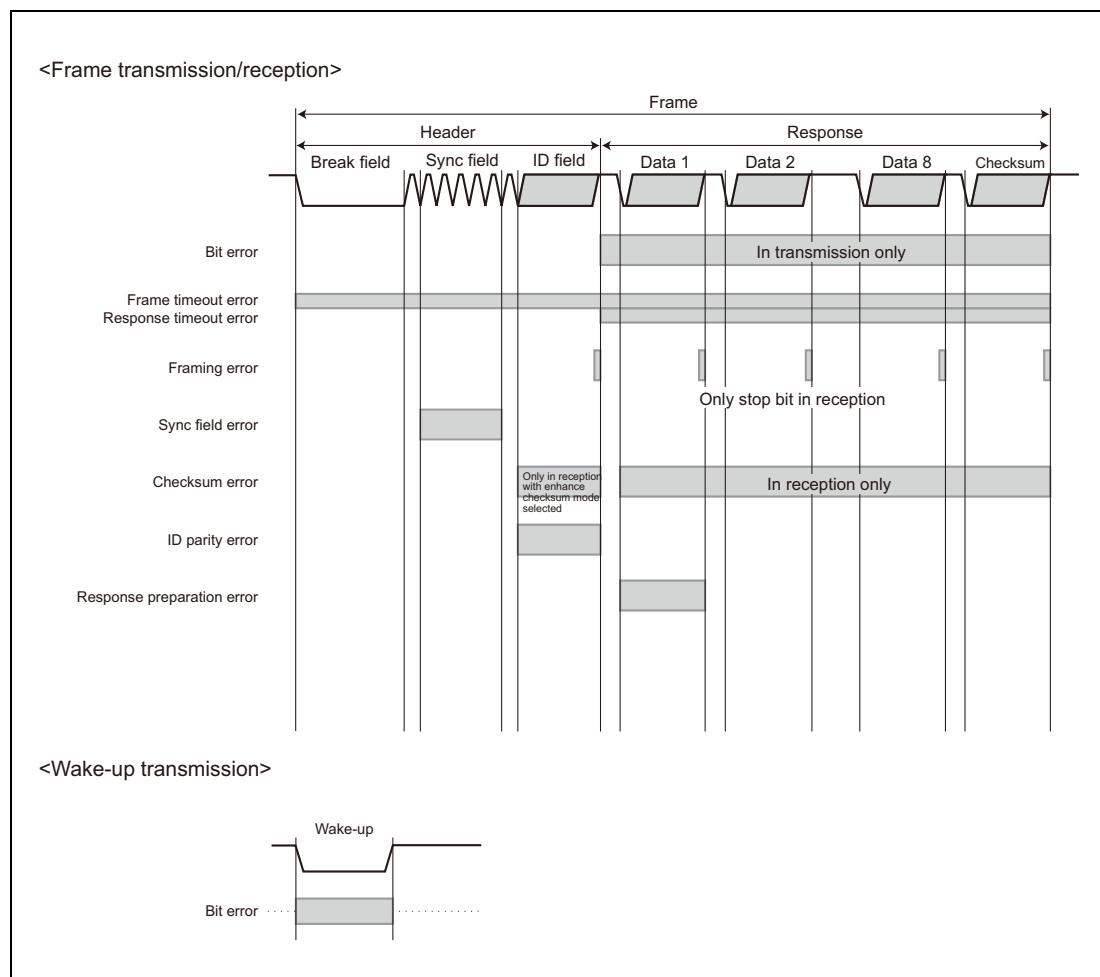


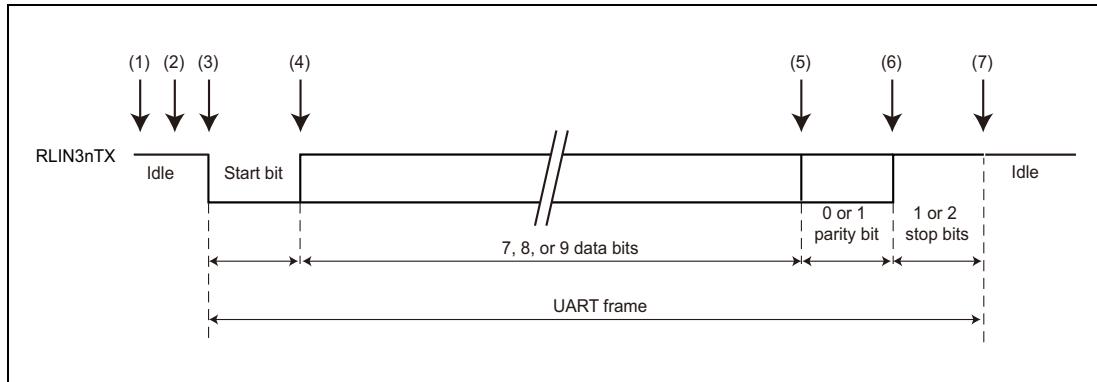
Figure 17.20 Target Time Domain for LIN Error Detection (LIN Slave Mode)

## 17.8 UART Mode

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 01<sub>B</sub> (UART mode) and the OM0 bit in the RLN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLN3nLMST register to 1.

### 17.8.1 Transmission

**Figure 17.21** shows the operation of the LIN/UART interface (in UART mode) during transmission. **Table 17.89** provides processing of the LIN/UART interface (in UART mode) during transmission.



**Figure 17.21** LIN/UART Interface (in UART Mode) Transmission Operation

**Table 17.89** LIN/UART Interface (UART Mode) Transmission Processing (1/2)

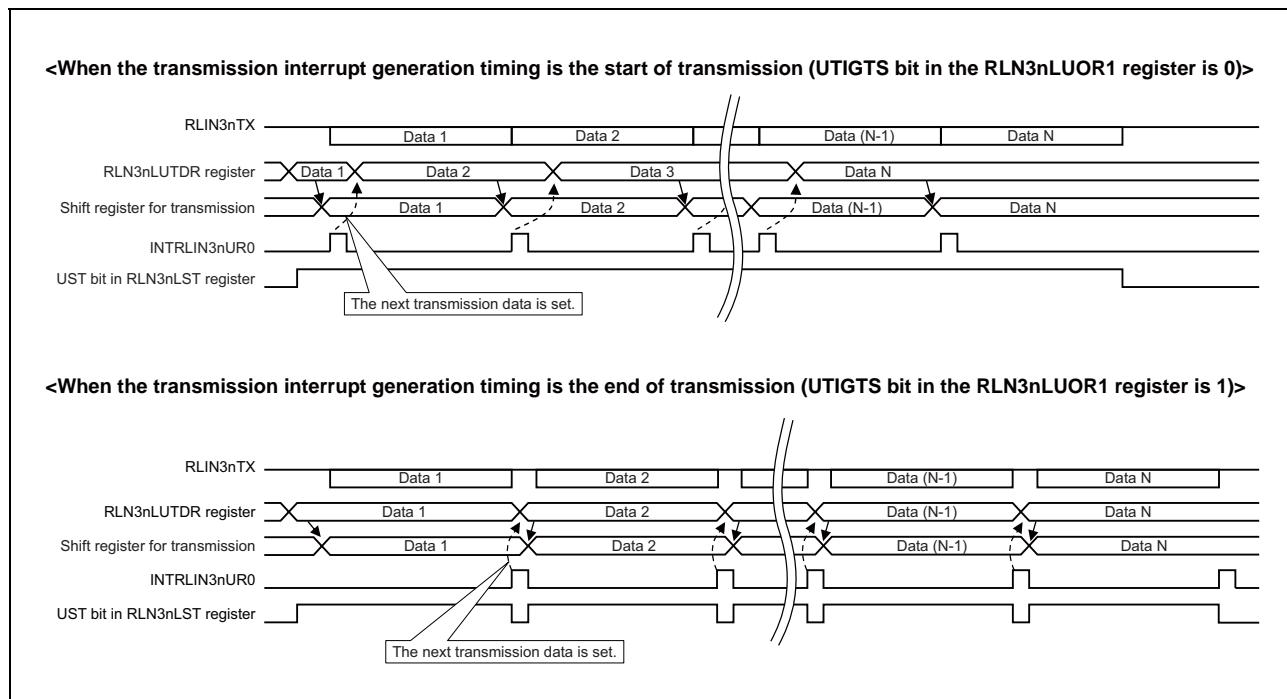
Software Processing	LIN/UART Interface Processing
(1) • Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Configures data format. • Sets an interrupt generation timing. • Exits from LIN reset mode. • Sets the transmission enable bit (UTOE bit) to 1.	• Waits for a transmission trigger (RLN3nLUTDR register) by software.
(2) • Specifies the data to be transmitted in the UART transmit data register (RLN3nLUTDR) or UART wait transmit data register (RLN3nLUWTDR).	• Sets the transmission status flag.
(3) • Waits for an interrupt request.	• Transmits a start bit (for switching from reception to transmission in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see <b>Section 17.8.1.4, Transmission Start Wait Function</b> ).  [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • Outputs a transmission interrupt.
(4) [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • When transmitting data continuously, specifies the next data to be transmitted in the UART transmission data register (RLN3nLUTDR) and waits for the generation of an interrupt request.	Transmits the data specified in the UART (wait) transmission data register.
(5)	Transmits a parity bit when parity is used.
(6)	Transmits 1 or 2 stop bits.

**Table 17.89 LIN/UART Interface (UART Mode) Transmission Processing (2/2)**

Software Processing	LIN/UART Interface Processing
(7) [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] <ul style="list-style-type: none"> <li>• If another item of transmission data is set, goes to step (3).</li> </ul>	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] <ul style="list-style-type: none"> <li>• If another piece of transmission data is set, goes to step (3).</li> <li>• If the next data to be transmitted is not specified, clears the transmission status flag.</li> </ul>
[When the UTIGTS bit is 1 (a transmission interrupt is output upon completion of transmission)] <ul style="list-style-type: none"> <li>• When transmitting data continuously, goes to step (2).</li> </ul>	[When the UTIGTS bit is 1 (a transmission interrupt is output upon completion of transmission)] <ul style="list-style-type: none"> <li>• Generates RLIN3n transmission interrupt request.</li> <li>• Clears the transmission status flag.</li> </ul>

### 17.8.1.1 Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLIN3nLUTDR register. **Figure 17.22** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the completion of transmission.



**Figure 17.22 Operation Examples of LIN/UART Interface (UART Mode) Continuous Transmission**

An interrupt can be generated at the completion of a transmission by changing the UTIGTS bit in the RLIN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided that the transmission interrupt generation timing is the start of transmission and the completion of transmission of final data needs to be known.

### 17.8.1.2 UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers and it is capable of performing continuous transmissions through the use of UART buffers.

**Figure 17.23** shows the UART buffer transmission operation of the LIN/UART interface (in UART mode). **Table 17.90** shows the UART buffer transmission processing.

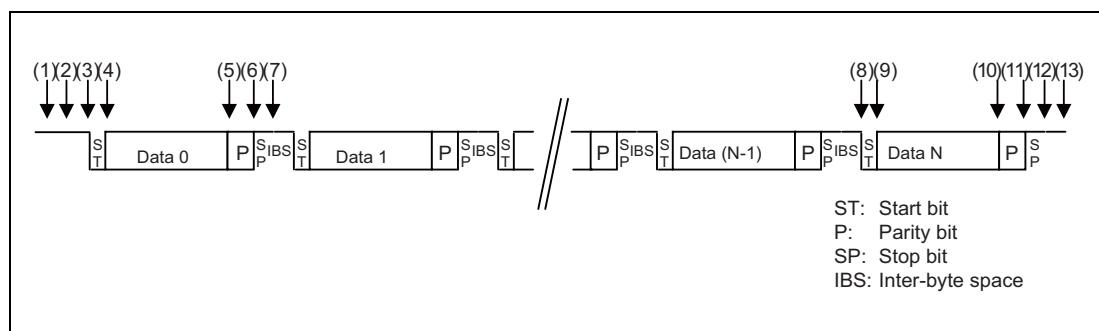


Figure 17.23 UART Buffer Transmission in LIN/UART Interface (in UART Mode)

Table 17.90 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode)  
(1/2)

Software Processing	LIN/UART Interface Processing
(1)	<ul style="list-style-type: none"> <li>Sets a baud rate</li> <li>Sets noise filter ON/OFF</li> <li>Enables error detection</li> <li>Configures data format</li> <li>Sets the interrupt generation timing to the completion of transmission.</li> <li>Exits from LIN reset mode.</li> <li>Sets the transmission enable bit (UTOE bit) to 1</li> </ul> <ul style="list-style-type: none"> <li>Waits for a transmission trigger (RTS bit) by software</li> </ul>
(2)	<ul style="list-style-type: none"> <li>Sets the UART buffer data length and whether the start of transmission must be waited.</li> <li>Specifies the data to be transmitted in the UART data 0 buffer register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b = 1 to 8)</li> <li>Sets the UART buffer transmission start bit (RTS).</li> </ul> <ul style="list-style-type: none"> <li>Sets the transmission status flag.</li> </ul>
(3)	Waits for an interrupt request. Transmits a start bit. (For switching from reception to transmission in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see <b>Section 17.8.1.4, Transmission Start Wait Function.</b> )
(4)	Transmits the data specified in the UART data buffer 0 register (RLN3nLUDB0) or the LIN/UART data buffer b register (RLN3nLDBRb).
(5)	Transmits a parity bit when parity is used.
(6)	Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length select bits is 1, proceeds to (12).)
(7)	Transmits an inter-byte space (idle).
	Repeats steps (3) to (7) until the number of data set in the UART buffer data length select bits -1 is reached.

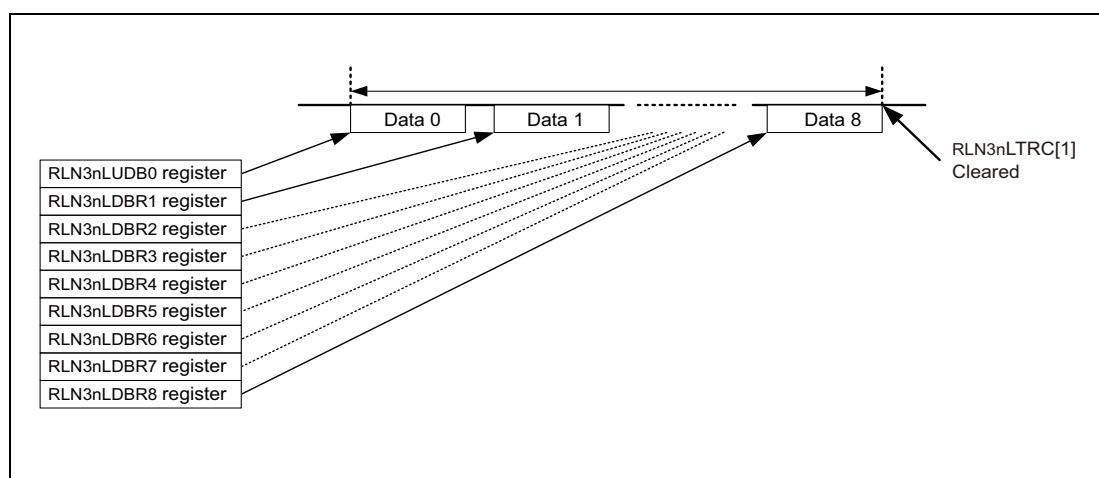
**Table 17.90 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) (2/2)**

Software Processing	LIN/UART Interface Processing
(8)	Transmits a start bit.
(9)	Transmits the data specified in the LIN/UART data buffer b register (RLN3nLDBRb).
(10)	Transmits a parity bit when parity is used.
(11)	Transmits 1 or 2 stop bits.
(12)	<ul style="list-style-type: none"> <li>• Sets the successful buffer transmission flag.</li> <li>• Clears the UART buffer transmission start bit (RTS).</li> <li>• Generates a transmission interrupt request signal.</li> <li>• Clears the transmission status flag.</li> </ul>
(13)	<ul style="list-style-type: none"> <li>• Checks the RLN3nLST register, and clears flags</li> <li>• When continuously transmitting data, goes to step (2).</li> </ul>

### (1) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data regions 0 to 8. The RLN3nLUDB0 register is used only if 9-byte transmission is specified. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending on the data length. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data regions 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the number of data specified in the MDL [3:0] bits of the RLN3nLDFC register is transmitted. The spaces between each transmitted data can be specified in the IBS bit of the RLN3nLSC register.

**Figure 17.24** shows a 9-byte UART buffer and the transmission processing.



**Figure 17.24** UART Buffer and Transmission Processing (for 9-Byte Transmission)

### 17.8.1.3 Data Transmission

One bit of data is transmitted per Tbit.

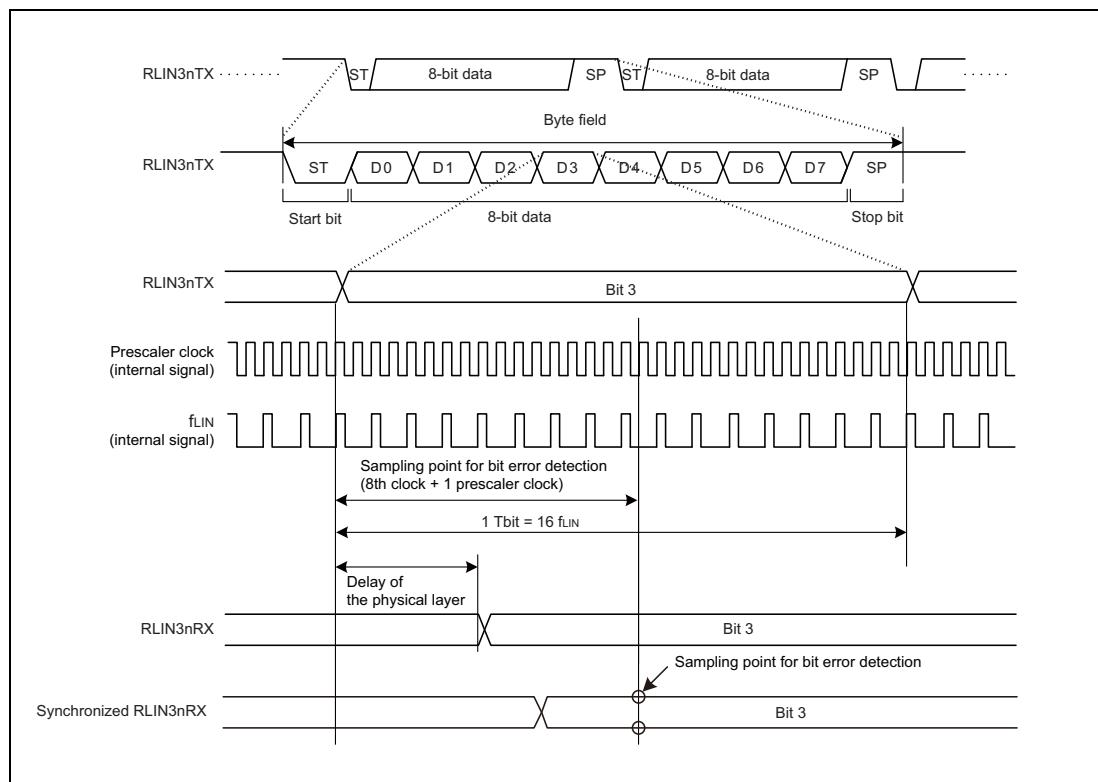
In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag of the RLN3nLEST register (see **Section 17.8.5, Error Statuses**). The timing at which the input pin is sampled during data transmission can vary depending on the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 17.91**.

**Table 17.91 Error Detection Timing in UART Mode**

Sampling Count Per Bit	Bit Error Detection Timing
6 samples	3rd clock cycle + 1 prescaler clock
7 samples	4th clock cycle + 1 prescaler clock
8 samples	4th clock cycle + 1 prescaler clock
9 samples	5th clock cycle + 1 prescaler clock
10 samples	5th clock cycle + 1 prescaler clock
11 samples	6th clock cycle + 1 prescaler clock
12 samples	6th clock cycle + 1 prescaler clock
13 samples	7th clock cycle + 1 prescaler clock
14 samples	7th clock cycle + 1 prescaler clock
15 samples	8th clock cycle + 1 prescaler clock
16 samples	8th clock cycle + 1 prescaler clock

Example of Data Transmission Timing (when 1 Tbit = 16 samplings) is shown in **Figure 17.25**



**Figure 17.25 Example of Data Transmission Timing (When 1 Tbit = 16 samplings)**

#### 17.8.1.4 Transmission Start Wait Function

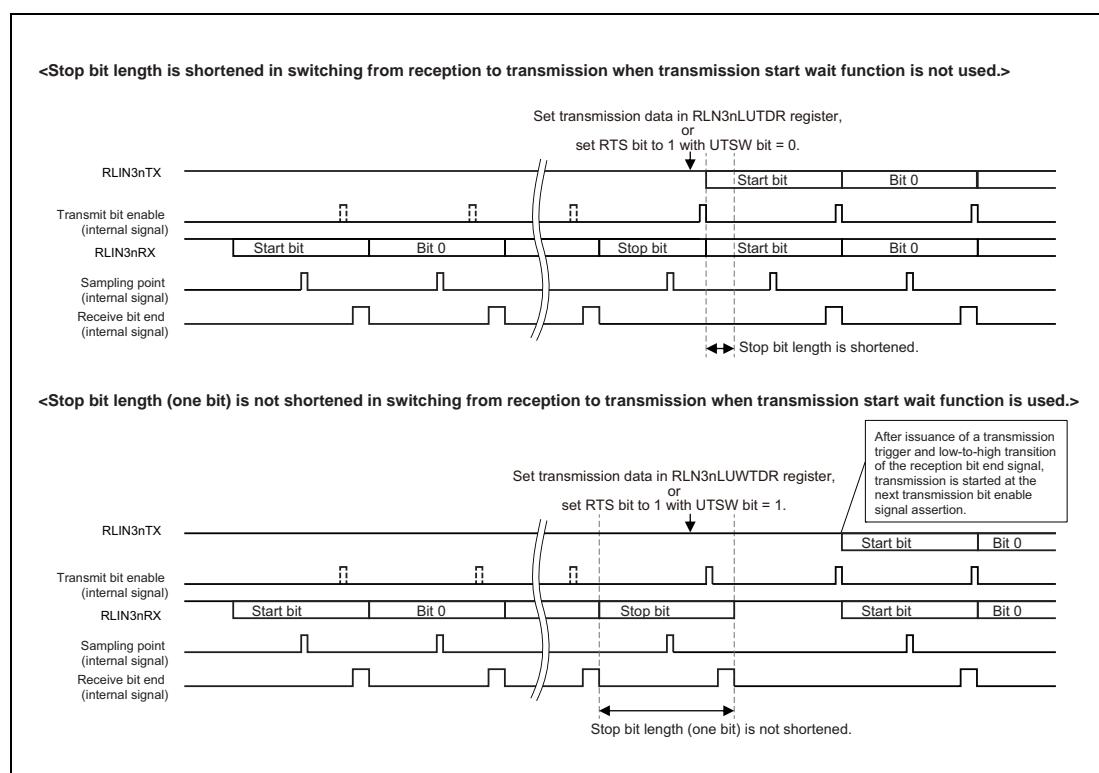
For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, specify the transmission data in the RLN3nLUWTDR register, which is used only for the wait function, instead of specifying it in the RLN3nLUTDR register as a transmission start request. When transmitting from the UART buffer, set the RTS bit in the RLN3nLTRC register to 1 (UART buffer transmission start) while the UTSW bit in the RLN3nLDFC register is set to 1.

In this case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

It should be noted that even if the UART stop bit length select bit (USBLS) in RLN3nBLFC register is 1 (stop bits = 2 bits), there is only a 1-bit delay.

**Figure 17.26** shows the operation of transmission wait function.



**Figure 17.26** Transmission Wait Function (Case When Transmit Data Is Set While Stop Bits Are Being Received)

## 17.8.2 Reception

**Figure 17.27** shows the LIN/UART interface (in UART mode) reception operation. **Table 17.92** shows the LIN/UART interface (in UART mode) reception processing.

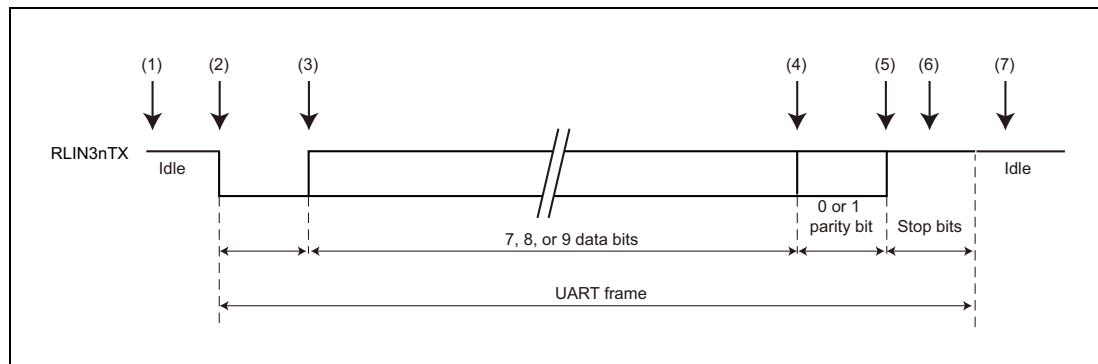


Figure 17.27 LIN/UART Interface (in UART Mode) Reception Operation

Table 17.92 LIN/UART Interface (in UART Mode) Reception Processing

Software Processing	LIN/UART Interface Processing
(1) • Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Configures data format. • Exits from LIN reset mode. • Sets the reception enable bit (UROE bit) to 1.	• Waits for the reception to be enabled by software. • Waits for detection of a start bit.
(2) Waits for an interrupt request.	• Waits for a falling edge from the reception pin, and detects a start bit. • Sets the reception status flag.
(3)	Receives data.
(4)	Receives a parity bit when parity is used.
(5)	Receives only 1 stop bit.
(6)	• Generates a RLIN3n successful reception interrupt request. • Clears the reception status flag.
(7) Checks the RLIN3nLST register, and clears flags	Waits for a falling edge from the reception pin.

### 17.8.2.1 Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

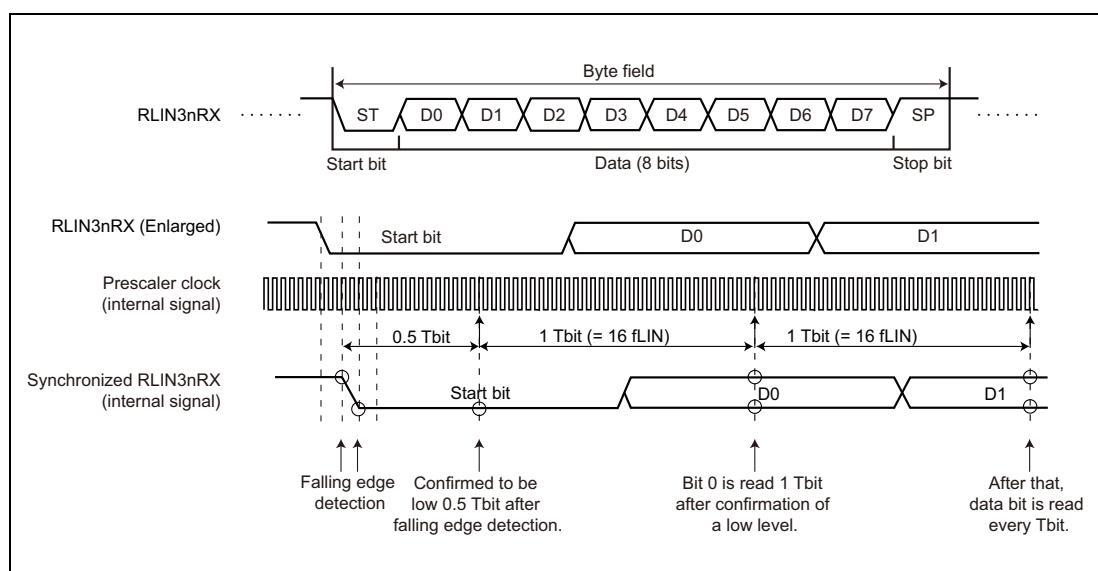
The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of sampling per 1 Tbit is even and  $\{( \text{number of sampling} + 1 ) / 2\} / (\text{number of sampling})$  Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filter function for received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-sampling majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

**Figure 17.28** shows an example of data reception timing.



**Figure 17.28 Example of Data Reception Timing (When 1 Tbit = 16 samplings)**

### 17.8.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLN3nLUOR1 register to 1.

#### 17.8.3.1 Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data by writing the 9-bit data to either the UART transmission data register (RLN3nLUTDR) or the UART wait transmission data register (RLN3nLUWTDR) when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1.

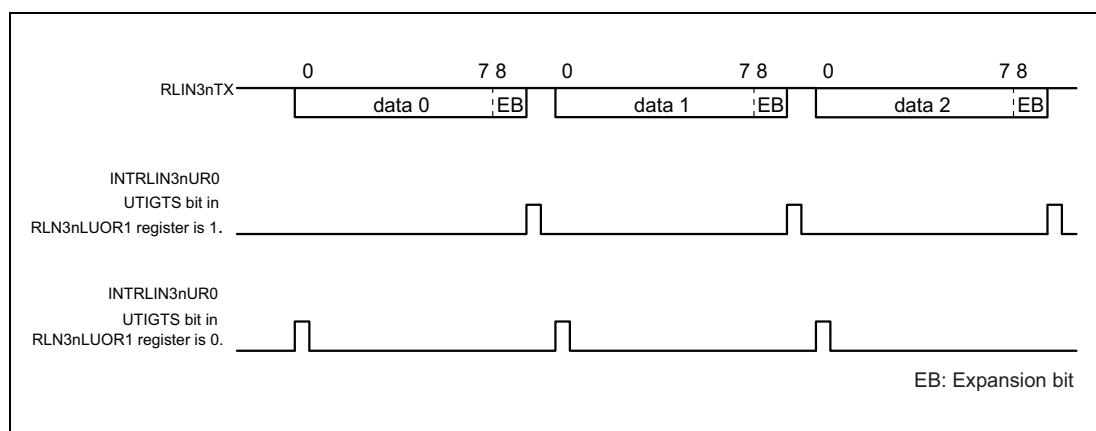


Figure 17.29 Transmission Example When Expansion Bit is Enabled (LSB First)

#### 17.8.3.2 Expansion Bit Reception

The LIN/UART interface (in UART mode) can always receive 9-bit data without comparing the expansion bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Regardless of the expansion bit detection level select bit (UEBDL) setting in the UART option register 1 (RLN3nLUOR1), a RLIN3n successful reception interrupt is generated when 9-bit data is received.

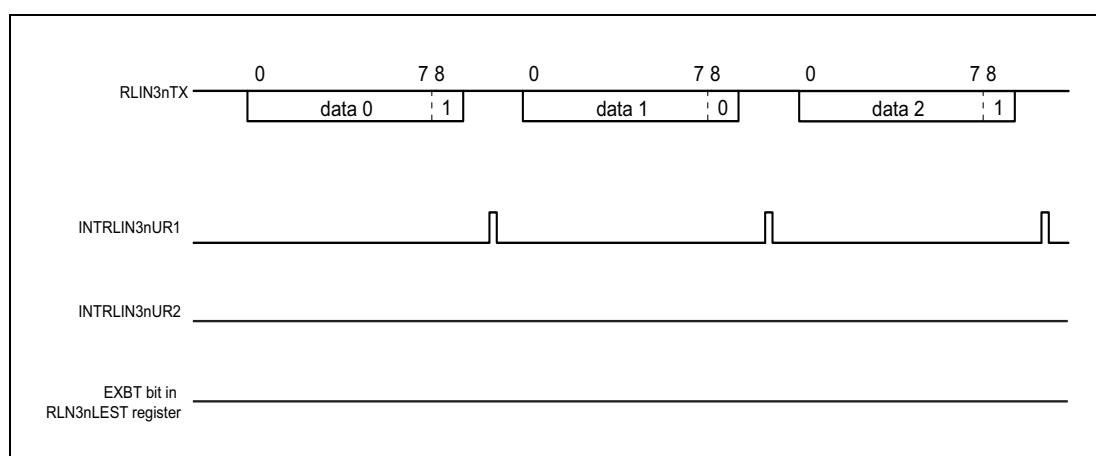


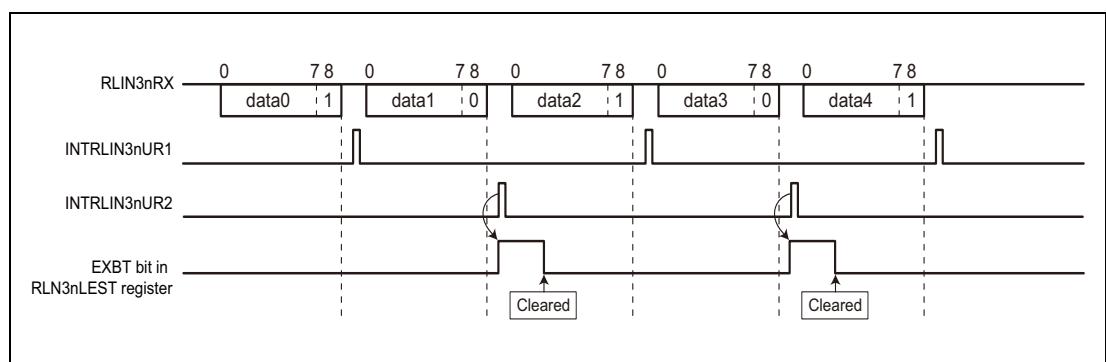
Figure 17.30 Expansion Bit Reception Example (LSB First)

### 17.8.3.3 Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level select bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, RLIN3n successful reception interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

**Figure 17.31** shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.



**Figure 17.31 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)**

#### NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a RLIN3n successful reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

#### 17.8.3.4 Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, when the level that was set by the expansion bit detection level select bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits of the received data excluding the expansion bit, with the a preset RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

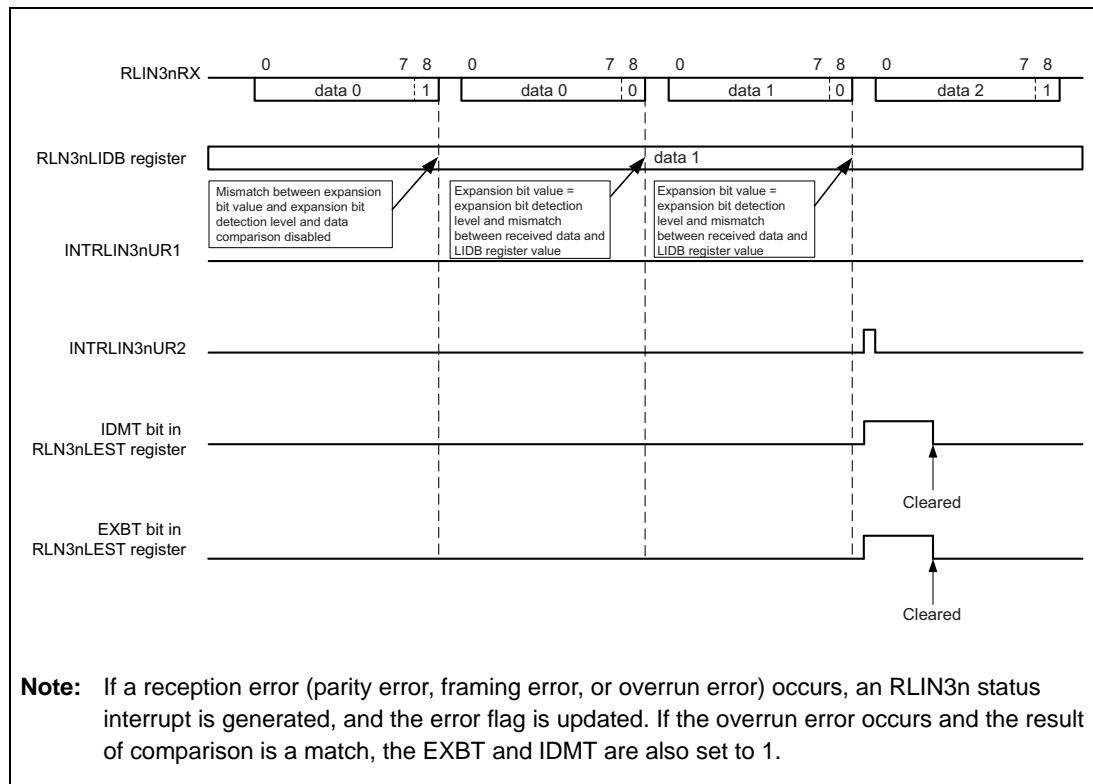
- Generates an RLIN3n status interrupt
- Sets the expansion bit detection flag (EXBT)
- Sets the ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, successful RLIN3n successful reception interrupt is not generated.

If the result of the comparison is not a match, no RLIN3n successful reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEVDCE bit to 0, make the change before the reception of the next data is completed.

**Figure 17.32** shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.



**Figure 17.32 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)**

### 17.8.4 Statuses

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

**Table 17.93** shows the types of statuses available in UART mode.

**Table 17.93 Types of Statuses in UART Mode**

Status	Status Set Condition	Status Clear Condition	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if the LIN/UART interface actually exits LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if the LIN/UART interface actually enters LIN reset mode.	OMM0 bit in RLN3nLMST register	—
Successful UART buffer transmission	<ul style="list-style-type: none"> <li>When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), if the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDTC register) is started.</li> <li>When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon completion of transmission), if the transmission of the data length specified by the MDL bit in the RLN3nLDTC register is completed.</li> </ul>	<ul style="list-style-type: none"> <li>When cleared by software</li> <li>After transition to LIN reset mode</li> </ul>	FTC flag in RLN3nLST register	✓
Error detection	If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	<ul style="list-style-type: none"> <li>When cleared by software<sup>*1</sup></li> <li>After transition to LIN reset mode</li> </ul>	ERR flag in RLN3nLST register	✓
Transmission status	<ul style="list-style-type: none"> <li>When data is written to the RLN3nLUTDR or RLN3nLUWTDR register.</li> <li>When 1 is written to the RTS bit in the RLN3nLTRC register.</li> </ul>	<ul style="list-style-type: none"> <li>When the transmission of the data specified in the RLN3nLUTDR or RLN3nLUWTDR register is completed, but the next transmission data is not specified.</li> <li>When the transmission of the data in the UART buffer is completed and the RTS bit in the RLN3nLTRC register is cleared.</li> <li>After transition to LIN reset mode</li> </ul>	UTS flag in RLN3nLST register	—
Reception status	<ul style="list-style-type: none"> <li>When a start bit is detected.</li> </ul>	<ul style="list-style-type: none"> <li>When a sampling point for stop bits is detected</li> <li>After transition to LIN reset mode</li> </ul>	URS flag in RLN3nLST register	—

Note 1. Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when not in the LIN reset mode sets the ERR flag in the RLN3nLST register to 0.

## 17.8.5 Error Statuses

### Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

**Table 17.94** lists applicable status types.

**Table 17.94 Types of Statuses in UART Mode**

Status	Error Detection Condition	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data monitored on the reception pin do not match <sup>*1</sup>	Continues until the transmission of the specified transmission data is completed.	Enabled	BER flag in RLN3nLEST register
Overrun error	After received data is stored in the RLN3nLURDR register, the next data is received before the previous data is read. (In this case, no data is stored in the RLN3nLURDR register).	— (Reception is completed by the time this error is detected)	Enabled	OER flag in RLN3nLEST register
Framing error	When the first stop bit is low level in the reception processing.	— (Reception is completed by the time this error is detected)	Enabled	FER flag in RLN3nLEST register
Parity error	The received parity value fails to match the parity value calculated from the received data	Continues until the data reception is completed.	Disabled <sup>*2</sup>	UPER flag in RLN3nLEST register
Expansion bit detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register.	—	Enabled	EXBT flag in RLN3nLEST register
ID match detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register and the 8-bit received data excluding the expansion bit matches the value of the RLN3nLIDB register.	—	Enabled	IDMT flag in RLN3nLEST register

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLN3nLBFC register to 10<sub>B</sub> (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

## 17.9 LIN Self-Test Mode

The LIN/UART interface provides a LIN self-test mode. When the LIN/UART interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and they are internally connected within the LIN/UART interface. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following four modes:

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception
- LIN slave self-test mode (transmission): Header reception and response transmission
- LIN slave self-test mode (reception): Header reception and response reception

In LIN self-test mode, the operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate setting is the LIN communication clock source/16 [bps]. (The NSPB bits in the RLN3nLWBR register should be set to 0000<sub>B</sub> or 1111<sub>B</sub>.)

In addition, in LIN self-test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- LIN slave mode (Auto baud rate)
- Frame/response timeout error

Do not use these functions.

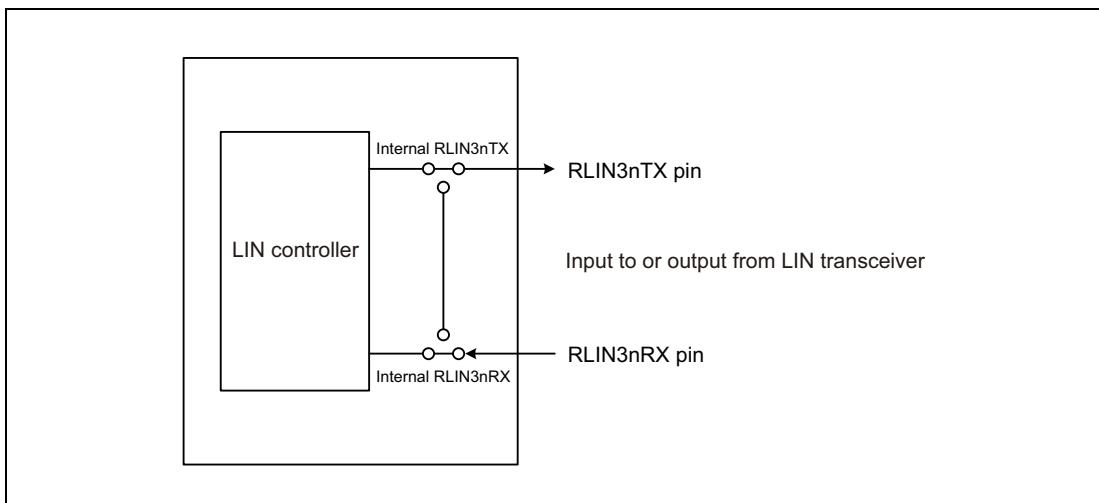


Figure 17.33 Connection in LIN Reset Mode, LIN Mode, and UART Mode

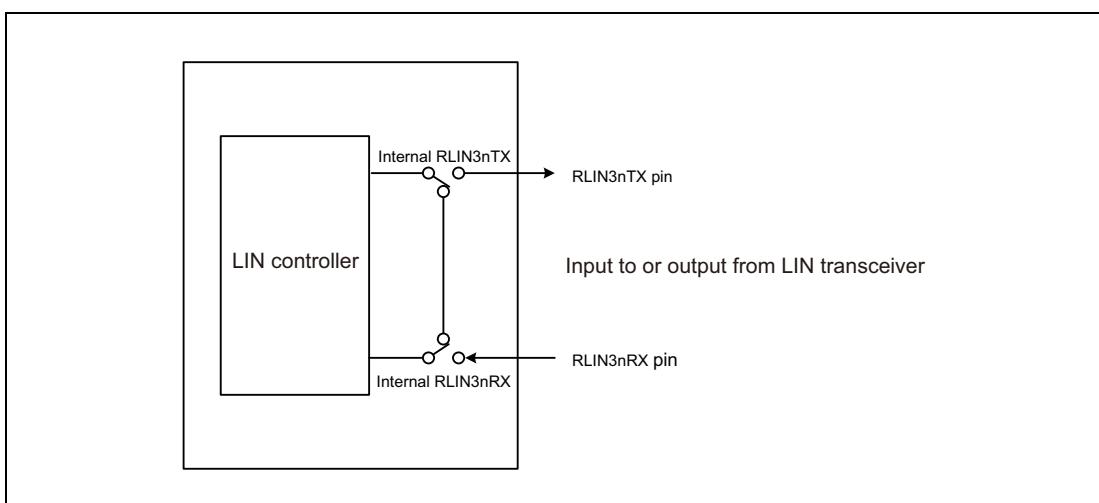


Figure 17.34 Connection in LIN Self-Test Mode

### 17.9.1 Transition to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.

The LSTM bit in the RLN3nLSTC register set to 1 indicates that the mode has transitioned to the LIN self-test mode.

When transitioning to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be consecutively written three times to the LIN self-test control register, as follows:

- Transition to LIN reset mode  
Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).  
Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode  
LMD bits in RLN3nLMD = 00<sub>B</sub> (LIN master mode) or 11<sub>B</sub> (LIN slave mode [fixed baud rate])
- 1st write: RLN3nLSTC register = 1010 0111<sub>B</sub> (A7<sub>H</sub>)
- 2nd write: RLN3nLSTC register = 0101 1000<sub>B</sub> (58<sub>H</sub>)
- 3rd write: RLN3nLSTC register = 0000 0001<sub>B</sub> (01<sub>H</sub>)
- Verify the transition to LIN self-test mode  
Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the 1st write (A7<sub>H</sub>) is written twice by mistake, the transition to LIN self-test mode is aborted. The above sequence should be retried from the 1st write step. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also aborted.

### 17.9.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLBRP0 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLBRP1 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLMD register = 00xx xx00<sub>B</sub><sup>\*1</sup>

- Configure the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx<sub>B</sub><sup>\*2</sup>

RLN3nLEDE register = x000 x0xx<sub>B</sub>

- Configure the break field and space related registers.

RLN3nLBFC register = 00xx xxxx<sub>B</sub>

RLN3nLSC register = 00xx 0xxx<sub>B</sub>

- Exit the LIN reset mode.

Write 11<sub>B</sub> to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11<sub>B</sub>.

- Configure the transmission frame related registers.

RLN3nLDFC register = 00x1 xxxx<sub>B</sub>

RLN3nLIDB register = xxxx xxxx<sub>B</sub>

RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx<sub>B</sub>

- Start header transmission → response transmission

Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).

The LIN master self-test mode (transmission) is executed, interrupt is generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/UART interface. To abort the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

**Note:** x: Arbitrary value can be specified.

**Note 1.** The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS bit in the RLN3nLMD register. Therefore, configuration of these registers is not necessary.

**Note 2.** If necessary, configure the related registers described in **Section 6, Exceptions/Interrupts**.

**Note 3.** When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).

The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up transmission flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

### 17.9.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLBRP0 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLBRP1 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLMD register = 00xx xx00<sub>B</sub><sup>\*1</sup>

- Configure the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx<sub>B</sub><sup>\*2</sup>

RLN3nLEDE register = x000 x0xx<sub>B</sub>

- Configure the break field and space related registers.

RLN3nLBFC register = 00xx xxxx<sub>B</sub>

RLN3nLSC register = 00xx 0xxx<sub>B</sub><sup>\*1</sup>

- Exit the LIN reset mode.

Write 11<sub>B</sub> to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11<sub>B</sub>.

- Configure the reception frame related registers.

RLN3nLDFC register = 00x0 xxxx<sub>B</sub>

RLN3nLIDB register = xxxx xxxx<sub>B</sub>

RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx xxxx<sub>B</sub>

RLN3nCBR register = xxxx xxxx<sub>B</sub>

Since the checksum value to be transmitted is not automatically calculated, perform the calculation and specify the calculated value in the RLN3nLCBR register.

- Start header transmission → response reception

Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).

The LIN master self-test mode (reception) is executed, interrupt is generated, and status and error status are also updated. To abort the LIN master self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDRB<sub>b</sub> (<sub>b</sub> = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the specified value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.

- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

**Note:** x: Arbitrary value can be specified.

- Note 1.** The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, configuration of these registers is not necessary.

**Note 2.** If necessary, configure the related registers described in **Section 6, Exceptions/Interrupts**.

**Note 3.** When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).

The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up reception flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

#### 17.9.4 Transmission in LIN Slave Self-Test Mode

To execute a self-test on LIN slave transmission, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 xxx0<sub>B</sub><sup>\*1</sup>

RLN3nLBRP0 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLBRP1 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLMD register = 00x x0011<sub>B</sub>

- Configure the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx<sub>B</sub><sup>\*2</sup>

RLN3nLEDE register = xx0x x00x<sub>B</sub>

- Configure the break field and space related registers.

RLN3nLBFC register = 0000 000x<sub>B</sub><sup>\*3</sup>

RLN3nLSC register = 00xx 0001<sub>B</sub>

- Exit the LIN reset mode.

Write 11<sub>B</sub> to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11<sub>B</sub>.

- Configure the transmission frame related registers.

RLN3nLDFC register = 00x1 xxxx<sub>B</sub>

RLN3nLIDB register = xxxx xxxx<sub>B</sub>

RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx<sub>B</sub>

- Header reception → response transmission started

Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started).

(The header reception and the response transmission are executed in this order, without manipulating the RTS bit in the RLN3nLTRC register.)

The LIN slave self-test mode (transmission) is executed, interrupt is generated, and status and error status are also updated.

The checksum is automatically calculated by the LIN/UART interface. To abort the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.

- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

**Note:** x: Arbitrary value can be specified.

**Note 1.** The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, and the RLN3nLBRP1 register. Therefore, configuration of these registers is not necessary.

**Note 2.** If necessary, configure the related registers described in **Section 6, Exceptions/Interrupts**

**Note 3.** According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

**Note 4.** When the successful header reception interrupt and the successful response transmission interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).

The time from setting of the successful header reception flag to setting of the successful response/wake-up transmission flag is calculated by using the following formula.

$$10 \times (\text{number of data bytes} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

### 17.9.5 Reception in LIN Slave Self-Test Mode

To execute a self-test on LIN slave reception, perform the following procedure:

- Configure the baud rate, noise filter, and interrupt output related registers.

RLN3nLWBR register = 0000 xxxx0<sub>B</sub><sup>\*1</sup>

RLN3nLBRP0 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLBRP1 register = xxxx xxxx<sub>B</sub><sup>\*1</sup>

RLN3nLMD register = 00xx 0011<sub>B</sub>

- Configure the interrupt enable and error enable related registers.

RLN3nLIE register = 0000 xxxx<sub>B</sub><sup>\*2</sup>

RLN3nLEDE register = xx0x x00x<sub>B</sub>

- Configure the break field and space related registers.

RLN3nLBFC register = 0000 000x<sub>B</sub><sup>\*3</sup>

RLN3nLSC register = 00xx 0001<sub>B</sub><sup>\*1</sup>

- Exit the LIN reset mode.

Write 11<sub>B</sub> to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11<sub>B</sub>.

- Configure the reception frame related registers.

RLN3nLDFA register = 00x0 xxxx<sub>B</sub>

RLN3nLIDB register = xxxx xxxx<sub>B</sub>

RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx<sub>B</sub>

RLN3nCBR register = xxxx xxxx<sub>B</sub>

Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.

- Header reception → response reception started

Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/reception started).

(Without any setting of the RTS bit in the RLN3nLTRC register, the header reception and the response reception are executed in this order.)

The LIN slave self-test mode (reception) is executed, interrupt is generated, and status and error status are also updated. To abort the LIN slave self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.

- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBR<sub>b</sub> (<sub>b</sub> = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the specified value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.

- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

**Note:** x: Arbitrary value can be specified.

**Note 1.** The settings of the following registers are not reflected to the operation of the LIN self-test mode: the LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register, and the IBS bit in the RLN3nLSC register. Therefore, configuration of these registers is not necessary.

**Note 2.** If necessary, configure the related registers described in **Section 6, Exceptions/Interrupts**.

**Note 3.** According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

**Note 4.** When the successful header reception interrupt and the successful response reception interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).

The time required from the set of the successful header reception flag to the set of the successful response/wake-up reception flag is calculated by the following formula.

$$10 \times (\text{number of data bytes} + 1) [\text{Tbit}]$$

$$1 \text{ Tbit} = 1/\text{frequency of LIN communication clock source} \times 16$$

### 17.9.6 Exiting LIN Self-Test Mode

To exit LIN self-test mode, perform the following procedure:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.  
If the OMM1 and OMM0 bits in the RLN3nLMST register are not  $11_B$ , write  $11_B$  to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register are set to  $11_B$ , transition to LIN reset mode.
- Verify that LIN/UART interface has exited LIN self-test mode.  
Read the LSTM bit in the RLN3nLSTC register; verify that it is 0 (not in LIN self-test mode)
- Verify the transition to LIN reset mode.  
Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

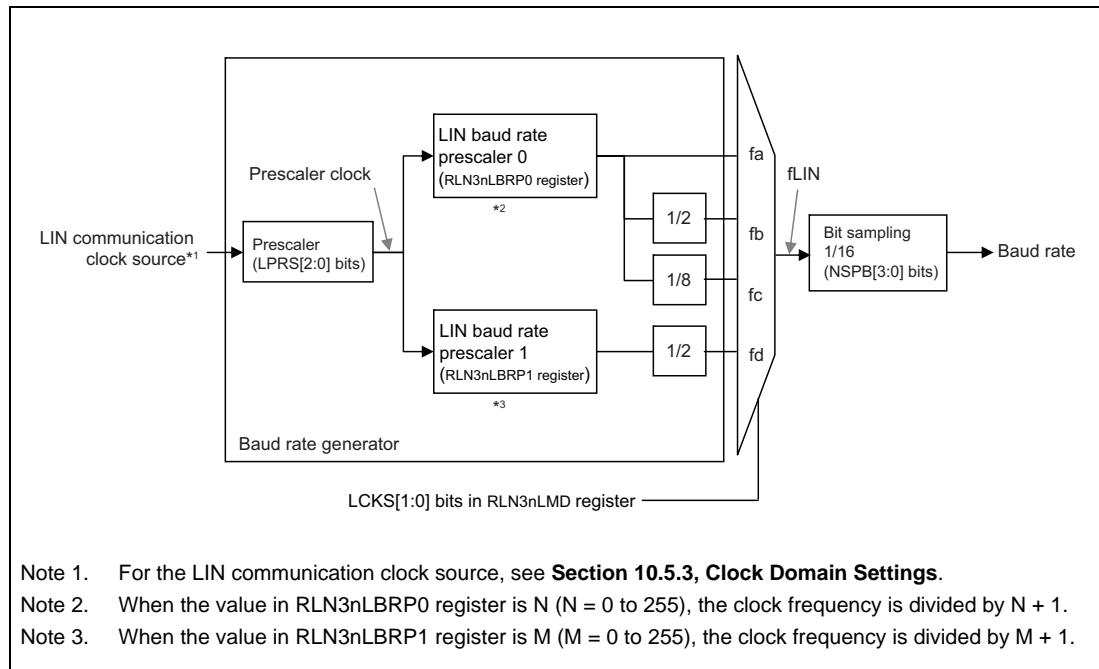
## 17.10 Baud Rate Generator

The prescaler clock is obtained by dividing the LIN communication clock source frequency by the prescaler, and the LIN system clock ( $f_{LIN}$ ) is obtained by dividing the prescaler clock frequency by the baud rate generator. The clock obtained by dividing the LIN system clock frequency ( $f_{LIN}$ ) by the number of samples is the baud rate. The inverse of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two types of baud rate generators. The baud rate generators are switched according to the mode used.

### 17.10.1 LIN Master Mode

**Figure 17.35** shows a block diagram of baud rate generation in LIN master mode.



**Figure 17.35 Block Diagram of Baud Rate Generation in LIN Master Mode**

By setting the RLN3nLBRP0 register so that  $fa$  is 307200 Hz ( $= 19200 \times 16$ ), the resulting bit rates are  $fa = 19200 \times 16$ ,  $fb = 9600 \times 16$ , and  $fc = 2400 \times 16$ . These bit rates are frequency-divided by 16 in the bit timing generator, enabling baud rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by configuring the RLN3nLBRP1 register so that  $fd$  is 166672 Hz ( $= 10417 \times 16$ ), the resulting bit rate is  $fd = 10417 \times 16$ . This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

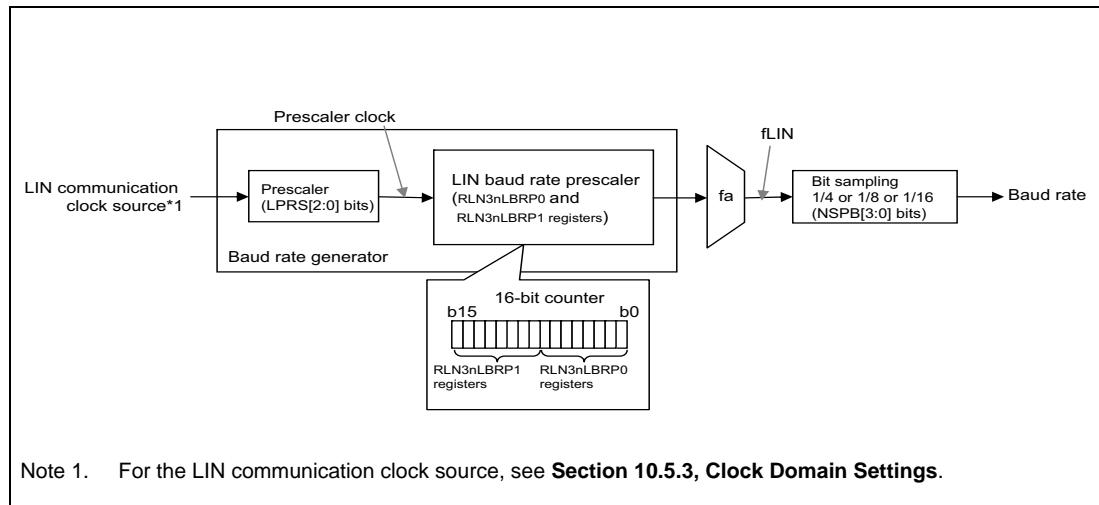
Baud rate of LIN master

$$\begin{aligned}
&= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \\
&\quad \div (\text{RLN3nLBRP0} + 1) \div 16 \text{ [bps]} \text{ (When } fa \text{ is selected for } f_{LIN}) \\
&= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \\
&\quad \div (\text{RLN3nLBRP0} + 1) \div 2 \div 16 \text{ [bps]} \text{ (When } fb \text{ is selected for } f_{LIN}) \\
&= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \\
&\quad \div (\text{RLN3nLBRP0} + 1) \div 8 \div 16 \text{ [bps]} \text{ (When } fc \text{ is selected for } f_{LIN})
\end{aligned}$$

$$\begin{aligned}
 &= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \\
 &\quad \div (\text{RLN3nLBRP1} + 1) \div 2 \div 16 \text{ [bps]} \text{ (When fd is selected for } f_{\text{LIN}})
 \end{aligned}$$

### 17.10.2 LIN Slave Mode

Figure 17.36 shows a block diagram of baud rate generation in LIN slave mode.



**Figure 17.36 Block Diagram of Baud Rate Generation in LIN Slave Mode**

In LIN slave mode (auto baud rate), the baud rate can be specified in the range of 1 kbps to 20 kbps. Configure the prescaler clock as follows according to the target baud rate:

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz *1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

**Note 1.** Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to "0011<sub>B</sub>" (4 samplings).

The formula for baud rate is described below.

Baud rate of LIN slave

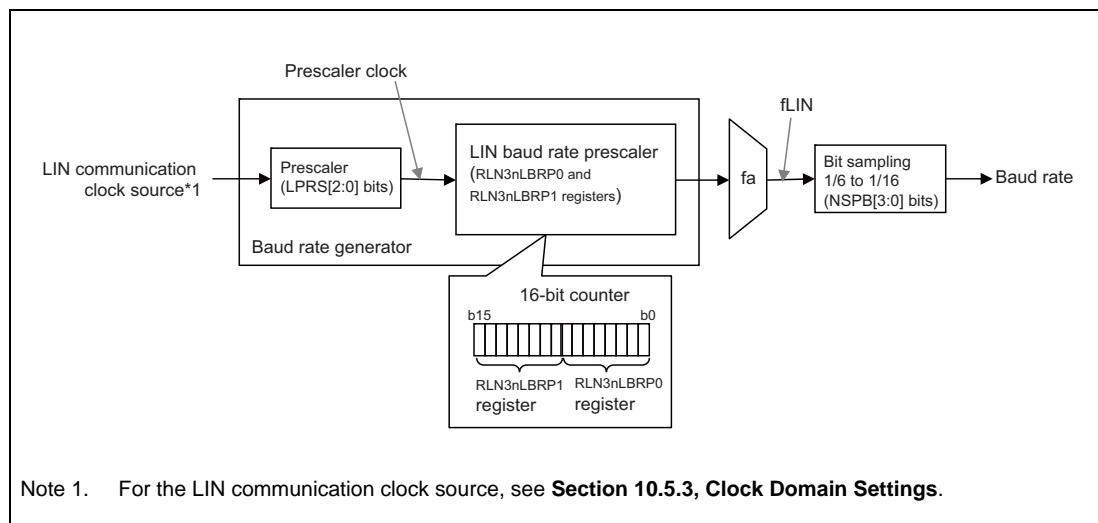
$$\begin{aligned}
 &= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \\
 &\quad \div (\text{RLN3nLBRP01} + 1) \div 16 \text{ [bps]} \text{ ([Fixed baud rate])} \\
 &= \{\text{Frequency of LIN communication clock source}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \\
 &\quad \div (\text{RLN3nLBRP01} + 1) \div 4 \text{ or } 8 \text{ [bps]} \text{ ([Auto baud rate])}
 \end{aligned}$$

**NOTE**

For a LIN slave with fixed baud rate, set the NSPB[3:0] bit to “0000<sub>B</sub>” (16 samplings) or “1111<sub>B</sub>” (16 samplings). For a LIN slave with auto baud rate, set the NSPB[3:0] bits to “0011<sub>B</sub>” (4 samplings) or “0100<sub>B</sub>” (8 samplings).

### 17.10.3 UART Mode

**Figure 17.37** shows a block diagram of baud rate generation in UART mode.



**Figure 17.37 Block Diagram of Baud Rate Generation in UART Mode**

UART baud rate is calculated with the following formula:

$$\begin{aligned} \text{UART baud rate} \\ = & \{\text{LIN communication clock source frequency}\} \times (\text{RLN3nLWBR.LPRS[2:0] selected clock}) \div \\ & (\text{RLN3nLBRP01} + 1) \div \{\text{RLN3nLWBR.NSPB[3:0] selected count}\} [\text{bps}] \end{aligned}$$

## 17.11 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLN3nLMD register to 0 (use the noise filter), the noise filter is activated. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

**Figure 17.38** shows the configuration of the noise filter, **Figure 17.39** shows an example of a noise filter circuit, and **Figure 17.40** shows the determination of the received data when the noise filter is used.

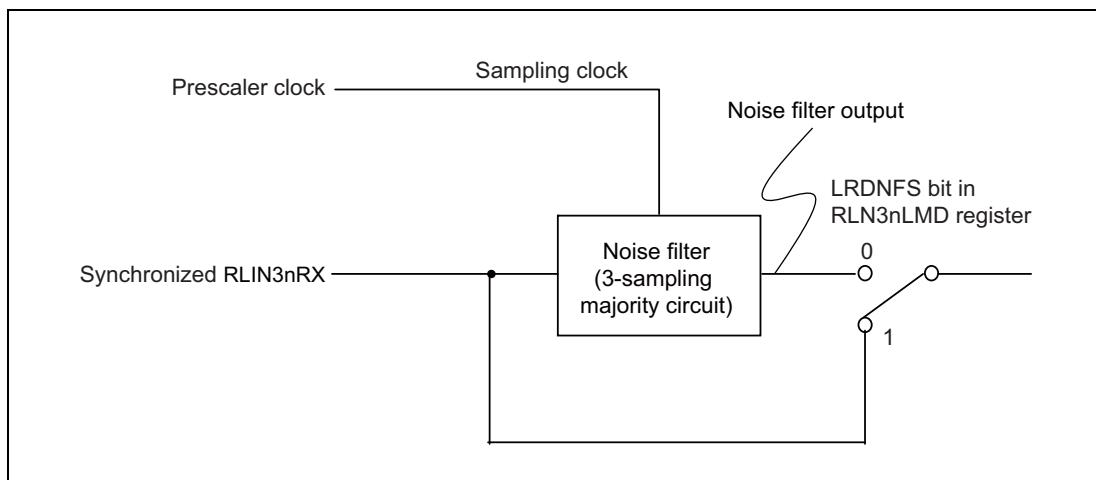


Figure 17.38 Configuration of Noise Filter

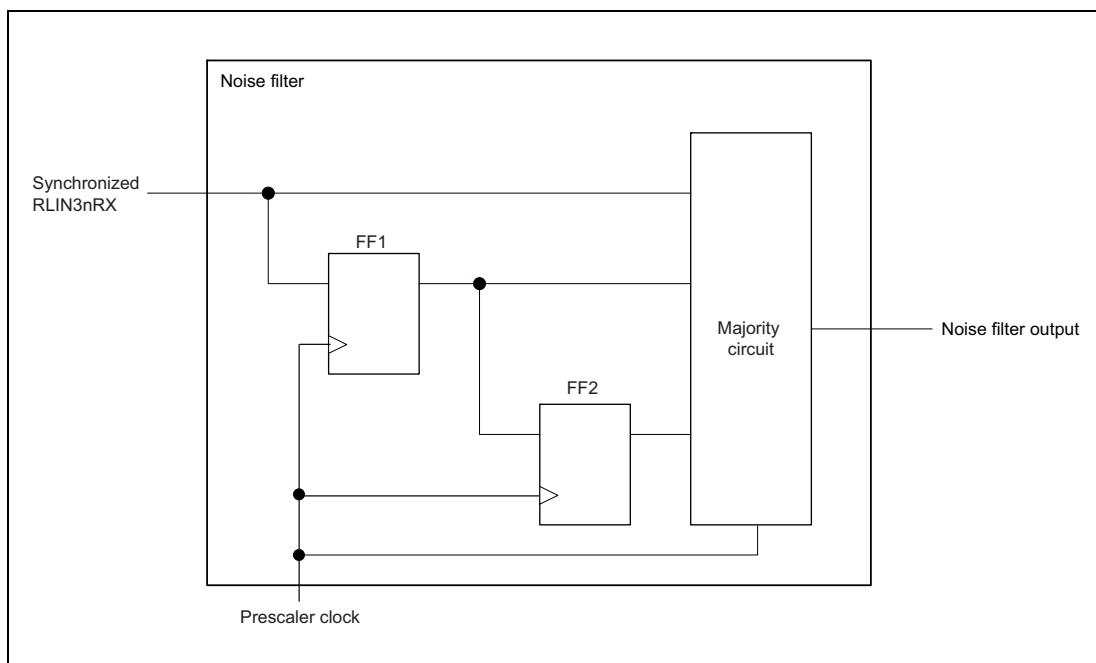


Figure 17.39 Example of Noise Filter Circuit

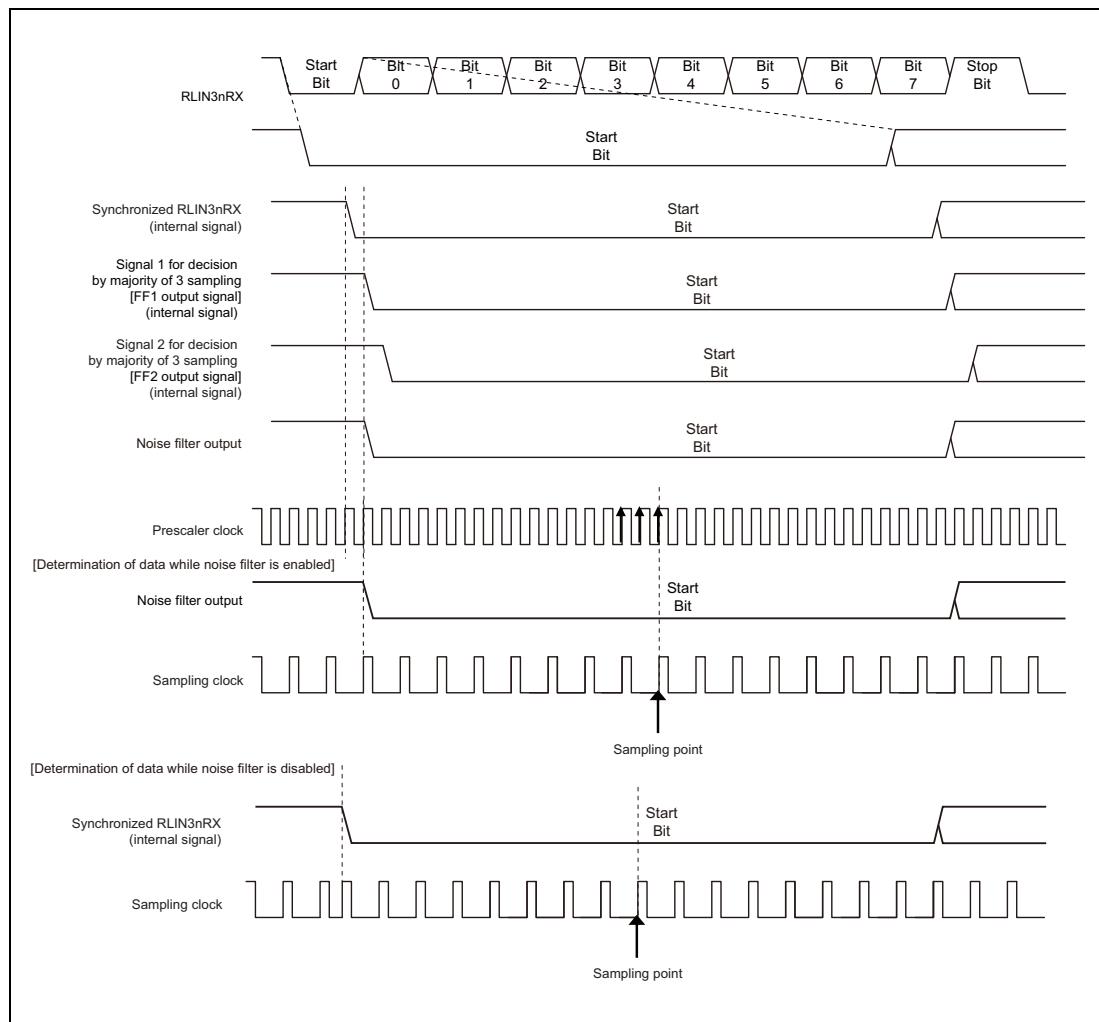


Figure 17.40 Determination of Received Data when Noise Filter is Used

## 17.12 Usage Notes

Applicable products of these notes are shown below.

**Table 17.95 Applicable products**

Series Name	Pin Number	Flash Size	Product Name	Applicable or Not
RH850/F1L	176pin	2MB	Except below products	Applicable
RH850/F1L for Gateway	144pin	1.5MB 1MB 768KB	R7F701xxxxAFP#YJ4 R7F701xxxxAFP#YK4 R7F701xxxxAFP#YB4 R7F701xxxxAFP#AA4 R7F701xxxxAFP#KA4	Not Applicable
	100pin 80pin 64pin 48pin	1MB 768KB	Except below products	Applicable
			R7F701xxxxAFP#YJ2 R7F701xxxxAFP#YK2 R7F701xxxxAFP#YB2 R7F701xxxxAFP#AA2 R7F701xxxxAFP#KA2	Not Applicable
		512KB 384KB 256KB	Except below products	Applicable
			R7F701xxxxAFP#YJ4 R7F701xxxxAFP#YK4 R7F701xxxxAFP#YB4 R7F701xxxxAFP#AA4 R7F701xxxxAFP#KA4	Not Applicable

### 17.12.1 Note on LIN Master mode

If in Frame separate mode (RLN3nLDFC.FSM=1) the start of the response transmission (Writing 1 to RLN3nLTRC.RTS) coincides with an error <sup>\*1</sup>occurrence, the response transmission will not be started.

Additionally, after receiving the header of next frame, unexpected data may be sent on the response transmission, even though "1" is not written to RLN3nLTRC.RTS. In case of an error occurrence, re-initialize the LIN/UART module (RLIN3) by transiting to LIN reset mode.

**Note 1.** Bit error, Physical bus error, Frame/Response timeout error, Framing error, Response preparation error.

### 17.12.2 Note on LIN Slave mode (Fixed baud rate)

If the low-level width, set by Reception Break (Low-Level) Detection Width Setting bit (RLN3nLBFC.LBLT), is detected during the response transmission (RLN3nLDFC.RCDS=1 and RLN3nLTRC.RTS=1), the response transmission will be interrupted and the RLIN3nTX pin will keep the same status as at the low-level width detection.

If in LIN Slave mode with fixed baud rate, set RLN3nLEDE.BERE=1 (Bit error detection is enable) and RLN3nLSC.IBHS [2:0] = 001<sub>B</sub> to 111<sub>B</sub> (Except 0 Tbit as response space).

If the Response Space Setting with 0 Tbit (RLN3nLSC.IBHS [2:0] = 000<sub>B</sub>), set RLN3nLEDE.BERE=1 and RLN3nLEDE.TERE=1 (Timeout error detection is enable). And transit to LIN reset mode after a timeout error occurrence to initialize the LIN/UART module (RLIN3).

### 17.12.3 Note on LIN Slave mode (Auto baud rate)

If the conditions (1.) and (2.) are met subsequently, the header might not be received due to false recognition of the ID field. However, the next header will be received correctly.

1. The falling edge is detected from the sampling point of the Stop-bit and Response/Inter-Byte-Space which Bit error is occurred to the end of its bit.
2. The falling edge of the start bit of the ID field is detected from the sampling point of the sync field of the next reception to the end of its bit.

## Section 18 I<sup>2</sup>C Bus Interface (RIIC)

This section contains a generic description of the I<sup>2</sup>C Bus Interface (RIIC).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RIIC.

### 18.1 Features of RH850/F1L RIIC

#### 18.1.1 Number of Units and Channels

This microcontroller has the following number of RIIC units.

Each RIIC unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

**Table 18.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units				1		
Name				RIICn (n = 0)		

**Table 18.2 Index**

Index	Description
n	Throughout this section, the individual RIIC units are identified by the index “n” (n = 0): for example, RIICnCR1 is the I <sup>2</sup> C bus control register 1.

#### 18.1.2 Register Base Address

RIIC base address is listed in the following table.

RIIC register addresses are given as offsets from the base address.

**Table 18.3 Register Base Address**

Base Address Name	Base Address
<RIIC0_base>	FFCA 0000 <sub>H</sub>

#### 18.1.3 Clock Supply

The RIIC clock supply is shown in the following table.

**Table 18.4 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
RIICn	PCLK* <sup>1</sup>	CPUCLK2
	Register access clock	CPUCLK2

Note 1. Set the period of PCLK no greater than 1/2 of the width at high level of the SCL clock.

### 18.1.4 Interrupt Requests

RIIC interrupt requests are listed in the following table.

**Table 18.5 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>RIIC0</b>			
INTIICnEE	RIIC communication error / event generation interrupt	71	—
INTIICnRI	RIIC Receive complete interrupt	70	20 (channels 0 to 7)
INTIICnTI	RIIC Transmit data empty interrupt	68	19 (channels 0 to 7)
INTIICnTEI	RIIC Transmit end interrupt	69	—

### 18.1.5 Reset Sources

RIIC reset sources are listed in the following table. RIIC is initialized by these reset sources.

**Table 18.6 Reset Sources**

Unit Name	Reset Source
RIIC0	All reset sources (ISORES)

### 18.1.6 External Input/Output Signals

External input/output signals of RIIC are listed below.

**Table 18.7 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal
<b>RIIC0</b>		
RIICnSCL	Serial clock I/O pin	RIIC0SCL
RIICnSDA	Serial data I/O pin	RIIC0SDA

When using these ports, the PBDCn register for the corresponding port and the corresponding bit in the PODCn register must be set to 1.

## 18.2 Overview

### 18.2.1 Functional Overview

#### Communications format

- I<sup>2</sup>C bus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times according to the specified transfer rate

#### Transfer rate

Up to 400 kbps

#### SCL clock

- For master operation, the duty cycle of the SCL clock is selectable in the following range:
  - 0% < Duty < 100%

#### Issuing and detecting conditions

Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.

#### Slave address

- Up to three slave-address settings can be made.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses and device ID addresses are detectable.

#### Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

### Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

### SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

### Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching of internal and line levels for transmit data.
- Loss of arbitration due to detection of the start condition while the bus is busy can be detected (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching can be detected.
- Loss of arbitration due to non-matching of internal and line levels for data can be detected in slave transmission.

### Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

### Noise removal

The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.

### Interrupt sources

- Four sources:
  - Error in transfer or occurrence of events (detection of arbitration loss, NACK, time-out, a start condition including a restart condition, or a stop condition)
  - Reception complete (including matching with a slave address)
  - Transmit-data-empty (including matching with a slave address)
  - Transmission complete

### 18.2.2 Block Diagram

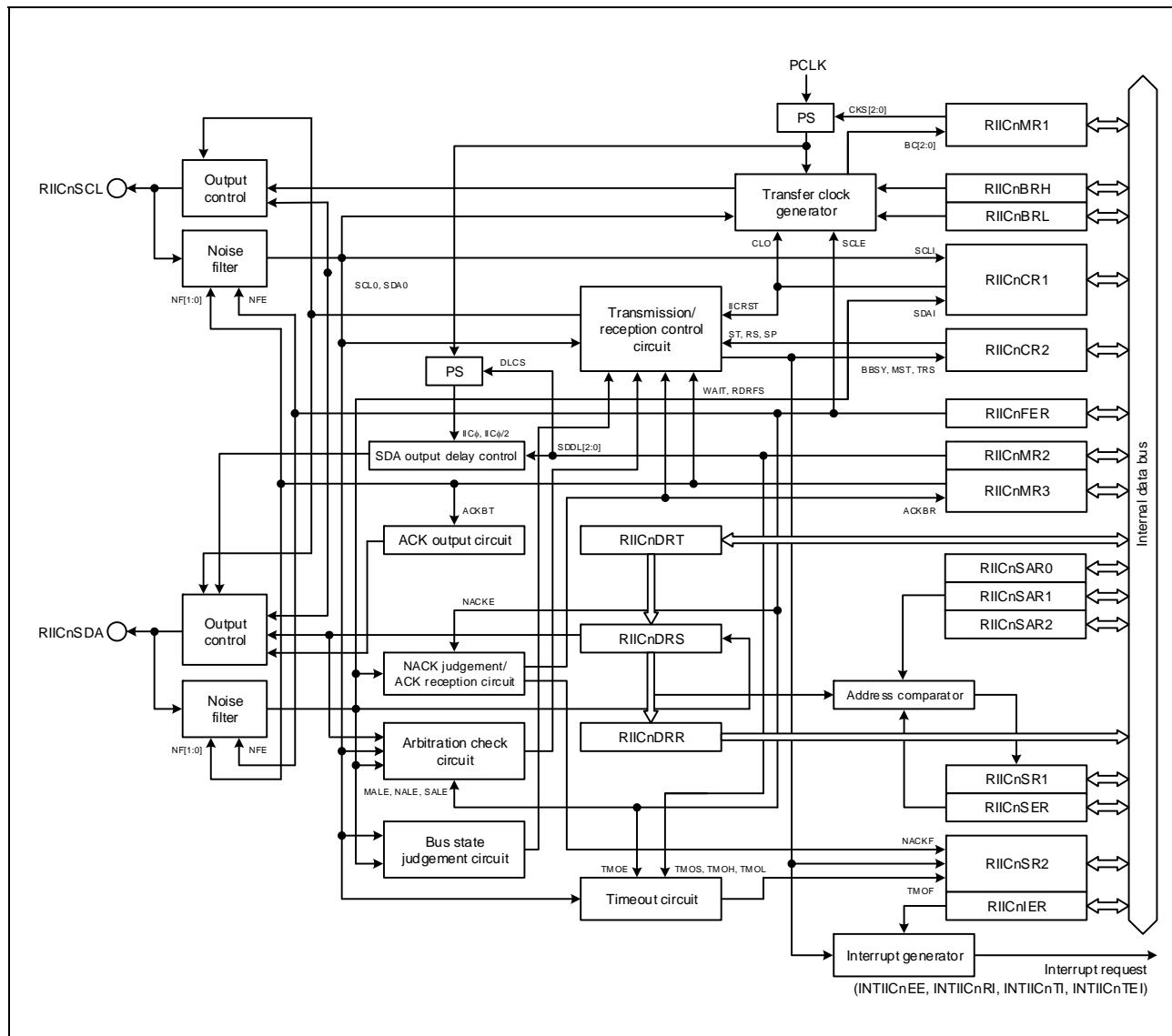
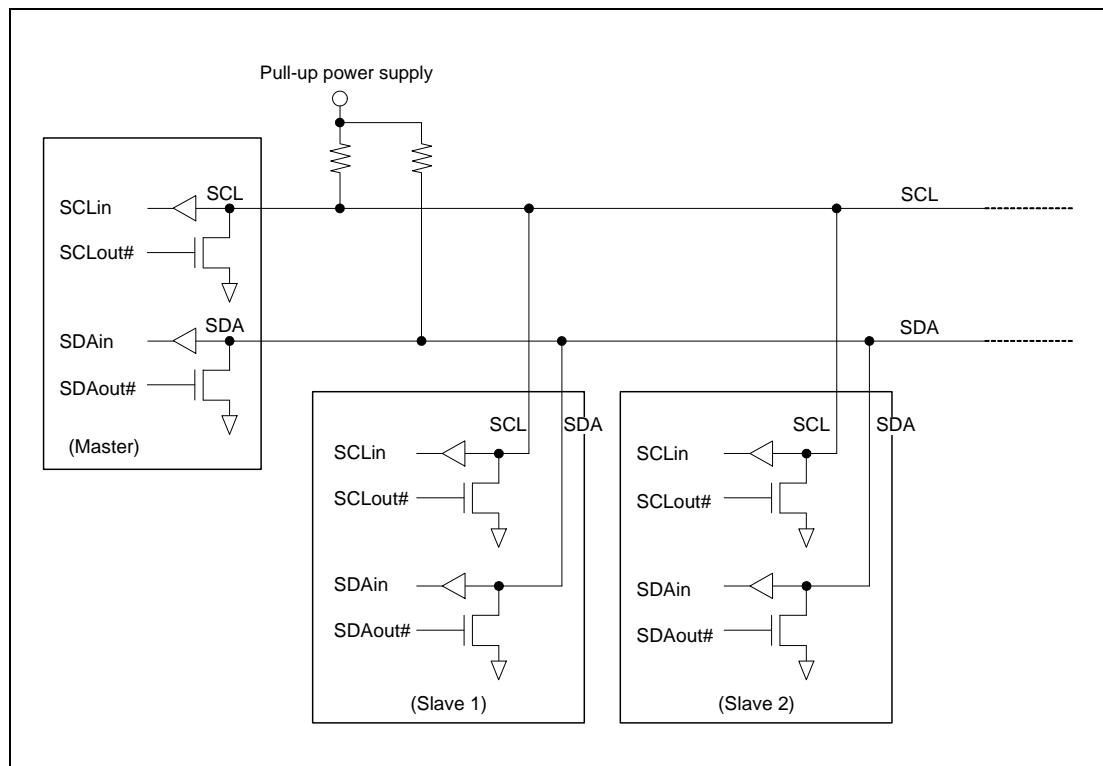


Figure 18.1 Block Diagram of RIIC



**Figure 18.2 Connections to the External Circuit by the I/O Pins (I<sup>2</sup>C Bus Configuration Example)**

## 18.3 Registers

### 18.3.1 List of Registers

RIIC registers are listed in the table below.

For details about <RIICn\_base>, see **Section 18.1.2, Register Base Address**.

Table 18.8 List of Registers

Module Name	Register Name	Symbol	Address
RIICn	I <sup>2</sup> C Bus Control Register 1	RIICnCR1	<RIICn_base> + 0000 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Control Register 2	RIICnCR2	<RIICn_base> + 0004 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Mode Register 1	RIICnMR1	<RIICn_base> + 0008 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Mode Register 2	RIICnMR2	<RIICn_base> + 000C <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Mode Register 3	RIICnMR3	<RIICn_base> + 0010 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Function Enable Register	RIICnFER	<RIICn_base> + 0014 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Status Enable Register	RIICnSER	<RIICn_base> + 0018 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Interrupt Enable Register	RIICnIER	<RIICn_base> + 001C <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Status Register 1	RIICnSR1	<RIICn_base> + 0020 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Status Register 2	RIICnSR2	<RIICn_base> + 0024 <sub>H</sub>
RIICn	I <sup>2</sup> C Slave Address Register 0	RIICnSAR0	<RIICn_base> + 0028 <sub>H</sub>
RIICn	I <sup>2</sup> C Slave Address Register 1	RIICnSAR1	<RIICn_base> + 002C <sub>H</sub>
RIICn	I <sup>2</sup> C Slave Address Register 2	RIICnSAR2	<RIICn_base> + 0030 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Bit Rate Low-Level Register	RIICnBRL	<RIICn_base> + 0034 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Bit Rate High-Level Register	RIICnBRH	<RIICn_base> + 0038 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Transmit Data Register	RIICnDRT	<RIICn_base> + 003C <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Receive Data Register	RIICnDRR	<RIICn_base> + 0040 <sub>H</sub>
RIICn	I <sup>2</sup> C Bus Shift Register	RIICnDRS	—

### 18.3.2 RIICnCR1 — I<sup>2</sup>C Bus Control Register 1

**Access:** RIICnCR1 register can be read or written in 32-bit units.  
 RIICnCR1L and RIICnCR1H registers can be read or written in 16-bit units.  
 RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH registers can be read or written in 8-bit units.

**Address:** RIICnCR1: <RIICn\_base> + 0000<sub>H</sub>  
 RIICnCR1L: <RIICn\_base> + 0000<sub>H</sub>, RIICnCR1H: <RIICn\_base> + 0002<sub>H</sub>  
 RIICnCR1LL: <RIICn\_base> + 0000<sub>H</sub>, RIICnCR1LH: <RIICn\_base> + 0001<sub>H</sub>,  
 RIICnCR1HL: <RIICn\_base> + 0002<sub>H</sub>, RIICnCR1HH: <RIICn\_base> + 0003<sub>H</sub>

**Value after reset:** 0000 001F<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	W	R/W	R/W	R	R	R

Table 18.9 RIICnCR1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ICE	I <sup>2</sup> C Bus Interface Enable 0: Disabled (the RIICnSCL and RIICnSDA pins are not driven). 1: Enabled (the RIICnSCL and RIICnSDA pins are driven). (This bit selects an RIIC reset or internal reset in combination with the IICRST bit.)
6	IICRST	I <sup>2</sup> C Bus Interface Internal Reset 0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)
5	CLO	Extra SCL Clock Cycle Output 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)
4	SOWP	SCLO/SDAO Write Protect 0: Bits SCLO and SDAO can be written. 1: Bits SCLO and SDAO are protected. (This bit is read as 1.)
3	SCLO	SCL Output Control/Monitor <ul style="list-style-type: none"> <li>• Read:<ul style="list-style-type: none"> <li>0: The RIIC has driven the RIICnSCL pin low.</li> <li>1: The RIIC has released the RIICnSCL pin.</li> </ul></li> <li>• Write:<ul style="list-style-type: none"> <li>0: The RIIC drives the RIICnSCL pin low.</li> <li>1: The RIIC releases the RIICnSCL pin.</li> </ul></li> </ul>

**Table 18.9 RIICnCR1 Register Contents (2/2)**

Bit Position	Bit Name	Function
2	SDAO	SDA Output Control/Monitor <ul style="list-style-type: none"> <li>• Read: 0: The RIIC has driven the RIICnSDA pin low. 1: The RIIC has released the RIICnSDA pin.</li> <li>• Write: 0: The RIIC drives the RIICnSDA pin low. 1: The RIIC releases the RIICnSDA pin.</li> </ul>
1	SCLI	SCL Line Monitor 0: RIICnSCL line is low. 1: RIICnSCL line is high.
0	SDAI	SDA Line Monitor 0: RIICnSDA line is low. 1: RIICnSDA line is high.

**SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)**

These bits are used to directly control the RIICnSDA and RIICnSCL signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

**CLO Bit (Extra SCL Clock Cycle Output)**

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see **Section 18.13.2, Extra SCL Clock Cycle Output Function**.

**IICRST Bit (I<sup>2</sup>C Bus Interface Internal Reset)**

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. **Table 18.10** lists the types of RIIC reset.

The RIIC reset resets all registers (except ICE and IICRST) including the RIICnCR2.BBSY flag and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the I<sup>2</sup>C bus shift register (RIICnDRS), and the I<sup>2</sup>C bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see **Section 18.14, Reset Function of RIIC**.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

#### CAUTION

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

**Table 18.10 RIIC Resets**

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers (except ICE and IICRST) and internal states of the RIIC.
	1	Internal reset	Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC.

#### ICE Bit (I<sup>2</sup>C Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see **Table 18.10, RIIC Resets**.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 stops driving of the RIICnSCL and RIICnSDA pins.

#### CAUTION

Though the output from RIICnSDA or RIICnSCL is disabled while the ICE bit is 0, the input to RIICnSDA or RIICnSCL is enabled. The RIICnSCL and RIICnSDA pin functions should not be assigned to the RIIC. If assigned, it causes the slave addresses to be compared.

### 18.3.3 RIICnCR2 — I<sup>2</sup>C Bus Control Register 2

**Access:** RIICnCR2 register can be read or written in 32-bit units.  
 RIICnCR2L and RIICnCR2H registers can be read or written in 16-bit units.  
 RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH registers can be read or written in 8-bit units.

**Address:** RIICnCR2: <RIICn\_base> + 0004<sub>H</sub>  
 RIICnCR2L: <RIICn\_base> + 0004<sub>H</sub>, RIICnCR2H: <RIICn\_base> + 0006<sub>H</sub>  
 RIICnCR2LL: <RIICn\_base> + 0004<sub>H</sub>, RIICnCR2LH: <RIICn\_base> + 0005<sub>H</sub>,  
 RIICnCR2HL: <RIICn\_base> + 0006<sub>H</sub>, RIICnCR2HH: <RIICn\_base> + 0007<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R

Table 18.11 RIICnCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	BBSY	Bus Busy Detection Flag 0: The I <sup>2</sup> C bus is released (the bus is free). 1: The I <sup>2</sup> C bus is occupied (the bus is busy).
6	MST <sup>*1</sup>	Master/Slave Mode 0: Slave mode 1: Master mode
5	TRS <sup>*1</sup>	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SP	Stop Condition Issuance Request 0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.
2	RS	Restart Condition Issuance Request 0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.
1	ST	Start Condition Issuance Request 0: Does not request to issue a start condition. 1: Requests to issue a start condition.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. When the RIICnMR1.MTWP bit is set to 1, the MST and TRS bits can be written to.

#### ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see **Section 18.12, Start Condition/Restart Condition/Stop Condition Issuing Function.**

[Setting condition]

When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

#### CAUTION

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).

Note that arbitration may be lost as the start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

### RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see **Section 18.12, Start Condition/Restart Condition/Stop Condition Issuing Function.**

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

#### CAUTIONS

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is recommended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the RS bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

### SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see **Section 18.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

#### CAUTIONS

1. Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

### TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode (1 or 0) by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc.

Although writing to the TRS bit is possible when the RIICnMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave transmit mode, a restart condition is detected (a restart condition is detected with RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)

- When 0 is written to the TRS bit with the RIICnMR1.MTWp bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode (1 or 0) by detection or issuing of a start condition, etc. Although writing to the MST bit is possible when the RIICnMR1.MTWp bit is set to 1, writing to this bit is not necessary during normal usage.

#### [Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the RIICnMR1.MTWp bit set to 1

#### [Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the RIICnMR1.MTWp bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I<sup>2</sup>C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

#### [Setting condition]

When a start condition is detected

#### [Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

### CAUTION

- When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
- When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

### 18.3.4 RIICnMR1 — I<sup>2</sup>C Bus Mode Register 1

**Access:** RIICnMR1 register can be read or written in 32-bit units.  
 RIICnMR1L and RIICnMR1H registers can be read or written in 16-bit units.  
 RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH registers can be read or written in 8-bit units.

**Address:** RIICnMR1: <RIICn\_base> + 0008<sub>H</sub>  
 RIICnMR1L: <RIICn\_base> + 0008<sub>H</sub>, RIICnMR1H: <RIICn\_base> + 000A<sub>H</sub>  
 RIICnMR1LL: <RIICn\_base> + 0008<sub>H</sub>, RIICnMR1LH: <RIICn\_base> + 0009<sub>H</sub>,  
 RIICnMR1HL: <RIICn\_base> + 000A<sub>H</sub>, RIICnMR1HH: <RIICn\_base> + 000B<sub>H</sub>

**Value after reset:** 0000 0008<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MTWP	CKS[2:0]	BCWP	BC[2:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Table 18.12 RIICnMR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MTWP	MST/TRS Write Protect 0: Disables writing to the RIICnCR2.MST and TRS bits. 1: Enables writing to the RIICnCR2.MST and TRS bits.
6 to 4	CKS[2:0]	Internal Reference Clock Selection (IIC $\phi$ ) b <sub>6</sub> b <sub>4</sub> 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock
3	BCWP <sup>1</sup>	BC Write Protect 0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.) 1: Protects the BC[2:0] bits.
2 to 0	BC[2:0]	Bit Counter b <sub>2</sub> b <sub>0</sub> 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

**BC[2:0] Bits (Bit Counter)**

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames. When setting any value other than 000<sub>H</sub>, set the value while the SCL line is at a low level.

**[Clearing conditions]**

- When 1 is written to the RIICnCR1.IICRST bit and a RIIC reset or internal reset is initiated.
- Data transfer including the acknowledge bit being completed.
- A start condition including a restart condition being detected.

### 18.3.5 RIICnMR2 — I<sup>2</sup>C Bus Mode Register 2

**Access:** RIICnMR2 register can be read or written in 32-bit units.  
 RIICnMR2L and RIICnMR2H registers can be read or written in 16-bit units.  
 RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH registers can be read or written in 8-bit units.

**Address:** RIICnMR2: <RIICn\_base> + 000C<sub>H</sub>  
 RIICnMR2L: <RIICn\_base> + 000C<sub>H</sub>, RIICnMR2H: <RIICn\_base> + 000E<sub>H</sub>  
 RIICnMR2LL: <RIICn\_base> + 000C<sub>H</sub>, RIICnMR2LH: <RIICn\_base> + 000D<sub>H</sub>,  
 RIICnMR2HL: <RIICn\_base> + 000E<sub>H</sub>, RIICnMR2HH: <RIICn\_base> + 000F<sub>H</sub>

**Value after reset:** 0000 0006<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DLCS	SDDL[2:0]	—	TMOH	TMOL	TMOS		
Value after reset	0	0	0	0	0	0	0	0	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 18.13 RIICnMR2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DLCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IIC $\phi$ ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IIC $\phi$ /2) is selected as the clock source of the SDA output delay counter.* <sup>1</sup>
6 to 4	SDDL[2:0]	SDA Output Delay Counter • When RIICnMR2.DLCS = 0 (IIC $\phi$ ) b6 b4 0 0 0: No output delay 0 0 1: 1 IIC $\phi$ cycle 0 1 0: 2 IIC $\phi$ cycles 0 1 1: 3 IIC $\phi$ cycles 1 0 0: 4 IIC $\phi$ cycles 1 0 1: 5 IIC $\phi$ cycles 1 1 0: 6 IIC $\phi$ cycles 1 1 1: 7 IIC $\phi$ cycles • When RIICnMR2.DLCS = 1 (IIC $\phi$ /2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IIC $\phi$ cycles 0 1 0: 3 or 4 IIC $\phi$ cycles 0 1 1: 5 or 6 IIC $\phi$ cycles 1 0 0: 7 or 8 IIC $\phi$ cycles 1 0 1: 9 or 10 IIC $\phi$ cycles 1 1 0: 11 or 12 IIC $\phi$ cycles 1 1 1: 13 or 14 IIC $\phi$ cycles
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOH	Timeout H Count Control 0: Disables counting while the SCL line is at a high level. 1: Enables counting while the SCL line is at a high level.

**Table 18.13 RIICnMR2 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
1	TMOL	Timeout L Count Control 0: Disables counting while the SCL line is at a low level. 1: Enables counting while the SCL line is at a low level.
0	TMOS	Timeout Detection Time Selection 0: Long mode is selected. 1: Short mode is selected.

Note 1. The setting DLCS = 1 (IIC $\phi$ /2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IIC $\phi$ ).

#### **TMOS Bit (Timeout Detection Time Selection)**

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IIC $\phi$ ) as a count source.

For details on the timeout function, see **Section 18.13.1, Timeout Function**.

#### **TMOL Bit (Timeout L Count Control)**

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

#### **TMOH Bit (Timeout H Count Control)**

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

#### **SDDL[2:0] Bits (SDA Output Delay Setup Counter)**

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see **Section 18.7, Facility for Delaying SDA Output**.

#### **CAUTION**

Set the SDA output delay time to meet the I<sup>2</sup>C bus standard (within the data enable time/acknowledge enable time<sup>\*1</sup>). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time  
3,450 ns (up to 100 kbps: standard mode [Sm])  
900 ns (up to 400 kbps: fast mode [Fm])

### 18.3.6 RIICnMR3 — I<sup>2</sup>C Bus Mode Register 3

**Access:** RIICnMR3 register can be read or written in 32-bit units.  
 RIICnMR3L and RIICnMR3H registers can be read or written in 16-bit units.  
 RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH registers can be read or written in 8-bit units.

**Address:** RIICnMR3: <RIICn\_base> + 0010<sub>H</sub>  
 RIICnMR3L: <RIICn\_base> + 0010<sub>H</sub>, RIICnMR3H: <RIICn\_base> + 0012<sub>H</sub>  
 RIICnMR3LL: <RIICn\_base> + 0010<sub>H</sub>, RIICnMR3LH: <RIICn\_base> + 0011<sub>H</sub>,  
 RIICnMR3HL: <RIICn\_base> + 0012<sub>H</sub>, RIICnMR3HH: <RIICn\_base> + 0013<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 18.14 RIICnMR3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	WAIT <sup>2</sup>	WAIT 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR.
5	RDRFS <sup>2</sup>	RDRF Flag Set Timing Selection 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.
4	ACKWP <sup>1</sup>	ACKBT Write Protect 0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.
3	ACKBT <sup>1</sup>	Transmit Acknowledge 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).
2	ACKBR	Receive Acknowledge 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).
1, 0	NF[1:0]	Digital noise Filter Stage Selection b1 b0 0 0: Noise of up to one IIC $\phi$ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IIC $\phi$ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IIC $\phi$ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IIC $\phi$ cycles is filtered out (4-stage filter).

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

**NF[1:0] Bits (Digital noise Filter Stage Selection)**

These bits are used to select the number of stages of the digital noise filter.

**CAUTION**

Set the noise range to be filtered out by the noise filter within a range less than the SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock synchronized (IIC $\phi$ ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

**ACKBR Bit (Receive Acknowledge)**

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

**[Setting condition]**

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

**[Clearing conditions]**

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

**ACKBT Bit (Transmit Acknowledge)**

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

**[Setting condition]**

When 1 is written to this bit with the ACKWP bit set to 1

**[Clearing conditions]**

- When 0 is written to this bit after ACKWP reading while the ACKWP bit is set to 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

**CAUTION**

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

**ACKWP Bit (ACKBT Write Protect)**

This bit is used to control the modification of the ACKBT bit.

### RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

### WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

### CAUTION

When the value of the WAIT bit is cleared to 0, be sure to read the RIICnDRR beforehand.

### 18.3.7 RIICnFER — I<sup>2</sup>C Bus Function Enable Register

**Access:** RIICnFER register can be read or written in 32-bit units.  
 RIICnFERL and RIICnFERH registers can be read or written in 16-bit units.  
 RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH registers can be read or written in 8-bit units.

**Address:** RIICnFER: <RIICn\_base> + 0014<sub>H</sub>  
 RIICnFERL: <RIICn\_base> + 0014<sub>H</sub>, RIICnFERH: <RIICn\_base> + 0016<sub>H</sub>  
 RIICnFERLL: <RIICn\_base> + 0014<sub>H</sub>, RIICnFERLH: <RIICn\_base> + 0015<sub>H</sub>,  
 RIICnFERHL: <RIICn\_base> + 0016<sub>H</sub>, RIICnFERHH: <RIICn\_base> + 0017<sub>H</sub>

**Value after reset:** 0000 0072<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.15 RIICnFER Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	SCLE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.
5	NFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.
4	NACKE	NACK Reception Transfer Suspension Enable 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).
3	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.
1	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.)
If 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode, the TRS bit is not cleared.		
0	TMOE	Timeout Function Enable 0: The timeout function is disabled. 1: The timeout function is enabled.

**TMOE Bit (Timeout Function Enable)**

This bit is used to enable or disable the timeout function.

For details on the timeout function, see **Section 18.13.1, Timeout Function**.

**MALE Bit (Master Arbitration-Lost Detection Enable)**

This bit is used to select enabling or disabling of the arbitration-lost detection function. Normally, set this bit to 1.

**NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)**

This bit is used to specify whether the detection of ACK during transmission of NACK in reception (such as when slaves with the same address are present on the bus and each is transmitting different data, or when two or more masters select the same slave device simultaneously with different numbers of bytes for reception) is judged to represent a loss in arbitration.

**SALE Bit (Slave Arbitration-Lost Detection Enable)**

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

**NACKE Bit (NACK Reception Transfer Suspension Enable)**

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

**SCLE Bit (SCL Synchronous Circuit Enable)**

This bit is used to specify whether to synchronize the SCL clock with a rising or falling edge on the SCL line. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the I<sup>2</sup>C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

### 18.3.8 RIICnSER — I<sup>2</sup>C Bus Status Enable Register

**Access:** RIICnSER register can be read or written in 32-bit units.  
 RIICnSERL and RIICnSERH registers can be read or written in 16-bit units.  
 RIICnSERLL, RIICnSERLH, RIICnSERHL, and RIICnSERHH registers can be read or written in 8-bit units.

**Address:** RIICnSER: <RIICn\_base> + 0018<sub>H</sub>  
 RIICnSERL: <RIICn\_base> + 0018<sub>H</sub>, RIICnSERH: <RIICn\_base> + 001A<sub>H</sub>  
 RIICnSERLL: <RIICn\_base> + 0018<sub>H</sub>, RIICnSERLH: <RIICn\_base> + 0019<sub>H</sub>,  
 RIICnSERHL: <RIICn\_base> + 001A<sub>H</sub>, RIICnSERHH: <RIICn\_base> + 001B<sub>H</sub>

**Value after reset:** 0000 0009<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DIDE	—	GCE	SAR2	SAR1	SAR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 18.16 RIICnSER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DIDE	Device-ID Address Detection Enable 0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCE	General Call Address Enable 0: General call address detection is disabled. 1: General call address detection is enabled.
2	SAR2	Slave Address Register 2 Enable 0: Slave address in RIICnSAR2 is disabled. 1: Slave address in RIICnSAR2 is enabled.
1	SAR1	Slave Address Register 1 Enable 0: Slave address in RIICnSAR1 is disabled. 1: Slave address in RIICnSAR1 is enabled.
0	SAR0	Slave Address Register 0 Enable 0: Slave address in RIICnSAR0 is disabled. 1: Slave address in RIICnSAR0 is enabled.

**SARy Bit (Slave Address Register y Enable) (y = 0 to 2)**

This bit is used to enable or disable the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

**GCAE Bit (General Call Address Enable)**

This bit is used to specify whether to ignore the general call address (0000 000<sub>B</sub> + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

**DIDE Bit (Device-ID Address Detection Enable)**

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100<sub>B</sub>) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 18.9.3, Device-ID Address Detection**.

### 18.3.9 RIICnIER — I<sup>2</sup>C Bus Interrupt Enable Register

**Access:** RIICnIER register can be read or written in 32-bit units.  
 RIICnIERL and RIICnIERH registers can be read or written in 16-bit units.  
 RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH registers can be read or written in 8-bit units.

**Address:** RIICnIER: <RIICn\_base> + 001C<sub>H</sub>  
 RIICnIERL: <RIICn\_base> + 001C<sub>H</sub>, RIICnIERH: <RIICn\_base> + 001E<sub>H</sub>  
 RIICnIERLL: <RIICn\_base> + 001C<sub>H</sub>, RIICnIERLH: <RIICn\_base> + 001D<sub>H</sub>,  
 RIICnIERHL: <RIICn\_base> + 001E<sub>H</sub>, RIICnIERHH: <RIICn\_base> + 001F<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.17 RIICnIER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TIE	Transmit Data Empty Interrupt Enable 0: Transmit data empty interrupt request (INTIICnTI) is disabled. 1: Transmit data empty interrupt request (INTIICnTI) is enabled.
6	TEIE	Transmit End Interrupt Enable 0: Transmit end interrupt request (INTIICnTEI) is disabled. 1: Transmit end interrupt request (INTIICnTEI) is enabled.
5	RIE	Receive Complete Interrupt Enable 0: Receive complete interrupt request (INTIICnRI) is disabled. 1: Receive complete interrupt request (INTIICnRI) is enabled.
4	NAKIE	NACK Reception Interrupt Enable 0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.
3	SPIE	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.
2	STIE	Start Condition Detection Interrupt Enable 0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.
1	ALIE	Arbitration-Lost Interrupt Enable 0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.
0	TMOIE	Timeout Interrupt Enable 0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.

#### TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the RIICnSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

**ALIE Bit (Arbitration-Lost Interrupt Enable)**

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the RIICnSR2.AL flag is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

**STIE Bit (Start Condition Detection Interrupt Enable)**

This bit is used to enable or disable start condition detection interrupt requests (STI) when the RIICnSR2.START flag is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

**SPIE Bit (Stop Condition Detection Interrupt Enable)**

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the RIICnSR2.STOP flag is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

**NAKIE Bit (NACK Reception Interrupt Enable)**

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the RIICnSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

**RIE Bit (Receive Complete Interrupt Enable)**

This bit is used to enable or disable receive complete interrupt requests (INTIICnRI) when the RIICnSR2.RDRF flag is set to 1.

**TEIE Bit (Transmit End Interrupt Enable)**

This bit is used to enable or disable transmit end interrupts (INTIICnTEI) when the RIICnSR2.TEND flag is set to 1. An INTIICnTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

**TIE Bit (Transmit Data Empty Interrupt Enable)**

This bit is used to enable or disable transmit data empty interrupts (INTIICnTI) when the RIICnSR2.TDRE flag is set to 1.

### 18.3.10 RIICnSR1 — I<sup>2</sup>C Bus Status Register 1

**Access:** RIICnSR1 register can be read or written in 32-bit units.  
 RIICnSR1L and RIICnSR1H registers can be read or written in 16-bit units.  
 RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH registers can be read or written in 8-bit units.

**Address:** RIICnSR1: <RIICn\_base> + 0020<sub>H</sub>  
 RIICnSR1L: <RIICn\_base> + 0020<sub>H</sub>, RIICnSR1H: <RIICn\_base> + 0022<sub>H</sub>  
 RIICnSR1LL: <RIICn\_base> + 0020<sub>H</sub>, RIICnSR1LH: <RIICn\_base> + 0021<sub>H</sub>,  
 RIICnSR1HL: <RIICn\_base> + 0022<sub>H</sub>, RIICnSR1HH: <RIICn\_base> + 0023<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R(W) <sub>1</sub>	R	R(W) <sub>1</sub>	R(W) <sub>1</sub>	R(W) <sub>1</sub>	R(W) <sub>1</sub>

Note 1. Only 0 can be written to this bit.

Table 18.18 RIICnSR1 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DID	Device-ID Address Detection Flag 0: Device-ID address is not detected. 1: Device-ID address is detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 is not detected. 1: Slave address 0 is detected.

#### AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARY bit set to 1 (slave address y detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of ( $1111\ 0_B + \text{RIICnSARy.SVA[9:8]}$ ) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of ( $1111\ 0_B + \text{RIICnSARy.SVA[9:8]}$ ) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

- When the received slave address matches a value of ( $1111\ 0_B + \text{RIICnSARy.SVA[9:8]}$ ) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

### GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address ( $0000\ 000_B + 0 [W]$ ) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

- When a restart condition is detected after a match with the device ID address and the device ID address ( $1111\ 100_B$ ) plus 1[R] has matched.

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address ( $0000\ 000_B + 0 [W]$ ) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### DID Flag (Device-ID Address Detection)

#### [Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100<sub>B</sub>) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address (1111 100<sub>B</sub>) plus 1[R] has matched while the setting of the RIICnSER.DIDE bit is 1 (device ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

#### [Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100<sub>B</sub>)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)  
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100<sub>B</sub>) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)  
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### 18.3.11 RIICnSR2 — I<sup>2</sup>C Bus Status Register 2

**Access:** RIICnSR2 register can be read or written in 32-bit units.  
 RIICnSR2L and RIICnSR2H registers can be read or written in 16-bit units.  
 RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH registers can be read or written in 8-bit units.

**Address:** RIICnSR2: <RIICn\_base> + 0024<sub>H</sub>  
 RIICnSR2L: <RIICn\_base> + 0024<sub>H</sub>, RIICnSR2H: <RIICn\_base> + 0026<sub>H</sub>  
 RIICnSR2LL: <RIICn\_base> + 0024<sub>H</sub>, RIICnSR2LH: <RIICn\_base> + 0025<sub>H</sub>,  
 RIICnSR2HL: <RIICn\_base> + 0026<sub>H</sub>, RIICnSR2HH: <RIICn\_base> + 0027<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R(1/W)	R(1/W)	R(1/W)	R(1/W)	R(1/W)	R(1/W)

Note 1. Only 0 can be written to this bit.

Table 18.19 RIICnSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	TEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RDRF	Receive Complete Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	NACKF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.
3	STOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	START	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	AL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	TMOF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.

### **TMOF Flag (Timeout Detection)**

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.

#### [Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (the RIICnCR2.BBSY flag is 1) in master mode (the RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (the RIICnSR1 register is not  $00_H$ ) and the bus is busy (the RIICnCR2.BBSY bit is 1) in slave mode (the RIICnCR2.MST bit is 0).
- The bus is free (the RIICnCR2.BBSY flag is 0) while generation of a start condition is requested (the RIICnCR2.ST bit is 1).

#### [Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### **AL Flag (Arbitration-Lost)**

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

#### [Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

#### [Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1

- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**Table 18.20 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions**

RIICnFER			RIICnSR2	Error	Arbitration-Lost Generation Source
MALE	NALE	SALE	AL		
1	—	—	1	Start condition issuance error	When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1
					When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1
—	1	—	1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
—	—	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

—: Don't care

### START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### NACKF Flag (NACK Detection)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTION**

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

**RDRF Flag (Receive Complete)**

[Setting conditions]

- When receive data has been transferred from RIICnDRS to RIICnDRR  
At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit)
- When the received slave address matches the address enabled in RIICnSER after a start condition (or a restart condition) is detected with the RIICnCR2.TRS bit cleared to 0
- In master mode, transition to master reception while the R/W# bit appended to the slave address is set to 1

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**TEND Flag (Transmit End)**

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**TDRE Flag (Transmit Data Empty)**

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition is detected
  - When 1 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When the received slave address matches the address enabled in RIICnSER after a start condition including a restart condition is detected with the RIICnCR2.TRS bit set to 1

**[Clearing conditions]**

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
  - When 0 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTION**

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the RIICnDRS register and the RIICnDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

### 18.3.12 RIICnSARy — I<sup>2</sup>C Slave Address Register y (y = 0 to 2)

**Access:** RIICnSARy register can be read or written in 32-bit units.  
 RIICnSARyL and RIICnSARyH registers can be read or written in 16-bit units.  
 RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH registers can be read or written in 8-bit units.

**Address:** RIICnSAR0: <RIICn\_base> + 0028<sub>H</sub>  
 RIICnSAR0L: <RIICn\_base> + 0028<sub>H</sub>, RIICnSAR0H: <RIICn\_base> + 002A<sub>H</sub>  
 RIICnSAR0LL: <RIICn\_base> + 0028<sub>H</sub>, RIICnSAR0LH: <RIICn\_base> + 0029<sub>H</sub>,  
 RIICnSAR0HL: <RIICn\_base> + 002A<sub>H</sub>, RIICnSAR0HH: <RIICn\_base> + 002B<sub>H</sub>  
 RIICnSAR1: <RIICn\_base> + 002C<sub>H</sub>  
 RIICnSAR1L: <RIICn\_base> + 002C<sub>H</sub>, RIICnSAR1H: <RIICn\_base> + 002E<sub>H</sub>  
 RIICnSAR1LL: <RIICn\_base> + 002C<sub>H</sub>, RIICnSAR1LH: <RIICn\_base> + 002D<sub>H</sub>,  
 RIICnSAR1HL: <RIICn\_base> + 002E<sub>H</sub>, RIICnSAR1HH: <RIICn\_base> + 002F<sub>H</sub>  
 RIICnSAR2: <RIICn\_base> + 0030<sub>H</sub>  
 RIICnSAR2L: <RIICn\_base> + 0030<sub>H</sub>, RIICnSAR2H: <RIICn\_base> + 0032<sub>H</sub>  
 RIICnSAR2LL: <RIICn\_base> + 0030<sub>H</sub>, RIICnSAR2LH: <RIICn\_base> + 0031<sub>H</sub>,  
 RIICnSAR2HL: <RIICn\_base> + 0032<sub>H</sub>, RIICnSAR2HH: <RIICn\_base> + 0033<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSy	—	—	—	—	—										SVA[9:1]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W									

Table 18.21 RIICnSARy Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	FSy	7-Bit/10-Bit Address Format Selection 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Upper Bits A slave address is set. <ul style="list-style-type: none"> <li>When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are valid and form a 7-bit slave address.</li> <li>When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit).</li> </ul>
0	SVA0	10-Bit Address LSB The least significant bit (LSB) of a 10-bit slave address is set. <ul style="list-style-type: none"> <li>When the FSy bit is 0 (7-bit address format), this bit is invalid.</li> <li>When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits).</li> </ul>

**SVA0 Bit (10-Bit Address LSB)**

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

**SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)**

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address.

When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.

While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

**FSy Bit (7-Bit/10-Bit Address Format Selection)**

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

### 18.3.13 RIICnBRL — I<sup>2</sup>C Bus Bit Rate Low-Level Register

**Access:** RIICnBRL register can be read or written in 32-bit units.  
 RIICnBRLL and RIICnBRLH registers can be read or written in 16-bit units.  
 RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH registers can be read or written in 8-bit units.

**Address:** RIICnBRL: <RIICn\_base> + 0034<sub>H</sub>  
 RIICnBRLL: <RIICn\_base> + 0034<sub>H</sub>, RIICnBRLH: <RIICn\_base> + 0036<sub>H</sub>  
 RIICnBRLLL: <RIICn\_base> + 0034<sub>H</sub>, RIICnBRLLH: <RIICn\_base> + 0035<sub>H</sub>,  
 RIICnBRLHL: <RIICn\_base> + 0036<sub>H</sub>, RIICnBRLHH: <RIICn\_base> + 0037<sub>H</sub>

**Value after reset:** 0000 00FF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRL[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

**Table 18.22 RIICnBRL Register Contents**

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock

The RIICnBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see **Section 18.10, Automatic Low-Hold Function for SCL**); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time<sup>\*1</sup>.

RIICnBRL counts the low-level period with the internal reference clock source (IIC $\phi$ ) specified by the RIICnMR1.CKS[2:0] bits.

- Note 1. Data setup time (tSU: DAT)  
 250 ns (up to 100 kbps: standard mode [Sm])  
 100 ns (up to 400 kbps: fast mode [Fm])

### 18.3.14 RIICnBRH — I<sup>2</sup>C Bus Bit Rate High-Level Register

**Access:** RIICnBRH register can be read or written in 32-bit units.  
 RIICnBRHL and RIICnBRHH registers can be read or written in 16-bit units.  
 RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH registers can be read or written in 8-bit units.

**Address:** RIICnBRH: <RIICn\_base> + 0038<sub>H</sub>  
 RIICnBRHL: <RIICn\_base> + 0038<sub>H</sub>, RIICnBRHH: <RIICn\_base> + 003A<sub>H</sub>  
 RIICnBRHLL: <RIICn\_base> + 0038<sub>H</sub>, RIICnBRHLH: <RIICn\_base> + 0039<sub>H</sub>,  
 RIICnBRHHL: <RIICn\_base> + 003A<sub>H</sub>, RIICnBRHHH: <RIICn\_base> + 003B<sub>H</sub>

**Value after reset:** 0000 00FF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	—	—	—	—	—	—	BRH[4:0]			
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	

Table 18.23 RIICnBRH Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock

RIICnBRH is a 5-bit register to set the high-level period of SCL clock. RIICnICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH counts the high-level period with the internal reference clock source (IICφ) specified by the RIICnMR1.CKS[2:0] bits.

The I<sup>2</sup>C transfer rate and the SCL clock duty are calculated using the following expression.

- (1) When RIICnFER.SCLE = 0

$$\text{Transfer rate} = 1 / \{[(\text{RIICnBRH}+1) + (\text{RIICnBRL}+1)] / \text{IIC}\phi^* + \text{tr} + \text{tf}\}$$

$$\text{Duty cycle} = \{\text{tr} + [(\text{RIICnBRH}+1) / \text{IIC}\phi]\} / \{ \text{tr} + \text{tf} + [(\text{RIICnBRH}+1) + (\text{RIICnBRL}+1)] / \text{IIC}\phi \}$$

- (2) When RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ = PCLK

$$\text{Transfer rate} = 1 / \{[(\text{RIICnBRH}+3) + (\text{RIICnBRL}+3)] / \text{IIC}\phi^* + \text{tr} + \text{tf}\}$$

$$\text{Duty cycle} = \{\text{tr} + [(\text{RIICnBRH}+3) / \text{IIC}\phi]\} / \{ \text{tr} + \text{tf} + [(\text{RIICnBRH}+3) + (\text{RIICnBRL}+3)] / \text{IIC}\phi \}$$

- (3) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ = PCLK

$$\text{Transfer rate} = 1 / \{[(\text{RIICnBRH}+3+nf) + (\text{RIICnBRL}+3+nf)] / \text{IIC}\phi^* + \text{tr} + \text{tf}\}$$

$$\text{Duty cycle} = \{\text{tr} + [(\text{RIICnBRH}+3+nf) / \text{IIC}\phi]\} / \{ \text{tr} + \text{tf} + [(\text{RIICnBRH}+3+nf) + (\text{RIICnBRL}+3+nf)] / \text{IIC}\phi \}$$

- (4) RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ < PCLK

$$\text{Transfer rate} = 1 / \{[(\text{RIICnBRH}+2) + (\text{RIICnBRL}+2)] / \text{IIC}\phi^* + \text{tr} + \text{tf}\}$$

$$\text{Duty cycle} = \{\text{tr} + [(\text{RIICnBRH}+2) / \text{IIC}\phi]\} / \{ \text{tr} + \text{tf} + [(\text{RIICnBRH}+2) + (\text{RIICnBRL}+2)] / \text{IIC}\phi \}$$

- (5) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ < PCLK  
Transfer rate = 1 / { [ (RIICnBRH+2+nf) + (RIICnBRL+2+nf) ] / IICφ \*<sup>1</sup> + tr + tf }  
Duty cycle = { tr +[(RIICnBRH+2+nf)/ IICφ] }/{ tr + tf + [(RIICnBRH+2+nf) + (RIICnBRL+2+nf)]/ IICφ }

tf: SCL line falling time [ns]\*<sup>2</sup>

tr: SCL line rising time [ns]\*<sup>2</sup>

nf: Digital noise filter stage

Duty cycle: 0% < Duty < 100%

**Note 1.** As for IICφ, see CKS[2:0] in **Section 18.3.4, RIICnMR1 — I<sup>2</sup>C Bus Mode Register 1**.

**Note 2.** The SCL line rising time [tr] and SCL line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I<sup>2</sup>C bus standard from NXP Semiconductors.

**Table 18.24** lists examples of the RIICnBRH and RIICnBRL register settings when the SCL synchronization circuit is not used.

**Table 18.24 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate**

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	8				10				12.5						
	CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL	
		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	100 <sub>B</sub>	22	F6 <sub>H</sub>	25	F9 <sub>H</sub>	101 <sub>B</sub>	13	ED <sub>H</sub>	15	EF <sub>H</sub>	101 <sub>B</sub>	16	F0 <sub>H</sub>	20	F4 <sub>H</sub>
50	010 <sub>B</sub>	16	F0 <sub>H</sub>	19	F3 <sub>H</sub>	010 <sub>B</sub>	21	F5 <sub>H</sub>	24	F8 <sub>H</sub>	011 <sub>B</sub>	12	EC <sub>H</sub>	15	EF <sub>H</sub>
100	001 <sub>B</sub>	15	EF <sub>H</sub>	18	F2 <sub>H</sub>	001 <sub>B</sub>	19	F3 <sub>H</sub>	23	F7 <sub>H</sub>	001 <sub>B</sub>	24	F8 <sub>H</sub>	29	FD <sub>H</sub>
400	000 <sub>B</sub>	4	E4 <sub>H</sub>	10	EA <sub>H</sub>	000 <sub>B</sub>	5	E5 <sub>H</sub>	12	EC <sub>H</sub>	000 <sub>B</sub>	7	E7 <sub>H</sub>	16	F0 <sub>H</sub>

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	16				20				25						
	CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL	
		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	101 <sub>B</sub>	22	F6 <sub>H</sub>	25	F9 <sub>H</sub>	110 <sub>B</sub>	13	ED <sub>H</sub>	15	EF <sub>H</sub>	110 <sub>B</sub>	16	F0 <sub>H</sub>	20	F4 <sub>H</sub>
50	011 <sub>B</sub>	16	F0 <sub>H</sub>	19	F3 <sub>H</sub>	011 <sub>B</sub>	21	F5 <sub>H</sub>	24	F8 <sub>H</sub>	100 <sub>B</sub>	12	EC <sub>H</sub>	15	EF <sub>H</sub>
100	010 <sub>B</sub>	15	EF <sub>H</sub>	18	F2 <sub>H</sub>	010 <sub>B</sub>	19	F3 <sub>H</sub>	23	F7 <sub>H</sub>	010 <sub>B</sub>	24	F8 <sub>H</sub>	29	FD <sub>H</sub>
400	000 <sub>B</sub>	9	E9 <sub>H</sub>	20	F4 <sub>H</sub>	000 <sub>B</sub>	11	EB <sub>H</sub>	25	F9 <sub>H</sub>	001 <sub>B</sub>	7	E7 <sub>H</sub>	16	F0 <sub>H</sub>

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	30				33										
	CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL		CKS [2:0]	RIICnBRH		RIICnBRL	
		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]		SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	110 <sub>B</sub>	20	F4 <sub>H</sub>	24	F8 <sub>H</sub>	110 <sub>B</sub>	22	F6 <sub>H</sub>	26	FA <sub>H</sub>					
50	100 <sub>B</sub>	15	EF <sub>H</sub>	18	F2 <sub>H</sub>	100 <sub>B</sub>	17	F1 <sub>H</sub>	20	F4 <sub>H</sub>					
100	011 <sub>B</sub>	14	EE <sub>H</sub>	17	F1 <sub>H</sub>	011 <sub>B</sub>	16	F0 <sub>H</sub>	19	F3 <sub>H</sub>					
400	001 <sub>B</sub>	8	E8 <sub>H</sub>	19	F3 <sub>H</sub>	001 <sub>B</sub>	9	E9 <sub>H</sub>	21	F5 <sub>H</sub>					

#### CAUTION

CBRH/ICBRL settings in these tables are calculated using the following values:

SCL line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns

SCL line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns

For the specified values of SCL line rising time (tr) and SCL line falling time (tf), see the I<sup>2</sup>C bus standard from NXP Semiconductors.

### 18.3.15 RIICnDRT — I<sup>2</sup>C Bus Transmit Data Register

**Access:** RIICnDRT register can be read or written in 32-bit units.  
 RIICnDRTL and RIICnDRTH registers can be read or written in 16-bit units.  
 RIICnDRTLH, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH registers can be read or written in 8-bit units.

**Address:** RIICnDRT: <RIICn\_base> + 003C<sub>H</sub>  
 RIICnDRTL: <RIICn\_base> + 003C<sub>H</sub>, RIICnDRTH: <RIICn\_base> + 003E<sub>H</sub>  
 RIICnDRTLH: <RIICn\_base> + 003D<sub>H</sub>, RIICnDRTHL: <RIICn\_base> + 003E<sub>H</sub>,  
 RIICnDRTH: <RIICn\_base> + 003F<sub>H</sub>

**Value after reset:** 0000 00FF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When RIICnDRT detects a space in the I<sup>2</sup>C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTIICnTI) request is generated. When writing to bit 31 to 8, write the value after reset.

### 18.3.16 RIICnDRR — I<sup>2</sup>C Bus Receive Data Register

**Access:** RIICnDRR register is a read-only register that can be read in 32-bit units.  
 RIICnDRRL and RIICnDRRH registers are the read-only registers that can be read in 16-bit units.  
 RIICnDRRL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RIICnDRR: <RIICn\_base> + 0040<sub>H</sub>  
 RIICnDRRL: <RIICn\_base> + 0040<sub>H</sub>, RIICnDRRH: <RIICn\_base> + 0042<sub>H</sub>  
 RIICnDRRLH: <RIICn\_base> + 0040<sub>H</sub>, RIICnDRRLH: <RIICn\_base> + 0041<sub>H</sub>,  
 RIICnDRRHL: <RIICn\_base> + 0042<sub>H</sub>, RIICnDRRHH: <RIICn\_base> + 0043<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When 1 byte of data has been received, the received data is transferred from the I<sup>2</sup>C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive complete interrupt (INTIICnRI) request is generated.

If RIIC receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

### 18.3.17 RIICnDRS — I<sup>2</sup>C Bus Shift Register

**Access:** This register is not accessible.

**Address:** —

**Value after reset:** 0000 00FF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

RIICnDRS is a shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

## 18.4 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive complete, transmit data empty, and transmit end.

**Table 18.25** lists details of the several interrupt requests. The receive complete and transmit data empty interrupt request are both capable of launching data transfer by the DMAC.

Table 18.25 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DMACA Launching	Interrupt Condition
INTIICnTI	Transmit Data Empty	TDRE	Possible	TDRE = 1 and TIE = 1
INTIICnTEI	Transmit End	TEND	Not possible	TEND = 1 and TEIE = 1
INTIICnRI	Receive Complete	RDRF	Possible	RDRF = 1 and RIE = 1
INTIICnEE	Transfer Error/ Event Generation	AL NACKF	Not possible	AL = 1 and ALIE = 1 NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1

Clear or mask the each flag during interrupt handling.

### CAUTIONS

1. There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since INTIICnTI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.TDRE flag (a condition for INTIICnTI) is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).
3. Since INTIICnRI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.RDRF flag (a condition for INTIICnRI) is automatically cleared to 0 when data are read from RIICnDRR.
4. When using the INTIICnTEI interrupt, clear the RIICnSR2.TEND flag in the INTIICnTEI interrupt processing.  
Note that the RIICnSR2.TEND flag is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).

## 18.5 Operation

### 18.5.1 Communication Data Format

The I<sup>2</sup>C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

**Figure 18.3** shows the I<sup>2</sup>C bus format, and **Figure 18.4** shows the I<sup>2</sup>C bus timing.

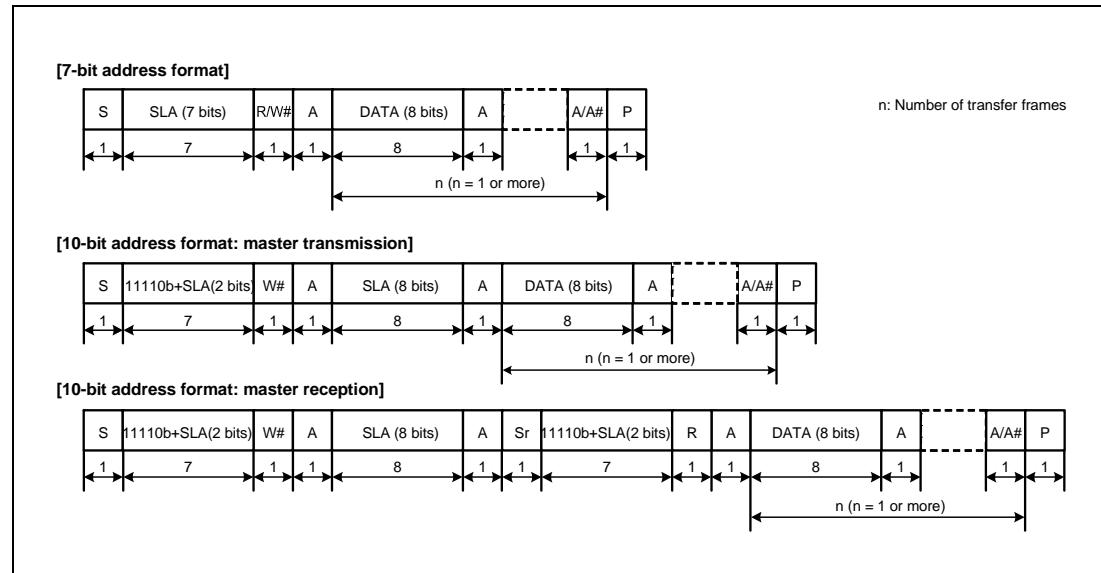


Figure 18.3 I<sup>2</sup>C Bus Format

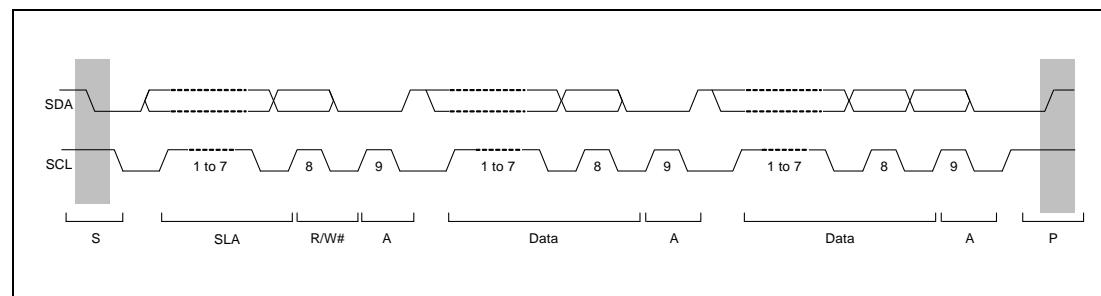


Figure 18.4 I<sup>2</sup>C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

### 18.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 18.5**. Make initial settings for the RIIC once when starting the RIIC.

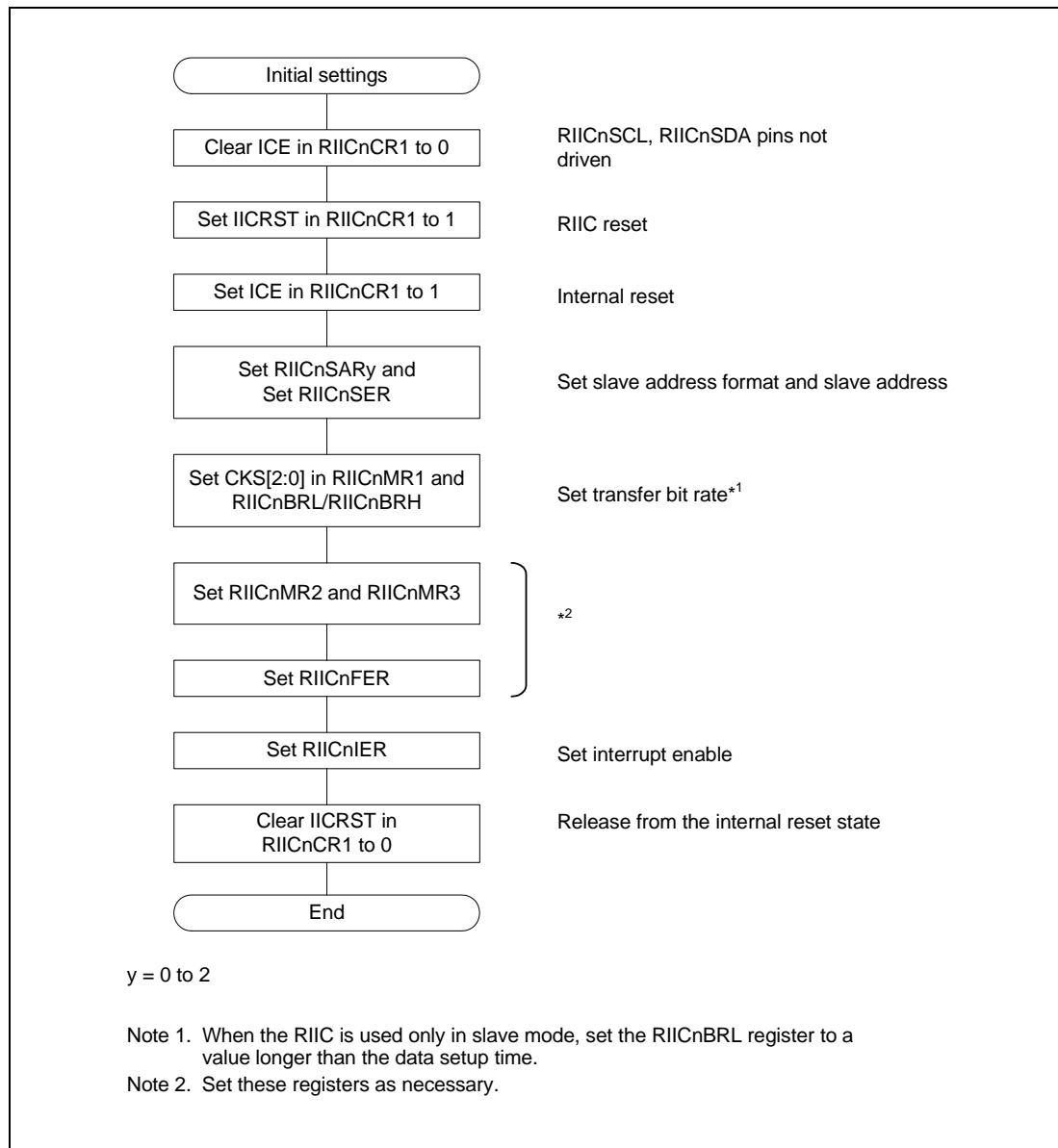


Figure 18.5 Example of RIIC Initialization Flowchart

### 18.5.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. **Figure 18.6** shows an example of usage of master transmission and **Figure 18.7** to **Figure 18.9** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

- (1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL ( $y = 0$  to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 18.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.
- (2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS and MST bits to 1.
- (3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.  
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.  
For data transmission with an address in the 10-bit format, start by writing 1111 0<sub>B</sub>, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.
- (4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.
- (5) After all bytes of data for transmission have been written to the RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

- (6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00<sub>B</sub> and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.
- (7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

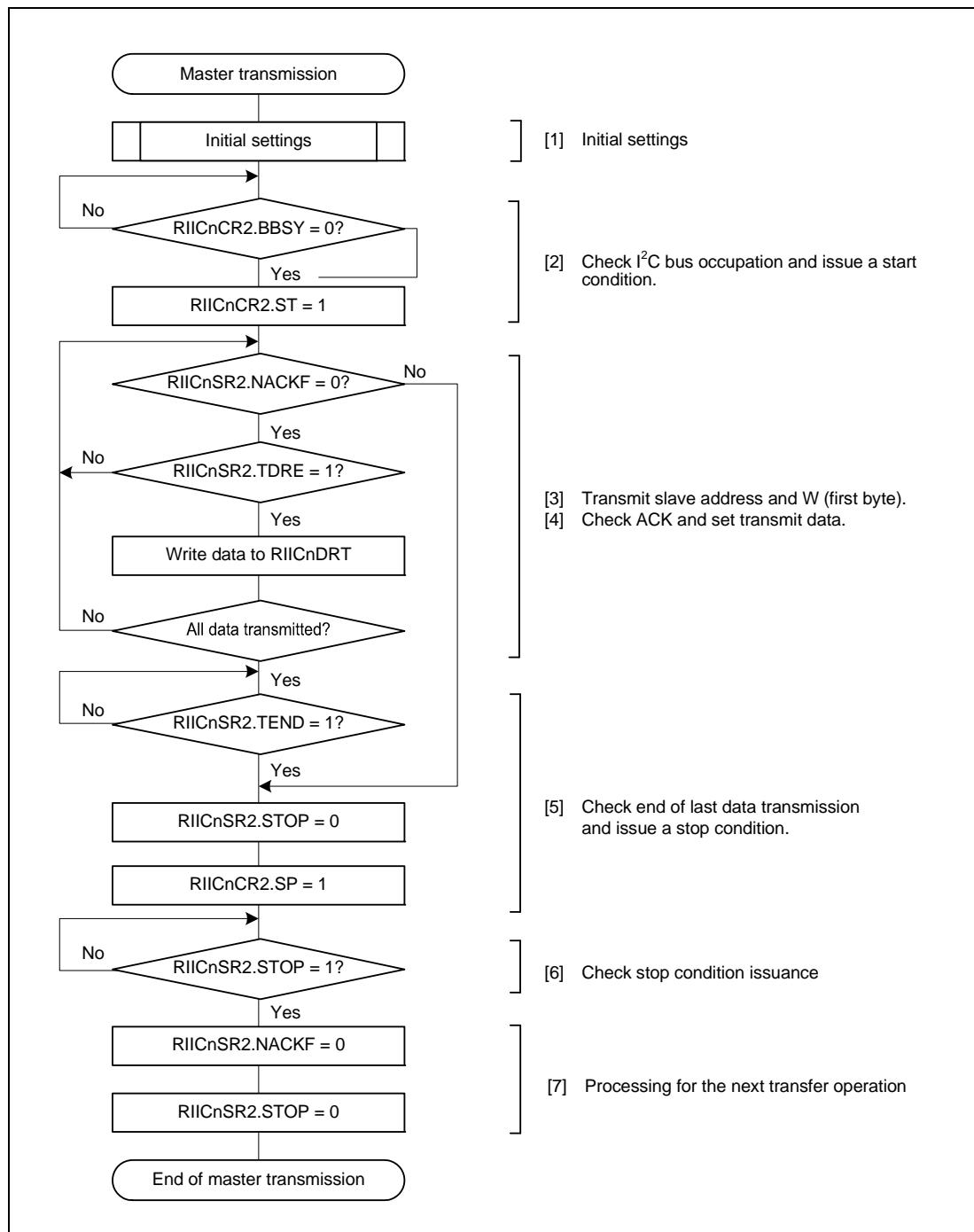


Figure 18.6 Example of Master Transmission Flowchart

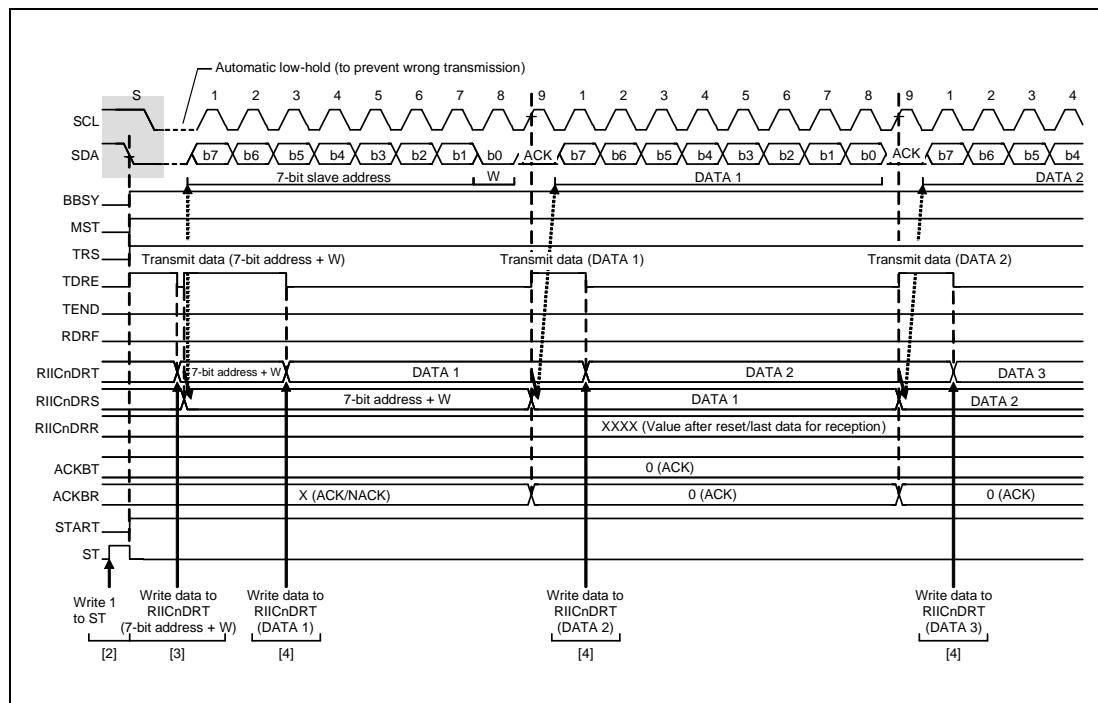


Figure 18.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

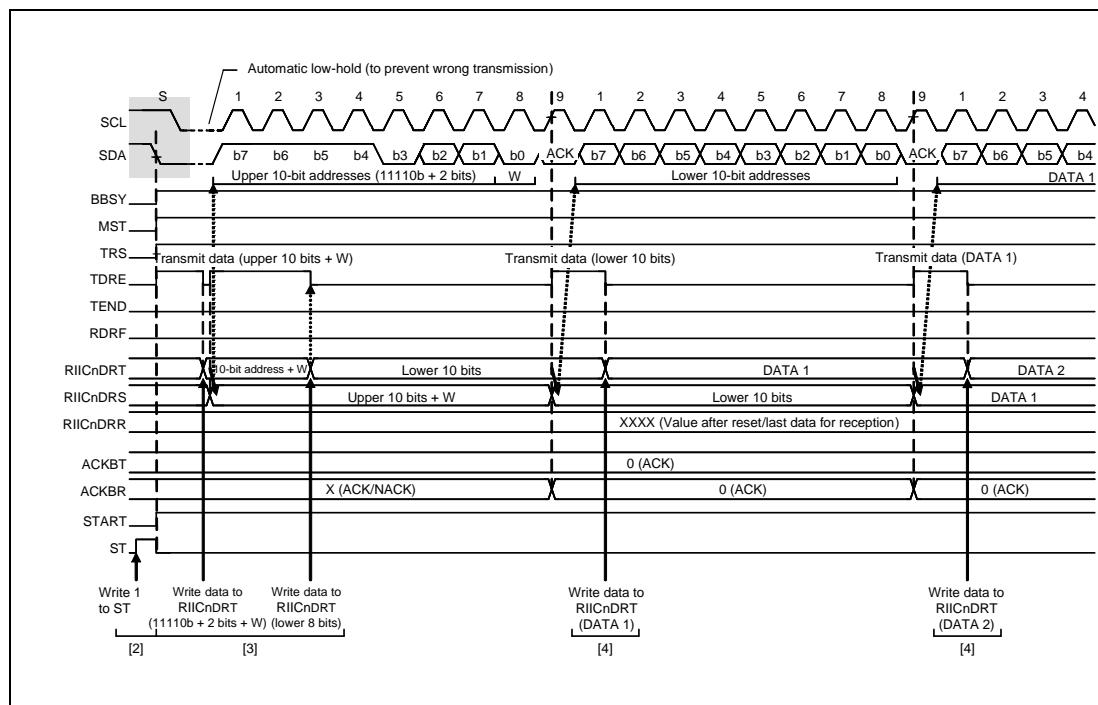


Figure 18.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

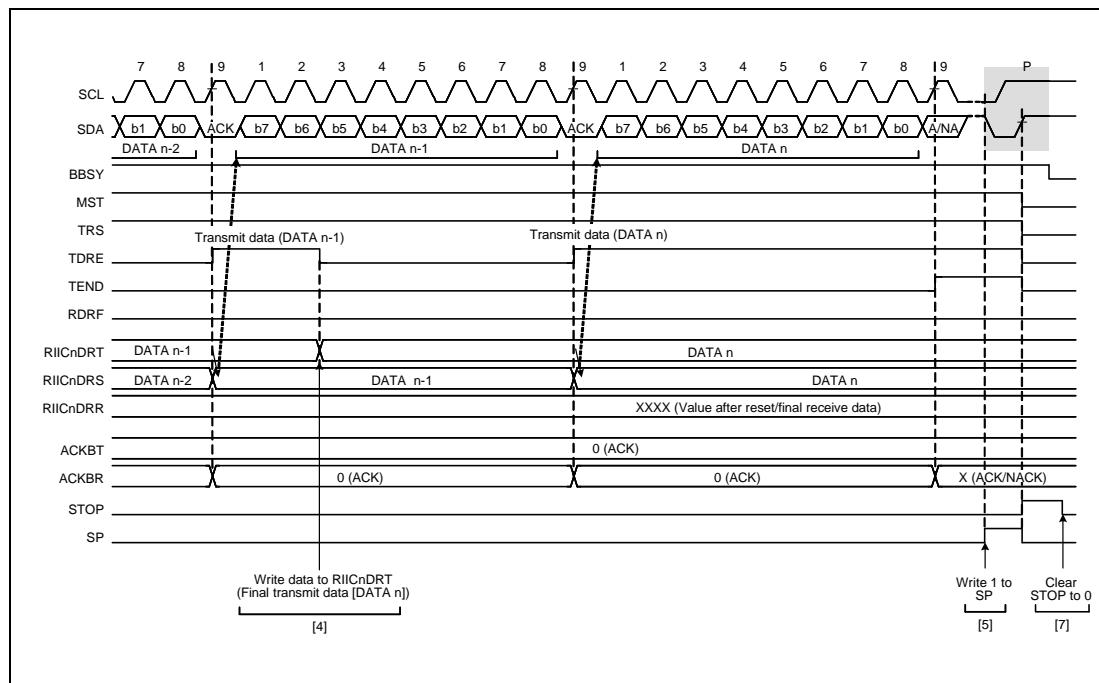


Figure 18.9 Master Transmit Operation Timing (3)

#### 18.5.4 Master Receive Operation

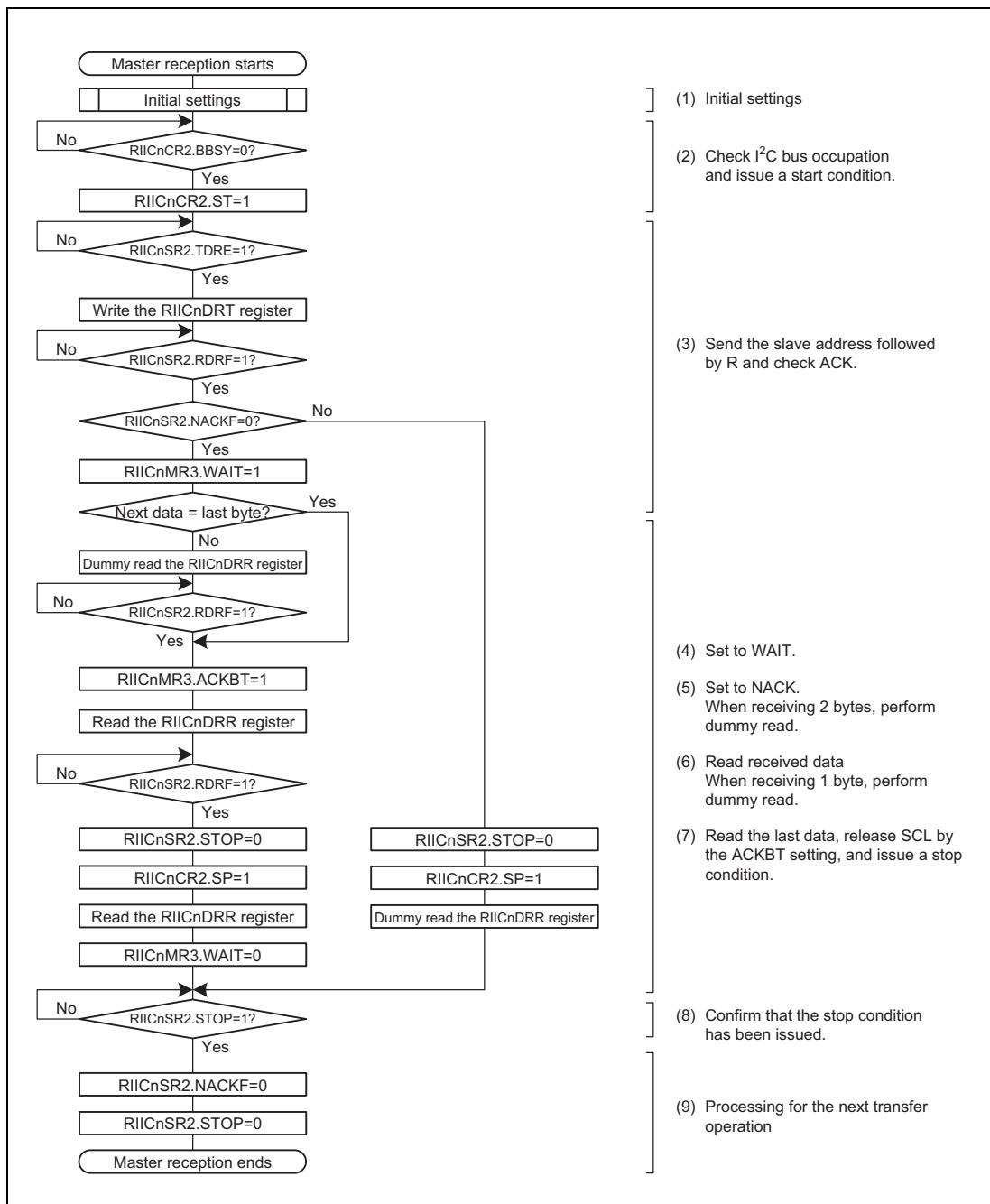
In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

**Figure 18.10** shows an example of master reception flowchart (7-bit address format, 1 or 2 bytes), **Figure 18.11** shows an example of master reception flowchart (7-bit address format, 3 bytes or more), and **Figure 18.12** to **Figure 18.14** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

- (1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 18.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.
- (2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
- (3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1. Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition. For master reception from a device with a 10-bit address, start by using master transmission to transmit the two higher-order bits of the slave address and then the eight lower-order bits of the slave address, and issue a restart condition following generation of the transmission end interrupt (or after TEND = 1) (see **Figure 18.13** for operation timing). After that, transmitting 1111 0<sub>B</sub> plus the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.
- (4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

- (5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.
- (6) When the RIICnMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).
- (7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.
- (8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00<sub>B</sub> and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.
- (9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.



**Figure 18.10 Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes)**

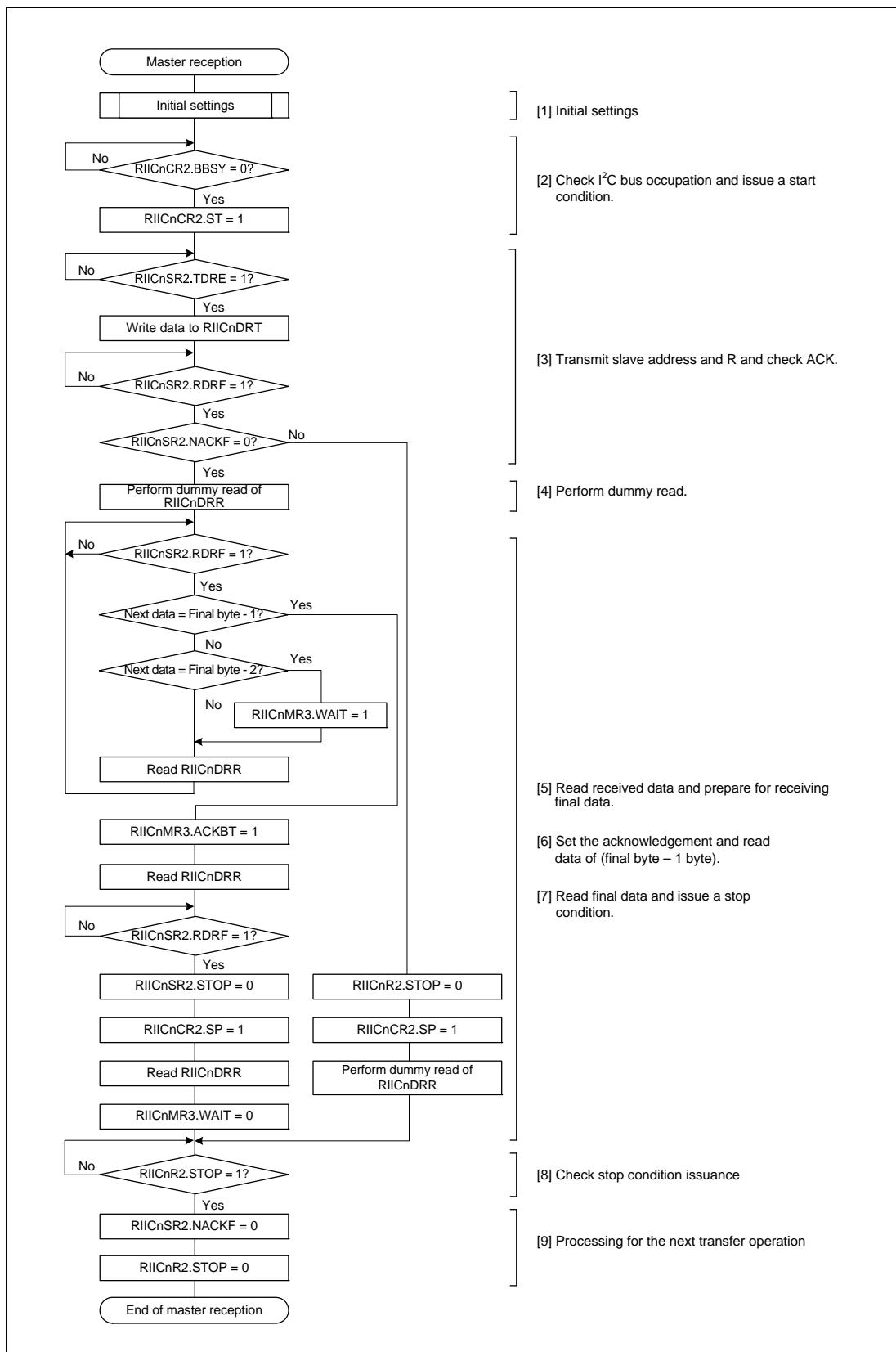


Figure 18.11 Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

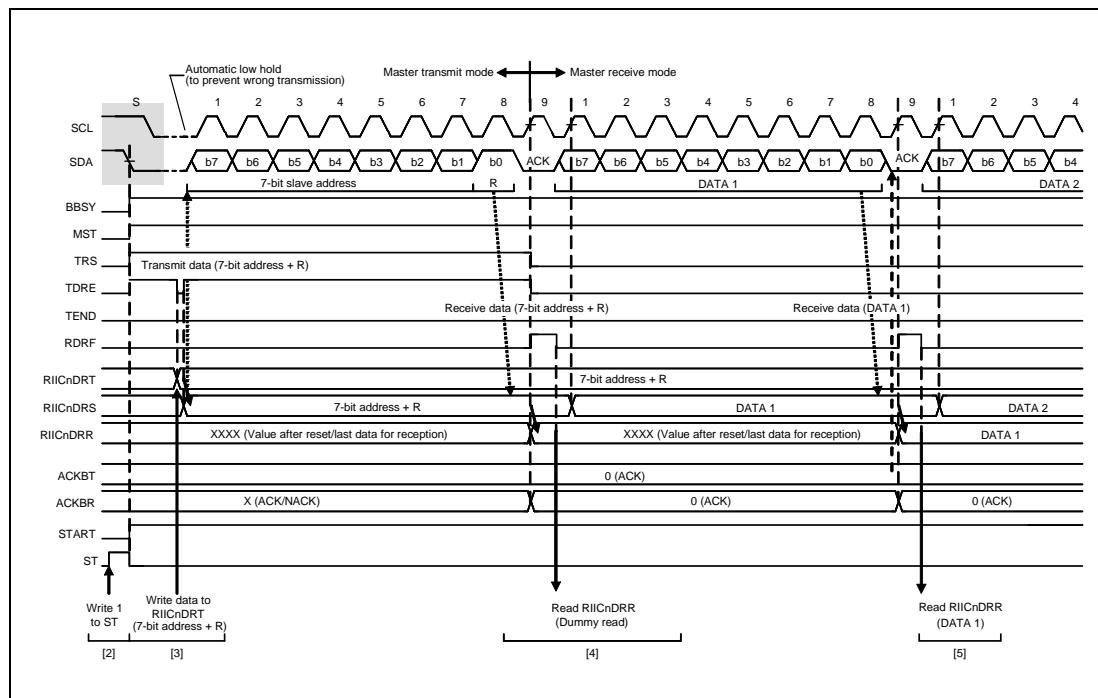


Figure 18.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

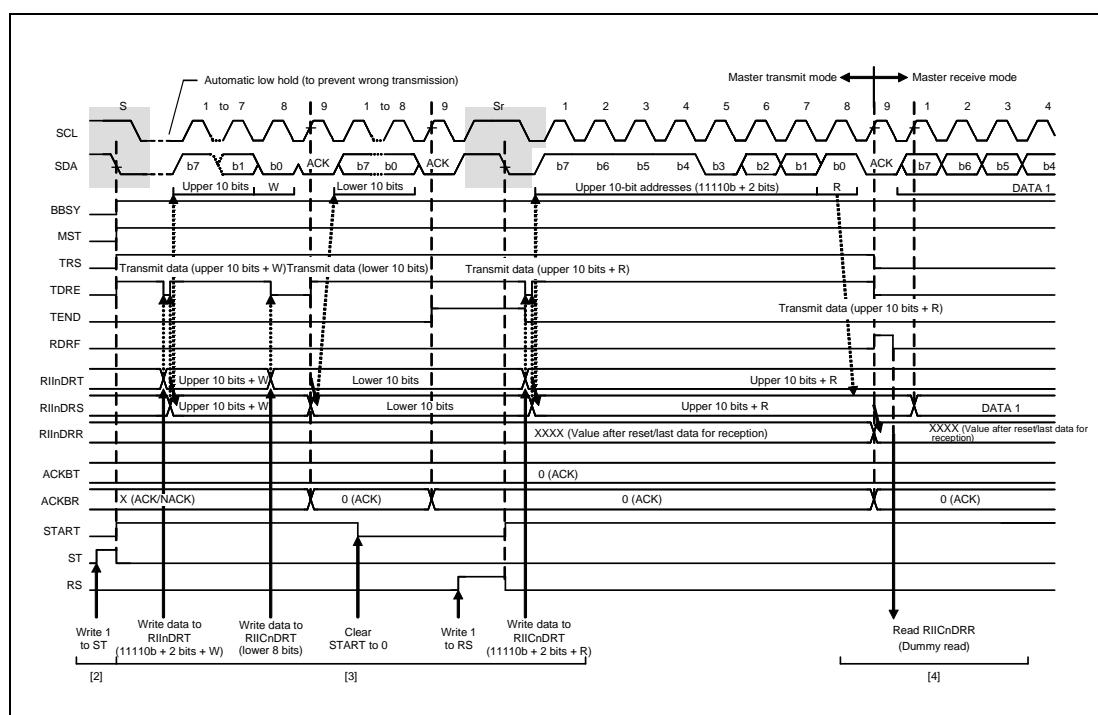


Figure 18.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

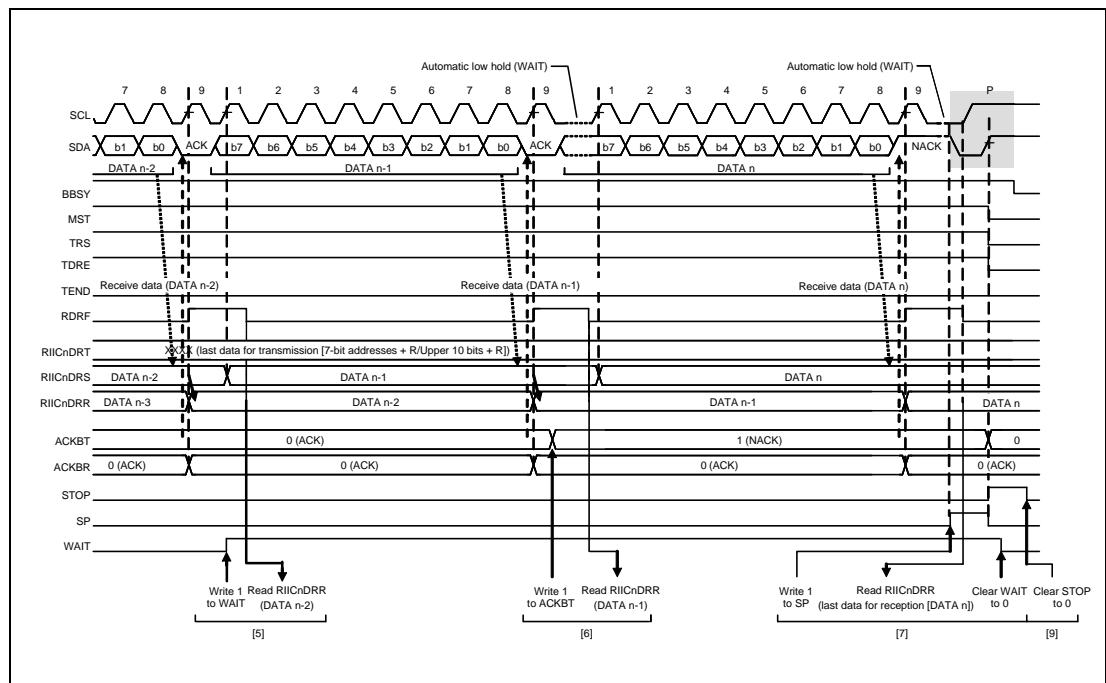


Figure 18.14 Master Receive Operation Timing (3) (when RDRFS = 0)

### 18.5.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

**Figure 18.5** shows an example of usage of slave transmission and **Figure 18.16** and **Figure 18.17** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

- (1) Follow the procedure in **Figure 18.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.
- (2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy ( $y = 0$  to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.
- (3) After the RIICnSR2.TDRE flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.
- (4) Wait until the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.
- (5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.
- (6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy ( $y = 0$  to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.
- (7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

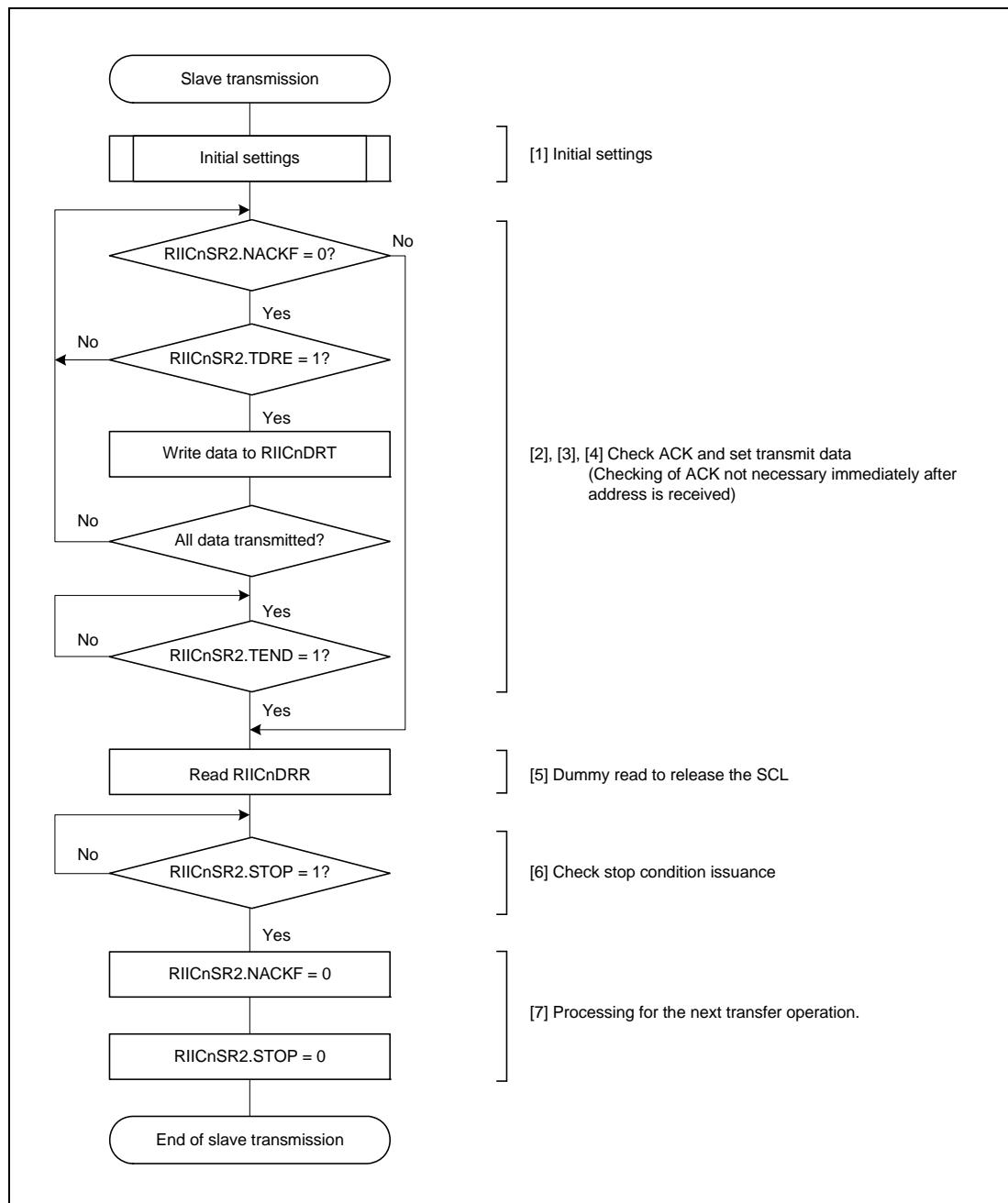


Figure 18.15 Example of Slave Transmission Flowchart

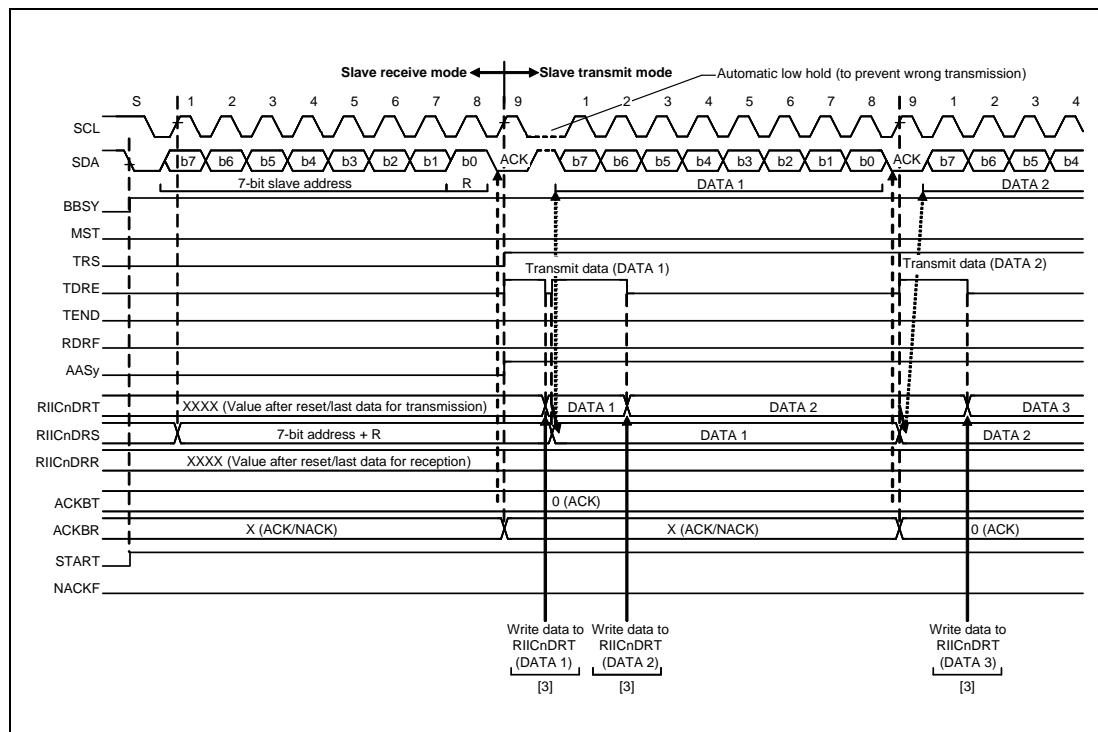


Figure 18.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

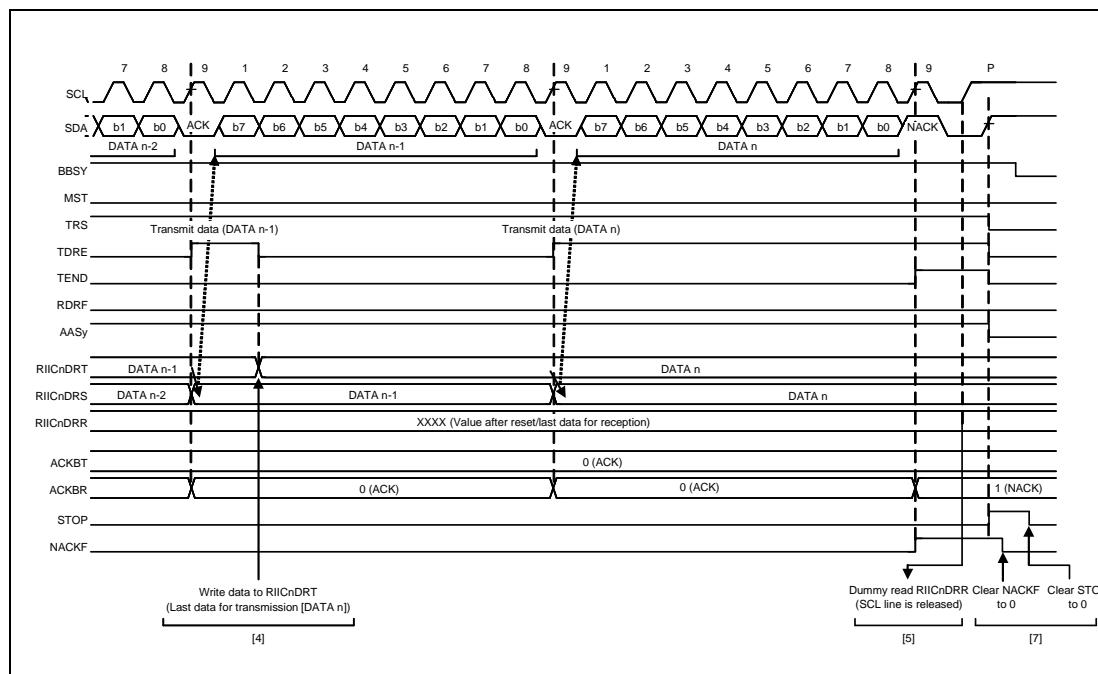


Figure 18.17 Slave Transmit Operation Timing (2)

### 18.5.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

**Figure 18.18** shows an example of usage of slave reception and **Figure 18.19** and **Figure 18.20** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

- (1) Follow the procedure in **Figure 18.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.
- (2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy ( $y = 0$  to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.
- (3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR as the first read operation (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).
- (4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.  
When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.
- (5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy ( $y = 0$  to 2) to 0. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.
- (6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

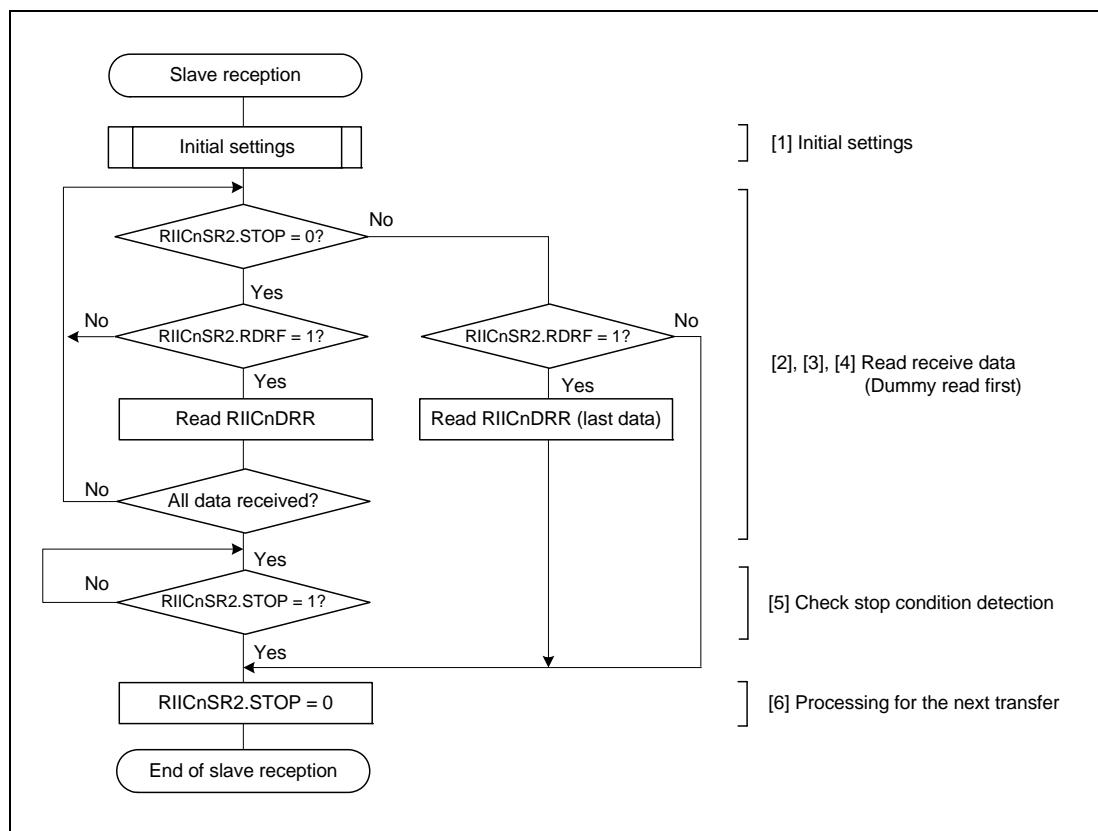


Figure 18.18 Example of Slave Reception Flowchart

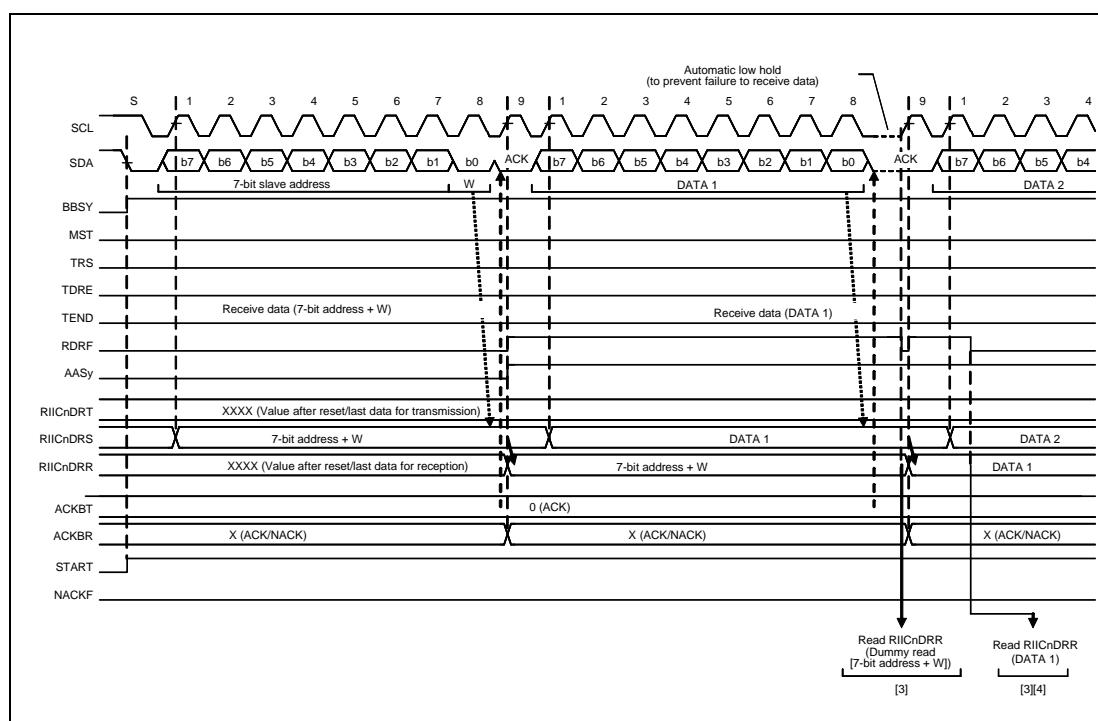
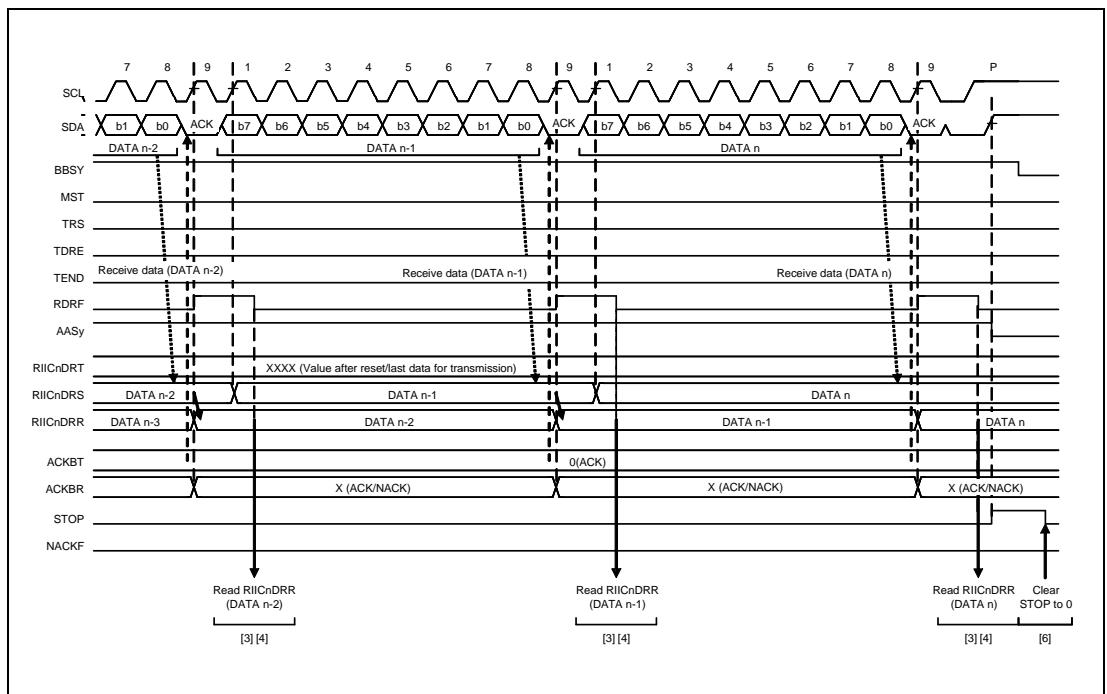


Figure 18.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)



**Figure 18.20 Slave Receive Operation Timing (2) (when RDRFS = 0)**

## 18.6 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I<sup>2</sup>C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.

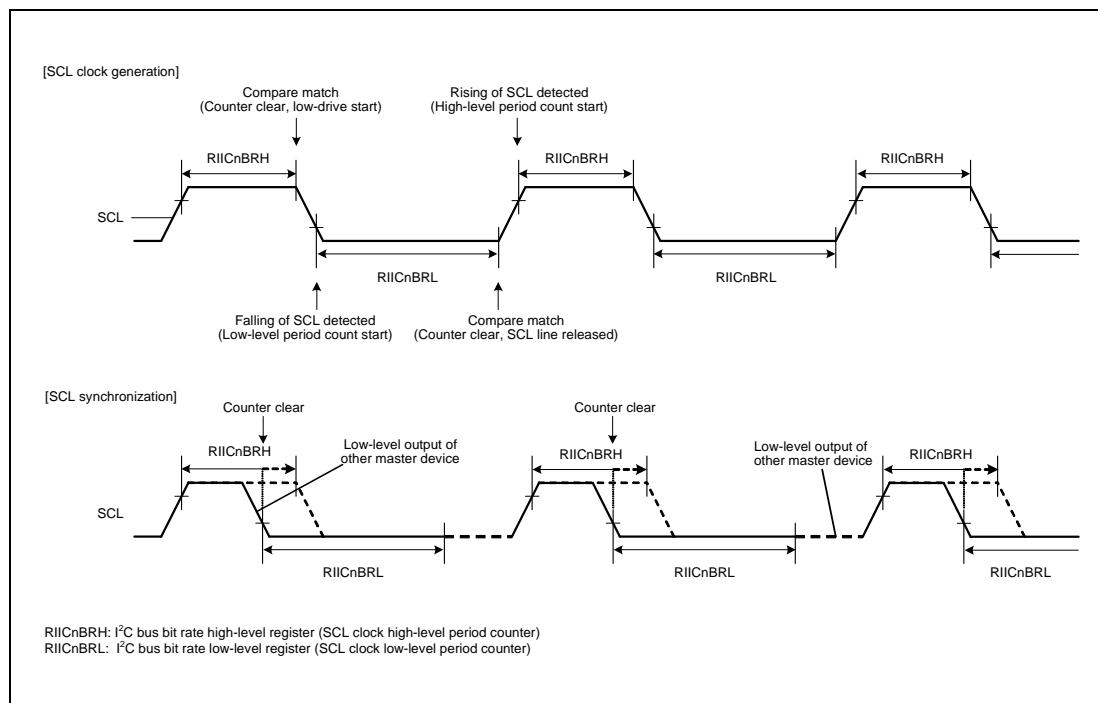


Figure 18.21 Generation and Synchronization of the SCL Signal from the RIIC

## 18.7 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000<sub>B</sub>, and disabled by setting the same bits to 000<sub>B</sub>.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits are set to any value other than 000<sub>B</sub>), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IIC $\phi$ ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IIC $\phi$ /2). The counter counts the number of cycles set in the SDDL[2:0] bits. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

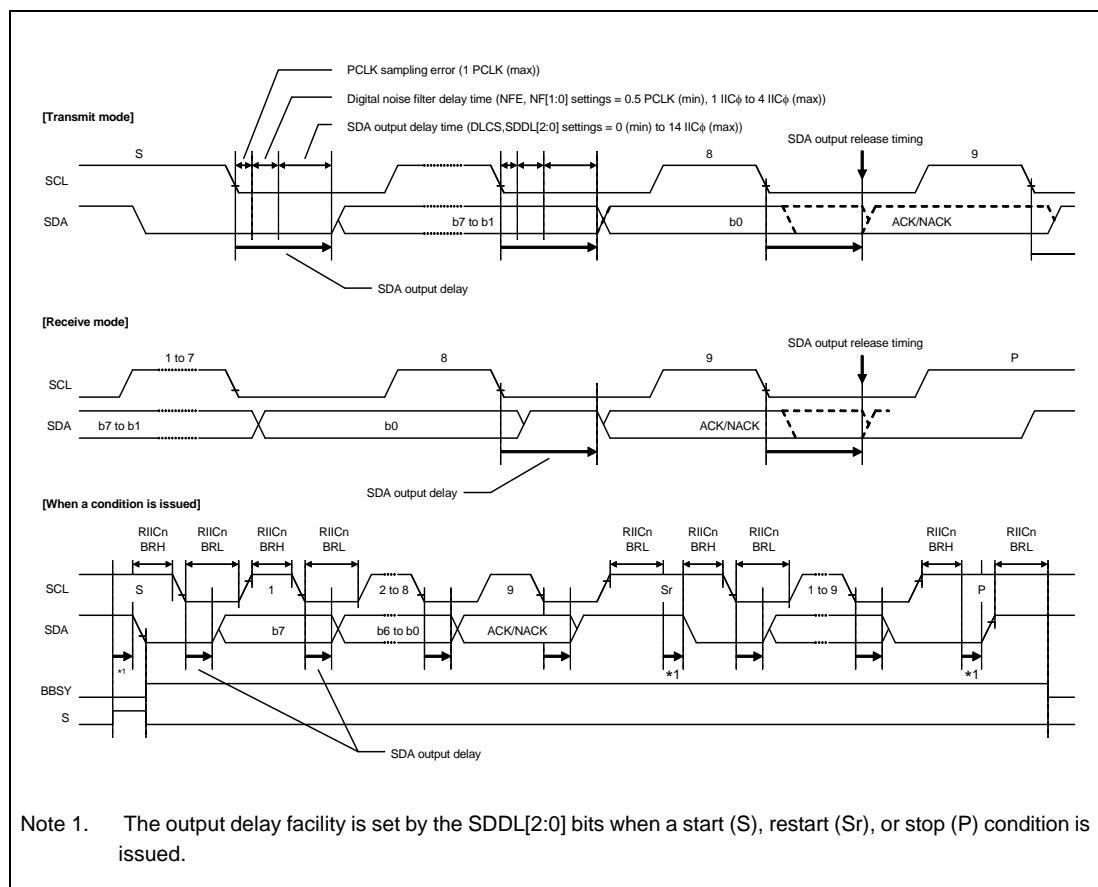


Figure 18.22 SDA Output Delay Facility

## 18.8 Digital Noise-Filter Circuits

The states of the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through the digital noise-filter circuit. **Figure 18.23** is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IIC $\phi$  cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IIC $\phi$  signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed as an internal signal. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small, the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the RIICnFER.NFE bit to 0).

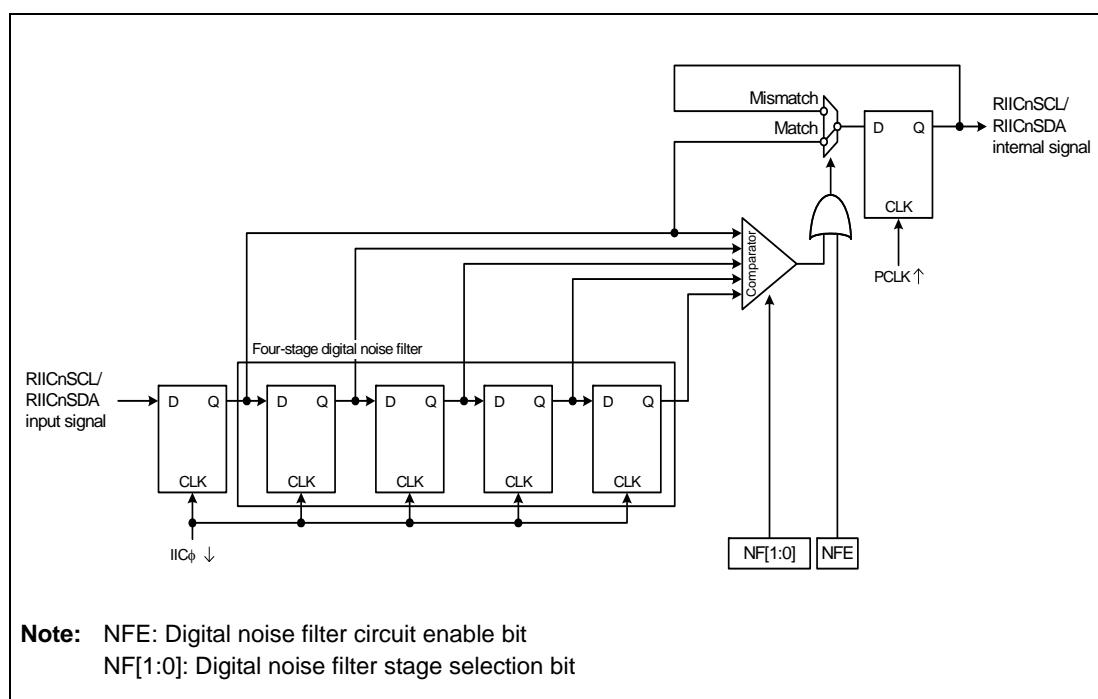


Figure 18.23 Block Diagram of Digital Noise Filter Circuit

## 18.9 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and device ID address, and also can set 7-bit or 10-bit slave addresses.

### 18.9.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit ( $y = 0$  to 2) is set to 1, the slave addresses set in RIICnSARy ( $y = 0$  to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag ( $y = 0$  to 2) is set to 1 at the rising edge of the ninth SCL clock cycle and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. The RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive complete interrupt (INTIICnRI) or transmit data empty interrupt (INTIICnTI) to be generated. The AASy flag is used to identify which slave address has been specified.

**Figure 18.24 to Figure 18.26** show the AASy flag set timing in three cases.

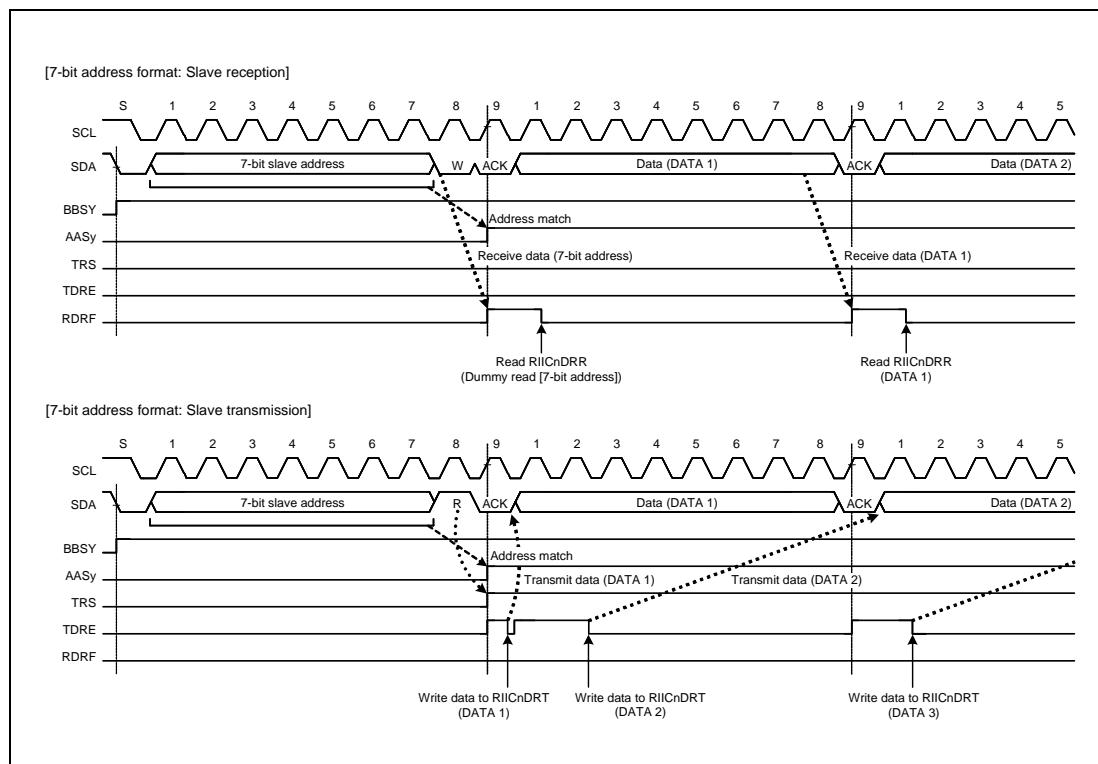


Figure 18.24 AASy Flag Set Timing with 7-Bit Address Format Selected

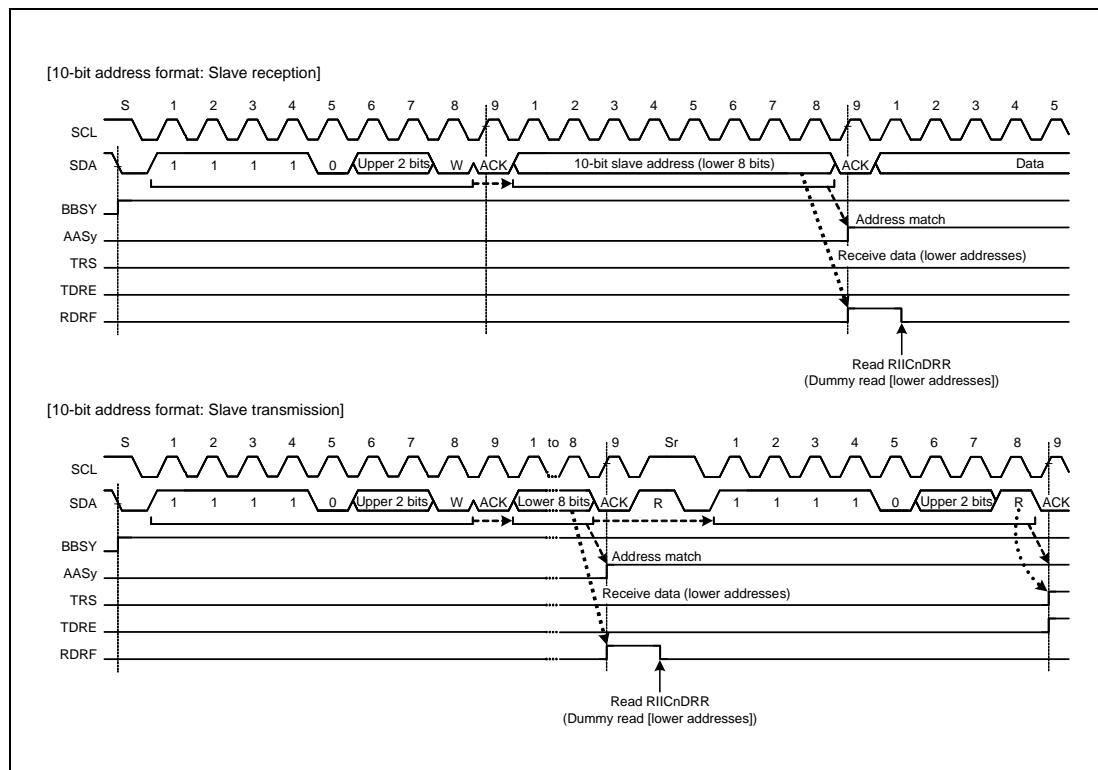


Figure 18.25 AASy Flag Set Timing with 10-Bit Address Format Selected

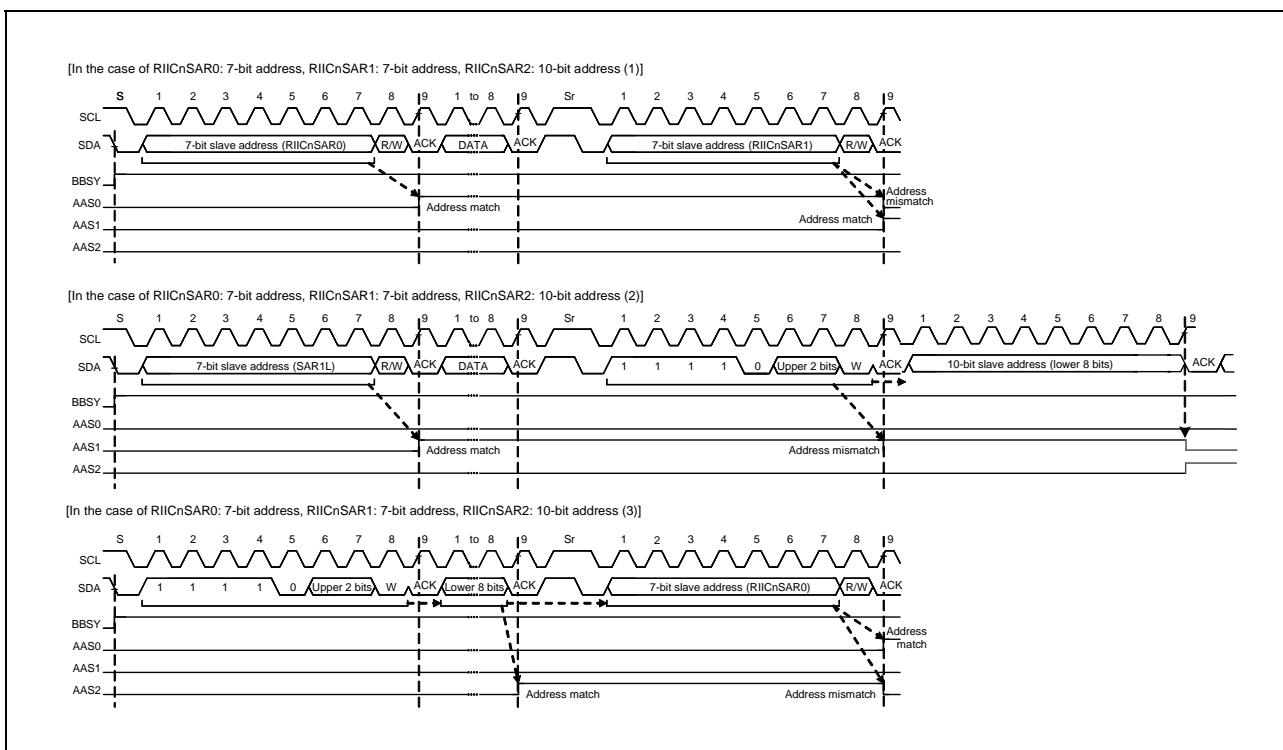


Figure 18.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

### 18.9.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address ( $0000\ 000_B + 0 [W]$ ). This is enabled by setting the RIICnSER.GCAE bit to 1.

If the address received after a start or restart condition is issued is  $0000\ 000_B + 1[R]$  (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive complete interrupt (INTIICnRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

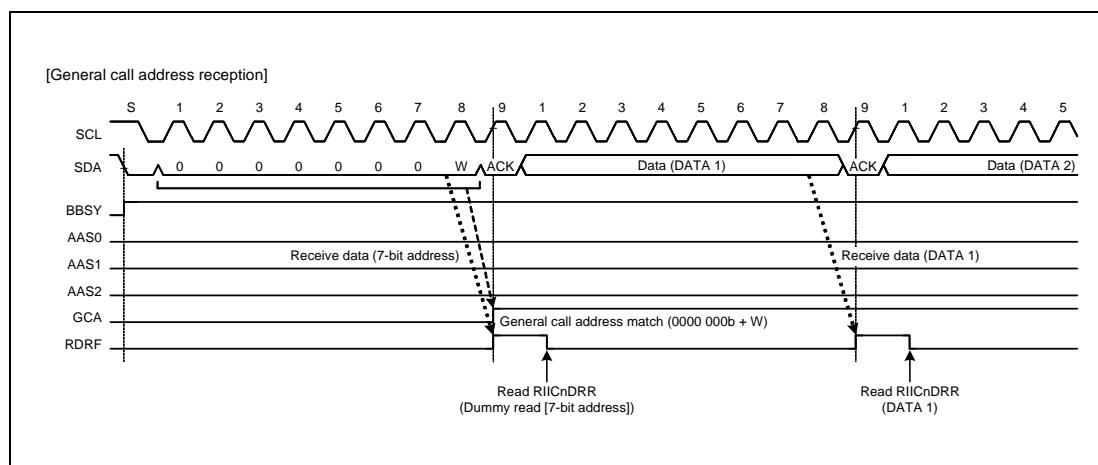


Figure 18.27 Timing of GCA Flag Setting during Reception of General Call Address

### 18.9.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conforming with the I<sup>2</sup>C bus specification (Rev. 03). When the RIIC receives 1111 100<sub>B</sub> as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100<sub>B</sub>) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100<sub>B</sub>) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details, see I<sup>2</sup>C Bus Standard from NXP Semiconductors.

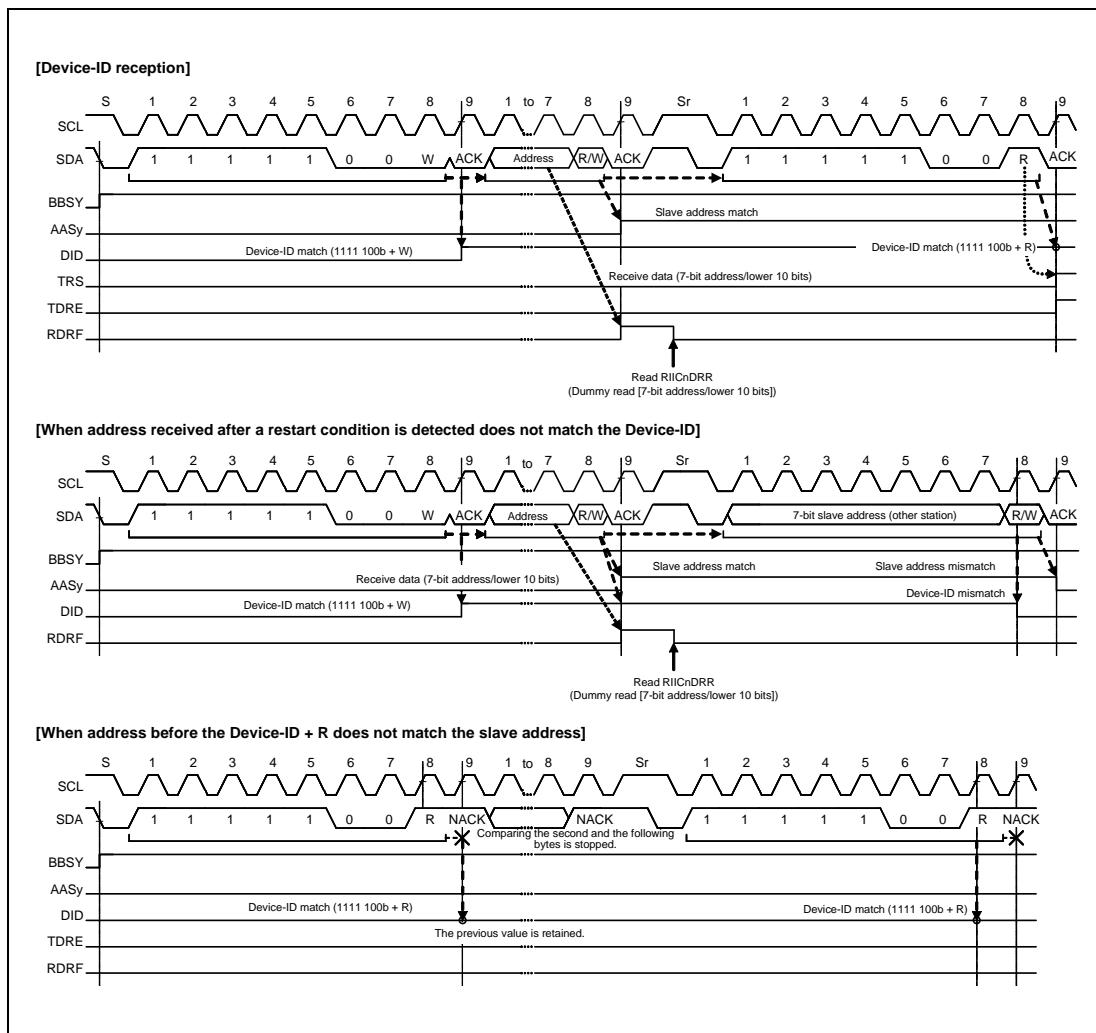


Figure 18.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

## 18.10 Automatic Low-Hold Function for SCL

### 18.10.1 Function to Prevent Wrong Transmission of Transmit Data

To prevent the unintended transmission of erroneous data, this low-hold period is extended until data for transmission have been written. In addition, the RIIC holds the SCL line low over the period until a stop condition is issued and also over the period until the RIICnDRR register is dummy read.

#### <Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle until a stop condition is issued

#### <Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle and the RIICnDRR register is dummy read

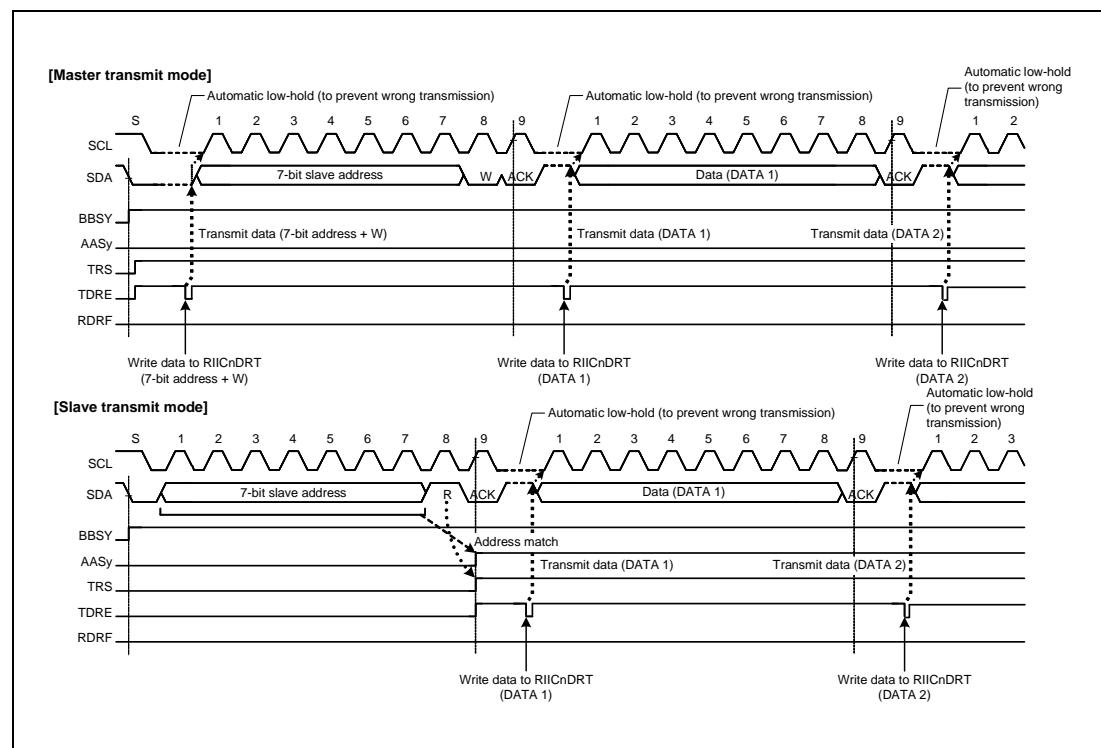


Figure 18.29 Automatic Low-Hold Operation in Transmit Mode

### 18.10.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

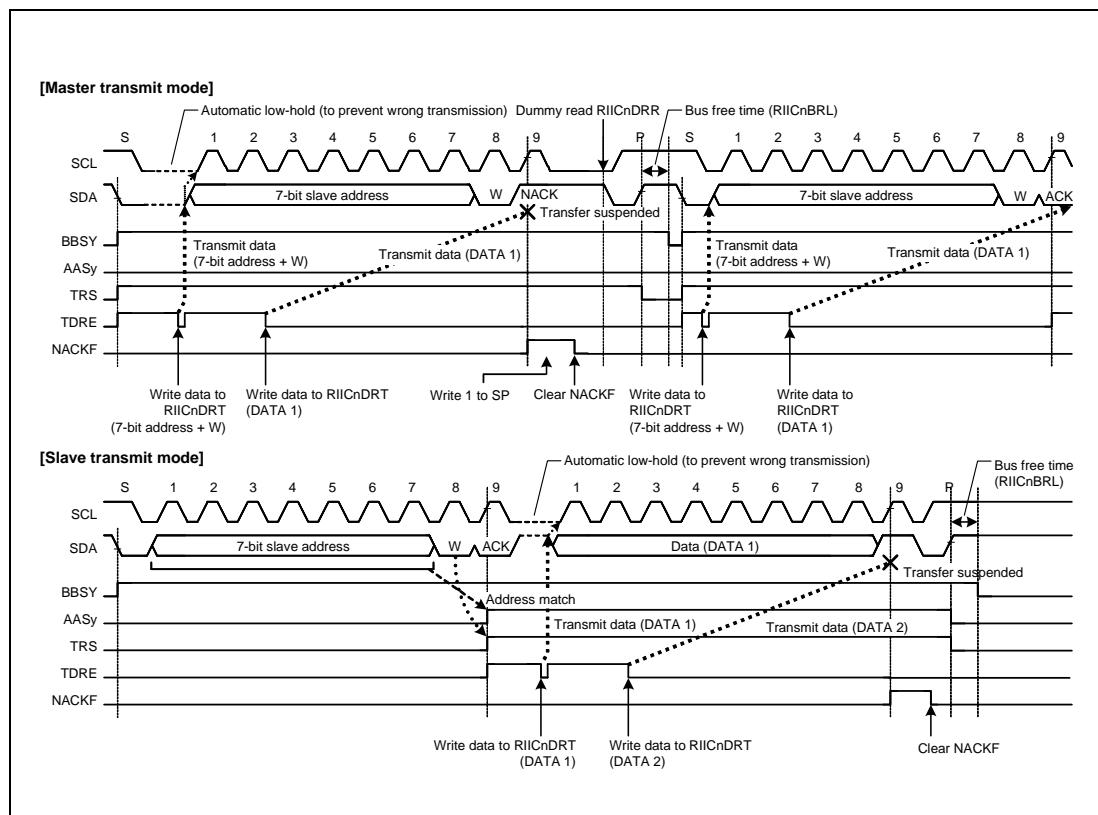


Figure 18.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

### 18.10.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive complete (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

#### (1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function.

Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

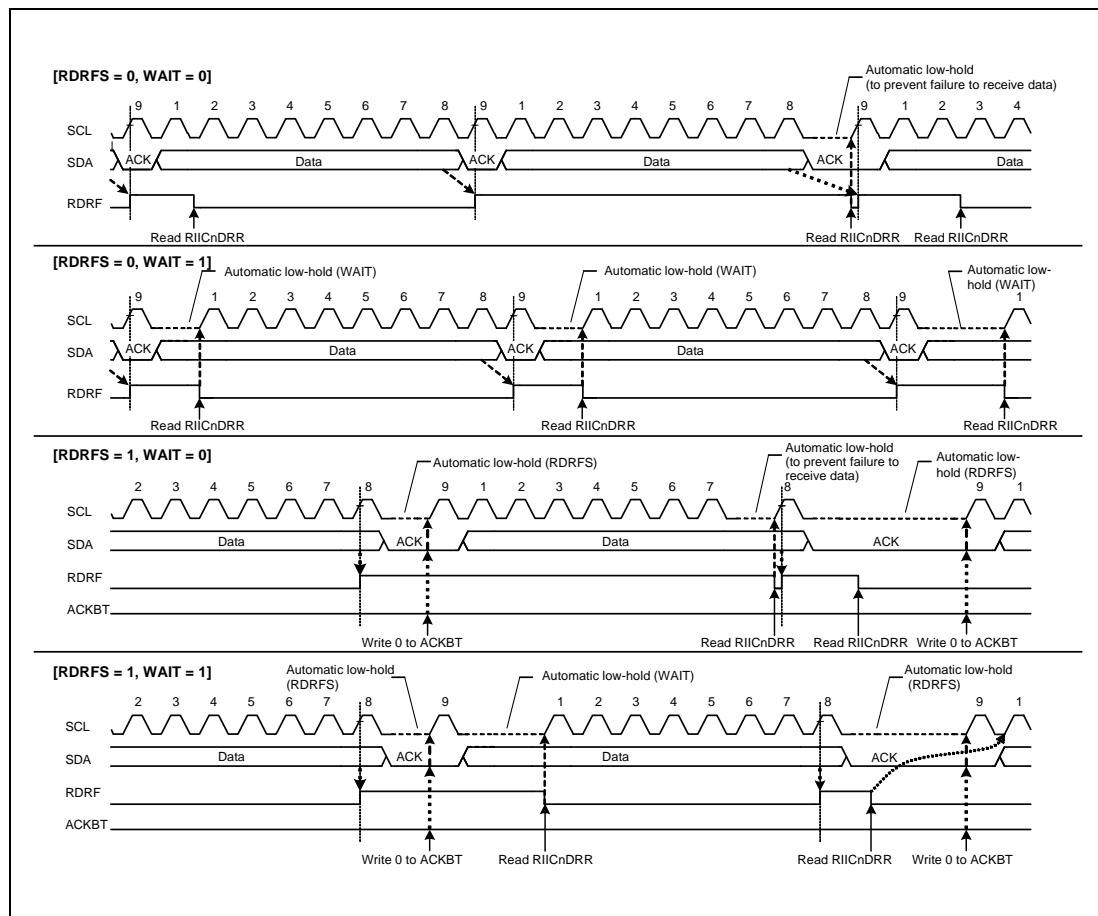
The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

#### (2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function.

When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive complete) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This lowhold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.



**Figure 18.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)**

## 18.11 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I<sup>2</sup>C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

### 18.11.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if the RIICnCR2.ST bit is set to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC considers itself to have lost in arbitration, so priority is given to transfer by the other master device and no start condition is generated.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line), the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11<sub>B</sub>)

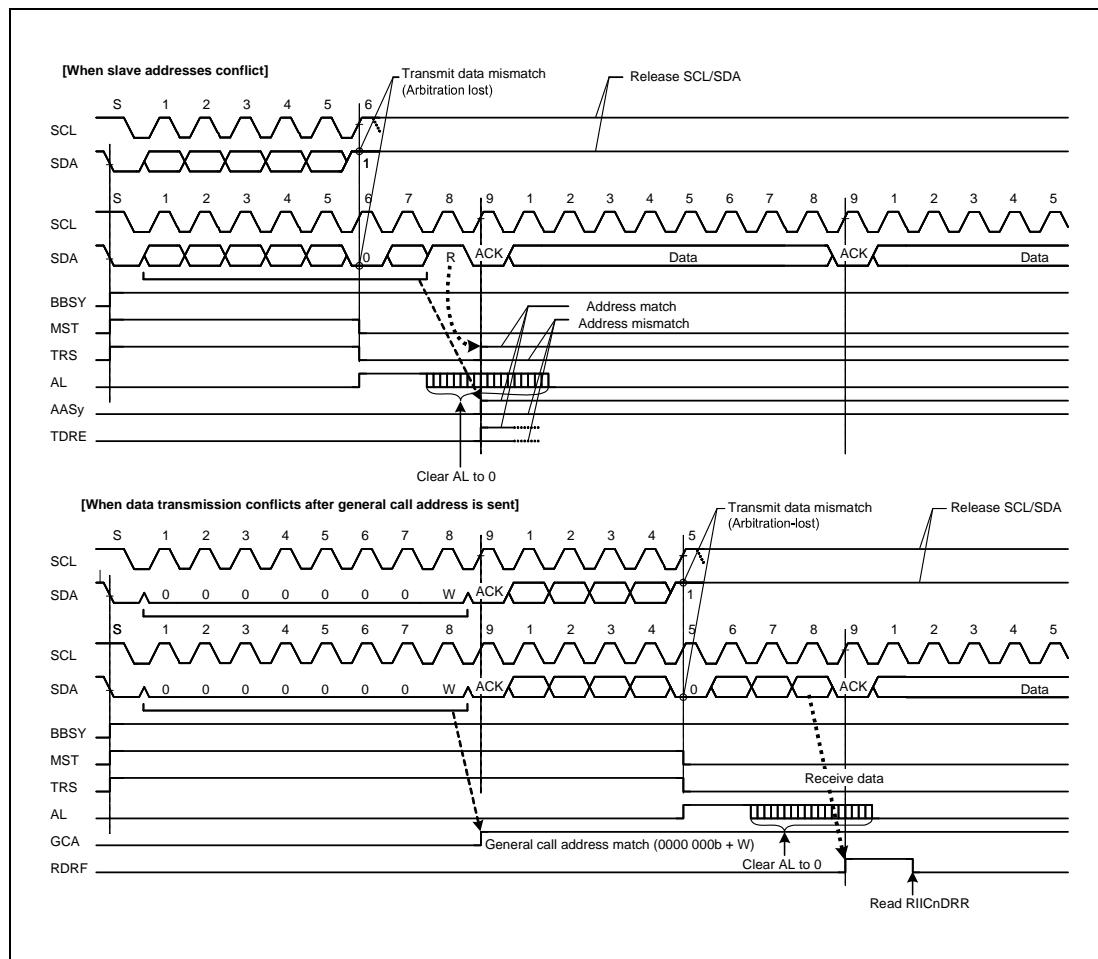


Figure 18.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

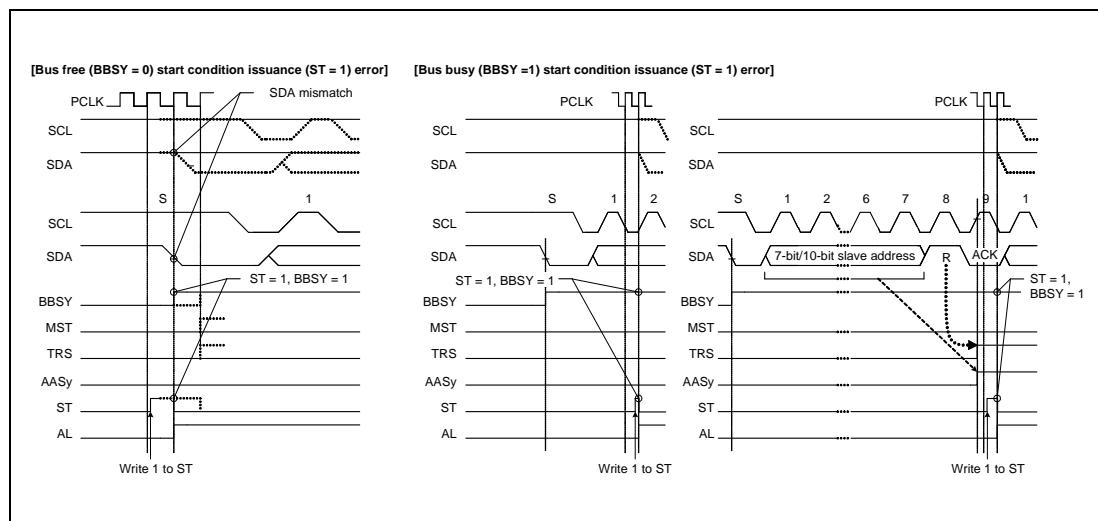
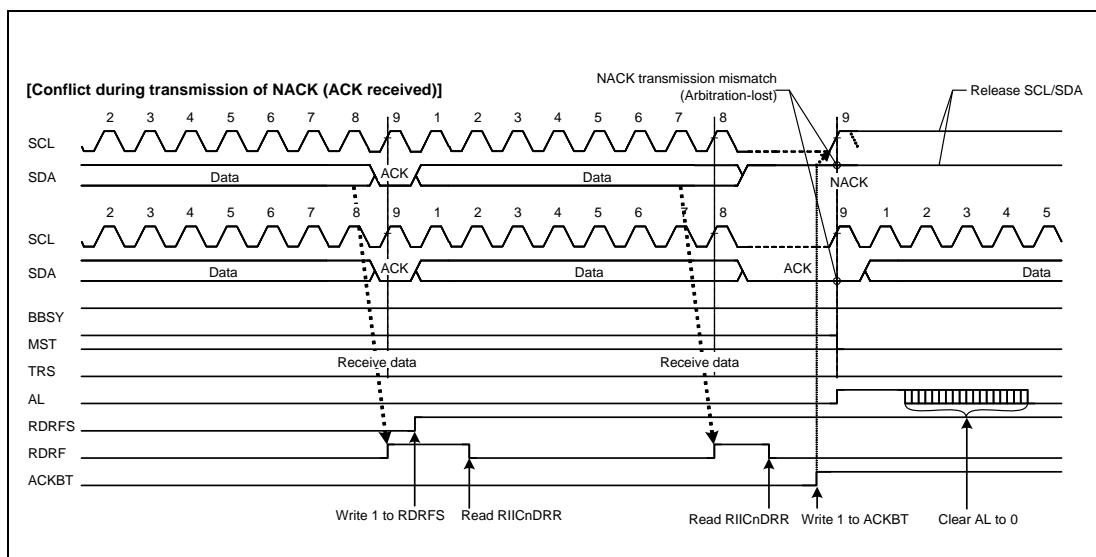


Figure 18.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

The TRS bit is not cleared if 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode.

### 18.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line) during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. **Figure 18.34** shows an example of arbitration-lost detection during transmission of NACK.



**Figure 18.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)**

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition.

Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1)

### 18.11.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the highimpedance state, and the low level is detected on the SDA line) in slave transmit mode.

When it loses slave arbitration, the RIIC enters slave receive mode.

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (RIICnCR2.MST and TRS bits = 01<sub>B</sub>)

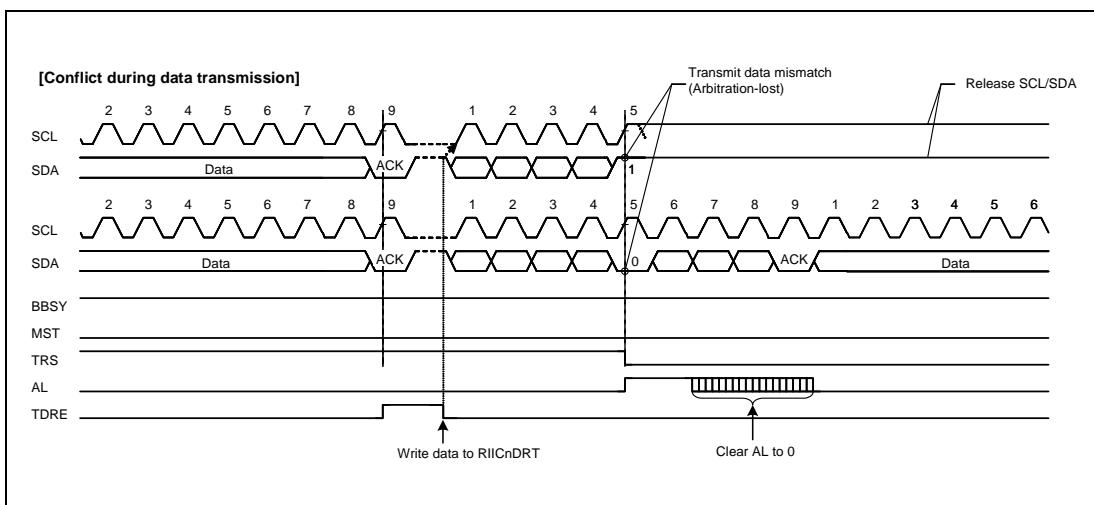


Figure 18.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

## 18.12 Start Condition/Restart Condition/Stop Condition Issuing Function

### 18.12.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

### 18.12.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made even during communication and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

(To detect the issuance of a restart condition, clear the RIICnSR2.START flag before a restart condition is issued.)

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

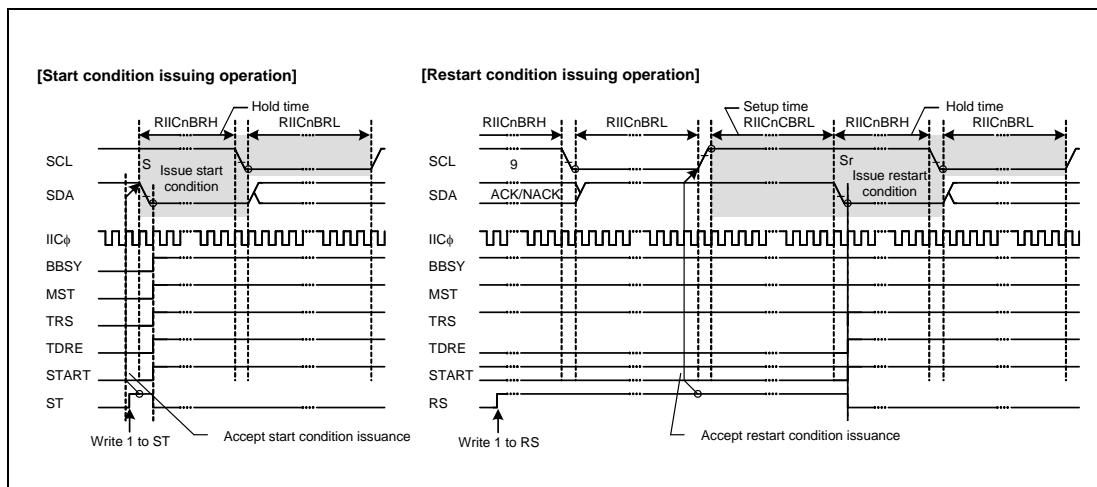


Figure 18.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

### 18.12.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

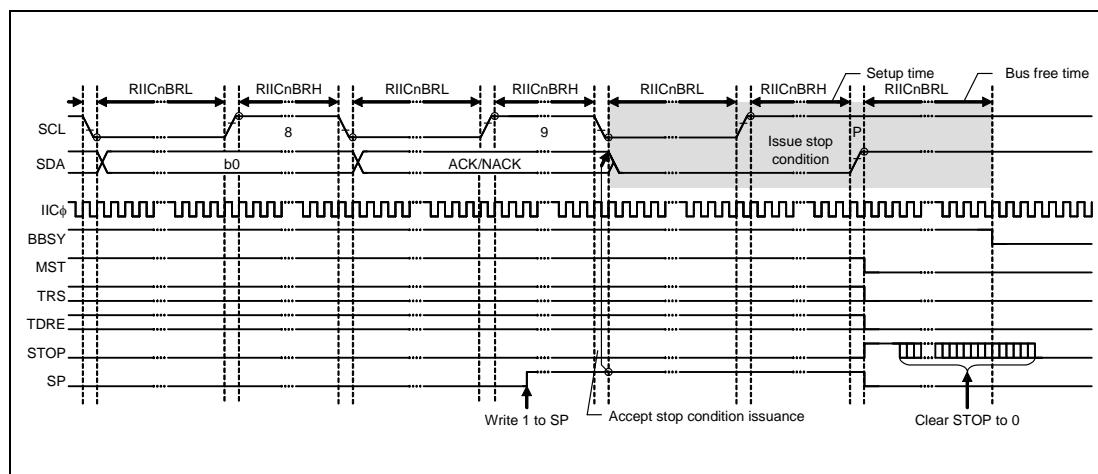


Figure 18.37 Stop Condition Issue Timing (SP Bit)

## 18.13 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I<sup>2</sup>C bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

### 18.13.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

The internal counter is cleared when one of the conditions is met.

- (1) When RIICnMR2.TMOH=0, and RIICnMR2.TMOL=1:  
The internal counter is cleared by SCL rising
- (2) When RIICnMR2.TMOH=1, and RIICnMR2.TMOL=0:  
The internal counter is cleared by SCL falling
- (3) When RIICnMR2.TMOH=RIICnMR2.TMOL=1:  
The internal counter is cleared by SCL rising or falling

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY flag is 1) in master mode (RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (RIICnSR1 register is not 00<sub>H</sub>) and the bus is busy (RIICnCR2.BBSY flag is 1) in slave mode (RIICnCR2.MST bit is 0).
- The bus is free (RIICnCR2.BBSY flag is 0) while generation of a START condition is requested (RIICnCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IIC $\phi$ ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

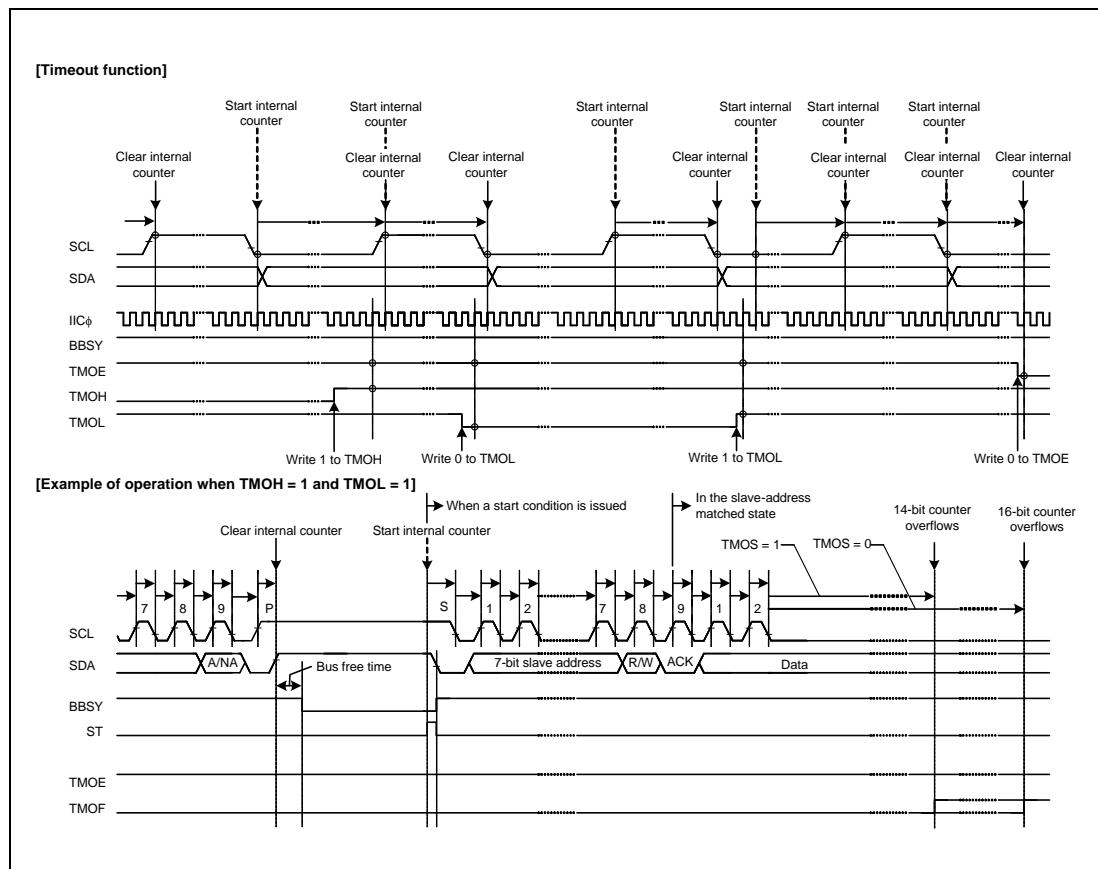


Figure 18.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

### 18.13.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

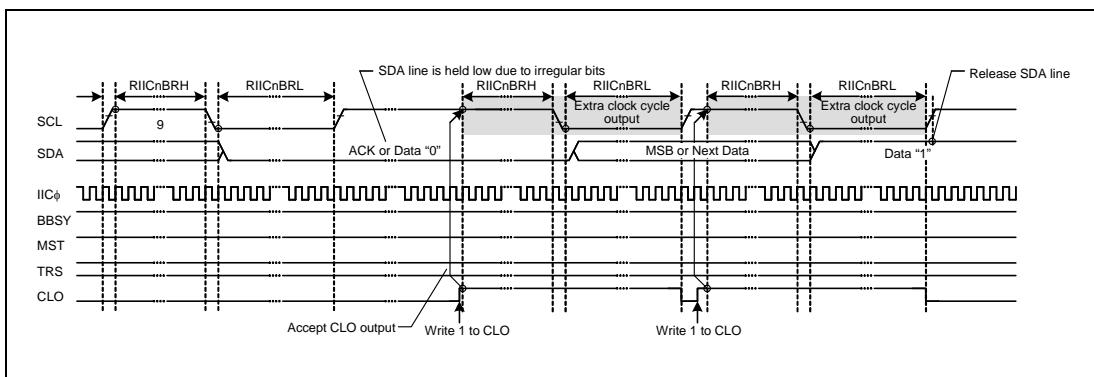
When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the RIICnFER.MALE bit (master arbitration-lost detection disabled) cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the RIICnCR1.SDAO bit does not match the state of the SDA line, so take care on this point.

[Additional output conditions for the SCL clock]

- In master mode and when the bus is free
- In master mode and the SCL line is not held low (the bus is busy)

**Figure 18.39** shows the operation timing of the extra SCL clock cycle output function (CLO bit).



**Figure 18.39** Extra SCL Clock Cycle Output Function (CLO Bit)

### 18.13.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings.

After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01<sub>B</sub>).

For a detailed description of the RIIC and internal resets, see **Section 18.14, Reset Function of RIIC**.

## 18.14 Reset Function of RIIC

The RIIC has RIIC reset, and internal reset functions. In addition RIIC is cleared by ISORES. **Table 18.26** lists the scope of each reset and reset conditions.

**Table 18.26 RIIC Reset Functions (1/2)**

UM		ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnCR1	ICE	Initialized	0	1	Retained	Retained
	IICRST	Initialized	1	1	Retained	Retained
	CLO	Initialized	Initialized	Retained	Retained	Retained
	SOWP	Initialized	Initialized	Retained	Retained	Retained
	SCLO	Initialized	Initialized	Initialized	Retained	Retained
	SDAO	Initialized	Initialized	Initialized	Retained	Retained
	SCLI	Initialized	Initialized	Retained	Retained	Retained
	SDAI	Initialized	Initialized	Retained	Retained	Retained
RIICnCR2	BBSY	Initialized	Initialized	Initialized *1	Operation	Retained
	MST	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	TRS	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	SP	Initialized	Initialized	Initialized	Initialized	Initialized
	RS	Initialized	Initialized	Initialized	Initialized	Initialized
	ST	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR1	MTWP	Initialized	Initialized	Retained	Retained	Retained
	CKS[2:0]	Initialized	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Initialized	Retained	Retained	Retained
	BC[2:0]	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR2		Initialized	Initialized	Retained	Retained	Retained
RIICnMR3	WAIT	Initialized	Initialized	Retained	Retained	Retained
	RDRFS	Initialized	Initialized	Retained	Retained	Retained
	ACKWP	Initialized	Initialized	Retained	Retained	Retained
	ACKBT	Initialized	Initialized	Retained	Retained	Initialized
	ACKBR	Initialized	Initialized	Retained	Retained	Retained
	NF[1:0]	Initialized	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Initialized	Retained	Retained	Retained
RIICnSR1	DID	Initialized	Initialized	Initialized	Retained	Initialized
	GCA	Initialized	Initialized	Initialized	Retained	Initialized
	AAS2	Initialized	Initialized	Initialized	Retained	Initialized
	AAS1	Initialized	Initialized	Initialized	Retained	Initialized
	AAS0	Initialized	Initialized	Initialized	Retained	Initialized
RIICnSR2	TDRE	Initialized	Initialized	Initialized	Retained	Initialized
	TEND	Initialized	Initialized	Initialized	Retained	Initialized
	RDRF	Initialized	Initialized	Initialized	Retained	Retained
	NACKF	Initialized	Initialized	Initialized	Retained	Retained
	STOP	Initialized	Initialized	Initialized	Retained	Operation
	START	Initialized	Initialized	Initialized	Operation	Initialized
	AL	Initialized	Initialized	Initialized	Retained	Retained
	TMOF	Initialized	Initialized	Initialized	Retained	Retained
RIICnSAR0, 1, 2		Initialized	Initialized	Retained	Retained	Retained

**Table 18.26 RIIC Reset Functions (2/2)**

<b>UM</b>	<b>ISORES</b>	<b>RIIC Reset (ICE = 0, IICRST = 1)</b>	<b>Internal Reset (ICE = 1, IICRST = 1)</b>	<b>Start/Restart Condition Detection</b>	<b>Stop Condition Detection</b>
RIICnBRH, RIICnBRL	Initialized	Initialized	Retained	Retained	Retained
RIICnDRT	Initialized	Initialized	Retained	Retained	Retained
RIICnDRR	Initialized	Initialized	Retained	Retained	Retained
RIICnDRS	Initialized	Initialized	Initialized	Retained	Retained

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.  
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared to 0.

## Section 19 CAN Interface (RS-CAN)

This section contains a generic description of the CAN Interface (RS-CAN).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RS-CAN.

### 19.1 Features of RH850/F1L RS-CAN

#### 19.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CAN units.

**Table 19.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units				1		
Name				RSCANn (n = 0)		

The individual products have the CAN interface channels listed below.

**Table 19.2 Unit Configurations and Channels**

Unit Name	Number of Channels per Unit	Channel Name	RH850/F1L 48 pins (1 ch)	RH850/F1L 64 pins (3 ch)	RH850/F1L 80 pins (3 ch)	RH850/F1L 100 pins (3 ch/6 ch <sup>*2</sup> )	RH850/F1L 144 pins (4 ch/6 ch <sup>*1</sup> )	RH850/F1L 176 pins (6 ch)
RSCAN0	0	CAN0	✓	✓	✓	✓	✓	✓
	1	CAN1		✓	✓	✓	✓	✓
	2	CAN2		✓	✓	✓	✓	✓
	3	CAN3				*2	✓	✓
	4	CAN4				*2	*1	✓
	5	CAN5				*2	*1	✓

Note 1. Only supported in devices with 1.5MB and 2MB code flash.

Note 2. Only supported in F1L for Gateway.

**Table 19.3 Indices (1/2)**

Index	Description
n	Throughout this section, the individual RS-CAN units are generically indicated by the index "n" (n = 0); for example, RSCANnGCTR is the global control register of the RSCANn unit.
m	Throughout this section, the individual channels of RS-CAN units are generically indicated by the index "m" (m = 0 to 5); for example, RSCAN0CmSTS is the channel m status register.
j	The individual registers associated with receive rule table are generically indicated by the index "j" (j = 0 to 15); for example, RSCAN0GAFLIDj is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index "k" (k = 0 to [channel m × 3 + 2]); for example, RSCAN0CFCCk is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers in the RS-CAN units are identified by the index "x" (x = 0 to 7); for example, RSCAN0RFSTSx is the receive FIFO buffer status register.

**Table 19.3 Indices (2/2)**

Index	Description
q	The individual receive buffers are generically indicated by the index "q" (q = 0 to [channel m × 16 + 15]); for example, RSCAN0RMIDq is the receive buffer ID register.
p	The individual transmit buffers are generically indicated by the index "p" (p = 0 to [channel m × 16 + 15]); for example, RSCAN0TMCP is the transmit buffer control register.
r	The individual RAM tests for CAN are generically indicated by the index "r" (r = 0 to 63); for example, RSCAN0RPGACCr is the RAM test page access register.
y	The registers not covered above are indicated by the index "y" (y = 0 to 2); for example, RSCAN0RMNDy is a receive buffer new data register.

**Note:** The functions and descriptions of registers in this section are for a 6-channel RS-CAN (m = 0 to 5). When referring to information with indices, regard the index values as the ones corresponding to your product. Also, note that, when writing a value to a register that will result in writing to bits outside the range of the index for the product you are using, write the value after reset to these bits.

The following table lists the values of indices for individual products.

**Table 19.4 Indices for Individual Products**

Indices for Individual Products						
48 pins	64 pins, 80 pins	100 pins	100 pins (F1L for Gateway Only)	144 pins (Except for 1.5 MB, 2 MB embedded)	144 pins (1.5 MB, 2 MB embedded)	176 pins
j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15
k = 0 to 2	k = 0 to 8	k = 0 to 8	k = 0 to 17	k = 0 to 11	k = 0 to 17	k = 0 to 17
x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7
q = 0 to 15	q = 0 to 47	q = 0 to 47	q = 0 to 95	q = 0 to 63	q = 0 to 95	q = 0 to 95
p = 0 to 15	p = 0 to 47	p = 0 to 47	p = 0 to 95	p = 0 to 63	p = 0 to 95	p = 0 to 95
r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63
y = 0	y = 0, 1	y = 0, 1	y = 0 to 2	y = 0, 1	y = 0 to 2	y = 0 to 2

## 19.1.2 Register Base Address

RSCAN0 base address is listed in the following table.

RSCAN0 register addresses are given as offsets from the base address.

**Table 19.5 Register Base Address**

Base Address Name	Base Address
<RSCAN0_base>	FFD0 0000H

### 19.1.3 Clock Supply

The RSCAN0 clock supply is shown in the following table.

**Table 19.6 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
RSCAN0	clk_xincan	CKSCLK_ICANOSC
	clkc	PPLLCLK2
	pclk	CKSCLK_ICAN
	Register access clock	CKSCLK_ICAN

The operating frequency of the RSCAN0 depends on the transfer rate and the number of channels in use. **Table 19.7** shows the range of the frequency.

**Table 19.7 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L**

Condition		Range of Operating Frequency		
Transfer Rate	No. of Channels in Use	pclk	clk_xincan <sup>*1, *3</sup>	clkc <sup>*1, *2</sup>
1 Mbps	6ch	pclk ≥ 53MHz	8 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	5ch	pclk ≥ 46MHz		
	4ch	pclk ≥ 40MHz		
	3ch	pclk ≥ 32MHz		
	2ch	pclk ≥ 26MHz		
	1ch	pclk ≥ 18MHz		
500 kbps	6ch	pclk ≥ 27MHz	4 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	5ch	pclk ≥ 23MHz		
	4ch	pclk ≥ 20MHz		
	3ch	pclk ≥ 16MHz		
	2ch	pclk ≥ 13MHz		
	1ch	pclk ≥ 8MHz		
125 kbps	6ch	pclk ≥ 8MHz	4 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	5ch			
	4ch			
	3ch			
	2ch			
	1ch			

Note 1. The DCS bit in RSCAN0GCFG allows the selection of either clk\_xincan or clkc. Configure the clock frequency to less than or equal to pclk/2 (up to 40 MHz).

Note 2. Select clk\_xincan when pclk < 25 MHz.

Note 3. The maximum frequency of clk\_xincan is 24 MHz.

#### CAUTION

When the RS-CAN module is used in STOP mode, set the MainOSC as the clock source of the RS-CAN module. For details about how to set the clock source, see Section 10.4.3.10, RS-CAN Clock Domains C\_ISO\_CAN and C\_ISO\_CANOSC.

### 19.1.4 Interrupt Requests

RSCAN0 interrupt requests are listed in the following table.

Table 19.8 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>RSCAN0</b>			
INTRCANGERR	CAN global error interrupt	14	—
INTRCANGRECC	CAN receive FIFO interrupt	15	—
<b>CAN0</b>			
INTRCANmERR (m = 0)	CAN0 error interrupt	16	—
INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	17	—
INTRCANmTRX (m = 0)	CAN0 transmit interrupt	18	—
<b>CAN1</b>			
INTRCANmERR (m = 1)	CAN1 error interrupt	105	—
INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	106	—
INTRCANmTRX (m = 1)	CAN1 transmit interrupt	107	—
<b>CAN2</b>			
INTRCANmERR (m = 2)	CAN2 error interrupt	209	—
INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	210	—
INTRCANmTRX (m = 2)	CAN2 transmit interrupt	211	—
<b>CAN3</b>			
INTRCANmERR (m = 3)	CAN3 error interrupt	212	—
INTRCANmREC (m = 3)	CAN3 transmit/receive FIFO receive completion interrupt	213	—
INTRCANmTRX (m = 3)	CAN3 transmit interrupt	214	—
<b>CAN4</b>			
INTRCANmERR (m = 4)	CAN4 error interrupt	264	—
INTRCANmREC (m = 4)	CAN4 transmit/receive FIFO receive completion interrupt	265	—
INTRCANmTRX (m = 4)	CAN4 transmit interrupt	266	—
<b>CAN5</b>			
INTRCANmERR (m = 5)	CAN5 error interrupt	279	—
INTRCANmREC (m = 5)	CAN5 transmit/receive FIFO receive completion interrupt	280	—
INTRCANmTRX (m = 5)	CAN5 transmit interrupt	281	—

#### NOTE

For the wake-up factors from standby mode, see **Section 11.1.2.1, Wake-Up Factors for Stand-By Modes**.

### 19.1.5 Reset Sources

RSCAN0 reset sources are listed in the following table. RSCAN0 is initialized by these reset sources.

**Table 19.9 Reset Sources**

Unit Name	Reset Source
RSCAN0	All reset sources (ISORES)

### 19.1.6 External Input/Output Signals

External input/output signals of RSCAN0 are listed below.

**Table 19.10 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal
<b>CAN0</b>		
CANmRX (m = 0)	CAN0 receive data input	CAN0RX
CANmTX (m = 0)	CAN0 transmit data output	CAN0TX
<b>CAN1</b>		
CANmRX (m = 1)	CAN1 receive data input	CAN1RX
CANmTX (m = 1)	CAN1 transmit data output	CAN1TX
<b>CAN2</b>		
CANmRX (m = 2)	CAN2 receive data input	CAN2RX
CANmTX (m = 2)	CAN2 transmit data output	CAN2TX
<b>CAN3</b>		
CANmRX (m = 3)	CAN3 receive data input	CAN3RX
CANmTX (m = 3)	CAN3 transmit data output	CAN3TX
<b>CAN4</b>		
CANmRX (m = 4)	CAN4 receive data input	CAN4RX
CANmTX (m = 4)	CAN4 transmit data output	CAN4TX
<b>CAN5</b>		
CANmRX (m = 5)	CAN5 receive data input	CAN5RX
CANmTX (m = 5)	CAN5 transmit data output	CAN5TX
<b>CAUTION</b>		
When port P0_0 is used as CAN0TX, port P0_0 (the <u>RESETOUT</u> signal) outputs a low level during a reset and after release from the reset state.		
For details, see <b>Section 2.11.1.1 P0_0: RESETOUT</b>		

## 19.2 Overview

### 19.2.1 Functional Overview

The RH850/F1L incorporates one CAN interface unit (RS-CAN) which implements six channels (CAN0 to CAN5) of CAN controllers conforming to the ISO11898-1 specifications. **Table 19.11** shows the RS-CAN module specifications. **Figure 19.1** shows the RS-CAN module block diagram.

**Table 19.11 RS-CAN Module Specifications (1/2)**

Item	Specification
Number of channels	6
Protocol	ISO11898-1 compliant
Communication speed	<ul style="list-style-type: none"> <li>Maximum 1 Mbps</li> </ul> $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ $\text{CANm bit time} = \text{CANmTq} \times \text{Tq count per bit}$ $\text{CANmTq} = \frac{(\text{BRP[9:0] bits in the RSCAN0CmCFG register} + 1)}{\text{fCAN}}$ <p>m = 0 to 5  Tq: Time quantum  fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCAN0GCFG register)</p>
Buffer	<p>480 buffers in total</p> <ul style="list-style-type: none"> <li>Dedicated buffers: 96 buffers (16 buffers × 6 channels)  Transmit buffer: 16 buffers per channel  Transmit queue: Single queue per channel (shared with the transmit buffer; up to 16 buffers allocatable)</li> <li>Shared between channels: 384 buffers for all channels  Receive buffer: 0 to 96 buffers  Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each)  Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each)</li> <li>Built-in ECC</li> </ul>
Reception function	<ul style="list-style-type: none"> <li>Receives data frames and remote frames.</li> <li>Selects ID format (standard ID, extended ID, or both IDs) to be received.</li> <li>Sets interrupt enable/disable for each FIFO.</li> <li>Mirror function (reception of messages transmitted from the own CAN node)</li> <li>Timestamp function (to record message reception time as a 16-bit timer value)</li> </ul>
Reception filter function	<ul style="list-style-type: none"> <li>Filters received messages according to 384 receive rules.</li> <li>Sets the number of receive rules (0 to 128) for each channel.</li> <li>Acceptance filter processing: Sets ID and mask for each receive rule.</li> <li>DLC filter processing: Enables DLC filter check for each reception rule.</li> </ul>
Received message transfer function	<ul style="list-style-type: none"> <li>Routing function  Transfers received messages to arbitrary destinations (can be transferred to up to 8 buffers)  Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer</li> <li>Labeling function  Stores label information together with the message in a receive buffer and FIFO buffer.</li> </ul>
Transmission function	<ul style="list-style-type: none"> <li>Transmits data frames and remote frames.</li> <li>Selects ID format (standard ID, extended ID, or both IDs) to be transmitted.</li> <li>Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer.</li> <li>Selects ID priority transmission or transmit buffer number priority transmission.</li> <li>Transmission can be aborted (possible to confirm with a flag)</li> <li>One-shot transmission function</li> </ul>
Interval transmission function	Transmits messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)

**Table 19.11 RS-CAN Module Specifications (2/2)**

Item	Specification
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmission-completed messages
Gateway function	Transmits a received message automatically.
Bus-off recovery mode selection	<p>Selects the method for recovering from bus-off state.</p> <ul style="list-style-type: none"> <li>• ISO11898-1 compliant</li> <li>• Automatic transition to channel halt mode at bus-off entry</li> <li>• Automatic transition to channel halt mode at bus-off end</li> <li>• Transition to channel halt mode by program request</li> <li>• Transition to the error-active state by program request (forcible recovery from the bus-off state)</li> </ul>
Error status monitoring	<ul style="list-style-type: none"> <li>• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock).</li> <li>• Detects error status transitions (error warning, error passive, bus-off entry, and bus-off recovery)</li> <li>• Reads the error counter.</li> <li>• Monitors DLC errors.</li> </ul>
Interrupt sources	<p>20 interrupts</p> <ul style="list-style-type: none"> <li>• Global interrupts (2 interrupts)           <ul style="list-style-type: none"> <li>Receive FIFO interrupt</li> <li>Global error interrupt</li> </ul> </li> <li>• Channel interrupts (3 interrupts/channel)           <ul style="list-style-type: none"> <li>CANm transmit interrupt (<math>m = 0</math> to 5)               <ul style="list-style-type: none"> <li>– CANm transmit complete interrupt</li> <li>– CANm transmit abort interrupt</li> <li>– CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)</li> <li>– CANm transmit history interrupt</li> <li>– CANm transmit queue interrupt</li> </ul> </li> <li>CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode)</li> <li>CANm error interrupt</li> </ul> </li> </ul>
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CAN module.
CAN clock source	Selects the clk_c or the clk_xincan. For the operating frequency range, see <b>Table 19.7</b> .
Test function	<p>Test function for user evaluation</p> <ul style="list-style-type: none"> <li>• Listen-only mode</li> <li>• Self-test mode 0 (external loopback)</li> <li>• Self-test mode 1 (internal loopback)</li> <li>• RAM test (read/write test)</li> <li>• Inter-channel communication test (48 pins product is not supported.)</li> </ul>

### 19.2.2 Block Diagram

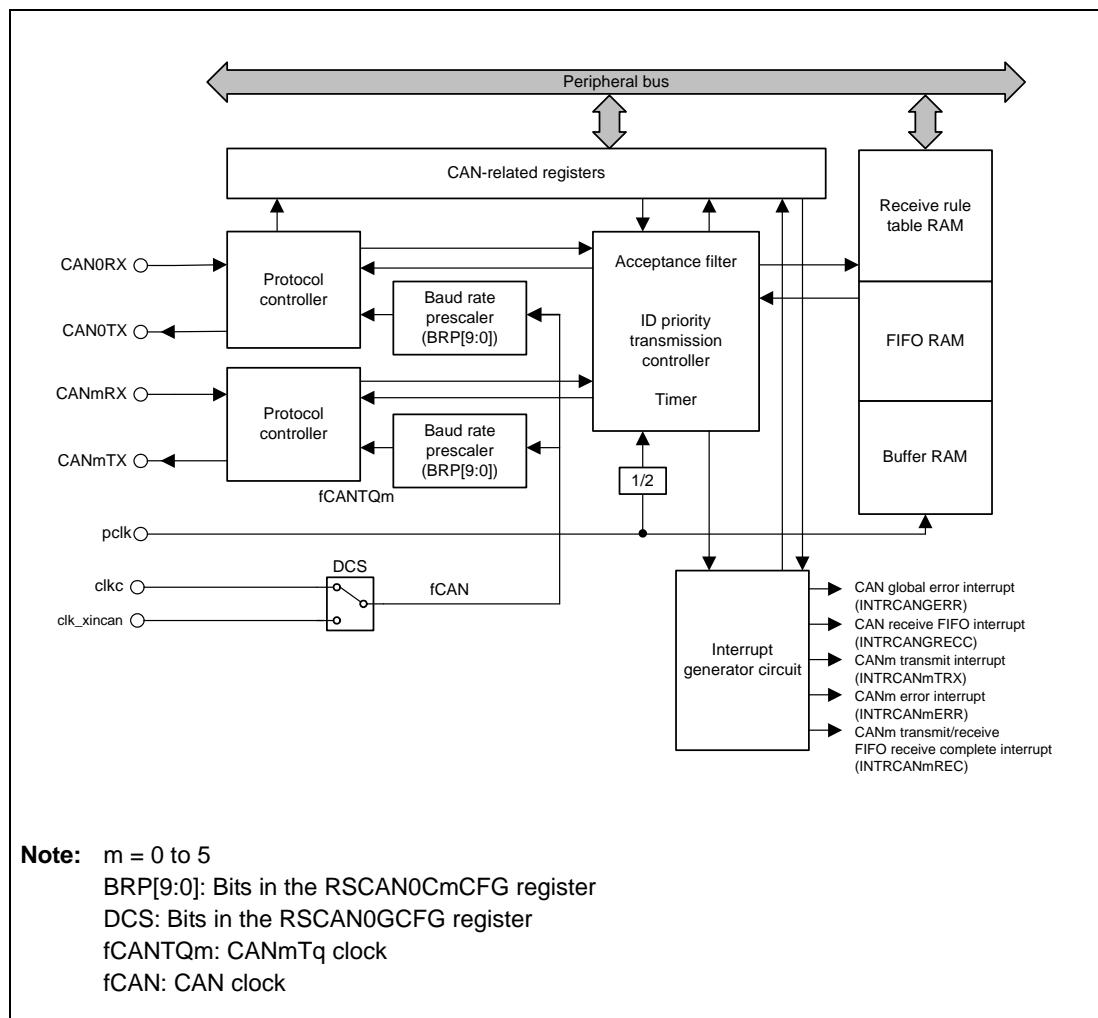


Figure 19.1 RS-CAN Module Block Diagram

## 19.3 Registers

### 19.3.1 List of Registers

RS-CAN registers are listed in the following table.

For details about <RSCAN0\_base>, see **Section 19.1.2, Register Base Address**.

Table 19.12 List of Registers (1/32)

Module	Register	Symbol	Address
RSCAN0	Channel 0 configuration register	RSCAN0C0CFG	<RSCAN0_base> + 0000H
RSCAN0	Channel 0 control register	RSCAN0C0CTR	<RSCAN0_base> + 0004H
RSCAN0	Channel 0 status register	RSCAN0C0STS	<RSCAN0_base> + 0008H
RSCAN0	Channel 0 error flag register	RSCAN0C0ERFL	<RSCAN0_base> + 000CH
RSCAN0	Channel 1 configuration register	RSCAN0C1CFG	<RSCAN0_base> + 0010H
RSCAN0	Channel 1 control register	RSCAN0C1CTR	<RSCAN0_base> + 0014H
RSCAN0	Channel 1 status register	RSCAN0C1STS	<RSCAN0_base> + 0018H
RSCAN0	Channel 1 error flag register	RSCAN0C1ERFL	<RSCAN0_base> + 001CH
RSCAN0	Channel 2 configuration register	RSCAN0C2CFG	<RSCAN0_base> + 0020H
RSCAN0	Channel 2 control register	RSCAN0C2CTR	<RSCAN0_base> + 0024H
RSCAN0	Channel 2 status register	RSCAN0C2STS	<RSCAN0_base> + 0028H
RSCAN0	Channel 2 error flag register	RSCAN0C2ERFL	<RSCAN0_base> + 002CH
RSCAN0	Channel 3 configuration register	RSCAN0C3CFG	<RSCAN0_base> + 0030H
RSCAN0	Channel 3 control register	RSCAN0C3CTR	<RSCAN0_base> + 0034H
RSCAN0	Channel 3 status register	RSCAN0C3STS	<RSCAN0_base> + 0038H
RSCAN0	Channel 3 error flag register	RSCAN0C3ERFL	<RSCAN0_base> + 003CH
RSCAN0	Channel 4 configuration register	RSCAN0C4CFG	<RSCAN0_base> + 0040H
RSCAN0	Channel 4 control register	RSCAN0C4CTR	<RSCAN0_base> + 0044H
RSCAN0	Channel 4 status register	RSCAN0C4STS	<RSCAN0_base> + 0048H
RSCAN0	Channel 4 error flag register	RSCAN0C4ERFL	<RSCAN0_base> + 004CH
RSCAN0	Channel 5 configuration register	RSCAN0C5CFG	<RSCAN0_base> + 0050H
RSCAN0	Channel 5 control register	RSCAN0C5CTR	<RSCAN0_base> + 0054H
RSCAN0	Channel 5 status register	RSCAN0C5STS	<RSCAN0_base> + 0058H
RSCAN0	Channel 5 error flag register	RSCAN0C5ERFL	<RSCAN0_base> + 005CH
RSCAN0	Global configuration register	RSCAN0GCFG	<RSCAN0_base> + 0084H
RSCAN0	Global control register	RSCAN0GCTR	<RSCAN0_base> + 0088H
RSCAN0	Global status register	RSCAN0GSTS	<RSCAN0_base> + 008CH
RSCAN0	Global error flag register	RSCAN0GERFL	<RSCAN0_base> + 0090H
RSCAN0	Global timestamp counter register	RSCAN0GTSC	<RSCAN0_base> + 0094H
RSCAN0	Receive rule entry control register	RSCAN0GAFLECTR	<RSCAN0_base> + 0098H
RSCAN0	Receive rule configuration register 0	RSCAN0GAFLCFG0	<RSCAN0_base> + 009CH
RSCAN0	Receive rule configuration register 1	RSCAN0GAFLCFG1	<RSCAN0_base> + 00A0H
RSCAN0	Receive buffer number register	RSCAN0RMNB	<RSCAN0_base> + 00A4H
RSCAN0	Receive buffer new data register 0	RSCAN0RMND0	<RSCAN0_base> + 00A8H
RSCAN0	Receive buffer new data register 1	RSCAN0RMND1	<RSCAN0_base> + 00ACH
RSCAN0	Receive buffer new data register 2	RSCAN0RMND2	<RSCAN0_base> + 00B0H
RSCAN0	Receive FIFO buffer configuration and control register 0	RSCAN0RFCC0	<RSCAN0_base> + 00B8H
RSCAN0	Receive FIFO buffer configuration and control register 1	RSCAN0RFCC1	<RSCAN0_base> + 00BCH

Table 19.12 List of Registers (2/32)

Module	Register	Symbol	Address
RSCAN0	Receive FIFO buffer configuration and control register 2	RSCAN0RFCC2	<RSCAN0_base> + 00C0H
RSCAN0	Receive FIFO buffer configuration and control register 3	RSCAN0RFCC3	<RSCAN0_base> + 00C4H
RSCAN0	Receive FIFO buffer configuration and control register 4	RSCAN0RFCC4	<RSCAN0_base> + 00C8H
RSCAN0	Receive FIFO buffer configuration and control register 5	RSCAN0RFCC5	<RSCAN0_base> + 00CCH
RSCAN0	Receive FIFO buffer configuration and control register 6	RSCAN0RFCC6	<RSCAN0_base> + 00D0H
RSCAN0	Receive FIFO buffer configuration and control register 7	RSCAN0RFCC7	<RSCAN0_base> + 00D4H
RSCAN0	Receive FIFO buffer status register 0	RSCAN0RFSTS0	<RSCAN0_base> + 00D8H
RSCAN0	Receive FIFO buffer status register 1	RSCAN0RFSTS1	<RSCAN0_base> + 00DCH
RSCAN0	Receive FIFO buffer status register 2	RSCAN0RFSTS2	<RSCAN0_base> + 00E0H
RSCAN0	Receive FIFO buffer status register 3	RSCAN0RFSTS3	<RSCAN0_base> + 00E4H
RSCAN0	Receive FIFO buffer status register 4	RSCAN0RFSTS4	<RSCAN0_base> + 00E8H
RSCAN0	Receive FIFO buffer status register 5	RSCAN0RFSTS5	<RSCAN0_base> + 00ECH
RSCAN0	Receive FIFO buffer status register 6	RSCAN0RFSTS6	<RSCAN0_base> + 00F0H
RSCAN0	Receive FIFO buffer status register 7	RSCAN0RFSTS7	<RSCAN0_base> + 00F4H
RSCAN0	Receive FIFO buffer pointer control register 0	RSCAN0RFPCTR0	<RSCAN0_base> + 00F8H
RSCAN0	Receive FIFO buffer pointer control register 1	RSCAN0RFPCTR1	<RSCAN0_base> + 00FCH
RSCAN0	Receive FIFO buffer pointer control register 2	RSCAN0RFPCTR2	<RSCAN0_base> + 0100H
RSCAN0	Receive FIFO buffer pointer control register 3	RSCAN0RFPCTR3	<RSCAN0_base> + 0104H
RSCAN0	Receive FIFO buffer pointer control register 4	RSCAN0RFPCTR4	<RSCAN0_base> + 0108H
RSCAN0	Receive FIFO buffer pointer control register 5	RSCAN0RFPCTR5	<RSCAN0_base> + 010CH
RSCAN0	Receive FIFO buffer pointer control register 6	RSCAN0RFPCTR6	<RSCAN0_base> + 0110H
RSCAN0	Receive FIFO buffer pointer control register 7	RSCAN0RFPCTR7	<RSCAN0_base> + 0114H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 0	RSCAN0CFCC0	<RSCAN0_base> + 0118H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 1	RSCAN0CFCC1	<RSCAN0_base> + 011CH
RSCAN0	Transmit/receive FIFO buffer configuration and control register 2	RSCAN0CFCC2	<RSCAN0_base> + 0120H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 3	RSCAN0CFCC3	<RSCAN0_base> + 0124H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 4	RSCAN0CFCC4	<RSCAN0_base> + 0128H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 5	RSCAN0CFCC5	<RSCAN0_base> + 012CH
RSCAN0	Transmit/receive FIFO buffer configuration and control register 6	RSCAN0CFCC6	<RSCAN0_base> + 0130H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 7	RSCAN0CFCC7	<RSCAN0_base> + 0134H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 8	RSCAN0CFCC8	<RSCAN0_base> + 0138H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 9	RSCAN0CFCC9	<RSCAN0_base> + 013CH
RSCAN0	Transmit/receive FIFO buffer configuration and control register 10	RSCAN0CFCC10	<RSCAN0_base> + 0140H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 11	RSCAN0CFCC11	<RSCAN0_base> + 0144H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 12	RSCAN0CFCC12	<RSCAN0_base> + 0148H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 13	RSCAN0CFCC13	<RSCAN0_base> + 014CH
RSCAN0	Transmit/receive FIFO buffer configuration and control register 14	RSCAN0CFCC14	<RSCAN0_base> + 0150H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 15	RSCAN0CFCC15	<RSCAN0_base> + 0154H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 16	RSCAN0CFCC16	<RSCAN0_base> + 0158H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 17	RSCAN0CFCC17	<RSCAN0_base> + 015CH
RSCAN0	Transmit/receive FIFO buffer status register 0	RSCAN0CFSTS0	<RSCAN0_base> + 0178H
RSCAN0	Transmit/receive FIFO buffer status register 1	RSCAN0CFSTS1	<RSCAN0_base> + 017CH
RSCAN0	Transmit/receive FIFO buffer status register 2	RSCAN0CFSTS2	<RSCAN0_base> + 0180H
RSCAN0	Transmit/receive FIFO buffer status register 3	RSCAN0CFSTS3	<RSCAN0_base> + 0184H

Table 19.12 List of Registers (3/32)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer status register 4	RSCAN0CFSTS4	<RSCAN0_base> + 0188 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 5	RSCAN0CFSTS5	<RSCAN0_base> + 018C <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 6	RSCAN0CFSTS6	<RSCAN0_base> + 0190 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 7	RSCAN0CFSTS7	<RSCAN0_base> + 0194 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 8	RSCAN0CFSTS8	<RSCAN0_base> + 0198 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 9	RSCAN0CFSTS9	<RSCAN0_base> + 019C <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 10	RSCAN0CFSTS10	<RSCAN0_base> + 01A0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 11	RSCAN0CFSTS11	<RSCAN0_base> + 01A4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 12	RSCAN0CFSTS12	<RSCAN0_base> + 01A8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 13	RSCAN0CFSTS13	<RSCAN0_base> + 01AC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 14	RSCAN0CFSTS14	<RSCAN0_base> + 01B0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 15	RSCAN0CFSTS15	<RSCAN0_base> + 01B4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 16	RSCAN0CFSTS16	<RSCAN0_base> + 01B8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer status register 17	RSCAN0CFSTS17	<RSCAN0_base> + 01BC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 0	RSCAN0CFPCTR0	<RSCAN0_base> + 01D8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 1	RSCAN0CFPCTR1	<RSCAN0_base> + 01DC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 2	RSCAN0CFPCTR2	<RSCAN0_base> + 01E0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 3	RSCAN0CFPCTR3	<RSCAN0_base> + 01E4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 4	RSCAN0CFPCTR4	<RSCAN0_base> + 01E8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 5	RSCAN0CFPCTR5	<RSCAN0_base> + 01EC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 6	RSCAN0CFPCTR6	<RSCAN0_base> + 01F0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 7	RSCAN0CFPCTR7	<RSCAN0_base> + 01F4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 8	RSCAN0CFPCTR8	<RSCAN0_base> + 01F8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 9	RSCAN0CFPCTR9	<RSCAN0_base> + 01FC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 10	RSCAN0CFPCTR10	<RSCAN0_base> + 0200 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 11	RSCAN0CFPCTR11	<RSCAN0_base> + 0204 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 12	RSCAN0CFPCTR12	<RSCAN0_base> + 0208 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 13	RSCAN0CFPCTR13	<RSCAN0_base> + 020C <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 14	RSCAN0CFPCTR14	<RSCAN0_base> + 0210 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 15	RSCAN0CFPCTR15	<RSCAN0_base> + 0214 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 16	RSCAN0CFPCTR16	<RSCAN0_base> + 0218 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer pointer control register 17	RSCAN0CFPCTR17	<RSCAN0_base> + 021C <sub>H</sub>
RSCAN0	FIFO empty status register	RSCAN0FESTS	<RSCAN0_base> + 0238 <sub>H</sub>
RSCAN0	FIFO full status register	RSCAN0FFSTS	<RSCAN0_base> + 023C <sub>H</sub>
RSCAN0	FIFO Msg lost status register	RSCAN0FMSTS	<RSCAN0_base> + 0240 <sub>H</sub>
RSCAN0	Receive FIFO buffer interrupt flag status register	RSCAN0RFISTS	<RSCAN0_base> + 0244 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer RX interrupt flag status register	RSCAN0CFRISTS	<RSCAN0_base> + 0248 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer TX interrupt flag status register	RSCAN0CFTISTS	<RSCAN0_base> + 024C <sub>H</sub>
RSCAN0	Transmit buffer control register 0	RSCAN0TMC0	<RSCAN0_base> + 0250 <sub>H</sub>
RSCAN0	Transmit buffer control register 1	RSCAN0TMC1	<RSCAN0_base> + 0251 <sub>H</sub>
RSCAN0	Transmit buffer control register 2	RSCAN0TMC2	<RSCAN0_base> + 0252 <sub>H</sub>
RSCAN0	Transmit buffer control register 3	RSCAN0TMC3	<RSCAN0_base> + 0253 <sub>H</sub>
RSCAN0	Transmit buffer control register 4	RSCAN0TMC4	<RSCAN0_base> + 0254 <sub>H</sub>
RSCAN0	Transmit buffer control register 5	RSCAN0TMC5	<RSCAN0_base> + 0255 <sub>H</sub>

Table 19.12 List of Registers (4/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 6	RSCAN0TMC6	<RSCAN0_base> + 0256H
RSCAN0	Transmit buffer control register 7	RSCAN0TMC7	<RSCAN0_base> + 0257H
RSCAN0	Transmit buffer control register 8	RSCAN0TMC8	<RSCAN0_base> + 0258H
RSCAN0	Transmit buffer control register 9	RSCAN0TMC9	<RSCAN0_base> + 0259H
RSCAN0	Transmit buffer control register 10	RSCAN0TMC10	<RSCAN0_base> + 025AH
RSCAN0	Transmit buffer control register 11	RSCAN0TMC11	<RSCAN0_base> + 025BH
RSCAN0	Transmit buffer control register 12	RSCAN0TMC12	<RSCAN0_base> + 025CH
RSCAN0	Transmit buffer control register 13	RSCAN0TMC13	<RSCAN0_base> + 025DH
RSCAN0	Transmit buffer control register 14	RSCAN0TMC14	<RSCAN0_base> + 025EH
RSCAN0	Transmit buffer control register 15	RSCAN0TMC15	<RSCAN0_base> + 025FH
RSCAN0	Transmit buffer control register 16	RSCAN0TMC16	<RSCAN0_base> + 0260H
RSCAN0	Transmit buffer control register 17	RSCAN0TMC17	<RSCAN0_base> + 0261H
RSCAN0	Transmit buffer control register 18	RSCAN0TMC18	<RSCAN0_base> + 0262H
RSCAN0	Transmit buffer control register 19	RSCAN0TMC19	<RSCAN0_base> + 0263H
RSCAN0	Transmit buffer control register 20	RSCAN0TMC20	<RSCAN0_base> + 0264H
RSCAN0	Transmit buffer control register 21	RSCAN0TMC21	<RSCAN0_base> + 0265H
RSCAN0	Transmit buffer control register 22	RSCAN0TMC22	<RSCAN0_base> + 0266H
RSCAN0	Transmit buffer control register 23	RSCAN0TMC23	<RSCAN0_base> + 0267H
RSCAN0	Transmit buffer control register 24	RSCAN0TMC24	<RSCAN0_base> + 0268H
RSCAN0	Transmit buffer control register 25	RSCAN0TMC25	<RSCAN0_base> + 0269H
RSCAN0	Transmit buffer control register 26	RSCAN0TMC26	<RSCAN0_base> + 026AH
RSCAN0	Transmit buffer control register 27	RSCAN0TMC27	<RSCAN0_base> + 026BH
RSCAN0	Transmit buffer control register 28	RSCAN0TMC28	<RSCAN0_base> + 026CH
RSCAN0	Transmit buffer control register 29	RSCAN0TMC29	<RSCAN0_base> + 026DH
RSCAN0	Transmit buffer control register 30	RSCAN0TMC30	<RSCAN0_base> + 026EH
RSCAN0	Transmit buffer control register 31	RSCAN0TMC31	<RSCAN0_base> + 026FH
RSCAN0	Transmit buffer control register 32	RSCAN0TMC32	<RSCAN0_base> + 0270H
RSCAN0	Transmit buffer control register 33	RSCAN0TMC33	<RSCAN0_base> + 0271H
RSCAN0	Transmit buffer control register 34	RSCAN0TMC34	<RSCAN0_base> + 0272H
RSCAN0	Transmit buffer control register 35	RSCAN0TMC35	<RSCAN0_base> + 0273H
RSCAN0	Transmit buffer control register 36	RSCAN0TMC36	<RSCAN0_base> + 0274H
RSCAN0	Transmit buffer control register 37	RSCAN0TMC37	<RSCAN0_base> + 0275H
RSCAN0	Transmit buffer control register 38	RSCAN0TMC38	<RSCAN0_base> + 0276H
RSCAN0	Transmit buffer control register 39	RSCAN0TMC39	<RSCAN0_base> + 0277H
RSCAN0	Transmit buffer control register 40	RSCAN0TMC40	<RSCAN0_base> + 0278H
RSCAN0	Transmit buffer control register 41	RSCAN0TMC41	<RSCAN0_base> + 0279H
RSCAN0	Transmit buffer control register 42	RSCAN0TMC42	<RSCAN0_base> + 027AH
RSCAN0	Transmit buffer control register 43	RSCAN0TMC43	<RSCAN0_base> + 027BH
RSCAN0	Transmit buffer control register 44	RSCAN0TMC44	<RSCAN0_base> + 027CH
RSCAN0	Transmit buffer control register 45	RSCAN0TMC45	<RSCAN0_base> + 027DH
RSCAN0	Transmit buffer control register 46	RSCAN0TMC46	<RSCAN0_base> + 027EH
RSCAN0	Transmit buffer control register 47	RSCAN0TMC47	<RSCAN0_base> + 027FH
RSCAN0	Transmit buffer control register 48	RSCAN0TMC48	<RSCAN0_base> + 0280H
RSCAN0	Transmit buffer control register 49	RSCAN0TMC49	<RSCAN0_base> + 0281H

Table 19.12 List of Registers (5/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 50	RSCAN0TMC50	<RSCAN0_base> + 0282 <sub>H</sub>
RSCAN0	Transmit buffer control register 51	RSCAN0TMC51	<RSCAN0_base> + 0283 <sub>H</sub>
RSCAN0	Transmit buffer control register 52	RSCAN0TMC52	<RSCAN0_base> + 0284 <sub>H</sub>
RSCAN0	Transmit buffer control register 53	RSCAN0TMC53	<RSCAN0_base> + 0285 <sub>H</sub>
RSCAN0	Transmit buffer control register 54	RSCAN0TMC54	<RSCAN0_base> + 0286 <sub>H</sub>
RSCAN0	Transmit buffer control register 55	RSCAN0TMC55	<RSCAN0_base> + 0287 <sub>H</sub>
RSCAN0	Transmit buffer control register 56	RSCAN0TMC56	<RSCAN0_base> + 0288 <sub>H</sub>
RSCAN0	Transmit buffer control register 57	RSCAN0TMC57	<RSCAN0_base> + 0289 <sub>H</sub>
RSCAN0	Transmit buffer control register 58	RSCAN0TMC58	<RSCAN0_base> + 028A <sub>H</sub>
RSCAN0	Transmit buffer control register 59	RSCAN0TMC59	<RSCAN0_base> + 028B <sub>H</sub>
RSCAN0	Transmit buffer control register 60	RSCAN0TMC60	<RSCAN0_base> + 028C <sub>H</sub>
RSCAN0	Transmit buffer control register 61	RSCAN0TMC61	<RSCAN0_base> + 028D <sub>H</sub>
RSCAN0	Transmit buffer control register 62	RSCAN0TMC62	<RSCAN0_base> + 028E <sub>H</sub>
RSCAN0	Transmit buffer control register 63	RSCAN0TMC63	<RSCAN0_base> + 028F <sub>H</sub>
RSCAN0	Transmit buffer control register 64	RSCAN0TMC64	<RSCAN0_base> + 0290 <sub>H</sub>
RSCAN0	Transmit buffer control register 65	RSCAN0TMC65	<RSCAN0_base> + 0291 <sub>H</sub>
RSCAN0	Transmit buffer control register 66	RSCAN0TMC66	<RSCAN0_base> + 0292 <sub>H</sub>
RSCAN0	Transmit buffer control register 67	RSCAN0TMC67	<RSCAN0_base> + 0293 <sub>H</sub>
RSCAN0	Transmit buffer control register 68	RSCAN0TMC68	<RSCAN0_base> + 0294 <sub>H</sub>
RSCAN0	Transmit buffer control register 69	RSCAN0TMC69	<RSCAN0_base> + 0295 <sub>H</sub>
RSCAN0	Transmit buffer control register 70	RSCAN0TMC70	<RSCAN0_base> + 0296 <sub>H</sub>
RSCAN0	Transmit buffer control register 71	RSCAN0TMC71	<RSCAN0_base> + 0297 <sub>H</sub>
RSCAN0	Transmit buffer control register 72	RSCAN0TMC72	<RSCAN0_base> + 0298 <sub>H</sub>
RSCAN0	Transmit buffer control register 73	RSCAN0TMC73	<RSCAN0_base> + 0299 <sub>H</sub>
RSCAN0	Transmit buffer control register 74	RSCAN0TMC74	<RSCAN0_base> + 029A <sub>H</sub>
RSCAN0	Transmit buffer control register 75	RSCAN0TMC75	<RSCAN0_base> + 029B <sub>H</sub>
RSCAN0	Transmit buffer control register 76	RSCAN0TMC76	<RSCAN0_base> + 029C <sub>H</sub>
RSCAN0	Transmit buffer control register 77	RSCAN0TMC77	<RSCAN0_base> + 029D <sub>H</sub>
RSCAN0	Transmit buffer control register 78	RSCAN0TMC78	<RSCAN0_base> + 029E <sub>H</sub>
RSCAN0	Transmit buffer control register 79	RSCAN0TMC79	<RSCAN0_base> + 029F <sub>H</sub>
RSCAN0	Transmit buffer control register 80	RSCAN0TMC80	<RSCAN0_base> + 02A0 <sub>H</sub>
RSCAN0	Transmit buffer control register 81	RSCAN0TMC81	<RSCAN0_base> + 02A1 <sub>H</sub>
RSCAN0	Transmit buffer control register 82	RSCAN0TMC82	<RSCAN0_base> + 02A2 <sub>H</sub>
RSCAN0	Transmit buffer control register 83	RSCAN0TMC83	<RSCAN0_base> + 02A3 <sub>H</sub>
RSCAN0	Transmit buffer control register 84	RSCAN0TMC84	<RSCAN0_base> + 02A4 <sub>H</sub>
RSCAN0	Transmit buffer control register 85	RSCAN0TMC85	<RSCAN0_base> + 02A5 <sub>H</sub>
RSCAN0	Transmit buffer control register 86	RSCAN0TMC86	<RSCAN0_base> + 02A6 <sub>H</sub>
RSCAN0	Transmit buffer control register 87	RSCAN0TMC87	<RSCAN0_base> + 02A7 <sub>H</sub>
RSCAN0	Transmit buffer control register 88	RSCAN0TMC88	<RSCAN0_base> + 02A8 <sub>H</sub>
RSCAN0	Transmit buffer control register 89	RSCAN0TMC89	<RSCAN0_base> + 02A9 <sub>H</sub>
RSCAN0	Transmit buffer control register 90	RSCAN0TMC90	<RSCAN0_base> + 02AA <sub>H</sub>
RSCAN0	Transmit buffer control register 91	RSCAN0TMC91	<RSCAN0_base> + 02AB <sub>H</sub>
RSCAN0	Transmit buffer control register 92	RSCAN0TMC92	<RSCAN0_base> + 02AC <sub>H</sub>
RSCAN0	Transmit buffer control register 93	RSCAN0TMC93	<RSCAN0_base> + 02AD <sub>H</sub>

Table 19.12 List of Registers (6/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 94	RSCAN0TMC94	<RSCAN0_base> + 02AE <sub>H</sub>
RSCAN0	Transmit buffer control register 95	RSCAN0TMC95	<RSCAN0_base> + 02AF <sub>H</sub>
RSCAN0	Transmit buffer status register 0	RSCAN0TMSTS0	<RSCAN0_base> + 02D0 <sub>H</sub>
RSCAN0	Transmit buffer status register 1	RSCAN0TMSTS1	<RSCAN0_base> + 02D1 <sub>H</sub>
RSCAN0	Transmit buffer status register 2	RSCAN0TMSTS2	<RSCAN0_base> + 02D2 <sub>H</sub>
RSCAN0	Transmit buffer status register 3	RSCAN0TMSTS3	<RSCAN0_base> + 02D3 <sub>H</sub>
RSCAN0	Transmit buffer status register 4	RSCAN0TMSTS4	<RSCAN0_base> + 02D4 <sub>H</sub>
RSCAN0	Transmit buffer status register 5	RSCAN0TMSTS5	<RSCAN0_base> + 02D5 <sub>H</sub>
RSCAN0	Transmit buffer status register 6	RSCAN0TMSTS6	<RSCAN0_base> + 02D6 <sub>H</sub>
RSCAN0	Transmit buffer status register 7	RSCAN0TMSTS7	<RSCAN0_base> + 02D7 <sub>H</sub>
RSCAN0	Transmit buffer status register 8	RSCAN0TMSTS8	<RSCAN0_base> + 02D8 <sub>H</sub>
RSCAN0	Transmit buffer status register 9	RSCAN0TMSTS9	<RSCAN0_base> + 02D9 <sub>H</sub>
RSCAN0	Transmit buffer status register 10	RSCAN0TMSTS10	<RSCAN0_base> + 02DA <sub>H</sub>
RSCAN0	Transmit buffer status register 11	RSCAN0TMSTS11	<RSCAN0_base> + 02DB <sub>H</sub>
RSCAN0	Transmit buffer status register 12	RSCAN0TMSTS12	<RSCAN0_base> + 02DC <sub>H</sub>
RSCAN0	Transmit buffer status register 13	RSCAN0TMSTS13	<RSCAN0_base> + 02DD <sub>H</sub>
RSCAN0	Transmit buffer status register 14	RSCAN0TMSTS14	<RSCAN0_base> + 02DE <sub>H</sub>
RSCAN0	Transmit buffer status register 15	RSCAN0TMSTS15	<RSCAN0_base> + 02DF <sub>H</sub>
RSCAN0	Transmit buffer status register 16	RSCAN0TMSTS16	<RSCAN0_base> + 02E0 <sub>H</sub>
RSCAN0	Transmit buffer status register 17	RSCAN0TMSTS17	<RSCAN0_base> + 02E1 <sub>H</sub>
RSCAN0	Transmit buffer status register 18	RSCAN0TMSTS18	<RSCAN0_base> + 02E2 <sub>H</sub>
RSCAN0	Transmit buffer status register 19	RSCAN0TMSTS19	<RSCAN0_base> + 02E3 <sub>H</sub>
RSCAN0	Transmit buffer status register 20	RSCAN0TMSTS20	<RSCAN0_base> + 02E4 <sub>H</sub>
RSCAN0	Transmit buffer status register 21	RSCAN0TMSTS21	<RSCAN0_base> + 02E5 <sub>H</sub>
RSCAN0	Transmit buffer status register 22	RSCAN0TMSTS22	<RSCAN0_base> + 02E6 <sub>H</sub>
RSCAN0	Transmit buffer status register 23	RSCAN0TMSTS23	<RSCAN0_base> + 02E7 <sub>H</sub>
RSCAN0	Transmit buffer status register 24	RSCAN0TMSTS24	<RSCAN0_base> + 02E8 <sub>H</sub>
RSCAN0	Transmit buffer status register 25	RSCAN0TMSTS25	<RSCAN0_base> + 02E9 <sub>H</sub>
RSCAN0	Transmit buffer status register 26	RSCAN0TMSTS26	<RSCAN0_base> + 02EA <sub>H</sub>
RSCAN0	Transmit buffer status register 27	RSCAN0TMSTS27	<RSCAN0_base> + 02EB <sub>H</sub>
RSCAN0	Transmit buffer status register 28	RSCAN0TMSTS28	<RSCAN0_base> + 02EC <sub>H</sub>
RSCAN0	Transmit buffer status register 29	RSCAN0TMSTS29	<RSCAN0_base> + 02ED <sub>H</sub>
RSCAN0	Transmit buffer status register 30	RSCAN0TMSTS30	<RSCAN0_base> + 02EE <sub>H</sub>
RSCAN0	Transmit buffer status register 31	RSCAN0TMSTS31	<RSCAN0_base> + 02EF <sub>H</sub>
RSCAN0	Transmit buffer status register 32	RSCAN0TMSTS32	<RSCAN0_base> + 02F0 <sub>H</sub>
RSCAN0	Transmit buffer status register 33	RSCAN0TMSTS33	<RSCAN0_base> + 02F1 <sub>H</sub>
RSCAN0	Transmit buffer status register 34	RSCAN0TMSTS34	<RSCAN0_base> + 02F2 <sub>H</sub>
RSCAN0	Transmit buffer status register 35	RSCAN0TMSTS35	<RSCAN0_base> + 02F3 <sub>H</sub>
RSCAN0	Transmit buffer status register 36	RSCAN0TMSTS36	<RSCAN0_base> + 02F4 <sub>H</sub>
RSCAN0	Transmit buffer status register 37	RSCAN0TMSTS37	<RSCAN0_base> + 02F5 <sub>H</sub>
RSCAN0	Transmit buffer status register 38	RSCAN0TMSTS38	<RSCAN0_base> + 02F6 <sub>H</sub>
RSCAN0	Transmit buffer status register 39	RSCAN0TMSTS39	<RSCAN0_base> + 02F7 <sub>H</sub>
RSCAN0	Transmit buffer status register 40	RSCAN0TMSTS40	<RSCAN0_base> + 02F8 <sub>H</sub>
RSCAN0	Transmit buffer status register 41	RSCAN0TMSTS41	<RSCAN0_base> + 02F9 <sub>H</sub>

Table 19.12 List of Registers (7/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer status register 42	RSCAN0TMSTS42	<RSCAN0_base> + 02FA <sub>H</sub>
RSCAN0	Transmit buffer status register 43	RSCAN0TMSTS43	<RSCAN0_base> + 02FB <sub>H</sub>
RSCAN0	Transmit buffer status register 44	RSCAN0TMSTS44	<RSCAN0_base> + 02FC <sub>H</sub>
RSCAN0	Transmit buffer status register 45	RSCAN0TMSTS45	<RSCAN0_base> + 02FD <sub>H</sub>
RSCAN0	Transmit buffer status register 46	RSCAN0TMSTS46	<RSCAN0_base> + 02FE <sub>H</sub>
RSCAN0	Transmit buffer status register 47	RSCAN0TMSTS47	<RSCAN0_base> + 02FF <sub>H</sub>
RSCAN0	Transmit buffer status register 48	RSCAN0TMSTS48	<RSCAN0_base> + 0300 <sub>H</sub>
RSCAN0	Transmit buffer status register 49	RSCAN0TMSTS49	<RSCAN0_base> + 0301 <sub>H</sub>
RSCAN0	Transmit buffer status register 50	RSCAN0TMSTS50	<RSCAN0_base> + 0302 <sub>H</sub>
RSCAN0	Transmit buffer status register 51	RSCAN0TMSTS51	<RSCAN0_base> + 0303 <sub>H</sub>
RSCAN0	Transmit buffer status register 52	RSCAN0TMSTS52	<RSCAN0_base> + 0304 <sub>H</sub>
RSCAN0	Transmit buffer status register 53	RSCAN0TMSTS53	<RSCAN0_base> + 0305 <sub>H</sub>
RSCAN0	Transmit buffer status register 54	RSCAN0TMSTS54	<RSCAN0_base> + 0306 <sub>H</sub>
RSCAN0	Transmit buffer status register 55	RSCAN0TMSTS55	<RSCAN0_base> + 0307 <sub>H</sub>
RSCAN0	Transmit buffer status register 56	RSCAN0TMSTS56	<RSCAN0_base> + 0308 <sub>H</sub>
RSCAN0	Transmit buffer status register 57	RSCAN0TMSTS57	<RSCAN0_base> + 0309 <sub>H</sub>
RSCAN0	Transmit buffer status register 58	RSCAN0TMSTS58	<RSCAN0_base> + 030A <sub>H</sub>
RSCAN0	Transmit buffer status register 59	RSCAN0TMSTS59	<RSCAN0_base> + 030B <sub>H</sub>
RSCAN0	Transmit buffer status register 60	RSCAN0TMSTS60	<RSCAN0_base> + 030C <sub>H</sub>
RSCAN0	Transmit buffer status register 61	RSCAN0TMSTS61	<RSCAN0_base> + 030D <sub>H</sub>
RSCAN0	Transmit buffer status register 62	RSCAN0TMSTS62	<RSCAN0_base> + 030E <sub>H</sub>
RSCAN0	Transmit buffer status register 63	RSCAN0TMSTS63	<RSCAN0_base> + 030F <sub>H</sub>
RSCAN0	Transmit buffer status register 64	RSCAN0TMSTS64	<RSCAN0_base> + 0310 <sub>H</sub>
RSCAN0	Transmit buffer status register 65	RSCAN0TMSTS65	<RSCAN0_base> + 0311 <sub>H</sub>
RSCAN0	Transmit buffer status register 66	RSCAN0TMSTS66	<RSCAN0_base> + 0312 <sub>H</sub>
RSCAN0	Transmit buffer status register 67	RSCAN0TMSTS67	<RSCAN0_base> + 0313 <sub>H</sub>
RSCAN0	Transmit buffer status register 68	RSCAN0TMSTS68	<RSCAN0_base> + 0314 <sub>H</sub>
RSCAN0	Transmit buffer status register 69	RSCAN0TMSTS69	<RSCAN0_base> + 0315 <sub>H</sub>
RSCAN0	Transmit buffer status register 70	RSCAN0TMSTS70	<RSCAN0_base> + 0316 <sub>H</sub>
RSCAN0	Transmit buffer status register 71	RSCAN0TMSTS71	<RSCAN0_base> + 0317 <sub>H</sub>
RSCAN0	Transmit buffer status register 72	RSCAN0TMSTS72	<RSCAN0_base> + 0318 <sub>H</sub>
RSCAN0	Transmit buffer status register 73	RSCAN0TMSTS73	<RSCAN0_base> + 0319 <sub>H</sub>
RSCAN0	Transmit buffer status register 74	RSCAN0TMSTS74	<RSCAN0_base> + 031A <sub>H</sub>
RSCAN0	Transmit buffer status register 75	RSCAN0TMSTS75	<RSCAN0_base> + 031B <sub>H</sub>
RSCAN0	Transmit buffer status register 76	RSCAN0TMSTS76	<RSCAN0_base> + 031C <sub>H</sub>
RSCAN0	Transmit buffer status register 77	RSCAN0TMSTS77	<RSCAN0_base> + 031D <sub>H</sub>
RSCAN0	Transmit buffer status register 78	RSCAN0TMSTS78	<RSCAN0_base> + 031E <sub>H</sub>
RSCAN0	Transmit buffer status register 79	RSCAN0TMSTS79	<RSCAN0_base> + 031F <sub>H</sub>
RSCAN0	Transmit buffer status register 80	RSCAN0TMSTS80	<RSCAN0_base> + 0320 <sub>H</sub>
RSCAN0	Transmit buffer status register 81	RSCAN0TMSTS81	<RSCAN0_base> + 0321 <sub>H</sub>
RSCAN0	Transmit buffer status register 82	RSCAN0TMSTS82	<RSCAN0_base> + 0322 <sub>H</sub>
RSCAN0	Transmit buffer status register 83	RSCAN0TMSTS83	<RSCAN0_base> + 0323 <sub>H</sub>
RSCAN0	Transmit buffer status register 84	RSCAN0TMSTS84	<RSCAN0_base> + 0324 <sub>H</sub>
RSCAN0	Transmit buffer status register 85	RSCAN0TMSTS85	<RSCAN0_base> + 0325 <sub>H</sub>

Table 19.12 List of Registers (8/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer status register 86	RSCAN0TMSTS86	<RSCAN0_base> + 0326 <sub>H</sub>
RSCAN0	Transmit buffer status register 87	RSCAN0TMSTS87	<RSCAN0_base> + 0327 <sub>H</sub>
RSCAN0	Transmit buffer status register 88	RSCAN0TMSTS88	<RSCAN0_base> + 0328 <sub>H</sub>
RSCAN0	Transmit buffer status register 89	RSCAN0TMSTS89	<RSCAN0_base> + 0329 <sub>H</sub>
RSCAN0	Transmit buffer status register 90	RSCAN0TMSTS90	<RSCAN0_base> + 032A <sub>H</sub>
RSCAN0	Transmit buffer status register 91	RSCAN0TMSTS91	<RSCAN0_base> + 032B <sub>H</sub>
RSCAN0	Transmit buffer status register 92	RSCAN0TMSTS92	<RSCAN0_base> + 032C <sub>H</sub>
RSCAN0	Transmit buffer status register 93	RSCAN0TMSTS93	<RSCAN0_base> + 032D <sub>H</sub>
RSCAN0	Transmit buffer status register 94	RSCAN0TMSTS94	<RSCAN0_base> + 032E <sub>H</sub>
RSCAN0	Transmit buffer status register 95	RSCAN0TMSTS95	<RSCAN0_base> + 032F <sub>H</sub>
RSCAN0	Transmit buffer transmit request status register 0	RSCAN0TMTRSTS0	<RSCAN0_base> + 0350 <sub>H</sub>
RSCAN0	Transmit buffer transmit request status register 1	RSCAN0TMTRSTS1	<RSCAN0_base> + 0354 <sub>H</sub>
RSCAN0	Transmit buffer transmit request status register 2	RSCAN0TMTRSTS2	<RSCAN0_base> + 0358 <sub>H</sub>
RSCAN0	Transmit buffer transmit abort request status register 0	RSCAN0TMTARSTS0	<RSCAN0_base> + 0360 <sub>H</sub>
RSCAN0	Transmit buffer transmit abort request status register 1	RSCAN0TMTARSTS1	<RSCAN0_base> + 0364 <sub>H</sub>
RSCAN0	Transmit buffer transmit abort request status register 2	RSCAN0TMTARSTS2	<RSCAN0_base> + 0368 <sub>H</sub>
RSCAN0	Transmit buffer transmit complete status register 0	RSCAN0TMCSTS0	<RSCAN0_base> + 0370 <sub>H</sub>
RSCAN0	Transmit buffer transmit complete status register 1	RSCAN0TMCSTS1	<RSCAN0_base> + 0374 <sub>H</sub>
RSCAN0	Transmit buffer transmit complete status register 2	RSCAN0TMCSTS2	<RSCAN0_base> + 0378 <sub>H</sub>
RSCAN0	Transmit buffer transmit abort status register 0	RSCAN0TMTASTS0	<RSCAN0_base> + 0380 <sub>H</sub>
RSCAN0	Transmit buffer transmit abort status register 1	RSCAN0TMTASTS1	<RSCAN0_base> + 0384 <sub>H</sub>
RSCAN0	Transmit buffer transmit abort status register 2	RSCAN0TMTASTS2	<RSCAN0_base> + 0388 <sub>H</sub>
RSCAN0	Transmit buffer interrupt enable configuration register 0	RSCAN0TMIEC0	<RSCAN0_base> + 0390 <sub>H</sub>
RSCAN0	Transmit buffer interrupt enable configuration register 1	RSCAN0TMIEC1	<RSCAN0_base> + 0394 <sub>H</sub>
RSCAN0	Transmit buffer interrupt enable configuration register 2	RSCAN0TMIEC2	<RSCAN0_base> + 0398 <sub>H</sub>
RSCAN0	Transmit queue configuration and control register 0	RSCAN0TXQCC0	<RSCAN0_base> + 03A0 <sub>H</sub>
RSCAN0	Transmit queue configuration and control register 1	RSCAN0TXQCC1	<RSCAN0_base> + 03A4 <sub>H</sub>
RSCAN0	Transmit queue configuration and control register 2	RSCAN0TXQCC2	<RSCAN0_base> + 03A8 <sub>H</sub>
RSCAN0	Transmit queue configuration and control register 3	RSCAN0TXQCC3	<RSCAN0_base> + 03AC <sub>H</sub>
RSCAN0	Transmit queue configuration and control register 4	RSCAN0TXQCC4	<RSCAN0_base> + 03B0 <sub>H</sub>
RSCAN0	Transmit queue configuration and control register 5	RSCAN0TXQCC5	<RSCAN0_base> + 03B4 <sub>H</sub>
RSCAN0	Transmit queue status register 0	RSCAN0TXQSTS0	<RSCAN0_base> + 03C0 <sub>H</sub>
RSCAN0	Transmit queue status register 1	RSCAN0TXQSTS1	<RSCAN0_base> + 03C4 <sub>H</sub>
RSCAN0	Transmit queue status register 2	RSCAN0TXQSTS2	<RSCAN0_base> + 03C8 <sub>H</sub>
RSCAN0	Transmit queue status register 3	RSCAN0TXQSTS3	<RSCAN0_base> + 03CCH
RSCAN0	Transmit queue status register 4	RSCAN0TXQSTS4	<RSCAN0_base> + 03D0 <sub>H</sub>
RSCAN0	Transmit queue status register 5	RSCAN0TXQSTS5	<RSCAN0_base> + 03D4 <sub>H</sub>
RSCAN0	Transmit queue pointer control register 0	RSCAN0TXQPCTR0	<RSCAN0_base> + 03E0 <sub>H</sub>
RSCAN0	Transmit queue pointer control register 1	RSCAN0TXQPCTR1	<RSCAN0_base> + 03E4 <sub>H</sub>
RSCAN0	Transmit queue pointer control register 2	RSCAN0TXQPCTR2	<RSCAN0_base> + 03E8 <sub>H</sub>
RSCAN0	Transmit queue pointer control register 3	RSCAN0TXQPCTR3	<RSCAN0_base> + 03EC <sub>H</sub>
RSCAN0	Transmit queue pointer control register 4	RSCAN0TXQPCTR4	<RSCAN0_base> + 03F0 <sub>H</sub>
RSCAN0	Transmit queue pointer control register 5	RSCAN0TXQPCTR5	<RSCAN0_base> + 03F4 <sub>H</sub>
RSCAN0	Transmit history configuration and control register 0	RSCAN0THLCC0	<RSCAN0_base> + 0400 <sub>H</sub>

Table 19.12 List of Registers (9/32)

Module	Register	Symbol	Address
RSCAN0	Transmit history configuration and control register 1	RSCAN0THLCC1	<RSCAN0_base> + 0404H
RSCAN0	Transmit history configuration and control register 2	RSCAN0THLCC2	<RSCAN0_base> + 0408H
RSCAN0	Transmit history configuration and control register 3	RSCAN0THLCC3	<RSCAN0_base> + 040CH
RSCAN0	Transmit history configuration and control register 4	RSCAN0THLCC4	<RSCAN0_base> + 0410H
RSCAN0	Transmit history configuration and control register 5	RSCAN0THLCC5	<RSCAN0_base> + 0414H
RSCAN0	Transmit history status register 0	RSCAN0THLSTS0	<RSCAN0_base> + 0420H
RSCAN0	Transmit history status register 1	RSCAN0THLSTS1	<RSCAN0_base> + 0424H
RSCAN0	Transmit history status register 2	RSCAN0THLSTS2	<RSCAN0_base> + 0428H
RSCAN0	Transmit history status register 3	RSCAN0THLSTS3	<RSCAN0_base> + 042CH
RSCAN0	Transmit history status register 4	RSCAN0THLSTS4	<RSCAN0_base> + 0430H
RSCAN0	Transmit history status register 5	RSCAN0THLSTS5	<RSCAN0_base> + 0434H
RSCAN0	Transmit history pointer control register 0	RSCAN0THLPCTR0	<RSCAN0_base> + 0440H
RSCAN0	Transmit history pointer control register 1	RSCAN0THLPCTR1	<RSCAN0_base> + 0444H
RSCAN0	Transmit history pointer control register 2	RSCAN0THLPCTR2	<RSCAN0_base> + 0448H
RSCAN0	Transmit history pointer control register 3	RSCAN0THLPCTR3	<RSCAN0_base> + 044CH
RSCAN0	Transmit history pointer control register 4	RSCAN0THLPCTR4	<RSCAN0_base> + 0450H
RSCAN0	Transmit history pointer control register 5	RSCAN0THLPCTR5	<RSCAN0_base> + 0454H
RSCAN0	Global TX interrupt status register 0	RSCAN0GTINTSTS0	<RSCAN0_base> + 0460H
RSCAN0	Global TX interrupt status register 1	RSCAN0GTINTSTS1	<RSCAN0_base> + 0464H
RSCAN0	Global test configuration register	RSCAN0GTSTCFG	<RSCAN0_base> + 0468H
RSCAN0	Global test control register	RSCAN0GTSTCTR	<RSCAN0_base> + 046CH
RSCAN0	Global lock key register	RSCAN0GLOCKK	<RSCAN0_base> + 047CH
RSCAN0	Receive rule ID register 0	RSCAN0GAFLID0	<RSCAN0_base> + 0500H
RSCAN0	Receive rule mask register 0	RSCAN0GAFLM0	<RSCAN0_base> + 0504H
RSCAN0	Receive rule pointer 0 register 0	RSCAN0GAFLP00	<RSCAN0_base> + 0508H
RSCAN0	Receive rule pointer 1 register 0	RSCAN0GAFLP10	<RSCAN0_base> + 050CH
RSCAN0	Receive rule ID register 1	RSCAN0GAFLID1	<RSCAN0_base> + 0510H
RSCAN0	Receive rule mask register 1	RSCAN0GAFLM1	<RSCAN0_base> + 0514H
RSCAN0	Receive rule pointer 0 register 1	RSCAN0GAFLP01	<RSCAN0_base> + 0518H
RSCAN0	Receive rule pointer 1 register 1	RSCAN0GAFLP11	<RSCAN0_base> + 051CH
RSCAN0	Receive rule ID register 2	RSCAN0GAFLID2	<RSCAN0_base> + 0520H
RSCAN0	Receive rule mask register 2	RSCAN0GAFLM2	<RSCAN0_base> + 0524H
RSCAN0	Receive rule pointer 0 register 2	RSCAN0GAFLP02	<RSCAN0_base> + 0528H
RSCAN0	Receive rule pointer 1 register 2	RSCAN0GAFLP12	<RSCAN0_base> + 052CH
RSCAN0	Receive rule ID register 3	RSCAN0GAFLID3	<RSCAN0_base> + 0530H
RSCAN0	Receive rule mask register 3	RSCAN0GAFLM3	<RSCAN0_base> + 0534H
RSCAN0	Receive rule pointer 0 register 3	RSCAN0GAFLP03	<RSCAN0_base> + 0538H
RSCAN0	Receive rule pointer 1 register 3	RSCAN0GAFLP13	<RSCAN0_base> + 053CH
RSCAN0	Receive rule ID register 4	RSCAN0GAFLID4	<RSCAN0_base> + 0540H
RSCAN0	Receive rule mask register 4	RSCAN0GAFLM4	<RSCAN0_base> + 0544H
RSCAN0	Receive rule pointer 0 register 4	RSCAN0GAFLP04	<RSCAN0_base> + 0548H
RSCAN0	Receive rule pointer 1 register 4	RSCAN0GAFLP14	<RSCAN0_base> + 054CH
RSCAN0	Receive rule ID register 5	RSCAN0GAFLID5	<RSCAN0_base> + 0550H
RSCAN0	Receive rule mask register 5	RSCAN0GAFLM5	<RSCAN0_base> + 0554H

Table 19.12 List of Registers (10/32)

Module	Register	Symbol	Address
RSCAN0	Receive rule pointer 0 register 5	RSCAN0GAFLP05	<RSCAN0_base> + 0558 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 5	RSCAN0GAFLP15	<RSCAN0_base> + 055C <sub>H</sub>
RSCAN0	Receive rule ID register 6	RSCAN0GAFLID6	<RSCAN0_base> + 0560 <sub>H</sub>
RSCAN0	Receive rule mask register 6	RSCAN0GAFLM6	<RSCAN0_base> + 0564 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 6	RSCAN0GAFLP06	<RSCAN0_base> + 0568 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 6	RSCAN0GAFLP16	<RSCAN0_base> + 056C <sub>H</sub>
RSCAN0	Receive rule ID register 7	RSCAN0GAFLID7	<RSCAN0_base> + 0570 <sub>H</sub>
RSCAN0	Receive rule mask register 7	RSCAN0GAFLM7	<RSCAN0_base> + 0574 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 7	RSCAN0GAFLP07	<RSCAN0_base> + 0578 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 7	RSCAN0GAFLP17	<RSCAN0_base> + 057C <sub>H</sub>
RSCAN0	Receive rule ID register 8	RSCAN0GAFLID8	<RSCAN0_base> + 0580 <sub>H</sub>
RSCAN0	Receive rule mask register 8	RSCAN0GAFLM8	<RSCAN0_base> + 0584 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 8	RSCAN0GAFLP08	<RSCAN0_base> + 0588 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 8	RSCAN0GAFLP18	<RSCAN0_base> + 058C <sub>H</sub>
RSCAN0	Receive rule ID register 9	RSCAN0GAFLID9	<RSCAN0_base> + 0590 <sub>H</sub>
RSCAN0	Receive rule mask register 9	RSCAN0GAFLM9	<RSCAN0_base> + 0594 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 9	RSCAN0GAFLP09	<RSCAN0_base> + 0598 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 9	RSCAN0GAFLP19	<RSCAN0_base> + 059C <sub>H</sub>
RSCAN0	Receive rule ID register 10	RSCAN0GAFLID10	<RSCAN0_base> + 05A0 <sub>H</sub>
RSCAN0	Receive rule mask register 10	RSCAN0GAFLM10	<RSCAN0_base> + 05A4 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 10	RSCAN0GAFLP010	<RSCAN0_base> + 05A8 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 10	RSCAN0GAFLP110	<RSCAN0_base> + 05AC <sub>H</sub>
RSCAN0	Receive rule ID register 11	RSCAN0GAFLID11	<RSCAN0_base> + 05B0 <sub>H</sub>
RSCAN0	Receive rule mask register 11	RSCAN0GAFLM11	<RSCAN0_base> + 05B4 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 11	RSCAN0GAFLP011	<RSCAN0_base> + 05B8 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 11	RSCAN0GAFLP111	<RSCAN0_base> + 05BC <sub>H</sub>
RSCAN0	Receive rule ID register 12	RSCAN0GAFLID12	<RSCAN0_base> + 05C0 <sub>H</sub>
RSCAN0	Receive rule mask register 12	RSCAN0GAFLM12	<RSCAN0_base> + 05C4 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 12	RSCAN0GAFLP012	<RSCAN0_base> + 05C8 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 12	RSCAN0GAFLP112	<RSCAN0_base> + 05CC <sub>H</sub>
RSCAN0	Receive rule ID register 13	RSCAN0GAFLID13	<RSCAN0_base> + 05D0 <sub>H</sub>
RSCAN0	Receive rule mask register 13	RSCAN0GAFLM13	<RSCAN0_base> + 05D4 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 13	RSCAN0GAFLP013	<RSCAN0_base> + 05D8 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 13	RSCAN0GAFLP113	<RSCAN0_base> + 05DC <sub>H</sub>
RSCAN0	Receive rule ID register 14	RSCAN0GAFLID14	<RSCAN0_base> + 05E0 <sub>H</sub>
RSCAN0	Receive rule mask register 14	RSCAN0GAFLM14	<RSCAN0_base> + 05E4 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 14	RSCAN0GAFLP014	<RSCAN0_base> + 05E8 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 14	RSCAN0GAFLP114	<RSCAN0_base> + 05EC <sub>H</sub>
RSCAN0	Receive rule ID register 15	RSCAN0GAFLID15	<RSCAN0_base> + 05F0 <sub>H</sub>
RSCAN0	Receive rule mask register 15	RSCAN0GAFLM15	<RSCAN0_base> + 05F4 <sub>H</sub>
RSCAN0	Receive rule pointer 0 register 15	RSCAN0GAFLP015	<RSCAN0_base> + 05F8 <sub>H</sub>
RSCAN0	Receive rule pointer 1 register 15	RSCAN0GAFLP115	<RSCAN0_base> + 05FC <sub>H</sub>
RSCAN0	Receive buffer ID register 0	RSCAN0RMID0	<RSCAN0_base> + 0600 <sub>H</sub>
RSCAN0	Receive buffer pointer register 0	RSCAN0RMPTR0	<RSCAN0_base> + 0604 <sub>H</sub>

Table 19.12 List of Registers (11/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 0	RSCAN0RMDF00	<RSCAN0_base> + 0608H
RSCAN0	Receive buffer data field 1 register 0	RSCAN0RMDF10	<RSCAN0_base> + 060CH
RSCAN0	Receive buffer ID register 1	RSCAN0RMID1	<RSCAN0_base> + 0610H
RSCAN0	Receive buffer pointer register 1	RSCAN0RMPTR1	<RSCAN0_base> + 0614H
RSCAN0	Receive buffer data field 0 register 1	RSCAN0RMDF01	<RSCAN0_base> + 0618H
RSCAN0	Receive buffer data field 1 register 1	RSCAN0RMDF11	<RSCAN0_base> + 061CH
RSCAN0	Receive buffer ID register 2	RSCAN0RMID2	<RSCAN0_base> + 0620H
RSCAN0	Receive buffer pointer register 2	RSCAN0RMPTR2	<RSCAN0_base> + 0624H
RSCAN0	Receive buffer data field 0 register 2	RSCAN0RMDF02	<RSCAN0_base> + 0628H
RSCAN0	Receive buffer data field 1 register 2	RSCAN0RMDF12	<RSCAN0_base> + 062CH
RSCAN0	Receive buffer ID register 3	RSCAN0RMID3	<RSCAN0_base> + 0630H
RSCAN0	Receive buffer pointer register 3	RSCAN0RMPTR3	<RSCAN0_base> + 0634H
RSCAN0	Receive buffer data field 0 register 3	RSCAN0RMDF03	<RSCAN0_base> + 0638H
RSCAN0	Receive buffer data field 1 register 3	RSCAN0RMDF13	<RSCAN0_base> + 063CH
RSCAN0	Receive buffer ID register 4	RSCAN0RMID4	<RSCAN0_base> + 0640H
RSCAN0	Receive buffer pointer register 4	RSCAN0RMPTR4	<RSCAN0_base> + 0644H
RSCAN0	Receive buffer data field 0 register 4	RSCAN0RMDF04	<RSCAN0_base> + 0648H
RSCAN0	Receive buffer data field 1 register 4	RSCAN0RMDF14	<RSCAN0_base> + 064CH
RSCAN0	Receive buffer ID register 5	RSCAN0RMID5	<RSCAN0_base> + 0650H
RSCAN0	Receive buffer pointer register 5	RSCAN0RMPTR5	<RSCAN0_base> + 0654H
RSCAN0	Receive buffer data field 0 register 5	RSCAN0RMDF05	<RSCAN0_base> + 0658H
RSCAN0	Receive buffer data field 1 register 5	RSCAN0RMDF15	<RSCAN0_base> + 065CH
RSCAN0	Receive buffer ID register 6	RSCAN0RMID6	<RSCAN0_base> + 0660H
RSCAN0	Receive buffer pointer register 6	RSCAN0RMPTR6	<RSCAN0_base> + 0664H
RSCAN0	Receive buffer data field 0 register 6	RSCAN0RMDF06	<RSCAN0_base> + 0668H
RSCAN0	Receive buffer data field 1 register 6	RSCAN0RMDF16	<RSCAN0_base> + 066CH
RSCAN0	Receive buffer ID register 7	RSCAN0RMID7	<RSCAN0_base> + 0670H
RSCAN0	Receive buffer pointer register 7	RSCAN0RMPTR7	<RSCAN0_base> + 0674H
RSCAN0	Receive buffer data field 0 register 7	RSCAN0RMDF07	<RSCAN0_base> + 0678H
RSCAN0	Receive buffer data field 1 register 7	RSCAN0RMDF17	<RSCAN0_base> + 067CH
RSCAN0	Receive buffer ID register 8	RSCAN0RMID8	<RSCAN0_base> + 0680H
RSCAN0	Receive buffer pointer register 8	RSCAN0RMPTR8	<RSCAN0_base> + 0684H
RSCAN0	Receive buffer data field 0 register 8	RSCAN0RMDF08	<RSCAN0_base> + 0688H
RSCAN0	Receive buffer data field 1 register 8	RSCAN0RMDF18	<RSCAN0_base> + 068CH
RSCAN0	Receive buffer ID register 9	RSCAN0RMID9	<RSCAN0_base> + 0690H
RSCAN0	Receive buffer pointer register 9	RSCAN0RMPTR9	<RSCAN0_base> + 0694H
RSCAN0	Receive buffer data field 0 register 9	RSCAN0RMDF09	<RSCAN0_base> + 0698H
RSCAN0	Receive buffer data field 1 register 9	RSCAN0RMDF19	<RSCAN0_base> + 069CH
RSCAN0	Receive buffer ID register 10	RSCAN0RMID10	<RSCAN0_base> + 06A0H
RSCAN0	Receive buffer pointer register 10	RSCAN0RMPTR10	<RSCAN0_base> + 06A4H
RSCAN0	Receive buffer data field 0 register 10	RSCAN0RMDF010	<RSCAN0_base> + 06A8H
RSCAN0	Receive buffer data field 1 register 10	RSCAN0RMDF110	<RSCAN0_base> + 06ACH
RSCAN0	Receive buffer ID register 11	RSCAN0RMID11	<RSCAN0_base> + 06B0H
RSCAN0	Receive buffer pointer register 11	RSCAN0RMPTR11	<RSCAN0_base> + 06B4H

Table 19.12 List of Registers (12/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 11	RSCAN0RMDF011	<RSCAN0_base> + 06B8H
RSCAN0	Receive buffer data field 1 register 11	RSCAN0RMDF111	<RSCAN0_base> + 06BCH
RSCAN0	Receive buffer ID register 12	RSCAN0RMID12	<RSCAN0_base> + 06C0H
RSCAN0	Receive buffer pointer register 12	RSCAN0RMPTR12	<RSCAN0_base> + 06C4H
RSCAN0	Receive buffer data field 0 register 12	RSCAN0RMDF012	<RSCAN0_base> + 06C8H
RSCAN0	Receive buffer data field 1 register 12	RSCAN0RMDF112	<RSCAN0_base> + 06CCH
RSCAN0	Receive buffer ID register 13	RSCAN0RMID13	<RSCAN0_base> + 06D0H
RSCAN0	Receive buffer pointer register 13	RSCAN0RMPTR13	<RSCAN0_base> + 06D4H
RSCAN0	Receive buffer data field 0 register 13	RSCAN0RMDF013	<RSCAN0_base> + 06D8H
RSCAN0	Receive buffer data field 1 register 13	RSCAN0RMDF113	<RSCAN0_base> + 06DCH
RSCAN0	Receive buffer ID register 14	RSCAN0RMID14	<RSCAN0_base> + 06E0H
RSCAN0	Receive buffer pointer register 14	RSCAN0RMPTR14	<RSCAN0_base> + 06E4H
RSCAN0	Receive buffer data field 0 register 14	RSCAN0RMDF014	<RSCAN0_base> + 06E8H
RSCAN0	Receive buffer data field 1 register 14	RSCAN0RMDF114	<RSCAN0_base> + 06ECH
RSCAN0	Receive buffer ID register 15	RSCAN0RMID15	<RSCAN0_base> + 06F0H
RSCAN0	Receive buffer pointer register 15	RSCAN0RMPTR15	<RSCAN0_base> + 06F4H
RSCAN0	Receive buffer data field 0 register 15	RSCAN0RMDF015	<RSCAN0_base> + 06F8H
RSCAN0	Receive buffer data field 1 register 15	RSCAN0RMDF115	<RSCAN0_base> + 06FCH
RSCAN0	Receive buffer ID register 16	RSCAN0RMID16	<RSCAN0_base> + 0700H
RSCAN0	Receive buffer pointer register 16	RSCAN0RMPTR16	<RSCAN0_base> + 0704H
RSCAN0	Receive buffer data field 0 register 16	RSCAN0RMDF016	<RSCAN0_base> + 0708H
RSCAN0	Receive buffer data field 1 register 16	RSCAN0RMDF116	<RSCAN0_base> + 070CH
RSCAN0	Receive buffer ID register 17	RSCAN0RMID17	<RSCAN0_base> + 0710H
RSCAN0	Receive buffer pointer register 17	RSCAN0RMPTR17	<RSCAN0_base> + 0714H
RSCAN0	Receive buffer data field 0 register 17	RSCAN0RMDF017	<RSCAN0_base> + 0718H
RSCAN0	Receive buffer data field 1 register 17	RSCAN0RMDF117	<RSCAN0_base> + 071CH
RSCAN0	Receive buffer ID register 18	RSCAN0RMID18	<RSCAN0_base> + 0720H
RSCAN0	Receive buffer pointer register 18	RSCAN0RMPTR18	<RSCAN0_base> + 0724H
RSCAN0	Receive buffer data field 0 register 18	RSCAN0RMDF018	<RSCAN0_base> + 0728H
RSCAN0	Receive buffer data field 1 register 18	RSCAN0RMDF118	<RSCAN0_base> + 072CH
RSCAN0	Receive buffer ID register 19	RSCAN0RMID19	<RSCAN0_base> + 0730H
RSCAN0	Receive buffer pointer register 19	RSCAN0RMPTR19	<RSCAN0_base> + 0734H
RSCAN0	Receive buffer data field 0 register 19	RSCAN0RMDF019	<RSCAN0_base> + 0738H
RSCAN0	Receive buffer data field 1 register 19	RSCAN0RMDF119	<RSCAN0_base> + 073CH
RSCAN0	Receive buffer ID register 20	RSCAN0RMID20	<RSCAN0_base> + 0740H
RSCAN0	Receive buffer pointer register 20	RSCAN0RMPTR20	<RSCAN0_base> + 0744H
RSCAN0	Receive buffer data field 0 register 20	RSCAN0RMDF020	<RSCAN0_base> + 0748H
RSCAN0	Receive buffer data field 1 register 20	RSCAN0RMDF120	<RSCAN0_base> + 074CH
RSCAN0	Receive buffer ID register 21	RSCAN0RMID21	<RSCAN0_base> + 0750H
RSCAN0	Receive buffer pointer register 21	RSCAN0RMPTR21	<RSCAN0_base> + 0754H
RSCAN0	Receive buffer data field 0 register 21	RSCAN0RMDF021	<RSCAN0_base> + 0758H
RSCAN0	Receive buffer data field 1 register 21	RSCAN0RMDF121	<RSCAN0_base> + 075CH
RSCAN0	Receive buffer ID register 22	RSCAN0RMID22	<RSCAN0_base> + 0760H
RSCAN0	Receive buffer pointer register 22	RSCAN0RMPTR22	<RSCAN0_base> + 0764H

Table 19.12 List of Registers (13/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 22	RSCAN0RMDF022	<RSCAN0_base> + 0768H
RSCAN0	Receive buffer data field 1 register 22	RSCAN0RMDF122	<RSCAN0_base> + 076CH
RSCAN0	Receive buffer ID register 23	RSCAN0RMID23	<RSCAN0_base> + 0770H
RSCAN0	Receive buffer pointer register 23	RSCAN0RMPTR23	<RSCAN0_base> + 0774H
RSCAN0	Receive buffer data field 0 register 23	RSCAN0RMDF023	<RSCAN0_base> + 0778H
RSCAN0	Receive buffer data field 1 register 23	RSCAN0RMDF123	<RSCAN0_base> + 077CH
RSCAN0	Receive buffer ID register 24	RSCAN0RMID24	<RSCAN0_base> + 0780H
RSCAN0	Receive buffer pointer register 24	RSCAN0RMPTR24	<RSCAN0_base> + 0784H
RSCAN0	Receive buffer data field 0 register 24	RSCAN0RMDF024	<RSCAN0_base> + 0788H
RSCAN0	Receive buffer data field 1 register 24	RSCAN0RMDF124	<RSCAN0_base> + 078CH
RSCAN0	Receive buffer ID register 25	RSCAN0RMID25	<RSCAN0_base> + 0790H
RSCAN0	Receive buffer pointer register 25	RSCAN0RMPTR25	<RSCAN0_base> + 0794H
RSCAN0	Receive buffer data field 0 register 25	RSCAN0RMDF025	<RSCAN0_base> + 0798H
RSCAN0	Receive buffer data field 1 register 25	RSCAN0RMDF125	<RSCAN0_base> + 079CH
RSCAN0	Receive buffer ID register 26	RSCAN0RMID26	<RSCAN0_base> + 07A0H
RSCAN0	Receive buffer pointer register 26	RSCAN0RMPTR26	<RSCAN0_base> + 07A4H
RSCAN0	Receive buffer data field 0 register 26	RSCAN0RMDF026	<RSCAN0_base> + 07A8H
RSCAN0	Receive buffer data field 1 register 26	RSCAN0RMDF126	<RSCAN0_base> + 07ACH
RSCAN0	Receive buffer ID register 27	RSCAN0RMID27	<RSCAN0_base> + 07B0H
RSCAN0	Receive buffer pointer register 27	RSCAN0RMPTR27	<RSCAN0_base> + 07B4H
RSCAN0	Receive buffer data field 0 register 27	RSCAN0RMDF027	<RSCAN0_base> + 07B8H
RSCAN0	Receive buffer data field 1 register 27	RSCAN0RMDF127	<RSCAN0_base> + 07BCH
RSCAN0	Receive buffer ID register 28	RSCAN0RMID28	<RSCAN0_base> + 07C0H
RSCAN0	Receive buffer pointer register 28	RSCAN0RMPTR28	<RSCAN0_base> + 07C4H
RSCAN0	Receive buffer data field 0 register 28	RSCAN0RMDF028	<RSCAN0_base> + 07C8H
RSCAN0	Receive buffer data field 1 register 28	RSCAN0RMDF128	<RSCAN0_base> + 07CCH
RSCAN0	Receive buffer ID register 29	RSCAN0RMID29	<RSCAN0_base> + 07D0H
RSCAN0	Receive buffer pointer register 29	RSCAN0RMPTR29	<RSCAN0_base> + 07D4H
RSCAN0	Receive buffer data field 0 register 29	RSCAN0RMDF029	<RSCAN0_base> + 07D8H
RSCAN0	Receive buffer data field 1 register 29	RSCAN0RMDF129	<RSCAN0_base> + 07DCH
RSCAN0	Receive buffer ID register 30	RSCAN0RMID30	<RSCAN0_base> + 07E0H
RSCAN0	Receive buffer pointer register 30	RSCAN0RMPTR30	<RSCAN0_base> + 07E4H
RSCAN0	Receive buffer data field 0 register 30	RSCAN0RMDF030	<RSCAN0_base> + 07E8H
RSCAN0	Receive buffer data field 1 register 30	RSCAN0RMDF130	<RSCAN0_base> + 07ECH
RSCAN0	Receive buffer ID register 31	RSCAN0RMID31	<RSCAN0_base> + 07F0H
RSCAN0	Receive buffer pointer register 31	RSCAN0RMPTR31	<RSCAN0_base> + 07F4H
RSCAN0	Receive buffer data field 0 register 31	RSCAN0RMDF031	<RSCAN0_base> + 07F8H
RSCAN0	Receive buffer data field 1 register 31	RSCAN0RMDF131	<RSCAN0_base> + 07FCH
RSCAN0	Receive buffer ID register 32	RSCAN0RMID32	<RSCAN0_base> + 0800H
RSCAN0	Receive buffer pointer register 32	RSCAN0RMPTR32	<RSCAN0_base> + 0804H
RSCAN0	Receive buffer data field 0 register 32	RSCAN0RMDF032	<RSCAN0_base> + 0808H
RSCAN0	Receive buffer data field 1 register 32	RSCAN0RMDF132	<RSCAN0_base> + 080CH
RSCAN0	Receive buffer ID register 33	RSCAN0RMID33	<RSCAN0_base> + 0810H
RSCAN0	Receive buffer pointer register 33	RSCAN0RMPTR33	<RSCAN0_base> + 0814H

Table 19.12 List of Registers (14/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 33	RSCAN0RMDF033	<RSCAN0_base> + 0818H
RSCAN0	Receive buffer data field 1 register 33	RSCAN0RMDF133	<RSCAN0_base> + 081CH
RSCAN0	Receive buffer ID register 34	RSCAN0RMID34	<RSCAN0_base> + 0820H
RSCAN0	Receive buffer pointer register 34	RSCAN0RMPTR34	<RSCAN0_base> + 0824H
RSCAN0	Receive buffer data field 0 register 34	RSCAN0RMDF034	<RSCAN0_base> + 0828H
RSCAN0	Receive buffer data field 1 register 34	RSCAN0RMDF134	<RSCAN0_base> + 082CH
RSCAN0	Receive buffer ID register 35	RSCAN0RMID35	<RSCAN0_base> + 0830H
RSCAN0	Receive buffer pointer register 35	RSCAN0RMPTR35	<RSCAN0_base> + 0834H
RSCAN0	Receive buffer data field 0 register 35	RSCAN0RMDF035	<RSCAN0_base> + 0838H
RSCAN0	Receive buffer data field 1 register 35	RSCAN0RMDF135	<RSCAN0_base> + 083CH
RSCAN0	Receive buffer ID register 36	RSCAN0RMID36	<RSCAN0_base> + 0840H
RSCAN0	Receive buffer pointer register 36	RSCAN0RMPTR36	<RSCAN0_base> + 0844H
RSCAN0	Receive buffer data field 0 register 36	RSCAN0RMDF036	<RSCAN0_base> + 0848H
RSCAN0	Receive buffer data field 1 register 36	RSCAN0RMDF136	<RSCAN0_base> + 084CH
RSCAN0	Receive buffer ID register 37	RSCAN0RMID37	<RSCAN0_base> + 0850H
RSCAN0	Receive buffer pointer register 37	RSCAN0RMPTR37	<RSCAN0_base> + 0854H
RSCAN0	Receive buffer data field 0 register 37	RSCAN0RMDF037	<RSCAN0_base> + 0858H
RSCAN0	Receive buffer data field 1 register 37	RSCAN0RMDF137	<RSCAN0_base> + 085CH
RSCAN0	Receive buffer ID register 38	RSCAN0RMID38	<RSCAN0_base> + 0860H
RSCAN0	Receive buffer pointer register 38	RSCAN0RMPTR38	<RSCAN0_base> + 0864H
RSCAN0	Receive buffer data field 0 register 38	RSCAN0RMDF038	<RSCAN0_base> + 0868H
RSCAN0	Receive buffer data field 1 register 38	RSCAN0RMDF138	<RSCAN0_base> + 086CH
RSCAN0	Receive buffer ID register 39	RSCAN0RMID39	<RSCAN0_base> + 0870H
RSCAN0	Receive buffer pointer register 39	RSCAN0RMPTR39	<RSCAN0_base> + 0874H
RSCAN0	Receive buffer data field 0 register 39	RSCAN0RMDF039	<RSCAN0_base> + 0878H
RSCAN0	Receive buffer data field 1 register 39	RSCAN0RMDF139	<RSCAN0_base> + 087CH
RSCAN0	Receive buffer ID register 40	RSCAN0RMID40	<RSCAN0_base> + 0880H
RSCAN0	Receive buffer pointer register 40	RSCAN0RMPTR40	<RSCAN0_base> + 0884H
RSCAN0	Receive buffer data field 0 register 40	RSCAN0RMDF040	<RSCAN0_base> + 0888H
RSCAN0	Receive buffer data field 1 register 40	RSCAN0RMDF140	<RSCAN0_base> + 088CH
RSCAN0	Receive buffer ID register 41	RSCAN0RMID41	<RSCAN0_base> + 0890H
RSCAN0	Receive buffer pointer register 41	RSCAN0RMPTR41	<RSCAN0_base> + 0894H
RSCAN0	Receive buffer data field 0 register 41	RSCAN0RMDF041	<RSCAN0_base> + 0898H
RSCAN0	Receive buffer data field 1 register 41	RSCAN0RMDF141	<RSCAN0_base> + 089CH
RSCAN0	Receive buffer ID register 42	RSCAN0RMID42	<RSCAN0_base> + 08A0H
RSCAN0	Receive buffer pointer register 42	RSCAN0RMPTR42	<RSCAN0_base> + 08A4H
RSCAN0	Receive buffer data field 0 register 42	RSCAN0RMDF042	<RSCAN0_base> + 08A8H
RSCAN0	Receive buffer data field 1 register 42	RSCAN0RMDF142	<RSCAN0_base> + 08ACH
RSCAN0	Receive buffer ID register 43	RSCAN0RMID43	<RSCAN0_base> + 08B0H
RSCAN0	Receive buffer pointer register 43	RSCAN0RMPTR43	<RSCAN0_base> + 08B4H
RSCAN0	Receive buffer data field 0 register 43	RSCAN0RMDF043	<RSCAN0_base> + 08B8H
RSCAN0	Receive buffer data field 1 register 43	RSCAN0RMDF143	<RSCAN0_base> + 08BCH
RSCAN0	Receive buffer ID register 44	RSCAN0RMID44	<RSCAN0_base> + 08C0H
RSCAN0	Receive buffer pointer register 44	RSCAN0RMPTR44	<RSCAN0_base> + 08C4H

Table 19.12 List of Registers (15/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 44	RSCAN0RMDF044	<RSCAN0_base> + 08C8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 44	RSCAN0RMDF144	<RSCAN0_base> + 08CC <sub>H</sub>
RSCAN0	Receive buffer ID register 45	RSCAN0RMID45	<RSCAN0_base> + 08D0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 45	RSCAN0RMPTR45	<RSCAN0_base> + 08D4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 45	RSCAN0RMDF045	<RSCAN0_base> + 08D8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 45	RSCAN0RMDF145	<RSCAN0_base> + 08DC <sub>H</sub>
RSCAN0	Receive buffer ID register 46	RSCAN0RMID46	<RSCAN0_base> + 08E0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 46	RSCAN0RMPTR46	<RSCAN0_base> + 08E4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 46	RSCAN0RMDF046	<RSCAN0_base> + 08E8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 46	RSCAN0RMDF146	<RSCAN0_base> + 08EC <sub>H</sub>
RSCAN0	Receive buffer ID register 47	RSCAN0RMID47	<RSCAN0_base> + 08F0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 47	RSCAN0RMPTR47	<RSCAN0_base> + 08F4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 47	RSCAN0RMDF047	<RSCAN0_base> + 08F8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 47	RSCAN0RMDF147	<RSCAN0_base> + 08FC <sub>H</sub>
RSCAN0	Receive buffer ID register 48	RSCAN0RMID48	<RSCAN0_base> + 0900 <sub>H</sub>
RSCAN0	Receive buffer pointer register 48	RSCAN0RMPTR48	<RSCAN0_base> + 0904 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 48	RSCAN0RMDF048	<RSCAN0_base> + 0908 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 48	RSCAN0RMDF148	<RSCAN0_base> + 090C <sub>H</sub>
RSCAN0	Receive buffer ID register 49	RSCAN0RMID49	<RSCAN0_base> + 0910 <sub>H</sub>
RSCAN0	Receive buffer pointer register 49	RSCAN0RMPTR49	<RSCAN0_base> + 0914 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 49	RSCAN0RMDF049	<RSCAN0_base> + 0918 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 49	RSCAN0RMDF149	<RSCAN0_base> + 091C <sub>H</sub>
RSCAN0	Receive buffer ID register 50	RSCAN0RMID50	<RSCAN0_base> + 0920 <sub>H</sub>
RSCAN0	Receive buffer pointer register 50	RSCAN0RMPTR50	<RSCAN0_base> + 0924 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 50	RSCAN0RMDF050	<RSCAN0_base> + 0928 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 50	RSCAN0RMDF150	<RSCAN0_base> + 092C <sub>H</sub>
RSCAN0	Receive buffer ID register 51	RSCAN0RMID51	<RSCAN0_base> + 0930 <sub>H</sub>
RSCAN0	Receive buffer pointer register 51	RSCAN0RMPTR51	<RSCAN0_base> + 0934 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 51	RSCAN0RMDF051	<RSCAN0_base> + 0938 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 51	RSCAN0RMDF151	<RSCAN0_base> + 093C <sub>H</sub>
RSCAN0	Receive buffer ID register 52	RSCAN0RMID52	<RSCAN0_base> + 0940 <sub>H</sub>
RSCAN0	Receive buffer pointer register 52	RSCAN0RMPTR52	<RSCAN0_base> + 0944 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 52	RSCAN0RMDF052	<RSCAN0_base> + 0948 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 52	RSCAN0RMDF152	<RSCAN0_base> + 094C <sub>H</sub>
RSCAN0	Receive buffer ID register 53	RSCAN0RMID53	<RSCAN0_base> + 0950 <sub>H</sub>
RSCAN0	Receive buffer pointer register 53	RSCAN0RMPTR53	<RSCAN0_base> + 0954 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 53	RSCAN0RMDF053	<RSCAN0_base> + 0958 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 53	RSCAN0RMDF153	<RSCAN0_base> + 095C <sub>H</sub>
RSCAN0	Receive buffer ID register 54	RSCAN0RMID54	<RSCAN0_base> + 0960 <sub>H</sub>
RSCAN0	Receive buffer pointer register 54	RSCAN0RMPTR54	<RSCAN0_base> + 0964 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 54	RSCAN0RMDF054	<RSCAN0_base> + 0968 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 54	RSCAN0RMDF154	<RSCAN0_base> + 096C <sub>H</sub>
RSCAN0	Receive buffer ID register 55	RSCAN0RMID55	<RSCAN0_base> + 0970 <sub>H</sub>
RSCAN0	Receive buffer pointer register 55	RSCAN0RMPTR55	<RSCAN0_base> + 0974 <sub>H</sub>

Table 19.12 List of Registers (16/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 55	RSCAN0RMDF055	<RSCAN0_base> + 0978H
RSCAN0	Receive buffer data field 1 register 55	RSCAN0RMDF155	<RSCAN0_base> + 097CH
RSCAN0	Receive buffer ID register 56	RSCAN0RMID56	<RSCAN0_base> + 0980H
RSCAN0	Receive buffer pointer register 56	RSCAN0RMPTR56	<RSCAN0_base> + 0984H
RSCAN0	Receive buffer data field 0 register 56	RSCAN0RMDF056	<RSCAN0_base> + 0988H
RSCAN0	Receive buffer data field 1 register 56	RSCAN0RMDF156	<RSCAN0_base> + 098CH
RSCAN0	Receive buffer ID register 57	RSCAN0RMID57	<RSCAN0_base> + 0990H
RSCAN0	Receive buffer pointer register 57	RSCAN0RMPTR57	<RSCAN0_base> + 0994H
RSCAN0	Receive buffer data field 0 register 57	RSCAN0RMDF057	<RSCAN0_base> + 0998H
RSCAN0	Receive buffer data field 1 register 57	RSCAN0RMDF157	<RSCAN0_base> + 099CH
RSCAN0	Receive buffer ID register 58	RSCAN0RMID58	<RSCAN0_base> + 09A0H
RSCAN0	Receive buffer pointer register 58	RSCAN0RMPTR58	<RSCAN0_base> + 09A4H
RSCAN0	Receive buffer data field 0 register 58	RSCAN0RMDF058	<RSCAN0_base> + 09A8H
RSCAN0	Receive buffer data field 1 register 58	RSCAN0RMDF158	<RSCAN0_base> + 09ACH
RSCAN0	Receive buffer ID register 59	RSCAN0RMID59	<RSCAN0_base> + 09B0H
RSCAN0	Receive buffer pointer register 59	RSCAN0RMPTR59	<RSCAN0_base> + 09B4H
RSCAN0	Receive buffer data field 0 register 59	RSCAN0RMDF059	<RSCAN0_base> + 09B8H
RSCAN0	Receive buffer data field 1 register 59	RSCAN0RMDF159	<RSCAN0_base> + 09BCH
RSCAN0	Receive buffer ID register 60	RSCAN0RMID60	<RSCAN0_base> + 09C0H
RSCAN0	Receive buffer pointer register 60	RSCAN0RMPTR60	<RSCAN0_base> + 09C4H
RSCAN0	Receive buffer data field 0 register 60	RSCAN0RMDF060	<RSCAN0_base> + 09C8H
RSCAN0	Receive buffer data field 1 register 60	RSCAN0RMDF160	<RSCAN0_base> + 09CCH
RSCAN0	Receive buffer ID register 61	RSCAN0RMID61	<RSCAN0_base> + 09D0H
RSCAN0	Receive buffer pointer register 61	RSCAN0RMPTR61	<RSCAN0_base> + 09D4H
RSCAN0	Receive buffer data field 0 register 61	RSCAN0RMDF061	<RSCAN0_base> + 09D8H
RSCAN0	Receive buffer data field 1 register 61	RSCAN0RMDF161	<RSCAN0_base> + 09DCH
RSCAN0	Receive buffer ID register 62	RSCAN0RMID62	<RSCAN0_base> + 09E0H
RSCAN0	Receive buffer pointer register 62	RSCAN0RMPTR62	<RSCAN0_base> + 09E4H
RSCAN0	Receive buffer data field 0 register 62	RSCAN0RMDF062	<RSCAN0_base> + 09E8H
RSCAN0	Receive buffer data field 1 register 62	RSCAN0RMDF162	<RSCAN0_base> + 09ECH
RSCAN0	Receive buffer ID register 63	RSCAN0RMID63	<RSCAN0_base> + 09F0H
RSCAN0	Receive buffer pointer register 63	RSCAN0RMPTR63	<RSCAN0_base> + 09F4H
RSCAN0	Receive buffer data field 0 register 63	RSCAN0RMDF063	<RSCAN0_base> + 09F8H
RSCAN0	Receive buffer data field 1 register 63	RSCAN0RMDF163	<RSCAN0_base> + 09FCH
RSCAN0	Receive buffer ID register 64	RSCAN0RMID64	<RSCAN0_base> + 0A00H
RSCAN0	Receive buffer pointer register 64	RSCAN0RMPTR64	<RSCAN0_base> + 0A04H
RSCAN0	Receive buffer data field 0 register 64	RSCAN0RMDF064	<RSCAN0_base> + 0A08H
RSCAN0	Receive buffer data field 1 register 64	RSCAN0RMDF164	<RSCAN0_base> + 0A0CH
RSCAN0	Receive buffer ID register 65	RSCAN0RMID65	<RSCAN0_base> + 0A10H
RSCAN0	Receive buffer pointer register 65	RSCAN0RMPTR65	<RSCAN0_base> + 0A14H
RSCAN0	Receive buffer data field 0 register 65	RSCAN0RMDF065	<RSCAN0_base> + 0A18H
RSCAN0	Receive buffer data field 1 register 65	RSCAN0RMDF165	<RSCAN0_base> + 0A1CH
RSCAN0	Receive buffer ID register 66	RSCAN0RMID66	<RSCAN0_base> + 0A20H
RSCAN0	Receive buffer pointer register 66	RSCAN0RMPTR66	<RSCAN0_base> + 0A24H

Table 19.12 List of Registers (17/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 66	RSCAN0RMDF066	<RSCAN0_base> + 0A28H
RSCAN0	Receive buffer data field 1 register 66	RSCAN0RMDF166	<RSCAN0_base> + 0A2CH
RSCAN0	Receive buffer ID register 67	RSCAN0RMID67	<RSCAN0_base> + 0A30H
RSCAN0	Receive buffer pointer register 67	RSCAN0RMPTR67	<RSCAN0_base> + 0A34H
RSCAN0	Receive buffer data field 0 register 67	RSCAN0RMDF067	<RSCAN0_base> + 0A38H
RSCAN0	Receive buffer data field 1 register 67	RSCAN0RMDF167	<RSCAN0_base> + 0A3CH
RSCAN0	Receive buffer ID register 68	RSCAN0RMID68	<RSCAN0_base> + 0A40H
RSCAN0	Receive buffer pointer register 68	RSCAN0RMPTR68	<RSCAN0_base> + 0A44H
RSCAN0	Receive buffer data field 0 register 68	RSCAN0RMDF068	<RSCAN0_base> + 0A48H
RSCAN0	Receive buffer data field 1 register 68	RSCAN0RMDF168	<RSCAN0_base> + 0A4CH
RSCAN0	Receive buffer ID register 69	RSCAN0RMID69	<RSCAN0_base> + 0A50H
RSCAN0	Receive buffer pointer register 69	RSCAN0RMPTR69	<RSCAN0_base> + 0A54H
RSCAN0	Receive buffer data field 0 register 69	RSCAN0RMDF069	<RSCAN0_base> + 0A58H
RSCAN0	Receive buffer data field 1 register 69	RSCAN0RMDF169	<RSCAN0_base> + 0A5CH
RSCAN0	Receive buffer ID register 70	RSCAN0RMID70	<RSCAN0_base> + 0A60H
RSCAN0	Receive buffer pointer register 70	RSCAN0RMPTR70	<RSCAN0_base> + 0A64H
RSCAN0	Receive buffer data field 0 register 70	RSCAN0RMDF070	<RSCAN0_base> + 0A68H
RSCAN0	Receive buffer data field 1 register 70	RSCAN0RMDF170	<RSCAN0_base> + 0A6CH
RSCAN0	Receive buffer ID register 71	RSCAN0RMID71	<RSCAN0_base> + 0A70H
RSCAN0	Receive buffer pointer register 71	RSCAN0RMPTR71	<RSCAN0_base> + 0A74H
RSCAN0	Receive buffer data field 0 register 71	RSCAN0RMDF071	<RSCAN0_base> + 0A78H
RSCAN0	Receive buffer data field 1 register 71	RSCAN0RMDF171	<RSCAN0_base> + 0A7CH
RSCAN0	Receive buffer ID register 72	RSCAN0RMID72	<RSCAN0_base> + 0A80H
RSCAN0	Receive buffer pointer register 72	RSCAN0RMPTR72	<RSCAN0_base> + 0A84H
RSCAN0	Receive buffer data field 0 register 72	RSCAN0RMDF072	<RSCAN0_base> + 0A88H
RSCAN0	Receive buffer data field 1 register 72	RSCAN0RMDF172	<RSCAN0_base> + 0A8CH
RSCAN0	Receive buffer ID register 73	RSCAN0RMID73	<RSCAN0_base> + 0A90H
RSCAN0	Receive buffer pointer register 73	RSCAN0RMPTR73	<RSCAN0_base> + 0A94H
RSCAN0	Receive buffer data field 0 register 73	RSCAN0RMDF073	<RSCAN0_base> + 0A98H
RSCAN0	Receive buffer data field 1 register 73	RSCAN0RMDF173	<RSCAN0_base> + 0A9CH
RSCAN0	Receive buffer ID register 74	RSCAN0RMID74	<RSCAN0_base> + 0AA0H
RSCAN0	Receive buffer pointer register 74	RSCAN0RMPTR74	<RSCAN0_base> + 0AA4H
RSCAN0	Receive buffer data field 0 register 74	RSCAN0RMDF074	<RSCAN0_base> + 0AA8H
RSCAN0	Receive buffer data field 1 register 74	RSCAN0RMDF174	<RSCAN0_base> + 0AACH
RSCAN0	Receive buffer ID register 75	RSCAN0RMID75	<RSCAN0_base> + 0AB0H
RSCAN0	Receive buffer pointer register 75	RSCAN0RMPTR75	<RSCAN0_base> + 0AB4H
RSCAN0	Receive buffer data field 0 register 75	RSCAN0RMDF075	<RSCAN0_base> + 0AB8H
RSCAN0	Receive buffer data field 1 register 75	RSCAN0RMDF175	<RSCAN0_base> + 0ABC <sub>H</sub>
RSCAN0	Receive buffer ID register 76	RSCAN0RMID76	<RSCAN0_base> + 0AC0H
RSCAN0	Receive buffer pointer register 76	RSCAN0RMPTR76	<RSCAN0_base> + 0AC4H
RSCAN0	Receive buffer data field 0 register 76	RSCAN0RMDF076	<RSCAN0_base> + 0AC8H
RSCAN0	Receive buffer data field 1 register 76	RSCAN0RMDF176	<RSCAN0_base> + 0ACC <sub>H</sub>
RSCAN0	Receive buffer ID register 77	RSCAN0RMID77	<RSCAN0_base> + 0AD0H
RSCAN0	Receive buffer pointer register 77	RSCAN0RMPTR77	<RSCAN0_base> + 0AD4H

Table 19.12 List of Registers (18/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 77	RSCAN0RMDF077	<RSCAN0_base> + 0AD8H
RSCAN0	Receive buffer data field 1 register 77	RSCAN0RMDF177	<RSCAN0_base> + 0ADC <sub>H</sub>
RSCAN0	Receive buffer ID register 78	RSCAN0RMID78	<RSCAN0_base> + 0AE0H
RSCAN0	Receive buffer pointer register 78	RSCAN0RMPTR78	<RSCAN0_base> + 0AE4H
RSCAN0	Receive buffer data field 0 register 78	RSCAN0RMDF078	<RSCAN0_base> + 0AE8H
RSCAN0	Receive buffer data field 1 register 78	RSCAN0RMDF178	<RSCAN0_base> + 0AEC <sub>H</sub>
RSCAN0	Receive buffer ID register 79	RSCAN0RMID79	<RSCAN0_base> + 0AF0H
RSCAN0	Receive buffer pointer register 79	RSCAN0RMPTR79	<RSCAN0_base> + 0AF4H
RSCAN0	Receive buffer data field 0 register 79	RSCAN0RMDF079	<RSCAN0_base> + 0AF8H
RSCAN0	Receive buffer data field 1 register 79	RSCAN0RMDF179	<RSCAN0_base> + 0AFC <sub>H</sub>
RSCAN0	Receive buffer ID register 80	RSCAN0RMID80	<RSCAN0_base> + 0B00H
RSCAN0	Receive buffer pointer register 80	RSCAN0RMPTR80	<RSCAN0_base> + 0B04H
RSCAN0	Receive buffer data field 0 register 80	RSCAN0RMDF080	<RSCAN0_base> + 0B08H
RSCAN0	Receive buffer data field 1 register 80	RSCAN0RMDF180	<RSCAN0_base> + 0B0CH
RSCAN0	Receive buffer ID register 81	RSCAN0RMID81	<RSCAN0_base> + 0B10H
RSCAN0	Receive buffer pointer register 81	RSCAN0RMPTR81	<RSCAN0_base> + 0B14H
RSCAN0	Receive buffer data field 0 register 81	RSCAN0RMDF081	<RSCAN0_base> + 0B18H
RSCAN0	Receive buffer data field 1 register 81	RSCAN0RMDF181	<RSCAN0_base> + 0B1CH
RSCAN0	Receive buffer ID register 82	RSCAN0RMID82	<RSCAN0_base> + 0B20H
RSCAN0	Receive buffer pointer register 82	RSCAN0RMPTR82	<RSCAN0_base> + 0B24H
RSCAN0	Receive buffer data field 0 register 82	RSCAN0RMDF082	<RSCAN0_base> + 0B28H
RSCAN0	Receive buffer data field 1 register 82	RSCAN0RMDF182	<RSCAN0_base> + 0B2CH
RSCAN0	Receive buffer ID register 83	RSCAN0RMID83	<RSCAN0_base> + 0B30H
RSCAN0	Receive buffer pointer register 83	RSCAN0RMPTR83	<RSCAN0_base> + 0B34H
RSCAN0	Receive buffer data field 0 register 83	RSCAN0RMDF083	<RSCAN0_base> + 0B38H
RSCAN0	Receive buffer data field 1 register 83	RSCAN0RMDF183	<RSCAN0_base> + 0B3CH
RSCAN0	Receive buffer ID register 84	RSCAN0RMID84	<RSCAN0_base> + 0B40H
RSCAN0	Receive buffer pointer register 84	RSCAN0RMPTR84	<RSCAN0_base> + 0B44H
RSCAN0	Receive buffer data field 0 register 84	RSCAN0RMDF084	<RSCAN0_base> + 0B48H
RSCAN0	Receive buffer data field 1 register 84	RSCAN0RMDF184	<RSCAN0_base> + 0B4CH
RSCAN0	Receive buffer ID register 85	RSCAN0RMID85	<RSCAN0_base> + 0B50H
RSCAN0	Receive buffer pointer register 85	RSCAN0RMPTR85	<RSCAN0_base> + 0B54H
RSCAN0	Receive buffer data field 0 register 85	RSCAN0RMDF085	<RSCAN0_base> + 0B58H
RSCAN0	Receive buffer data field 1 register 85	RSCAN0RMDF185	<RSCAN0_base> + 0B5CH
RSCAN0	Receive buffer ID register 86	RSCAN0RMID86	<RSCAN0_base> + 0B60H
RSCAN0	Receive buffer pointer register 86	RSCAN0RMPTR86	<RSCAN0_base> + 0B64H
RSCAN0	Receive buffer data field 0 register 86	RSCAN0RMDF086	<RSCAN0_base> + 0B68H
RSCAN0	Receive buffer data field 1 register 86	RSCAN0RMDF186	<RSCAN0_base> + 0B6CH
RSCAN0	Receive buffer ID register 87	RSCAN0RMID87	<RSCAN0_base> + 0B70H
RSCAN0	Receive buffer pointer register 87	RSCAN0RMPTR87	<RSCAN0_base> + 0B74H
RSCAN0	Receive buffer data field 0 register 87	RSCAN0RMDF087	<RSCAN0_base> + 0B78H
RSCAN0	Receive buffer data field 1 register 87	RSCAN0RMDF187	<RSCAN0_base> + 0B7CH
RSCAN0	Receive buffer ID register 88	RSCAN0RMID88	<RSCAN0_base> + 0B80H
RSCAN0	Receive buffer pointer register 88	RSCAN0RMPTR88	<RSCAN0_base> + 0B84H

Table 19.12 List of Registers (19/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 88	RSCAN0RMDF088	<RSCAN0_base> + 0B88 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 88	RSCAN0RMDF188	<RSCAN0_base> + 0B8C <sub>H</sub>
RSCAN0	Receive buffer ID register 89	RSCAN0RMID89	<RSCAN0_base> + 0B90 <sub>H</sub>
RSCAN0	Receive buffer pointer register 89	RSCAN0RMPTR89	<RSCAN0_base> + 0B94 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 89	RSCAN0RMDF089	<RSCAN0_base> + 0B98 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 89	RSCAN0RMDF189	<RSCAN0_base> + 0B9C <sub>H</sub>
RSCAN0	Receive buffer ID register 90	RSCAN0RMID90	<RSCAN0_base> + 0BA0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 90	RSCAN0RMPTR90	<RSCAN0_base> + 0BA4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 90	RSCAN0RMDF090	<RSCAN0_base> + 0BA8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 90	RSCAN0RMDF190	<RSCAN0_base> + 0BAC <sub>H</sub>
RSCAN0	Receive buffer ID register 91	RSCAN0RMID91	<RSCAN0_base> + 0BB0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 91	RSCAN0RMPTR91	<RSCAN0_base> + 0BB4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 91	RSCAN0RMDF091	<RSCAN0_base> + 0BB8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 91	RSCAN0RMDF191	<RSCAN0_base> + 0BBC <sub>H</sub>
RSCAN0	Receive buffer ID register 92	RSCAN0RMID92	<RSCAN0_base> + 0BC0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 92	RSCAN0RMPTR92	<RSCAN0_base> + 0BC4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 92	RSCAN0RMDF092	<RSCAN0_base> + 0BC8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 92	RSCAN0RMDF192	<RSCAN0_base> + 0BCC <sub>H</sub>
RSCAN0	Receive buffer ID register 93	RSCAN0RMID93	<RSCAN0_base> + 0BD0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 93	RSCAN0RMPTR93	<RSCAN0_base> + 0BD4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 93	RSCAN0RMDF093	<RSCAN0_base> + 0BD8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 93	RSCAN0RMDF193	<RSCAN0_base> + 0BDC <sub>H</sub>
RSCAN0	Receive buffer ID register 94	RSCAN0RMID94	<RSCAN0_base> + 0BE0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 94	RSCAN0RMPTR94	<RSCAN0_base> + 0BE4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 94	RSCAN0RMDF094	<RSCAN0_base> + 0BE8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 94	RSCAN0RMDF194	<RSCAN0_base> + 0BEC <sub>H</sub>
RSCAN0	Receive buffer ID register 95	RSCAN0RMID95	<RSCAN0_base> + 0BF0 <sub>H</sub>
RSCAN0	Receive buffer pointer register 95	RSCAN0RMPTR95	<RSCAN0_base> + 0BF4 <sub>H</sub>
RSCAN0	Receive buffer data field 0 register 95	RSCAN0RMDF095	<RSCAN0_base> + 0BF8 <sub>H</sub>
RSCAN0	Receive buffer data field 1 register 95	RSCAN0RMDF195	<RSCAN0_base> + 0BFC <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 0	RSCAN0RFID0	<RSCAN0_base> + 0E00 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 0	RSCAN0RFPTR0	<RSCAN0_base> + 0E04 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 0	RSCAN0RFDF00	<RSCAN0_base> + 0E08 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 0	RSCAN0RFDF10	<RSCAN0_base> + 0E0C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 1	RSCAN0RFID1	<RSCAN0_base> + 0E10 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 1	RSCAN0RFPTR1	<RSCAN0_base> + 0E14 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 1	RSCAN0RFDF01	<RSCAN0_base> + 0E18 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 1	RSCAN0RFDF11	<RSCAN0_base> + 0E1C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 2	RSCAN0RFID2	<RSCAN0_base> + 0E20 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 2	RSCAN0RFPTR2	<RSCAN0_base> + 0E24 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 2	RSCAN0RFDF02	<RSCAN0_base> + 0E28 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 2	RSCAN0RFDF12	<RSCAN0_base> + 0E2C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 3	RSCAN0RFID3	<RSCAN0_base> + 0E30 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 3	RSCAN0RFPTR3	<RSCAN0_base> + 0E34 <sub>H</sub>

Table 19.12 List of Registers (20/32)

Module	Register	Symbol	Address
RSCAN0	Receive FIFO buffer access data field 0 register 3	RSCAN0RFDF03	<RSCAN0_base> + 0E38 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 3	RSCAN0RFDF13	<RSCAN0_base> + 0E3C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 4	RSCAN0RFID4	<RSCAN0_base> + 0E40 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 4	RSCAN0RFPTR4	<RSCAN0_base> + 0E44 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 4	RSCAN0RFDF04	<RSCAN0_base> + 0E48 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 4	RSCAN0RFDF14	<RSCAN0_base> + 0E4C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 5	RSCAN0RFID5	<RSCAN0_base> + 0E50 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 5	RSCAN0RFPTR5	<RSCAN0_base> + 0E54 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 5	RSCAN0RFDF05	<RSCAN0_base> + 0E58 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 5	RSCAN0RFDF15	<RSCAN0_base> + 0E5C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 6	RSCAN0RFID6	<RSCAN0_base> + 0E60 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 6	RSCAN0RFPTR6	<RSCAN0_base> + 0E64 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 6	RSCAN0RFDF06	<RSCAN0_base> + 0E68 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 6	RSCAN0RFDF16	<RSCAN0_base> + 0E6C <sub>H</sub>
RSCAN0	Receive FIFO buffer access ID register 7	RSCAN0RFID7	<RSCAN0_base> + 0E70 <sub>H</sub>
RSCAN0	Receive FIFO buffer access pointer register 7	RSCAN0RFPTR7	<RSCAN0_base> + 0E74 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 0 register 7	RSCAN0RFDF07	<RSCAN0_base> + 0E78 <sub>H</sub>
RSCAN0	Receive FIFO buffer access data field 1 register 7	RSCAN0RFDF17	<RSCAN0_base> + 0E7C <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 0	RSCAN0CFID0	<RSCAN0_base> + 0E80 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 0	RSCAN0CFPTR0	<RSCAN0_base> + 0E84 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 0	RSCAN0CFDF00	<RSCAN0_base> + 0E88 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 0	RSCAN0CFDF10	<RSCAN0_base> + 0E8C <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 1	RSCAN0CFID1	<RSCAN0_base> + 0E90 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 1	RSCAN0CFPTR1	<RSCAN0_base> + 0E94 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 1	RSCAN0CFDF01	<RSCAN0_base> + 0E98 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 1	RSCAN0CFDF11	<RSCAN0_base> + 0E9C <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 2	RSCAN0CFID2	<RSCAN0_base> + 0EA0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 2	RSCAN0CFPTR2	<RSCAN0_base> + 0EA4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 2	RSCAN0CFDF02	<RSCAN0_base> + 0EA8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 2	RSCAN0CFDF12	<RSCAN0_base> + 0EAC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 3	RSCAN0CFID3	<RSCAN0_base> + 0EB0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 3	RSCAN0CFPTR3	<RSCAN0_base> + 0EB4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 3	RSCAN0CFDF03	<RSCAN0_base> + 0EB8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 3	RSCAN0CFDF13	<RSCAN0_base> + 0EBC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 4	RSCAN0CFID4	<RSCAN0_base> + 0EC0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 4	RSCAN0CFPTR4	<RSCAN0_base> + 0EC4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 4	RSCAN0CFDF04	<RSCAN0_base> + 0EC8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 4	RSCAN0CFDF14	<RSCAN0_base> + 0ECC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 5	RSCAN0CFID5	<RSCAN0_base> + 0ED0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 5	RSCAN0CFPTR5	<RSCAN0_base> + 0ED4 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 5	RSCAN0CFDF05	<RSCAN0_base> + 0ED8 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 5	RSCAN0CFDF15	<RSCAN0_base> + 0EDC <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access ID register 6	RSCAN0CFID6	<RSCAN0_base> + 0EE0 <sub>H</sub>
RSCAN0	Transmit/receive FIFO buffer access pointer register 6	RSCAN0CFPTR6	<RSCAN0_base> + 0EE4 <sub>H</sub>

Table 19.12 List of Registers (21/32)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 6	RSCAN0CFDF06	<RSCAN0_base> + 0EE8H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 6	RSCAN0CFDF16	<RSCAN0_base> + 0EECH
RSCAN0	Transmit/receive FIFO buffer access ID register 7	RSCAN0CFID7	<RSCAN0_base> + 0EF0H
RSCAN0	Transmit/receive FIFO buffer access pointer register 7	RSCAN0CFPTR7	<RSCAN0_base> + 0EF4H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 7	RSCAN0CFDF07	<RSCAN0_base> + 0EF8H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 7	RSCAN0CFDF17	<RSCAN0_base> + 0EFCH
RSCAN0	Transmit/receive FIFO buffer access ID register 8	RSCAN0CFID8	<RSCAN0_base> + 0F00H
RSCAN0	Transmit/receive FIFO buffer access pointer register 8	RSCAN0CFPTR8	<RSCAN0_base> + 0F04H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 8	RSCAN0CFDF08	<RSCAN0_base> + 0F08H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 8	RSCAN0CFDF18	<RSCAN0_base> + 0F0CH
RSCAN0	Transmit/receive FIFO buffer access ID register 9	RSCAN0CFID9	<RSCAN0_base> + 0F10H
RSCAN0	Transmit/receive FIFO buffer access pointer register 9	RSCAN0CFPTR9	<RSCAN0_base> + 0F14H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 9	RSCAN0CFDF09	<RSCAN0_base> + 0F18H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 9	RSCAN0CFDF19	<RSCAN0_base> + 0F1CH
RSCAN0	Transmit/receive FIFO buffer access ID register 10	RSCAN0CFID10	<RSCAN0_base> + 0F20H
RSCAN0	Transmit/receive FIFO buffer access pointer register 10	RSCAN0CFPTR10	<RSCAN0_base> + 0F24H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 10	RSCAN0CFDF010	<RSCAN0_base> + 0F28H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 10	RSCAN0CFDF110	<RSCAN0_base> + 0F2CH
RSCAN0	Transmit/receive FIFO buffer access ID register 11	RSCAN0CFID11	<RSCAN0_base> + 0F30H
RSCAN0	Transmit/receive FIFO buffer access pointer register 11	RSCAN0CFPTR11	<RSCAN0_base> + 0F34H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 11	RSCAN0CFDF011	<RSCAN0_base> + 0F38H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 11	RSCAN0CFDF111	<RSCAN0_base> + 0F3CH
RSCAN0	Transmit/receive FIFO buffer access ID register 12	RSCAN0CFID12	<RSCAN0_base> + 0F40H
RSCAN0	Transmit/receive FIFO buffer access pointer register 12	RSCAN0CFPTR12	<RSCAN0_base> + 0F44H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 12	RSCAN0CFDF012	<RSCAN0_base> + 0F48H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 12	RSCAN0CFDF112	<RSCAN0_base> + 0F4CH
RSCAN0	Transmit/receive FIFO buffer access ID register 13	RSCAN0CFID13	<RSCAN0_base> + 0F50H
RSCAN0	Transmit/receive FIFO buffer access pointer register 13	RSCAN0CFPTR13	<RSCAN0_base> + 0F54H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 13	RSCAN0CFDF013	<RSCAN0_base> + 0F58H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 13	RSCAN0CFDF113	<RSCAN0_base> + 0F5CH
RSCAN0	Transmit/receive FIFO buffer access ID register 14	RSCAN0CFID14	<RSCAN0_base> + 0F60H
RSCAN0	Transmit/receive FIFO buffer access pointer register 14	RSCAN0CFPTR14	<RSCAN0_base> + 0F64H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 14	RSCAN0CFDF014	<RSCAN0_base> + 0F68H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 14	RSCAN0CFDF114	<RSCAN0_base> + 0F6CH
RSCAN0	Transmit/receive FIFO buffer access ID register 15	RSCAN0CFID15	<RSCAN0_base> + 0F70H
RSCAN0	Transmit/receive FIFO buffer access pointer register 15	RSCAN0CFPTR15	<RSCAN0_base> + 0F74H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 15	RSCAN0CFDF015	<RSCAN0_base> + 0F78H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 15	RSCAN0CFDF115	<RSCAN0_base> + 0F7CH
RSCAN0	Transmit/receive FIFO buffer access ID register 16	RSCAN0CFID16	<RSCAN0_base> + 0F80H
RSCAN0	Transmit/receive FIFO buffer access pointer register 16	RSCAN0CFPTR16	<RSCAN0_base> + 0F84H
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 16	RSCAN0CFDF016	<RSCAN0_base> + 0F88H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 16	RSCAN0CFDF116	<RSCAN0_base> + 0F8CH
RSCAN0	Transmit/receive FIFO buffer access ID register 17	RSCAN0CFID17	<RSCAN0_base> + 0F90H
RSCAN0	Transmit/receive FIFO buffer access pointer register 17	RSCAN0CFPTR17	<RSCAN0_base> + 0F94H

Table 19.12 List of Registers (22/32)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 17	RSCAN0CFDF017	<RSCAN0_base> + 0F98H
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 17	RSCAN0CFDF117	<RSCAN0_base> + 0F9CH
RSCAN0	Transmit buffer ID register 0	RSCAN0TMID0	<RSCAN0_base> + 1000H
RSCAN0	Transmit buffer pointer register 0	RSCAN0TMRPTR0	<RSCAN0_base> + 1004H
RSCAN0	Transmit buffer data field 0 register 0	RSCAN0TMDF00	<RSCAN0_base> + 1008H
RSCAN0	Transmit buffer data field 1 register 0	RSCAN0TMDF10	<RSCAN0_base> + 100CH
RSCAN0	Transmit buffer ID register 1	RSCAN0TMID1	<RSCAN0_base> + 1010H
RSCAN0	Transmit buffer pointer register 1	RSCAN0TMRPTR1	<RSCAN0_base> + 1014H
RSCAN0	Transmit buffer data field 0 register 1	RSCAN0TMDF01	<RSCAN0_base> + 1018H
RSCAN0	Transmit buffer data field 1 register 1	RSCAN0TMDF11	<RSCAN0_base> + 101CH
RSCAN0	Transmit buffer ID register 2	RSCAN0TMID2	<RSCAN0_base> + 1020H
RSCAN0	Transmit buffer pointer register 2	RSCAN0TMRPTR2	<RSCAN0_base> + 1024H
RSCAN0	Transmit buffer data field 0 register 2	RSCAN0TMDF02	<RSCAN0_base> + 1028H
RSCAN0	Transmit buffer data field 1 register 2	RSCAN0TMDF12	<RSCAN0_base> + 102CH
RSCAN0	Transmit buffer ID register 3	RSCAN0TMID3	<RSCAN0_base> + 1030H
RSCAN0	Transmit buffer pointer register 3	RSCAN0TMRPTR3	<RSCAN0_base> + 1034H
RSCAN0	Transmit buffer data field 0 register 3	RSCAN0TMDF03	<RSCAN0_base> + 1038H
RSCAN0	Transmit buffer data field 1 register 3	RSCAN0TMDF13	<RSCAN0_base> + 103CH
RSCAN0	Transmit buffer ID register 4	RSCAN0TMID4	<RSCAN0_base> + 1040H
RSCAN0	Transmit buffer pointer register 4	RSCAN0TMRPTR4	<RSCAN0_base> + 1044H
RSCAN0	Transmit buffer data field 0 register 4	RSCAN0TMDF04	<RSCAN0_base> + 1048H
RSCAN0	Transmit buffer data field 1 register 4	RSCAN0TMDF14	<RSCAN0_base> + 104CH
RSCAN0	Transmit buffer ID register 5	RSCAN0TMID5	<RSCAN0_base> + 1050H
RSCAN0	Transmit buffer pointer register 5	RSCAN0TMRPTR5	<RSCAN0_base> + 1054H
RSCAN0	Transmit buffer data field 0 register 5	RSCAN0TMDF05	<RSCAN0_base> + 1058H
RSCAN0	Transmit buffer data field 1 register 5	RSCAN0TMDF15	<RSCAN0_base> + 105CH
RSCAN0	Transmit buffer ID register 6	RSCAN0TMID6	<RSCAN0_base> + 1060H
RSCAN0	Transmit buffer pointer register 6	RSCAN0TMRPTR6	<RSCAN0_base> + 1064H
RSCAN0	Transmit buffer data field 0 register 6	RSCAN0TMDF06	<RSCAN0_base> + 1068H
RSCAN0	Transmit buffer data field 1 register 6	RSCAN0TMDF16	<RSCAN0_base> + 106CH
RSCAN0	Transmit buffer ID register 7	RSCAN0TMID7	<RSCAN0_base> + 1070H
RSCAN0	Transmit buffer pointer register 7	RSCAN0TMRPTR7	<RSCAN0_base> + 1074H
RSCAN0	Transmit buffer data field 0 register 7	RSCAN0TMDF07	<RSCAN0_base> + 1078H
RSCAN0	Transmit buffer data field 1 register 7	RSCAN0TMDF17	<RSCAN0_base> + 107CH
RSCAN0	Transmit buffer ID register 8	RSCAN0TMID8	<RSCAN0_base> + 1080H
RSCAN0	Transmit buffer pointer register 8	RSCAN0TMRPTR8	<RSCAN0_base> + 1084H
RSCAN0	Transmit buffer data field 0 register 8	RSCAN0TMDF08	<RSCAN0_base> + 1088H
RSCAN0	Transmit buffer data field 1 register 8	RSCAN0TMDF18	<RSCAN0_base> + 108CH
RSCAN0	Transmit buffer ID register 9	RSCAN0TMID9	<RSCAN0_base> + 1090H
RSCAN0	Transmit buffer pointer register 9	RSCAN0TMRPTR9	<RSCAN0_base> + 1094H
RSCAN0	Transmit buffer data field 0 register 9	RSCAN0TMDF09	<RSCAN0_base> + 1098H
RSCAN0	Transmit buffer data field 1 register 9	RSCAN0TMDF19	<RSCAN0_base> + 109CH
RSCAN0	Transmit buffer ID register 10	RSCAN0TMID10	<RSCAN0_base> + 10A0H
RSCAN0	Transmit buffer pointer register 10	RSCAN0TMRPTR10	<RSCAN0_base> + 10A4H

Table 19.12 List of Registers (23/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 10	RSCAN0TMDF010	<RSCAN0_base> + 10A8H
RSCAN0	Transmit buffer data field 1 register 10	RSCAN0TMDF110	<RSCAN0_base> + 10ACH
RSCAN0	Transmit buffer ID register 11	RSCAN0TMID11	<RSCAN0_base> + 10B0H
RSCAN0	Transmit buffer pointer register 11	RSCAN0TMRPTR11	<RSCAN0_base> + 10B4H
RSCAN0	Transmit buffer data field 0 register 11	RSCAN0TMDF011	<RSCAN0_base> + 10B8H
RSCAN0	Transmit buffer data field 1 register 11	RSCAN0TMDF111	<RSCAN0_base> + 10BCH
RSCAN0	Transmit buffer ID register 12	RSCAN0TMID12	<RSCAN0_base> + 10C0H
RSCAN0	Transmit buffer pointer register 12	RSCAN0TMRPTR12	<RSCAN0_base> + 10C4H
RSCAN0	Transmit buffer data field 0 register 12	RSCAN0TMDF012	<RSCAN0_base> + 10C8H
RSCAN0	Transmit buffer data field 1 register 12	RSCAN0TMDF112	<RSCAN0_base> + 10CCH
RSCAN0	Transmit buffer ID register 13	RSCAN0TMID13	<RSCAN0_base> + 10D0H
RSCAN0	Transmit buffer pointer register 13	RSCAN0TMRPTR13	<RSCAN0_base> + 10D4H
RSCAN0	Transmit buffer data field 0 register 13	RSCAN0TMDF013	<RSCAN0_base> + 10D8H
RSCAN0	Transmit buffer data field 1 register 13	RSCAN0TMDF113	<RSCAN0_base> + 10DCH
RSCAN0	Transmit buffer ID register 14	RSCAN0TMID14	<RSCAN0_base> + 10E0H
RSCAN0	Transmit buffer pointer register 14	RSCAN0TMRPTR14	<RSCAN0_base> + 10E4H
RSCAN0	Transmit buffer data field 0 register 14	RSCAN0TMDF014	<RSCAN0_base> + 10E8H
RSCAN0	Transmit buffer data field 1 register 14	RSCAN0TMDF114	<RSCAN0_base> + 10ECH
RSCAN0	Transmit buffer ID register 15	RSCAN0TMID15	<RSCAN0_base> + 10F0H
RSCAN0	Transmit buffer pointer register 15	RSCAN0TMRPTR15	<RSCAN0_base> + 10F4H
RSCAN0	Transmit buffer data field 0 register 15	RSCAN0TMDF015	<RSCAN0_base> + 10F8H
RSCAN0	Transmit buffer data field 1 register 15	RSCAN0TMDF115	<RSCAN0_base> + 10FCH
RSCAN0	Transmit buffer ID register 16	RSCAN0TMID16	<RSCAN0_base> + 1100H
RSCAN0	Transmit buffer pointer register 16	RSCAN0TMRPTR16	<RSCAN0_base> + 1104H
RSCAN0	Transmit buffer data field 0 register 16	RSCAN0TMDF016	<RSCAN0_base> + 1108H
RSCAN0	Transmit buffer data field 1 register 16	RSCAN0TMDF116	<RSCAN0_base> + 110CH
RSCAN0	Transmit buffer ID register 17	RSCAN0TMID17	<RSCAN0_base> + 1110H
RSCAN0	Transmit buffer pointer register 17	RSCAN0TMRPTR17	<RSCAN0_base> + 1114H
RSCAN0	Transmit buffer data field 0 register 17	RSCAN0TMDF017	<RSCAN0_base> + 1118H
RSCAN0	Transmit buffer data field 1 register 17	RSCAN0TMDF117	<RSCAN0_base> + 111CH
RSCAN0	Transmit buffer ID register 18	RSCAN0TMID18	<RSCAN0_base> + 1120H
RSCAN0	Transmit buffer pointer register 18	RSCAN0TMRPTR18	<RSCAN0_base> + 1124H
RSCAN0	Transmit buffer data field 0 register 18	RSCAN0TMDF018	<RSCAN0_base> + 1128H
RSCAN0	Transmit buffer data field 1 register 18	RSCAN0TMDF118	<RSCAN0_base> + 112CH
RSCAN0	Transmit buffer ID register 19	RSCAN0TMID19	<RSCAN0_base> + 1130H
RSCAN0	Transmit buffer pointer register 19	RSCAN0TMRPTR19	<RSCAN0_base> + 1134H
RSCAN0	Transmit buffer data field 0 register 19	RSCAN0TMDF019	<RSCAN0_base> + 1138H
RSCAN0	Transmit buffer data field 1 register 19	RSCAN0TMDF119	<RSCAN0_base> + 113CH
RSCAN0	Transmit buffer ID register 20	RSCAN0TMID20	<RSCAN0_base> + 1140H
RSCAN0	Transmit buffer pointer register 20	RSCAN0TMRPTR20	<RSCAN0_base> + 1144H
RSCAN0	Transmit buffer data field 0 register 20	RSCAN0TMDF020	<RSCAN0_base> + 1148H
RSCAN0	Transmit buffer data field 1 register 20	RSCAN0TMDF120	<RSCAN0_base> + 114CH
RSCAN0	Transmit buffer ID register 21	RSCAN0TMID21	<RSCAN0_base> + 1150H
RSCAN0	Transmit buffer pointer register 21	RSCAN0TMRPTR21	<RSCAN0_base> + 1154H

Table 19.12 List of Registers (24/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 21	RSCAN0TMDF021	<RSCAN0_base> + 1158H
RSCAN0	Transmit buffer data field 1 register 21	RSCAN0TMDF121	<RSCAN0_base> + 115CH
RSCAN0	Transmit buffer ID register 22	RSCAN0TMID22	<RSCAN0_base> + 1160H
RSCAN0	Transmit buffer pointer register 22	RSCAN0TMRPTR22	<RSCAN0_base> + 1164H
RSCAN0	Transmit buffer data field 0 register 22	RSCAN0TMDF022	<RSCAN0_base> + 1168H
RSCAN0	Transmit buffer data field 1 register 22	RSCAN0TMDF122	<RSCAN0_base> + 116CH
RSCAN0	Transmit buffer ID register 23	RSCAN0TMID23	<RSCAN0_base> + 1170H
RSCAN0	Transmit buffer pointer register 23	RSCAN0TMRPTR23	<RSCAN0_base> + 1174H
RSCAN0	Transmit buffer data field 0 register 23	RSCAN0TMDF023	<RSCAN0_base> + 1178H
RSCAN0	Transmit buffer data field 1 register 23	RSCAN0TMDF123	<RSCAN0_base> + 117CH
RSCAN0	Transmit buffer ID register 24	RSCAN0TMID24	<RSCAN0_base> + 1180H
RSCAN0	Transmit buffer pointer register 24	RSCAN0TMRPTR24	<RSCAN0_base> + 1184H
RSCAN0	Transmit buffer data field 0 register 24	RSCAN0TMDF024	<RSCAN0_base> + 1188H
RSCAN0	Transmit buffer data field 1 register 24	RSCAN0TMDF124	<RSCAN0_base> + 118CH
RSCAN0	Transmit buffer ID register 25	RSCAN0TMID25	<RSCAN0_base> + 1190H
RSCAN0	Transmit buffer pointer register 25	RSCAN0TMRPTR25	<RSCAN0_base> + 1194H
RSCAN0	Transmit buffer data field 0 register 25	RSCAN0TMDF025	<RSCAN0_base> + 1198H
RSCAN0	Transmit buffer data field 1 register 25	RSCAN0TMDF125	<RSCAN0_base> + 119CH
RSCAN0	Transmit buffer ID register 26	RSCAN0TMID26	<RSCAN0_base> + 11A0H
RSCAN0	Transmit buffer pointer register 26	RSCAN0TMRPTR26	<RSCAN0_base> + 11A4H
RSCAN0	Transmit buffer data field 0 register 26	RSCAN0TMDF026	<RSCAN0_base> + 11A8H
RSCAN0	Transmit buffer data field 1 register 26	RSCAN0TMDF126	<RSCAN0_base> + 11ACH
RSCAN0	Transmit buffer ID register 27	RSCAN0TMID27	<RSCAN0_base> + 11B0H
RSCAN0	Transmit buffer pointer register 27	RSCAN0TMRPTR27	<RSCAN0_base> + 11B4H
RSCAN0	Transmit buffer data field 0 register 27	RSCAN0TMDF027	<RSCAN0_base> + 11B8H
RSCAN0	Transmit buffer data field 1 register 27	RSCAN0TMDF127	<RSCAN0_base> + 11BCH
RSCAN0	Transmit buffer ID register 28	RSCAN0TMID28	<RSCAN0_base> + 11C0H
RSCAN0	Transmit buffer pointer register 28	RSCAN0TMRPTR28	<RSCAN0_base> + 11C4H
RSCAN0	Transmit buffer data field 0 register 28	RSCAN0TMDF028	<RSCAN0_base> + 11C8H
RSCAN0	Transmit buffer data field 1 register 28	RSCAN0TMDF128	<RSCAN0_base> + 11CCH
RSCAN0	Transmit buffer ID register 29	RSCAN0TMID29	<RSCAN0_base> + 11D0H
RSCAN0	Transmit buffer pointer register 29	RSCAN0TMRPTR29	<RSCAN0_base> + 11D4H
RSCAN0	Transmit buffer data field 0 register 29	RSCAN0TMDF029	<RSCAN0_base> + 11D8H
RSCAN0	Transmit buffer data field 1 register 29	RSCAN0TMDF129	<RSCAN0_base> + 11DCH
RSCAN0	Transmit buffer ID register 30	RSCAN0TMID30	<RSCAN0_base> + 11E0H
RSCAN0	Transmit buffer pointer register 30	RSCAN0TMRPTR30	<RSCAN0_base> + 11E4H
RSCAN0	Transmit buffer data field 0 register 30	RSCAN0TMDF030	<RSCAN0_base> + 11E8H
RSCAN0	Transmit buffer data field 1 register 30	RSCAN0TMDF130	<RSCAN0_base> + 11ECH
RSCAN0	Transmit buffer ID register 31	RSCAN0TMID31	<RSCAN0_base> + 11F0H
RSCAN0	Transmit buffer pointer register 31	RSCAN0TMRPTR31	<RSCAN0_base> + 11F4H
RSCAN0	Transmit buffer data field 0 register 31	RSCAN0TMDF031	<RSCAN0_base> + 11F8H
RSCAN0	Transmit buffer data field 1 register 31	RSCAN0TMDF131	<RSCAN0_base> + 11FCH
RSCAN0	Transmit buffer ID register 32	RSCAN0TMID32	<RSCAN0_base> + 1200H
RSCAN0	Transmit buffer pointer register 32	RSCAN0TMRPTR32	<RSCAN0_base> + 1204H

Table 19.12 List of Registers (25/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 32	RSCAN0TMDF032	<RSCAN0_base> + 1208 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 32	RSCAN0TMDF132	<RSCAN0_base> + 120C <sub>H</sub>
RSCAN0	Transmit buffer ID register 33	RSCAN0TMID33	<RSCAN0_base> + 1210 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 33	RSCAN0TMPTR33	<RSCAN0_base> + 1214 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 33	RSCAN0TMDF033	<RSCAN0_base> + 1218 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 33	RSCAN0TMDF133	<RSCAN0_base> + 121C <sub>H</sub>
RSCAN0	Transmit buffer ID register 34	RSCAN0TMID34	<RSCAN0_base> + 1220 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 34	RSCAN0TMPTR34	<RSCAN0_base> + 1224 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 34	RSCAN0TMDF034	<RSCAN0_base> + 1228 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 34	RSCAN0TMDF134	<RSCAN0_base> + 122C <sub>H</sub>
RSCAN0	Transmit buffer ID register 35	RSCAN0TMID35	<RSCAN0_base> + 1230 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 35	RSCAN0TMPTR35	<RSCAN0_base> + 1234 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 35	RSCAN0TMDF035	<RSCAN0_base> + 1238 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 35	RSCAN0TMDF135	<RSCAN0_base> + 123C <sub>H</sub>
RSCAN0	Transmit buffer ID register 36	RSCAN0TMID36	<RSCAN0_base> + 1240 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 36	RSCAN0TMPTR36	<RSCAN0_base> + 1244 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 36	RSCAN0TMDF036	<RSCAN0_base> + 1248 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 36	RSCAN0TMDF136	<RSCAN0_base> + 124C <sub>H</sub>
RSCAN0	Transmit buffer ID register 37	RSCAN0TMID37	<RSCAN0_base> + 1250 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 37	RSCAN0TMPTR37	<RSCAN0_base> + 1254 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 37	RSCAN0TMDF037	<RSCAN0_base> + 1258 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 37	RSCAN0TMDF137	<RSCAN0_base> + 125C <sub>H</sub>
RSCAN0	Transmit buffer ID register 38	RSCAN0TMID38	<RSCAN0_base> + 1260 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 38	RSCAN0TMPTR38	<RSCAN0_base> + 1264 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 38	RSCAN0TMDF038	<RSCAN0_base> + 1268 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 38	RSCAN0TMDF138	<RSCAN0_base> + 126C <sub>H</sub>
RSCAN0	Transmit buffer ID register 39	RSCAN0TMID39	<RSCAN0_base> + 1270 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 39	RSCAN0TMPTR39	<RSCAN0_base> + 1274 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 39	RSCAN0TMDF039	<RSCAN0_base> + 1278 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 39	RSCAN0TMDF139	<RSCAN0_base> + 127C <sub>H</sub>
RSCAN0	Transmit buffer ID register 40	RSCAN0TMID40	<RSCAN0_base> + 1280 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 40	RSCAN0TMPTR40	<RSCAN0_base> + 1284 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 40	RSCAN0TMDF040	<RSCAN0_base> + 1288 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 40	RSCAN0TMDF140	<RSCAN0_base> + 128C <sub>H</sub>
RSCAN0	Transmit buffer ID register 41	RSCAN0TMID41	<RSCAN0_base> + 1290 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 41	RSCAN0TMPTR41	<RSCAN0_base> + 1294 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 41	RSCAN0TMDF041	<RSCAN0_base> + 1298 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 41	RSCAN0TMDF141	<RSCAN0_base> + 129C <sub>H</sub>
RSCAN0	Transmit buffer ID register 42	RSCAN0TMID42	<RSCAN0_base> + 12A0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 42	RSCAN0TMPTR42	<RSCAN0_base> + 12A4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 42	RSCAN0TMDF042	<RSCAN0_base> + 12A8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 42	RSCAN0TMDF142	<RSCAN0_base> + 12AC <sub>H</sub>
RSCAN0	Transmit buffer ID register 43	RSCAN0TMID43	<RSCAN0_base> + 12B0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 43	RSCAN0TMPTR43	<RSCAN0_base> + 12B4 <sub>H</sub>

Table 19.12 List of Registers (26/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 43	RSCAN0TMDF043	<RSCAN0_base> + 12B8H
RSCAN0	Transmit buffer data field 1 register 43	RSCAN0TMDF143	<RSCAN0_base> + 12BCH
RSCAN0	Transmit buffer ID register 44	RSCAN0TMID44	<RSCAN0_base> + 12C0H
RSCAN0	Transmit buffer pointer register 44	RSCAN0TMRPTR44	<RSCAN0_base> + 12C4H
RSCAN0	Transmit buffer data field 0 register 44	RSCAN0TMDF044	<RSCAN0_base> + 12C8H
RSCAN0	Transmit buffer data field 1 register 44	RSCAN0TMDF144	<RSCAN0_base> + 12CCH
RSCAN0	Transmit buffer ID register 45	RSCAN0TMID45	<RSCAN0_base> + 12D0H
RSCAN0	Transmit buffer pointer register 45	RSCAN0TMRPTR45	<RSCAN0_base> + 12D4H
RSCAN0	Transmit buffer data field 0 register 45	RSCAN0TMDF045	<RSCAN0_base> + 12D8H
RSCAN0	Transmit buffer data field 1 register 45	RSCAN0TMDF145	<RSCAN0_base> + 12DCH
RSCAN0	Transmit buffer ID register 46	RSCAN0TMID46	<RSCAN0_base> + 12E0H
RSCAN0	Transmit buffer pointer register 46	RSCAN0TMRPTR46	<RSCAN0_base> + 12E4H
RSCAN0	Transmit buffer data field 0 register 46	RSCAN0TMDF046	<RSCAN0_base> + 12E8H
RSCAN0	Transmit buffer data field 1 register 46	RSCAN0TMDF146	<RSCAN0_base> + 12ECH
RSCAN0	Transmit buffer ID register 47	RSCAN0TMID47	<RSCAN0_base> + 12F0H
RSCAN0	Transmit buffer pointer register 47	RSCAN0TMRPTR47	<RSCAN0_base> + 12F4H
RSCAN0	Transmit buffer data field 0 register 47	RSCAN0TMDF047	<RSCAN0_base> + 12F8H
RSCAN0	Transmit buffer data field 1 register 47	RSCAN0TMDF147	<RSCAN0_base> + 12FCH
RSCAN0	Transmit buffer ID register 48	RSCAN0TMID48	<RSCAN0_base> + 1300H
RSCAN0	Transmit buffer pointer register 48	RSCAN0TMRPTR48	<RSCAN0_base> + 1304H
RSCAN0	Transmit buffer data field 0 register 48	RSCAN0TMDF048	<RSCAN0_base> + 1308H
RSCAN0	Transmit buffer data field 1 register 48	RSCAN0TMDF148	<RSCAN0_base> + 130CH
RSCAN0	Transmit buffer ID register 49	RSCAN0TMID49	<RSCAN0_base> + 1310H
RSCAN0	Transmit buffer pointer register 49	RSCAN0TMRPTR49	<RSCAN0_base> + 1314H
RSCAN0	Transmit buffer data field 0 register 49	RSCAN0TMDF049	<RSCAN0_base> + 1318H
RSCAN0	Transmit buffer data field 1 register 49	RSCAN0TMDF149	<RSCAN0_base> + 131CH
RSCAN0	Transmit buffer ID register 50	RSCAN0TMID50	<RSCAN0_base> + 1320H
RSCAN0	Transmit buffer pointer register 50	RSCAN0TMRPTR50	<RSCAN0_base> + 1324H
RSCAN0	Transmit buffer data field 0 register 50	RSCAN0TMDF050	<RSCAN0_base> + 1328H
RSCAN0	Transmit buffer data field 1 register 50	RSCAN0TMDF150	<RSCAN0_base> + 132CH
RSCAN0	Transmit buffer ID register 51	RSCAN0TMID51	<RSCAN0_base> + 1330H
RSCAN0	Transmit buffer pointer register 51	RSCAN0TMRPTR51	<RSCAN0_base> + 1334H
RSCAN0	Transmit buffer data field 0 register 51	RSCAN0TMDF051	<RSCAN0_base> + 1338H
RSCAN0	Transmit buffer data field 1 register 51	RSCAN0TMDF151	<RSCAN0_base> + 133CH
RSCAN0	Transmit buffer ID register 52	RSCAN0TMID52	<RSCAN0_base> + 1340H
RSCAN0	Transmit buffer pointer register 52	RSCAN0TMRPTR52	<RSCAN0_base> + 1344H
RSCAN0	Transmit buffer data field 0 register 52	RSCAN0TMDF052	<RSCAN0_base> + 1348H
RSCAN0	Transmit buffer data field 1 register 52	RSCAN0TMDF152	<RSCAN0_base> + 134CH
RSCAN0	Transmit buffer ID register 53	RSCAN0TMID53	<RSCAN0_base> + 1350H
RSCAN0	Transmit buffer pointer register 53	RSCAN0TMRPTR53	<RSCAN0_base> + 1354H
RSCAN0	Transmit buffer data field 0 register 53	RSCAN0TMDF053	<RSCAN0_base> + 1358H
RSCAN0	Transmit buffer data field 1 register 53	RSCAN0TMDF153	<RSCAN0_base> + 135CH
RSCAN0	Transmit buffer ID register 54	RSCAN0TMID54	<RSCAN0_base> + 1360H
RSCAN0	Transmit buffer pointer register 54	RSCAN0TMRPTR54	<RSCAN0_base> + 1364H

Table 19.12 List of Registers (27/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 54	RSCAN0TMDF054	<RSCAN0_base> + 1368 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 54	RSCAN0TMDF154	<RSCAN0_base> + 136C <sub>H</sub>
RSCAN0	Transmit buffer ID register 55	RSCAN0TMID55	<RSCAN0_base> + 1370 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 55	RSCAN0TMPTR55	<RSCAN0_base> + 1374 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 55	RSCAN0TMDF055	<RSCAN0_base> + 1378 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 55	RSCAN0TMDF155	<RSCAN0_base> + 137C <sub>H</sub>
RSCAN0	Transmit buffer ID register 56	RSCAN0TMID56	<RSCAN0_base> + 1380 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 56	RSCAN0TMPTR56	<RSCAN0_base> + 1384 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 56	RSCAN0TMDF056	<RSCAN0_base> + 1388 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 56	RSCAN0TMDF156	<RSCAN0_base> + 138C <sub>H</sub>
RSCAN0	Transmit buffer ID register 57	RSCAN0TMID57	<RSCAN0_base> + 1390 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 57	RSCAN0TMPTR57	<RSCAN0_base> + 1394 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 57	RSCAN0TMDF057	<RSCAN0_base> + 1398 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 57	RSCAN0TMDF157	<RSCAN0_base> + 139C <sub>H</sub>
RSCAN0	Transmit buffer ID register 58	RSCAN0TMID58	<RSCAN0_base> + 13A0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 58	RSCAN0TMPTR58	<RSCAN0_base> + 13A4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 58	RSCAN0TMDF058	<RSCAN0_base> + 13A8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 58	RSCAN0TMDF158	<RSCAN0_base> + 13AC <sub>H</sub>
RSCAN0	Transmit buffer ID register 59	RSCAN0TMID59	<RSCAN0_base> + 13B0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 59	RSCAN0TMPTR59	<RSCAN0_base> + 13B4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 59	RSCAN0TMDF059	<RSCAN0_base> + 13B8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 59	RSCAN0TMDF159	<RSCAN0_base> + 13BC <sub>H</sub>
RSCAN0	Transmit buffer ID register 60	RSCAN0TMID60	<RSCAN0_base> + 13C0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 60	RSCAN0TMPTR60	<RSCAN0_base> + 13C4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 60	RSCAN0TMDF060	<RSCAN0_base> + 13C8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 60	RSCAN0TMDF160	<RSCAN0_base> + 13CC <sub>H</sub>
RSCAN0	Transmit buffer ID register 61	RSCAN0TMID61	<RSCAN0_base> + 13D0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 61	RSCAN0TMPTR61	<RSCAN0_base> + 13D4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 61	RSCAN0TMDF061	<RSCAN0_base> + 13D8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 61	RSCAN0TMDF161	<RSCAN0_base> + 13DC <sub>H</sub>
RSCAN0	Transmit buffer ID register 62	RSCAN0TMID62	<RSCAN0_base> + 13E0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 62	RSCAN0TMPTR62	<RSCAN0_base> + 13E4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 62	RSCAN0TMDF062	<RSCAN0_base> + 13E8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 62	RSCAN0TMDF162	<RSCAN0_base> + 13EC <sub>H</sub>
RSCAN0	Transmit buffer ID register 63	RSCAN0TMID63	<RSCAN0_base> + 13F0 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 63	RSCAN0TMPTR63	<RSCAN0_base> + 13F4 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 63	RSCAN0TMDF063	<RSCAN0_base> + 13F8 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 63	RSCAN0TMDF163	<RSCAN0_base> + 13FC <sub>H</sub>
RSCAN0	Transmit buffer ID register 64	RSCAN0TMID64	<RSCAN0_base> + 1400 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 64	RSCAN0TMPTR64	<RSCAN0_base> + 1404 <sub>H</sub>
RSCAN0	Transmit buffer data field 0 register 64	RSCAN0TMDF064	<RSCAN0_base> + 1408 <sub>H</sub>
RSCAN0	Transmit buffer data field 1 register 64	RSCAN0TMDF164	<RSCAN0_base> + 140C <sub>H</sub>
RSCAN0	Transmit buffer ID register 65	RSCAN0TMID65	<RSCAN0_base> + 1410 <sub>H</sub>
RSCAN0	Transmit buffer pointer register 65	RSCAN0TMPTR65	<RSCAN0_base> + 1414 <sub>H</sub>

Table 19.12 List of Registers (28/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 65	RSCAN0TMDF065	<RSCAN0_base> + 1418H
RSCAN0	Transmit buffer data field 1 register 65	RSCAN0TMDF165	<RSCAN0_base> + 141CH
RSCAN0	Transmit buffer ID register 66	RSCAN0TMID66	<RSCAN0_base> + 1420H
RSCAN0	Transmit buffer pointer register 66	RSCAN0TMRPTR66	<RSCAN0_base> + 1424H
RSCAN0	Transmit buffer data field 0 register 66	RSCAN0TMDF066	<RSCAN0_base> + 1428H
RSCAN0	Transmit buffer data field 1 register 66	RSCAN0TMDF166	<RSCAN0_base> + 142CH
RSCAN0	Transmit buffer ID register 67	RSCAN0TMID67	<RSCAN0_base> + 1430H
RSCAN0	Transmit buffer pointer register 67	RSCAN0TMRPTR67	<RSCAN0_base> + 1434H
RSCAN0	Transmit buffer data field 0 register 67	RSCAN0TMDF067	<RSCAN0_base> + 1438H
RSCAN0	Transmit buffer data field 1 register 67	RSCAN0TMDF167	<RSCAN0_base> + 143CH
RSCAN0	Transmit buffer ID register 68	RSCAN0TMID68	<RSCAN0_base> + 1440H
RSCAN0	Transmit buffer pointer register 68	RSCAN0TMRPTR68	<RSCAN0_base> + 1444H
RSCAN0	Transmit buffer data field 0 register 68	RSCAN0TMDF068	<RSCAN0_base> + 1448H
RSCAN0	Transmit buffer data field 1 register 68	RSCAN0TMDF168	<RSCAN0_base> + 144CH
RSCAN0	Transmit buffer ID register 69	RSCAN0TMID69	<RSCAN0_base> + 1450H
RSCAN0	Transmit buffer pointer register 69	RSCAN0TMRPTR69	<RSCAN0_base> + 1454H
RSCAN0	Transmit buffer data field 0 register 69	RSCAN0TMDF069	<RSCAN0_base> + 1458H
RSCAN0	Transmit buffer data field 1 register 69	RSCAN0TMDF169	<RSCAN0_base> + 145CH
RSCAN0	Transmit buffer ID register 70	RSCAN0TMID70	<RSCAN0_base> + 1460H
RSCAN0	Transmit buffer pointer register 70	RSCAN0TMRPTR70	<RSCAN0_base> + 1464H
RSCAN0	Transmit buffer data field 0 register 70	RSCAN0TMDF070	<RSCAN0_base> + 1468H
RSCAN0	Transmit buffer data field 1 register 70	RSCAN0TMDF170	<RSCAN0_base> + 146CH
RSCAN0	Transmit buffer ID register 71	RSCAN0TMID71	<RSCAN0_base> + 1470H
RSCAN0	Transmit buffer pointer register 71	RSCAN0TMRPTR71	<RSCAN0_base> + 1474H
RSCAN0	Transmit buffer data field 0 register 71	RSCAN0TMDF071	<RSCAN0_base> + 1478H
RSCAN0	Transmit buffer data field 1 register 71	RSCAN0TMDF171	<RSCAN0_base> + 147CH
RSCAN0	Transmit buffer ID register 72	RSCAN0TMID72	<RSCAN0_base> + 1480H
RSCAN0	Transmit buffer pointer register 72	RSCAN0TMRPTR72	<RSCAN0_base> + 1484H
RSCAN0	Transmit buffer data field 0 register 72	RSCAN0TMDF072	<RSCAN0_base> + 1488H
RSCAN0	Transmit buffer data field 1 register 72	RSCAN0TMDF172	<RSCAN0_base> + 148CH
RSCAN0	Transmit buffer ID register 73	RSCAN0TMID73	<RSCAN0_base> + 1490H
RSCAN0	Transmit buffer pointer register 73	RSCAN0TMRPTR73	<RSCAN0_base> + 1494H
RSCAN0	Transmit buffer data field 0 register 73	RSCAN0TMDF073	<RSCAN0_base> + 1498H
RSCAN0	Transmit buffer data field 1 register 73	RSCAN0TMDF173	<RSCAN0_base> + 149CH
RSCAN0	Transmit buffer ID register 74	RSCAN0TMID74	<RSCAN0_base> + 14A0H
RSCAN0	Transmit buffer pointer register 74	RSCAN0TMRPTR74	<RSCAN0_base> + 14A4H
RSCAN0	Transmit buffer data field 0 register 74	RSCAN0TMDF074	<RSCAN0_base> + 14A8H
RSCAN0	Transmit buffer data field 1 register 74	RSCAN0TMDF174	<RSCAN0_base> + 14ACH
RSCAN0	Transmit buffer ID register 75	RSCAN0TMID75	<RSCAN0_base> + 14B0H
RSCAN0	Transmit buffer pointer register 75	RSCAN0TMRPTR75	<RSCAN0_base> + 14B4H
RSCAN0	Transmit buffer data field 0 register 75	RSCAN0TMDF075	<RSCAN0_base> + 14B8H
RSCAN0	Transmit buffer data field 1 register 75	RSCAN0TMDF175	<RSCAN0_base> + 14BCH
RSCAN0	Transmit buffer ID register 76	RSCAN0TMID76	<RSCAN0_base> + 14C0H
RSCAN0	Transmit buffer pointer register 76	RSCAN0TMRPTR76	<RSCAN0_base> + 14C4H

Table 19.12 List of Registers (29/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 76	RSCAN0TMDF076	<RSCAN0_base> + 14C8H
RSCAN0	Transmit buffer data field 1 register 76	RSCAN0TMDF176	<RSCAN0_base> + 14CCH
RSCAN0	Transmit buffer ID register 77	RSCAN0TMID77	<RSCAN0_base> + 14D0H
RSCAN0	Transmit buffer pointer register 77	RSCAN0TMRP77	<RSCAN0_base> + 14D4H
RSCAN0	Transmit buffer data field 0 register 77	RSCAN0TMDF077	<RSCAN0_base> + 14D8H
RSCAN0	Transmit buffer data field 1 register 77	RSCAN0TMDF177	<RSCAN0_base> + 14DCH
RSCAN0	Transmit buffer ID register 78	RSCAN0TMID78	<RSCAN0_base> + 14E0H
RSCAN0	Transmit buffer pointer register 78	RSCAN0TMRP78	<RSCAN0_base> + 14E4H
RSCAN0	Transmit buffer data field 0 register 78	RSCAN0TMDF078	<RSCAN0_base> + 14E8H
RSCAN0	Transmit buffer data field 1 register 78	RSCAN0TMDF178	<RSCAN0_base> + 14ECH
RSCAN0	Transmit buffer ID register 79	RSCAN0TMID79	<RSCAN0_base> + 14F0H
RSCAN0	Transmit buffer pointer register 79	RSCAN0TMRP79	<RSCAN0_base> + 14F4H
RSCAN0	Transmit buffer data field 0 register 79	RSCAN0TMDF079	<RSCAN0_base> + 14F8H
RSCAN0	Transmit buffer data field 1 register 79	RSCAN0TMDF179	<RSCAN0_base> + 14FCH
RSCAN0	Transmit buffer ID register 80	RSCAN0TMID80	<RSCAN0_base> + 1500H
RSCAN0	Transmit buffer pointer register 80	RSCAN0TMRP80	<RSCAN0_base> + 1504H
RSCAN0	Transmit buffer data field 0 register 80	RSCAN0TMDF080	<RSCAN0_base> + 1508H
RSCAN0	Transmit buffer data field 1 register 80	RSCAN0TMDF180	<RSCAN0_base> + 150CH
RSCAN0	Transmit buffer ID register 81	RSCAN0TMID81	<RSCAN0_base> + 1510H
RSCAN0	Transmit buffer pointer register 81	RSCAN0TMRP81	<RSCAN0_base> + 1514H
RSCAN0	Transmit buffer data field 0 register 81	RSCAN0TMDF081	<RSCAN0_base> + 1518H
RSCAN0	Transmit buffer data field 1 register 81	RSCAN0TMDF181	<RSCAN0_base> + 151CH
RSCAN0	Transmit buffer ID register 82	RSCAN0TMID82	<RSCAN0_base> + 1520H
RSCAN0	Transmit buffer pointer register 82	RSCAN0TMRP82	<RSCAN0_base> + 1524H
RSCAN0	Transmit buffer data field 0 register 82	RSCAN0TMDF082	<RSCAN0_base> + 1528H
RSCAN0	Transmit buffer data field 1 register 82	RSCAN0TMDF182	<RSCAN0_base> + 152CH
RSCAN0	Transmit buffer ID register 83	RSCAN0TMID83	<RSCAN0_base> + 1530H
RSCAN0	Transmit buffer pointer register 83	RSCAN0TMRP83	<RSCAN0_base> + 1534H
RSCAN0	Transmit buffer data field 0 register 83	RSCAN0TMDF083	<RSCAN0_base> + 1538H
RSCAN0	Transmit buffer data field 1 register 83	RSCAN0TMDF183	<RSCAN0_base> + 153CH
RSCAN0	Transmit buffer ID register 84	RSCAN0TMID84	<RSCAN0_base> + 1540H
RSCAN0	Transmit buffer pointer register 84	RSCAN0TMRP84	<RSCAN0_base> + 1544H
RSCAN0	Transmit buffer data field 0 register 84	RSCAN0TMDF084	<RSCAN0_base> + 1548H
RSCAN0	Transmit buffer data field 1 register 84	RSCAN0TMDF184	<RSCAN0_base> + 154CH
RSCAN0	Transmit buffer ID register 85	RSCAN0TMID85	<RSCAN0_base> + 1550H
RSCAN0	Transmit buffer pointer register 85	RSCAN0TMRP85	<RSCAN0_base> + 1554H
RSCAN0	Transmit buffer data field 0 register 85	RSCAN0TMDF085	<RSCAN0_base> + 1558H
RSCAN0	Transmit buffer data field 1 register 85	RSCAN0TMDF185	<RSCAN0_base> + 155CH
RSCAN0	Transmit buffer ID register 86	RSCAN0TMID86	<RSCAN0_base> + 1560H
RSCAN0	Transmit buffer pointer register 86	RSCAN0TMRP86	<RSCAN0_base> + 1564H
RSCAN0	Transmit buffer data field 0 register 86	RSCAN0TMDF086	<RSCAN0_base> + 1568H
RSCAN0	Transmit buffer data field 1 register 86	RSCAN0TMDF186	<RSCAN0_base> + 156CH
RSCAN0	Transmit buffer ID register 87	RSCAN0TMID87	<RSCAN0_base> + 1570H
RSCAN0	Transmit buffer pointer register 87	RSCAN0TMRP87	<RSCAN0_base> + 1574H

Table 19.12 List of Registers (30/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 87	RSCAN0TMDF087	<RSCAN0_base> + 1578H
RSCAN0	Transmit buffer data field 1 register 87	RSCAN0TMDF187	<RSCAN0_base> + 157CH
RSCAN0	Transmit buffer ID register 88	RSCAN0TMID88	<RSCAN0_base> + 1580H
RSCAN0	Transmit buffer pointer register 88	RSCAN0TMRPTR88	<RSCAN0_base> + 1584H
RSCAN0	Transmit buffer data field 0 register 88	RSCAN0TMDF088	<RSCAN0_base> + 1588H
RSCAN0	Transmit buffer data field 1 register 88	RSCAN0TMDF188	<RSCAN0_base> + 158CH
RSCAN0	Transmit buffer ID register 89	RSCAN0TMID89	<RSCAN0_base> + 1590H
RSCAN0	Transmit buffer pointer register 89	RSCAN0TMRPTR89	<RSCAN0_base> + 1594H
RSCAN0	Transmit buffer data field 0 register 89	RSCAN0TMDF089	<RSCAN0_base> + 1598H
RSCAN0	Transmit buffer data field 1 register 89	RSCAN0TMDF189	<RSCAN0_base> + 159CH
RSCAN0	Transmit buffer ID register 90	RSCAN0TMID90	<RSCAN0_base> + 15A0H
RSCAN0	Transmit buffer pointer register 90	RSCAN0TMRPTR90	<RSCAN0_base> + 15A4H
RSCAN0	Transmit buffer data field 0 register 90	RSCAN0TMDF090	<RSCAN0_base> + 15A8H
RSCAN0	Transmit buffer data field 1 register 90	RSCAN0TMDF190	<RSCAN0_base> + 15ACH
RSCAN0	Transmit buffer ID register 91	RSCAN0TMID91	<RSCAN0_base> + 15B0H
RSCAN0	Transmit buffer pointer register 91	RSCAN0TMRPTR91	<RSCAN0_base> + 15B4H
RSCAN0	Transmit buffer data field 0 register 91	RSCAN0TMDF091	<RSCAN0_base> + 15B8H
RSCAN0	Transmit buffer data field 1 register 91	RSCAN0TMDF191	<RSCAN0_base> + 15BCH
RSCAN0	Transmit buffer ID register 92	RSCAN0TMID92	<RSCAN0_base> + 15C0H
RSCAN0	Transmit buffer pointer register 92	RSCAN0TMRPTR92	<RSCAN0_base> + 15C4H
RSCAN0	Transmit buffer data field 0 register 92	RSCAN0TMDF092	<RSCAN0_base> + 15C8H
RSCAN0	Transmit buffer data field 1 register 92	RSCAN0TMDF192	<RSCAN0_base> + 15CCH
RSCAN0	Transmit buffer ID register 93	RSCAN0TMID93	<RSCAN0_base> + 15D0H
RSCAN0	Transmit buffer pointer register 93	RSCAN0TMRPTR93	<RSCAN0_base> + 15D4H
RSCAN0	Transmit buffer data field 0 register 93	RSCAN0TMDF093	<RSCAN0_base> + 15D8H
RSCAN0	Transmit buffer data field 1 register 93	RSCAN0TMDF193	<RSCAN0_base> + 15DCH
RSCAN0	Transmit buffer ID register 94	RSCAN0TMID94	<RSCAN0_base> + 15E0H
RSCAN0	Transmit buffer pointer register 94	RSCAN0TMRPTR94	<RSCAN0_base> + 15E4H
RSCAN0	Transmit buffer data field 0 register 94	RSCAN0TMDF094	<RSCAN0_base> + 15E8H
RSCAN0	Transmit buffer data field 1 register 94	RSCAN0TMDF194	<RSCAN0_base> + 15ECH
RSCAN0	Transmit buffer ID register 95	RSCAN0TMID95	<RSCAN0_base> + 15F0H
RSCAN0	Transmit buffer pointer register 95	RSCAN0TMRPTR95	<RSCAN0_base> + 15F4H
RSCAN0	Transmit buffer data field 0 register 95	RSCAN0TMDF095	<RSCAN0_base> + 15F8H
RSCAN0	Transmit buffer data field 1 register 95	RSCAN0TMDF195	<RSCAN0_base> + 15FCH
RSCAN0	Transmit history access register 0	RSCAN0THLACC0	<RSCAN0_base> + 1800H
RSCAN0	Transmit history access register 1	RSCAN0THLACC1	<RSCAN0_base> + 1804H
RSCAN0	Transmit history access register 2	RSCAN0THLACC2	<RSCAN0_base> + 1808H
RSCAN0	Transmit history access register 3	RSCAN0THLACC3	<RSCAN0_base> + 180CH
RSCAN0	Transmit history access register 4	RSCAN0THLACC4	<RSCAN0_base> + 1810H
RSCAN0	Transmit history access register 5	RSCAN0THLACC5	<RSCAN0_base> + 1814H
RSCAN0	RAM test page access register 0	RSCAN0RPGACC0	<RSCAN0_base> + 1900H
RSCAN0	RAM test page access register 1	RSCAN0RPGACC1	<RSCAN0_base> + 1904H
RSCAN0	RAM test page access register 2	RSCAN0RPGACC2	<RSCAN0_base> + 1908H
RSCAN0	RAM test page access register 3	RSCAN0RPGACC3	<RSCAN0_base> + 190CH

Table 19.12 List of Registers (31/32)

Module	Register	Symbol	Address
RSCAN0	RAM test page access register 4	RSCAN0RPGACC4	<RSCAN0_base> + 1910H
RSCAN0	RAM test page access register 5	RSCAN0RPGACC5	<RSCAN0_base> + 1914H
RSCAN0	RAM test page access register 6	RSCAN0RPGACC6	<RSCAN0_base> + 1918H
RSCAN0	RAM test page access register 7	RSCAN0RPGACC7	<RSCAN0_base> + 191CH
RSCAN0	RAM test page access register 8	RSCAN0RPGACC8	<RSCAN0_base> + 1920H
RSCAN0	RAM test page access register 9	RSCAN0RPGACC9	<RSCAN0_base> + 1924H
RSCAN0	RAM test page access register 10	RSCAN0RPGACC10	<RSCAN0_base> + 1928H
RSCAN0	RAM test page access register 11	RSCAN0RPGACC11	<RSCAN0_base> + 192CH
RSCAN0	RAM test page access register 12	RSCAN0RPGACC12	<RSCAN0_base> + 1930H
RSCAN0	RAM test page access register 13	RSCAN0RPGACC13	<RSCAN0_base> + 1934H
RSCAN0	RAM test page access register 14	RSCAN0RPGACC14	<RSCAN0_base> + 1938H
RSCAN0	RAM test page access register 15	RSCAN0RPGACC15	<RSCAN0_base> + 193CH
RSCAN0	RAM test page access register 16	RSCAN0RPGACC16	<RSCAN0_base> + 1940H
RSCAN0	RAM test page access register 17	RSCAN0RPGACC17	<RSCAN0_base> + 1944H
RSCAN0	RAM test page access register 18	RSCAN0RPGACC18	<RSCAN0_base> + 1948H
RSCAN0	RAM test page access register 19	RSCAN0RPGACC19	<RSCAN0_base> + 194CH
RSCAN0	RAM test page access register 20	RSCAN0RPGACC20	<RSCAN0_base> + 1950H
RSCAN0	RAM test page access register 21	RSCAN0RPGACC21	<RSCAN0_base> + 1954H
RSCAN0	RAM test page access register 22	RSCAN0RPGACC22	<RSCAN0_base> + 1958H
RSCAN0	RAM test page access register 23	RSCAN0RPGACC23	<RSCAN0_base> + 195CH
RSCAN0	RAM test page access register 24	RSCAN0RPGACC24	<RSCAN0_base> + 1960H
RSCAN0	RAM test page access register 25	RSCAN0RPGACC25	<RSCAN0_base> + 1964H
RSCAN0	RAM test page access register 26	RSCAN0RPGACC26	<RSCAN0_base> + 1968H
RSCAN0	RAM test page access register 27	RSCAN0RPGACC27	<RSCAN0_base> + 196CH
RSCAN0	RAM test page access register 28	RSCAN0RPGACC28	<RSCAN0_base> + 1970H
RSCAN0	RAM test page access register 29	RSCAN0RPGACC29	<RSCAN0_base> + 1974H
RSCAN0	RAM test page access register 30	RSCAN0RPGACC30	<RSCAN0_base> + 1978H
RSCAN0	RAM test page access register 31	RSCAN0RPGACC31	<RSCAN0_base> + 197CH
RSCAN0	RAM test page access register 32	RSCAN0RPGACC32	<RSCAN0_base> + 1980H
RSCAN0	RAM test page access register 33	RSCAN0RPGACC33	<RSCAN0_base> + 1984H
RSCAN0	RAM test page access register 34	RSCAN0RPGACC34	<RSCAN0_base> + 1988H
RSCAN0	RAM test page access register 35	RSCAN0RPGACC35	<RSCAN0_base> + 198CH
RSCAN0	RAM test page access register 36	RSCAN0RPGACC36	<RSCAN0_base> + 1990H
RSCAN0	RAM test page access register 37	RSCAN0RPGACC37	<RSCAN0_base> + 1994H
RSCAN0	RAM test page access register 38	RSCAN0RPGACC38	<RSCAN0_base> + 1998H
RSCAN0	RAM test page access register 39	RSCAN0RPGACC39	<RSCAN0_base> + 199CH
RSCAN0	RAM test page access register 40	RSCAN0RPGACC40	<RSCAN0_base> + 19A0H
RSCAN0	RAM test page access register 41	RSCAN0RPGACC41	<RSCAN0_base> + 19A4H
RSCAN0	RAM test page access register 42	RSCAN0RPGACC42	<RSCAN0_base> + 19A8H
RSCAN0	RAM test page access register 43	RSCAN0RPGACC43	<RSCAN0_base> + 19ACH
RSCAN0	RAM test page access register 44	RSCAN0RPGACC44	<RSCAN0_base> + 19B0H
RSCAN0	RAM test page access register 45	RSCAN0RPGACC45	<RSCAN0_base> + 19B4H
RSCAN0	RAM test page access register 46	RSCAN0RPGACC46	<RSCAN0_base> + 19B8H
RSCAN0	RAM test page access register 47	RSCAN0RPGACC47	<RSCAN0_base> + 19BCH

**Table 19.12 List of Registers (32/32)**

<b>Module</b>	<b>Register</b>	<b>Symbol</b>	<b>Address</b>
RSCAN0	RAM test page access register 48	RSCAN0RPGACC48	<RSCAN0_base> + 19C0 <sub>H</sub>
RSCAN0	RAM test page access register 49	RSCAN0RPGACC49	<RSCAN0_base> + 19C4 <sub>H</sub>
RSCAN0	RAM test page access register 50	RSCAN0RPGACC50	<RSCAN0_base> + 19C8 <sub>H</sub>
RSCAN0	RAM test page access register 51	RSCAN0RPGACC51	<RSCAN0_base> + 19CC <sub>H</sub>
RSCAN0	RAM test page access register 52	RSCAN0RPGACC52	<RSCAN0_base> + 19D0 <sub>H</sub>
RSCAN0	RAM test page access register 53	RSCAN0RPGACC53	<RSCAN0_base> + 19D4 <sub>H</sub>
RSCAN0	RAM test page access register 54	RSCAN0RPGACC54	<RSCAN0_base> + 19D8 <sub>H</sub>
RSCAN0	RAM test page access register 55	RSCAN0RPGACC55	<RSCAN0_base> + 19DC <sub>H</sub>
RSCAN0	RAM test page access register 56	RSCAN0RPGACC56	<RSCAN0_base> + 19E0 <sub>H</sub>
RSCAN0	RAM test page access register 57	RSCAN0RPGACC57	<RSCAN0_base> + 19E4 <sub>H</sub>
RSCAN0	RAM test page access register 58	RSCAN0RPGACC58	<RSCAN0_base> + 19E8 <sub>H</sub>
RSCAN0	RAM test page access register 59	RSCAN0RPGACC59	<RSCAN0_base> + 19EC <sub>H</sub>
RSCAN0	RAM test page access register 60	RSCAN0RPGACC60	<RSCAN0_base> + 19F0 <sub>H</sub>
RSCAN0	RAM test page access register 61	RSCAN0RPGACC61	<RSCAN0_base> + 19F4 <sub>H</sub>
RSCAN0	RAM test page access register 62	RSCAN0RPGACC62	<RSCAN0_base> + 19F8 <sub>H</sub>
RSCAN0	RAM test page access register 63	RSCAN0RPGACC63	<RSCAN0_base> + 19FC <sub>H</sub>

**Table 19.13** Transmit Buffer p Allocated to Each Channel

CANm	
Transmit buffer p	Transmit buffer $16 \times m + 0$
	Transmit buffer $16 \times m + 1$
	Transmit buffer $16 \times m + 2$
	Transmit buffer $16 \times m + 3$
	Transmit buffer $16 \times m + 4$
	Transmit buffer $16 \times m + 5$
	Transmit buffer $16 \times m + 6$
	Transmit buffer $16 \times m + 7$
	Transmit buffer $16 \times m + 8$
	Transmit buffer $16 \times m + 9$
	Transmit buffer $16 \times m + 10$
	Transmit buffer $16 \times m + 11$
	Transmit buffer $16 \times m + 12$
	Transmit buffer $16 \times m + 13$
	Transmit buffer $16 \times m + 14$
	Transmit buffer $16 \times m + 15$

**Table 19.14** Transmit/Receive FIFO Buffer k Allocated to Each Channel

CANm	
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer $3 \times m + 0$
	Transmit/receive FIFO buffer $3 \times m + 1$
	Transmit/receive FIFO buffer $3 \times m + 2$

**Table 19.15** Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000 <sub>B</sub>	Transmit buffer $16 \times m + 0$
0001 <sub>B</sub>	Transmit buffer $16 \times m + 1$
0010 <sub>B</sub>	Transmit buffer $16 \times m + 2$
0011 <sub>B</sub>	Transmit buffer $16 \times m + 3$
0100 <sub>B</sub>	Transmit buffer $16 \times m + 4$
0101 <sub>B</sub>	Transmit buffer $16 \times m + 5$
0110 <sub>B</sub>	Transmit buffer $16 \times m + 6$
0111 <sub>B</sub>	Transmit buffer $16 \times m + 7$
1000 <sub>B</sub>	Transmit buffer $16 \times m + 8$
1001 <sub>B</sub>	Transmit buffer $16 \times m + 9$
1010 <sub>B</sub>	Transmit buffer $16 \times m + 10$
1011 <sub>B</sub>	Transmit buffer $16 \times m + 11$
1100 <sub>B</sub>	Transmit buffer $16 \times m + 12$
1101 <sub>B</sub>	Transmit buffer $16 \times m + 13$
1110 <sub>B</sub>	Transmit buffer $16 \times m + 14$
1111 <sub>B</sub>	Transmit buffer $16 \times m + 15$

**Table 19.16 Transmit Buffer p Allocated to the Transmit Queue of Each Channel**

<b>Setting of Bits TXQDC[3:0]</b>	<b>Transmit Buffer p Allocated to the Transmit Queue</b>
0000 <sub>B</sub>	Setting prohibited
0001 <sub>B</sub>	Setting prohibited
0010 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 13
0011 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 12
0100 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 11
0101 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 10
0110 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 9
0111 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 8
1000 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 7
1001 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 6
1010 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 5
1011 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 4
1100 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 3
1101 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 2
1110 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 1
1111 <sub>B</sub>	Transmit buffer 16 × m + 15 to 16 × m + 0

### 19.3.2 RSCAN0CmCFG — Channel Configuration Register (m = 0 to 5)

**Access:** RSCAN0CmCFG register can be read or written in 32-bit units.  
 RSCAN0CmCFG<sub>L</sub>, RSCAN0CmCFG<sub>H</sub> registers can be read or written in 16-bit units.  
 RSCAN0CmCFG<sub>LL</sub>, RSCAN0CmCFG<sub>LH</sub>, RSCAN0CmCFG<sub>HL</sub>, RSCAN0CmCFG<sub>HH</sub> registers can be read or written in 8-bit units.

**Address:** RSCAN0CmCFG: <RSCAN0\_base> + 0000<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmCFG<sub>L</sub>: <RSCAN0\_base> + 0000<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCFG<sub>H</sub>: <RSCAN0\_base> + 0002<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmCFG<sub>LL</sub>: <RSCAN0\_base> + 0000<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCFG<sub>LH</sub>: <RSCAN0\_base> + 0001<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCFG<sub>HL</sub>: <RSCAN0\_base> + 0002<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCFG<sub>HH</sub>: <RSCAN0\_base> + 0003<sub>H</sub> + (10<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SJW[1:0]	—	TSEG2[2:0]	—	TSEG1[3:0]	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BRP[9:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.17 RSCAN0CmCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	SJW[1:0]	Resynchronization Jump Width Control b <sub>25</sub> b <sub>24</sub> 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 20	TSEG2[2:0]	Time Segment 2 Control b <sub>22</sub> b <sub>21</sub> b <sub>20</sub> 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

**Table 19.17 RSCAN0CmCFG Register Contents (2/2)**

Bit Position	Bit Name	Function
19 to 16	TSEG1[3:0]	Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	BRP[9:0]	Prescaler Division Ratio Set When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

Modify the RSCAN0CmCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode before shifting to channel communication mode or channel halt mode. For a description of the bit timing parameters and settings, see **Section 19.10.1, Initial Settings**.

### SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.

Set a value less than or equal to the value of the TSEG2 bits.

### TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE\_SEG2).

Allowed values are 2 Tq to 8 Tq, inclusive.

Set a value smaller than the value of the TSEG1 bits.

### TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation segment (PROP\_SEG) and phase segment 1 (PHASE\_SEG1).

Allowed values are 4 Tq to 16 Tq, inclusive.

### BRP[9:0] Bits

The CANmTq clock (fCANTQm) is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler, ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

### 19.3.3 RSCAN0CmCTR — Channel Control Register (m = 0 to 5)

**Access:** RSCAN0CmCTR register can be read or written in 32-bit units.  
 RSCAN0CmCTRL, RSCAN0CmCTRH registers can be read or written in 16-bit units.  
 RSCAN0CmCTRLL, RSCAN0CmCTRLH, RSCAN0CmCTRHL, RSCAN0CmCTRHH registers can be read or written in 8-bit units.

**Address:** RSCAN0CmCTR: <RSCAN0\_base> + 0004<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmCTRL: <RSCAN0\_base> + 0004<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCTRH: <RSCAN0\_base> + 0006<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmCTRLL: <RSCAN0\_base> + 0004<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCTRLH: <RSCAN0\_base> + 0005<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCTRHL: <RSCAN0\_base> + 0006<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmCTRHH: <RSCAN0\_base> + 0007<sub>H</sub> + (10<sub>H</sub> × m)

**Value after reset:** 0000 0005<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CTMS[1:0]	CTME	ERRD	BOM[1:0]	—	—	—	—	—	TAIE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 19.18 RSCAN0CmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in RSCAN0CmERFL are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus-off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request
20 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.

**Table 19.18 RSCAN0CmCTR Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus-off Recovery Interrupt Enable 0: Bus-off recovery interrupt is disabled. 1: Bus-off recovery interrupt is enabled.
11	BOEIE	Bus-off Entry Interrupt Enable 0: Bus-off entry interrupt is disabled. 1: Bus-off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RTBO	Forcible Recovery from Bus-off When this bit is set to 1, forcible recovery from the bus-off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

**CTMS[1:0] Bits**

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

**CTME Bit**

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

**ERRD Bit**

This bit is used to control the display mode of bits 14 to 8 in the RSCAN0CmERFL register. When this bit is clear to 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

### BOM[1:0] Bits

These bits are used to select the bus-off recovery mode of the RS-CAN module.

When the BOM[1:0] bits are set to  $00_B$ , recovery from the bus-off state to the error active state is compliant with the CAN specifications. That is, the RS-CAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus-off recovery interrupt request is generated at the time of recovery from the bus-off state. Even if the CHMDC[1:0] bits are set to  $10_B$  (channel halt mode) before recessive bits are detected 128 times, the RS-CAN module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CAN module reaches the bus-off state when the BOM[1:0] bits are set to  $01_B$ , the CHMDC[1:0] bits in the RSCAN0CmCTR register ( $m = 0$  to 5) are set to  $10_B$  and the RS-CAN module transitions to channel halt mode. No bus-off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are cleared to  $00_H$ .

When the RS-CAN module reaches the bus-off state when the BOM[1:0] bits are set to  $10_B$ , the CHMDC[1:0] bits are set to  $10_B$  and the RS-CAN module transitions to channel halt mode after recovery from the bus-off state (11 consecutive recessive bits are detected 128 times). A bus-off recovery interrupt request is generated at the time of recovery from the bus-off state and the TEC[7:0] and REC[7:0] bits are cleared to  $00_H$ .

When the BOM[1:0] bits are set to  $11_B$  and the CHMDC[1:0] bits are set to  $10_B$  while the RS-CAN module is in the bus-off state, the RS-CAN module transitions to channel halt mode. No bus-off recovery interrupt request is generated at the time of recovery from the bus-off state and the TEC[7:0] and REC[7:0] bits are cleared to  $00_H$ . However, if 11 consecutive recessive bits are detected 128 times and the RS-CAN module has recovered to the error active state from the bus-off state before the CHMDC[1:0] bits are set to  $10_B$ , a bus-off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CAN module transition to channel halt mode (at bus-off entry when the BOM[1:0] bits are  $01B$  or at bus-off end when the BOM[1:0] bits are  $10_B$ ), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

### TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

### ALIE Bit

When the ALF flag in the RSCAN0CmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

### BLIE Bit

When the BLF flag in the RSCAN0CmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

### OLIE Bit

When the OVLF flag in the RSCAN0CmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

**BORIE Bit**

When the BORF flag in the RSCAN0CmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

**BOEIE Bit**

When the BOEF flag in the RSCAN0CmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

**EPIE Bit**

When the EPF flag in the RSCAN0CmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

**EWIE Bit**

When the EWF flag in the RSCAN0CmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

**BEIE Bit**

When the BEF flag in the RSCAN0CmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

**RTBO Bit**

Setting this bit to 1 in the bus-off state forcibly returns the state from the bus-off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register to  $00_H$  and also clears the BOSTS flag in the RSCAN0CmSTS register to 0 (not in bus-off state). The other registers remain unchanged. No bus-off recovery interrupt request is generated upon recovery from the bus-off state in this case. Use this bit only when the BOM[1:0] bits in the RSCAN0CmCTR register are  $00_B$  (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RS-CAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

**CSLPR Bit**

Setting this bit to 1 places the channel into channel stop mode.

Clearing this bit to 0 makes the channel exit channel stop mode.

This bit should not be modified in channel communication mode or channel halt mode.

**CHMDC[1:0] Bits**

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 19.5.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to  $11_B$ . When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically become  $10_B$ .

### 19.3.4 RSCAN0CmSTS — Channel Status Register (m = 0 to 5)

**Access:** RSCAN0CmSTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0CmSTS<sub>L</sub>, RSCAN0CmSTS<sub>H</sub> registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0CmSTS<sub>LL</sub>, RSCAN0CmSTS<sub>SH</sub>, RSCAN0CmSTS<sub>HH</sub> registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0CmSTS: <RSCAN0\_base> + 0008<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmSTS<sub>L</sub>: <RSCAN0\_base> + 0008<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmSTS<sub>H</sub>: <RSCAN0\_base> + 000A<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmSTS<sub>LL</sub>: <RSCAN0\_base> + 0008<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmSTS<sub>SH</sub>: <RSCAN0\_base> + 000A<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmSTS<sub>HH</sub>: <RSCAN0\_base> + 000B<sub>H</sub> + (10<sub>H</sub> × m)

**Value after reset:** 0000 0005<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]										REC[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.19 RSCAN0CmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 8	Reserved	When read, the value after reset is returned.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus-off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus-off state
4	BOSTS	Bus-off Status Flag 0: Not in bus-off state 1: In bus-off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

**TEC[7:0] Bits**

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

**REC[7:0] Bits**

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

**COMSTS Flag**

This bit indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

**RECSTS Flag**

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

**TRMSTS Flag**

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus-off state.

**BOSTS Flag**

This flag is set to 1 when the bus-off state ( $TEC[7:0] > 255$ ) is entered. It is cleared to 0 when the CAN module has exited the bus-off state.

**EPSTS Flag**

This flag is set to 1 when the RS-CAN module has entered the error passive state ( $(128 \leq TEC[7:0] \leq 255)$  or  $(128 \leq REC[7:0])$ ). It is cleared to 0 when the RS-CAN module has exited the error passive state or has entered channel reset mode.

**CSLPSTS Flag**

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

**CHLTSTS Flag**

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

**CRSTSTS Flag**

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

### 19.3.5 RSCAN0CmERFL — Channel Error Flag Register (m = 0 to 5)

**Access:** RSCAN0CmERFL register can be read or written in 32-bit units.  
 RSCAN0CmERFLL register can be read or written in 16-bit units.  
 RSCAN0CmERFLH register is a read-only register that can be read in 16-bit units.  
 RSCAN0CmERFLLL, RSCAN0CmERFLH registers can be read or written in 8-bit units.  
 RSCAN0CmERFLHL, RSCAN0CmERFLHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0CmERFL: <RSCAN0\_base> + 000C<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmERFLL: <RSCAN0\_base> + 000C<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmERFLH: <RSCAN0\_base> + 000E<sub>H</sub> + (10<sub>H</sub> × m)

RSCAN0CmERFLLL: <RSCAN0\_base> + 000C<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmERFLHH: <RSCAN0\_base> + 000D<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmERFLHL: <RSCAN0\_base> + 000E<sub>H</sub> + (10<sub>H</sub> × m),  
 RSCAN0CmERFLHH: <RSCAN0\_base> + 000F<sub>H</sub> + (10<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W* <sup>1</sup>														

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 19.20 RSCAN0CmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data A CRC value calculated based on the transmit message or received message is indicated.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.

**Table 19.20 RSCAN0CmERFL Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Bus Lock Flag 0: No channel bus is detected. 1: Channel bus is detected.
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus-off Recovery Flag 0: No bus-off recovery is detected. 1: Bus-off recovery is detected.
3	BOEF	Bus-off Entry Flag 0: No bus-off entry is detected. 1: Bus-off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 0 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCAN0CmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCAN0CmERFL is detected, the flag bits are only set by the error event if bits 14 to 8 were all 0 at the when time the error occurred.

### **CRCREG[14:0] Flag**

When the CTME bit in the RSCAN0CmCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0.

### **ADERR Flag**

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

### **B0ERR Flag**

This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

### **B1ERR Flag**

This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

**CERR Flag**

This flag is set to 1 when a CRC error has been detected.

**AERR Flag**

This flag is set to 1 when an ACK error has been detected.

**FERR Flag**

This flag is set to 1 when a form error has been detected.

**SERR Flag**

This flag is set to 1 when a stuff error has been detected.

**ALF Flag**

This flag is set to 1 when an arbitration-lost has been detected.

**BLF Flag**

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a dominant lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

**OVLF Flag**

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

**BORF Flag**

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus-off state. However, this flag is not set to 1 if the CAN module returns from the bus-off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01<sub>B</sub> (channel reset mode).
- The RTBO bit in the RSCAN0CmCTR register is set to 1 (forcible recovery from the bus-off state is made).
- The BOM[1:0] bits in the RSCAN0CmCTR register are set to 01<sub>B</sub> (transition to channel halt mode at bus-off entry).
- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 10<sub>B</sub> (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11<sub>B</sub> (transition to channel halt mode upon a request from the program during bus-off).

**BOEF Flag**

This flag is set to 1 when the bus-off state is reached (TEC[7:0] value > 255). This flag is also set to 1 if the bus-off state is reached when the BOM[1:0] bits in the RSCAN0CmCTR register ( $m = 0$  to 5) set to  $01_B$  (transition to channel halt mode at bus-off entry).

**EPF Flag**

This flag becomes 1 when the error passive state is reached (REC[7:0] or TEC[7:0] value > 127). This flag becomes 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

**EWF Flag**

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

**BEF Flag**

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCAN0CmERFL register is set to 1.

**NOTE**

To clear the flag of this register to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

### 19.3.6 RSCAN0GCFG — Global Configuration Register

**Access:** RSCAN0GCFG register can be read or written in 32-bit units.  
 RSCAN0GCFGGL, RSCAN0GCFGH registers can be read or written in 16-bit units.  
 RSCAN0GCFGLL, RSCAN0GCFGLH, RSCAN0GCFGHL, RSCAN0GCFGHH registers can be read or written in 8-bit units.

**Address:** RSCAN0GCFG: <RSCAN0\_base> + 0084<sub>H</sub>  
 RSCAN0GCFGGL: <RSCAN0\_base> + 0084<sub>H</sub>, RSCAN0GCFGH: <RSCAN0\_base> + 0086<sub>H</sub>  
 RSCAN0GCFGLL: <RSCAN0\_base> + 0084<sub>H</sub>, RSCAN0GCFGLH: <RSCAN0\_base> + 0085<sub>H</sub>,  
 RSCAN0GCFGHL: <RSCAN0\_base> + 0086<sub>H</sub>, RSCAN0GCFGHH: <RSCAN0\_base> + 0087<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ITRCP[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TSBTCS[2:0] TSSS TSP[3:0] — DCS MME DRE DCE TPRI																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.21 RSCAN0GCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000 <sub>H</sub> is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 bit time clock 0 0 1: Channel 1 bit time clock 0 1 0: Channel 2 bit time clock 0 1 1: Channel 3 bit time clock 1 0 0: Channel 4 bit time clock 1 0 1: Channel 5 bit time clock 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2 <sup>*1</sup> 1: Bit time clock
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768

**Table 19.21 RSCAN0GCFG Register Contents (2/2)**

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	DCS	CAN Clock Source Select <sup>*2</sup> 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000<sub>B</sub>.

Note 2. For the CAN clock frequency settings, see **Table 19.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L**.

Modify the RSCAN0GCFG register only in global reset mode.

### ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See **Section 19.7.3.1, Interval Transmission Function**.

### TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

### TSSS Bit

This bit is used to select a clock source of the timestamp counter.

### TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

### DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk\_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see **Table 19.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L**.

### MME Bit

Setting this bit to 1 makes the mirror function available.

**DRE Bit**

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of  $00_H$  is stored in each data byte beyond the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

**DCE Bit**

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCAN0GAFLP0j register to  $0000_B$  before clearing the DCE bit in the RSCAN0GCFG register to 0.

**TPRI Bit**

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the lowest transmit buffer number of those has the highest priority.

While the transmit queue is in use, this bit should be set to 0.

### 19.3.7 RSCAN0GCTR — Global Control Register

**Access:** RSCAN0GCTR register can be read or written in 32-bit units.  
 RSCAN0GCTRL, RSCAN0GCTRH registers can be read or written in 16-bit units.  
 RSCAN0GCTRLL, RSCAN0GCTRLHL, RSCAN0GCTRHL registers can be read or written in 8-bit units.

**Address:** RSCAN0GCTR: <RSCAN0\_base> + 0088<sub>H</sub>  
 RSCAN0GCTRL: <RSCAN0\_base> + 0088<sub>H</sub>, RSCAN0GCTRH: <RSCAN0\_base> + 008A<sub>H</sub>  
 RSCAN0GCTRLL: <RSCAN0\_base> + 0088<sub>H</sub>, RSCAN0GCTRLHL: <RSCAN0\_base> + 0089<sub>H</sub>,  
 RSCAN0GCTRHL: <RSCAN0\_base> + 008A<sub>H</sub>

**Value after reset:** 0000 0005<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 19.22 RSCAN0GCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

#### TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCAN0GTSC register is cleared to 0000<sub>H</sub>.

**THLEIE Bit**

When the THLEIE bit is set to 1 and the THLES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

**MEIE Bit**

When the MEIE bit is set to 1 and the MES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

**DEIE Bit**

When the DEIE bit is set to 1 and the DEF flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

**GSLPR Bit**

Setting this bit to 1 places the RS-CAN module into global stop mode.

Clearing this bit to 0 makes the RS-CAN module leave from global stop mode.

This bit should not be modified in global operating mode or global test mode.

**GMDC[1:0] Bits**

These bits are used to select the mode of entire RS-CAN module (global operating mode, global reset mode, or global test mode). For details, see **Section 19.5.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CAN module into global stop mode.

### 19.3.8 RSCAN0GSTS — Global Status Register

**Access:** RSCAN0GSTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0GSTSL register is a read-only register that can be read in 16-bit units.  
 RSCAN0GSTSLL register is a read-only register that can be read in 8-bit units.

**Address:** RSCAN0GSTS: <RSCAN0\_base> + 008C<sub>H</sub>  
 RSCAN0GSTSL: <RSCAN0\_base> + 008C<sub>H</sub>  
 RSCAN0GSTSLL: <RSCAN0\_base> + 008C<sub>H</sub>

**Value after reset:** 0000 000D<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLSTS	GRSTS TS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.23 RSCAN0GSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTS TS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

#### GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

#### GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

#### GHLSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

**GRSTSTS Flag**

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

**19.3.9 RSCAN0GERFL — Global Error Flag Register**

**Access:** RSCAN0GERFL register can be read or written in 32-bit units.  
 RSCAN0GERFLL register can be read or written in 16-bit units.  
 RSCAN0GERFLLL register can be read or written in 8-bit units.

**Address:** RSCAN0GERFL: <RSCAN0\_base> + 0090<sub>H</sub>  
 RSCAN0GERFLL: <RSCAN0\_base> + 0090<sub>H</sub>  
 RSCAN0GERFLLL: <RSCAN0\_base> + 0090<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W <sup>*1</sup>

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

**Table 19.24 RSCAN0GERFL Register Contents**

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	Reserved	When read, the undefined value is returned. When writing, write the value after reset.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	Reserved	When read, the undefined value is returned. When writing, write the value after reset.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCAN0GERFL register are cleared to 0 in global reset mode.

**THLES Flag**

The THLES flag is set to 1 when any one of the THLELT flags in the RSCAN0THLSTS<sub>m</sub> register ( $m = 0$  to 5) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

**MES Flag**

The MES flag is set to 1 when any one of the RFMLT flags in the RSCAN0RFSTS<sub>x</sub> register ( $x = 0$  to 7) or the CFMLT flags in the RSCAN0CFSTS<sub>k</sub> register ( $k = 0$  to 17) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

**DEF Flag**

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

**NOTE**

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to "0" to "0", and the bits not to be set to "0" to "1".

### 19.3.10 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0

**Access:** RSCAN0GTINTSTS0 register is a read-only register that can be read in 32-bit units. RSCAN0GTINTSTS0L, RSCAN0GTINTSTS0H registers are the read-only registers that can be read in 16-bit units. RSCAN0GTINTSTS0LL, RSCAN0GTINTSTS0LH, RSCAN0GTINTSTS0HL, RSCAN0GTINTSTS0HH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0GTINTSTS0: <RSCAN0\_base> + 0460<sub>H</sub>  
 RSCAN0GTINTSTS0L: <RSCAN0\_base> + 0460<sub>H</sub>, RSCAN0GTINTSTS0H: <RSCAN0\_base> + 0462<sub>H</sub>  
 RSCAN0GTINTSTS0LL: <RSCAN0\_base> + 0460<sub>H</sub>, RSCAN0GTINTSTS0LH: <RSCAN0\_base> + 0461<sub>H</sub>,  
 RSCAN0GTINTSTS0HL: <RSCAN0\_base> + 0462<sub>H</sub>, RSCAN0GTINTSTS0HH: <RSCAN0\_base> + 0463<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R* <sup>1</sup>	R	R	R	R* <sup>1</sup>								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R* <sup>1</sup>	R	R	R	R* <sup>1</sup>								

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 19.25 RSCAN0GTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned.
28	THIF3	Channel 3 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
27	CFTIF3	Channel 3 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
26	TQIF3	Channel 3 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
25	TAIF3	Channel 3 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested
24	TSIF3	Channel 3 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
23 to 21	Reserved	When read, the value after reset is returned.
20	THIF2	Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.

Table 19.25 RSCAN0GTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	TSIF2	Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	Reserved	When read, the value after reset is returned.
12	THIF1	Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF1	Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	When read, the value after reset is returned.
4	THIF0	Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

### TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to  $10_B$  (transmit completed without abort request) or  $11_B$  (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to  $00_B$  under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIE bit to 0 also clears this flag to 0.

### TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCAN0CmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to  $01_B$  (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to  $00_B$  after the transmit abort is completed.

**TQIFm Bits**

When the TXQIE bit in the RSCAN0TXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCAN0TXQSTS<sub>m</sub> register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCAN0TXQSTS<sub>m</sub> register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

**CFTIFm Bits**

When the CFTXIE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCAN0CFSTS<sub>k</sub> register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

**THIFm Bits**

When the THLIE bit in the RSCAN0THLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCAN0THLSTS<sub>m</sub> register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCAN0THLSTS<sub>m</sub> register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

### 19.3.11 RSCAN0GTINTSTS1 — Global TX Interrupt Status Register 1

**Access:** RSCAN0GTINTSTS1 register is a read-only register that can be read in 32-bit units.  
 RSCAN0GTINTSTS1L register is a read-only register that can be read in 16-bit units.  
 RSCAN0GTINTSTS1LL, RSCAN0GTINTSTS1LH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0GTINTSTS1: <RSCAN0\_base> + 0464<sub>H</sub>  
 RSCAN0GTINTSTS1L: <RSCAN0\_base> + 0464<sub>H</sub>  
 RSCAN0GTINTSTS1LL: <RSCAN0\_base> + 0464<sub>H</sub>, RSCAN0GTINTSTS1LH: <RSCAN0\_base> + 0465<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF5	CFTIF5	TQIF5	TAIF5	TSIF5	—	—	—	THIF4	CFTIF4	TQIF4	TAIF4	TSIF4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 19.26 RSCAN0GTINTSTS1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12	THIF5	Channel 5 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF5	Channel 5 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF5	Channel 5 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF5	Channel 5 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF5	Channel 5 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	When read, the value after reset is returned.
4	THIF4	Channel 4 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF4	Channel 4 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF4	Channel 4 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF4	Channel 4 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.

**Table 19.26 RSCAN0GTINTSTS1 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
0	TSIF4	Channel 4 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

**TSIFm Bits**

The TSIFm bit is set to 1 when the TMIEp bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to  $10_B$  (transmit completed without abort request) or  $11_B$  (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to  $00_B$  under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIE bit to 0 also clears this flag to 0.

**TAIFm Bits**

The TAIFm bit is set to 1 when the TAIE bit in the RSCAN0CmCTR register is set to 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to  $01_B$  (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to  $00_B$  after the transmit abort is completed.

**TQIFm Bits**

When the TXQIE bit in the RSCAN0TXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCAN0TXQSTS register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCAN0TXQSTS register is cleared to 0, this bit is cleared to 0. Clearing the TXQIE bit to 0 also clears this flag to 0.

**CFTIFm Bits**

When the CFTXIE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCAN0CFSTS register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

**THIFm Bits**

When the THLIE bit in the RSCAN0THLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCAN0THLSTS register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCAN0THLSTS register is cleared to 0, this bit is cleared to 0.

This flag is also cleared to 0 when the THLIE bit is cleared to 0.

### 19.3.12 RSCAN0GTSC — Global Timestamp Counter Register

**Access:** RSCAN0GTSC register is a read-only register that can be read in 32-bit units.  
RSCAN0GTSCL register is a read-only register that can be read in 16-bit units.

**Address:** RSCAN0GTSC: <RSCAN0\_base> + 0094<sub>H</sub>  
RSCAN0GTSCL: <RSCAN0\_base> + 0094<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
TS[15:0]																

Table 19.27 RSCAN0GTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000 <sub>H</sub> to FFFF <sub>H</sub>

#### TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCAN0GCFG register is 0 (pclk):  
The timestamp counter starts counting when the RS-CAN module has transitioned to global operating mode.  
This counter stops counting when the RS-CAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm bit time clock):  
The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.  
This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

### 19.3.13 RSCAN0GAFLECTR — Receive Rule Entry Control Register

**Access:** RSCAN0GAFLECTR register can be read or written in 32-bit units.  
 RSCAN0GAFLECTRL register can be read or written in 16-bit units.  
 RSCAN0GAFLECTRL, RSCAN0GAFLECTRLH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLECTR: <RSCAN0\_base> + 0098<sub>H</sub>

RSCAN0GAFLECTRL: <RSCAN0\_base> + 0098<sub>H</sub>

RSCAN0GAFLECTRLH: <RSCAN0\_base> + 0098<sub>H</sub>, RSCAN0GAFLECTRLH: <RSCAN0\_base> + 0099<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.28 RSCAN0GAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000 <sub>B</sub> ) to page 23 (10111 <sub>B</sub> ).

#### AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

#### AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000<sub>B</sub> to 10111<sub>B</sub>.

### 19.3.14 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0

**Access:** RSCAN0GAFLCFG0 register can be read or written in 32-bit units.  
 RSCAN0GAFLCFG0L, RSCAN0GAFLCFG0H registers can be read or written in 16-bit units.  
 RSCAN0GAFLCFG0LL, RSCAN0GAFLCFG0LH, RSCAN0GAFLCFG0HL, RSCAN0GAFLCFG0HH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLCFG0: <RSCAN0\_base> + 009C<sub>H</sub>  
 RSCAN0GAFLCFG0L: <RSCAN0\_base> + 009C<sub>H</sub>, RSCAN0GAFLCFG0H: <RSCAN0\_base> + 009E<sub>H</sub>  
 RSCAN0GAFLCFG0LL: <RSCAN0\_base> + 009C<sub>H</sub>, RSCAN0GAFLCFG0LH: <RSCAN0\_base> + 009D<sub>H</sub>,  
 RSCAN0GAFLCFG0HL: <RSCAN0\_base> + 009E<sub>H</sub>, RSCAN0GAFLCFG0HH: <RSCAN0\_base> + 009F<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								RNC3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.29 RSCAN0GAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	RNC3[7:0]	Number of Rules for Channel 3 Set the number of receive rules exclusively used for channel 3.

Modify the RSCAN0GAFLCFG0 register only in global reset mode.

Up to  $64 \times$  (number of channels) rules can be registered in the receive rule table as the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered in the entire unit.

#### RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00<sub>H</sub> to 80<sub>H</sub>.

#### RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00<sub>H</sub> to 80<sub>H</sub>.

**RNC2[7:0] Bits**

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of  $00_H$  to  $80_H$ .

**RNC3[7:0] Bits**

These bits are used to set the number of rules to be registered in the channel 3 receive rule table.

Set these bits to a value within the range of  $00_H$  to  $80_H$ .

### 19.3.15 RSCAN0GAFLCFG1 — Receive Rule Configuration Register 1

**Access:** RSCAN0GAFLCFG1 register can be read or written in 32-bit units.  
 RSCAN0GAFLCFG1H register can be read or written in 16-bit units.  
 RSCAN0GAFLCFG1HL, RSCAN0GAFLCFG1HH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLCFG1: <RSCAN0\_base> + 00A0<sub>H</sub>  
 RSCAN0GAFLCFG1H: <RSCAN0\_base> + 00A2<sub>H</sub>  
 RSCAN0GAFLCFG1HL: <RSCAN0\_base> + 00A2<sub>H</sub>, RSCAN0GAFLCFG1HH: <RSCAN0\_base> + 00A3<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC4[7:0]										RNC5[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.30 RSCAN0GAFLCFG1 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC4[7:0]	Number of Rules for Channel 4 Set the number of receive rules exclusively used for channel 4.
23 to 16	RNC5[7:0]	Number of Rules for Channel 5 Set the number of receive rules exclusively used for channel 5.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Modify the RSCAN0GAFLCFG1 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table as the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered in the entire unit.

#### RNC4[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 4 receive rule table.

Set these bits to a value within the range of 00<sub>H</sub> to 80<sub>H</sub>.

#### RNC5[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 5 receive rule table.

Set these bits to a value within the range of 00<sub>H</sub> to 80<sub>H</sub>.

### 19.3.16 RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15)

**Access:** RSCAN0GAFLIDj register can be read or written in 32-bit units.  
 RSCAN0GAFLIDjL, RSCAN0GAFLIDjH registers can be read or written in 16-bit units.  
 RSCAN0GAFLIDjLL, RSCAN0GAFLIDjLH, RSCAN0GAFLIDjHL, RSCAN0GAFLIDjHH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLIDj: <RSCAN0\_base> + 0500<sub>H</sub> + (10<sub>H</sub> × j)

RSCAN0GAFLIDjL: <RSCAN0\_base> + 0500<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLIDjH: <RSCAN0\_base> + 0502<sub>H</sub> + (10<sub>H</sub> × j)

RSCAN0GAFLIDjLL: <RSCAN0\_base> + 0500<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLIDjLH: <RSCAN0\_base> + 0501<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLIDjHL: <RSCAN0\_base> + 0502<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLIDjHH: <RSCAN0\_base> + 0503<sub>H</sub> + (10<sub>H</sub> × j)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLID E	GAFLR TR	GAFLL B	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.31 RSCAN0GAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCAN0GAFLIDj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

#### GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

#### GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

**GAFLLB Bit**

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

**GAFLID[28:0] Bits**

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

### 19.3.17 RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15)

**Access:** RSCAN0GAFLMj register can be read or written in 32-bit units.  
 RSCAN0GAFLMjL, RSCAN0GAFLMjH registers can be read or written in 16-bit units.  
 RSCAN0GAFLMjLL, RSCAN0GAFLMjLH, RSCAN0GAFLMjHL, RSCAN0GAFLMjHH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLMj: <RSCAN0\_base> + 0504<sub>H</sub> + (10<sub>H</sub> × j)  
 RSCAN0GAFLMjL: <RSCAN0\_base> + 0504<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLMjH: <RSCAN0\_base> + 0506<sub>H</sub> + (10<sub>H</sub> × j)  
 RSCAN0GAFLMjLL: <RSCAN0\_base> + 0504<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLMjLH: <RSCAN0\_base> + 0505<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLMjHL: <RSCAN0\_base> + 0506<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLMjHH: <RSCAN0\_base> + 0507<sub>H</sub> + (10<sub>H</sub> × j)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—													GAFLIDM[28:16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																GAFLIDM[15:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.32 RSCAN0GAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared
29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCAN0GAFLMj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

#### GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCAN0GAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:0] bits to all 0 at the same time.

#### GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

#### GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

### 19.3.18 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15)

**Access:** RSCAN0GAFLP0j register can be read or written in 32-bit units.  
 RSCAN0GAFLP0jL, RSCAN0GAFLP0jH registers can be read or written in 16-bit units.  
 RSCAN0GAFLP0jLH, RSCAN0GAFLP0jHL, RSCAN0GAFLP0jHH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLP0j: <RSCAN0\_base> + 0508<sub>H</sub> + (10<sub>H</sub> × j)  
 RSCAN0GAFLP0jL: <RSCAN0\_base> + 0508<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP0jH: <RSCAN0\_base> + 050A<sub>H</sub> + (10<sub>H</sub> × j)  
 RSCAN0GAFLP0jLH: <RSCAN0\_base> + 0509<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP0jHL: <RSCAN0\_base> + 050A<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP0jHH: <RSCAN0\_base> + 050B<sub>H</sub> + (10<sub>H</sub> × j)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV		GAFLRMDP[6:0]						—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 19.33 RSCAN0GAFLP0j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store received message.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Modify the RSCAN0GAFLP0j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

#### GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000<sub>B</sub> disables the DLC check function allowing messages with any data length to pass the DLC check.

**GAFLPTR[11:0] Bits**

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

**GAFLRMV Bit**

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

**GAFLRMDP[6:0] Bits**

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCAN0RMNB register.

### 19.3.19 RSCAN0GAFLP1j—Receive Rule Pointer 1 Register (j = 0 to 15)

**Access:** RSCAN0GAFLP1j register can be read or written in 32-bit units.  
 RSCAN0GAFLP1jL, RSCAN0GAFLP1jH registers can be read or written in 16-bit units.  
 RSCAN0GAFLP1jLL, RSCAN0GAFLP1jLH, RSCAN0GAFLP1jHL, RSCAN0GAFLP1jHH registers can be read or written in 8-bit units.

**Address:** RSCAN0GAFLP1j: <RSCAN0\_base> + 050C<sub>H</sub> + (10<sub>H</sub> × j)  
 RSCAN0GAFLP1jL: <RSCAN0\_base> + 050C<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP1jH: <RSCAN0\_base> + 050E<sub>H</sub> + (10<sub>H</sub> × j)  
 RSCAN0GAFLP1jLL: <RSCAN0\_base> + 050C<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP1jLH: <RSCAN0\_base> + 050D<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP1jHL: <RSCAN0\_base> + 050E<sub>H</sub> + (10<sub>H</sub> × j),  
 RSCAN0GAFLP1jHH: <RSCAN0\_base> + 050F<sub>H</sub> + (10<sub>H</sub> × j)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
	—	—	—	—	—	—		GAFLFDP[25:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
	GAFLFDP[15:0]																						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							

Table 19.34 RSCAN0GAFLP1j Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 8	GAFLFDP[25:8]	Transmit/Receive FIFO Buffer k Select (Bit position –8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCAN0GAFLP1j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

#### GAFLFDP [25:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers are selectable. However, when the GAFLRMV bit in the RSCAN0GAFLP0j register is set to 1 (a message is stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCAN0CFCCk register are set to 00<sub>B</sub> (receive mode) or 10<sub>B</sub> (gateway mode) are selectable.

### 19.3.20 RSCAN0RMNB — Receive Buffer Number Register

**Access:** RSCAN0RMNB register can be read or written in 32-bit units.  
 RSCAN0RMNBL register can be read or written in 16-bit units.  
 RSCAN0RMNBLL register can be read or written in 8-bit units.

**Address:** RSCAN0RMNB: <RSCAN0\_base> + 00A4H  
 RSCAN0RMNBL: <RSCAN0\_base> + 00A4H  
 RSCAN0RMNBLL: <RSCAN0\_base> + 00A4H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.35 RSCAN0RMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 96.

Modify the RSCAN0RMNB register only in global reset mode.

#### NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CAN module. The maximum value is  $16 \times (\text{number of channels})$ .

Setting these bits all to 0 makes receive buffers unavailable.

### 19.3.21 RSCAN0RMNDy — Receive Buffer New Data Register (y = 0 to 2)

**Access:** RSCAN0RMNDy register can be read or written in 32-bit units.  
 RSCAN0RMNDyL, RSCAN0RMNDyH registers can be read or written in 16-bit units.  
 RSCAN0RMNDyLL, RSCAN0RMNDyLH, RSCAN0RMNDyHL, RSCAN0RMNDyHH registers can be read or written in 8-bit units.

**Address:** RSCAN0RMNDy: <RSCAN0\_base> + 00A8<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0RMNDyL: <RSCAN0\_base> + 00A8<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0RMNDyH: <RSCAN0\_base> + 00AA<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0RMNDyLL: <RSCAN0\_base> + 00A8<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0RMNDyLH: <RSCAN0\_base> + 00A9<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0RMNDyHL: <RSCAN0\_base> + 00AA<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0RMNDyHH: <RSCAN0\_base> + 00AB<sub>H</sub> + (04<sub>H</sub> × y)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.36 RSCAN0RMNDy Register Contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCAN0RMNDy register in global operating mode or global test mode.

#### RMNSq Flags (q = 0 to 95)

Each RMNS flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. It takes ten clock cycles of pclk to store a message.

These flags are cleared to 0 in global reset mode.

### 19.3.22 RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 95)

**Access:** RSCAN0RMIDq register is a read-only register that can be read in 32-bit units.  
 RSCAN0RMIDqL, RSCAN0RMIDqH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0RMIDqLL, RSCAN0RMIDqLH, RSCAN0RMIDqHL, RSCAN0RMIDqHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RMIDq: <RSCAN0\_base> + 0600<sub>H</sub> + (10<sub>H</sub> × q)

RSCAN0RMIDqL: <RSCAN0\_base> + 0600<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMIDqH: <RSCAN0\_base> + 0602<sub>H</sub> + (10<sub>H</sub> × q)

RSCAN0RMIDqLL: <RSCAN0\_base> + 0600<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMIDqLH: <RSCAN0\_base> + 0601<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMIDqHL: <RSCAN0\_base> + 0602<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMIDqHH: <RSCAN0\_base> + 0603<sub>H</sub> + (10<sub>H</sub> × q)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	RMIDE	RMRTR	—	RMID[28:16]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	RMID[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 19.37 RSCAN0RMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	When read, the value after reset is returned.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

#### RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

#### RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

#### RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

### 19.3.23 RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 95)

**Access:** RSCAN0RMPTRq register is a read-only register that can be read in 32-bit units.  
 RSCAN0RMPTRqL, RSCAN0RMPTRqH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0RMPTRqLL, RSCAN0RMPTRqLH, RSCAN0RMPTRqHL, RSCAN0RMPTRqHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RMPTRq: <RSCAN0\_base> + 0604<sub>H</sub> + (10<sub>H</sub> × q)  
 RSCAN0RMPTRqL: <RSCAN0\_base> + 0604<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMPTRqH: <RSCAN0\_base> + 0606<sub>H</sub> + (10<sub>H</sub> × q)  
 RSCAN0RMPTRqLL: <RSCAN0\_base> + 0604<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMPTRqLH: <RSCAN0\_base> + 0605<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMPTRqHL: <RSCAN0\_base> + 0606<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMPTRqHH: <RSCAN0\_base> + 0607<sub>H</sub> + (10<sub>H</sub> × q)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.38 RSCAN0RMPTRq Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: No data byte 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.

#### RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

#### RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

#### RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

### 19.3.24 RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 95)

**Access:** RSCAN0RMDF0q register is a read-only register that can be read in 32-bit units.  
 RSCAN0RMDF0qL, RSCAN0RMDF0qH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0RMDF0qLL, RSCAN0RMDF0qLH, RSCAN0RMDF0qHL, RSCAN0RMDF0qHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RMDF0q: <RSCAN0\_base> + 0608<sub>H</sub> + (10<sub>H</sub> × q)

RSCAN0RMDF0qL: <RSCAN0\_base> + 0608<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF0qH: <RSCAN0\_base> + 060A<sub>H</sub> + (10<sub>H</sub> × q)

RSCAN0RMDF0qLL: <RSCAN0\_base> + 0608<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF0qLH: <RSCAN0\_base> + 0609<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF0qHL: <RSCAN0\_base> + 060A<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF0qHH: <RSCAN0\_base> + 060B<sub>H</sub> + (10<sub>H</sub> × q)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB3[7:0]										RMDB2[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB1[7:0]										RMDB0[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.39 RSCAN0RMDF0q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB3[7:0]	Receive Buffer Data Byte 3
23 to 16	RMDB2[7:0]	Receive Buffer Data Byte 2
15 to 8	RMDB1[7:0]	Receive Buffer Data Byte 1
7 to 0	RMDB0[7:0]	Receive Buffer Data Byte 0
		Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCAN0RMPTRq register is smaller than 1000<sub>B</sub>, data bytes for which no data is set are read as 00<sub>H</sub>.

### 19.3.25 RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 95)

**Access:** RSCAN0RMDF1q register is a read-only register that can be read in 32-bit units.  
 RSCAN0RMDF1qL, RSCAN0RMDF1qH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0RMDF1qLL, RSCAN0RMDF1qLH, RSCAN0RMDF1qHL, RSCAN0RMDF1qHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RMDF1q: <RSCAN0\_base> + 060C<sub>H</sub> + (10<sub>H</sub> × q)  
 RSCAN0RMDF1qL: <RSCAN0\_base> + 060C<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF1qH: <RSCAN0\_base> + 060E<sub>H</sub> + (10<sub>H</sub> × q)  
 RSCAN0RMDF1qLL: <RSCAN0\_base> + 060C<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF1qLH: <RSCAN0\_base> + 060D<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF1qHL: <RSCAN0\_base> + 060E<sub>H</sub> + (10<sub>H</sub> × q),  
 RSCAN0RMDF1qHH: <RSCAN0\_base> + 060F<sub>H</sub> + (10<sub>H</sub> × q)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB7[7:0]										RMDB6[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB5[7:0]										RMDB4[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.40 RSCAN0RMDF1q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB7[7:0]	Receive Buffer Data Byte 7
23 to 16	RMDB6[7:0]	Receive Buffer Data Byte 6
15 to 8	RMDB5[7:0]	Receive Buffer Data Byte 5
7 to 0	RMDB4[7:0]	Receive Buffer Data Byte 4
		Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCAN0RMPTRq register is smaller than 1000<sub>B</sub>, data bytes for which no data is set are read as 00<sub>H</sub>.

### 19.3.26 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

**Access:** RSCAN0RFCCx register can be read or written in 32-bit units.  
 RSCAN0RFCCxL register can be read or written in 16-bit units.  
 RSCAN0RFCCxLL, RSCAN0RFCCxLH registers can be read or written in 8-bit units.

**Address:** RSCAN0RFCCx: <RSCAN0\_base> + 00B8<sub>H</sub> + (04<sub>H</sub> × x)

RSCAN0RFCCxL: <RSCAN0\_base> + 00B8<sub>H</sub> + (04<sub>H</sub> × x)

RSCAN0RFCCxLL: <RSCAN0\_base> + 00B8<sub>H</sub> + (04<sub>H</sub> × x),  
 RSCAN0RFCCxLH: <RSCAN0\_base> + 00B9<sub>H</sub> + (04<sub>H</sub> × x)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 19.41 RSCAN0RFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.

**Table 19.41 RSCAN0RFCCx Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

**RFIGCV[2:0] Bits**

These bits are used to specify the number of received messages for generating a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001<sub>B</sub> (4 messages), set the RFIGCV[2:0] bits to 001<sub>B</sub>, 011<sub>B</sub>, 101<sub>B</sub>, or 111<sub>B</sub>. Modify these bits only in global reset mode.

**RFIM Bit**

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

**RFDC[2:0] Bits**

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000<sub>B</sub>, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

**RFIE Bit**

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit set to 0 (no receive FIFO buffer is used).

**RFE Bit**

Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RSCAN0RFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode.

### 19.3.27 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

**Access:** RSCAN0RFSTSx register can be read or written in 32-bit units.  
 RSCAN0RFSTSxL register can be read or written in 16-bit units.  
 RSCAN0RFSTSxLL register can be read or written in 8-bit units.  
 RSCAN0RFSTSxLH register is a read-only register that can be read in 8-bit units.

**Address:** RSCAN0RFSTSx: <RSCAN0\_base> + 00D8<sub>H</sub> + (04<sub>H</sub> × x)  
 RSCAN0RFSTSxL: <RSCAN0\_base> + 00D8<sub>H</sub> + (04<sub>H</sub> × x)

RSCAN0RFSTSxLL: <RSCAN0\_base> + 00D8<sub>H</sub> + (04<sub>H</sub> × x),  
 RSCAN0RFSTSxLH: <RSCAN0\_base> + 00D9<sub>H</sub> + (04<sub>H</sub> × x)

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W <sup>*1</sup>	R/W <sup>*1</sup>	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 19.42 RSCAN0RFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

#### RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00<sub>H</sub> when the RFE bit in the RSCAN0RFCCx register is set to 0.

### **RFIF Flag**

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCAN0RFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **RFMLT Flag**

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **RFFLL Flag**

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCAN0RFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCAN0RFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

### **RFEMP Flag**

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCAN0RFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

#### **NOTE**

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

### 19.3.28 RSCAN0RFPCTR $x$ — Receive FIFO Buffer Pointer Control Register ( $x = 0$ to 7)

**Access:** RSCAN0RFPCTR $x$  register is a write-only register that can be written in 32-bit units.  
 RSCAN0RFPCTRXL register is a write-only register that can be written in 16-bit units.  
 RSCAN0RFPCTRLL register is a write-only register that can be written in 8-bit units.

**Address:** RSCAN0RFPCTR $x$ : <RSCAN0\_base> + 00F8<sub>H</sub> + (04<sub>H</sub> ×  $x$ )

RSCAN0RFPCTRXL: <RSCAN0\_base> + 00F8<sub>H</sub> + (04<sub>H</sub> ×  $x$ )

RSCAN0RFPCTRLL: <RSCAN0\_base> + 00F8<sub>H</sub> + (04<sub>H</sub> ×  $x$ )

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 19.43 RSCAN0RFPCTR $x$  Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF <sub>H</sub> , the read pointer moves to the next unread message in the receive FIFO buffer.

#### RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF<sub>H</sub>, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCAN0RFSTS $x$  register is decremented. Read the RSCAN0RFID $x$ , RSCAN0RFPTR $x$ , RSCAN0RFDF0 $x$ , and RSCAN0RFDF1 $x$  registers to read messages in the receive FIFO buffer, and then write FF<sub>H</sub> to the RFPC[7:0] bits.

When writing FF<sub>H</sub> to these bits, make sure that the RFE bit in the RSCAN0RFCC $x$  register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCAN0RFSTS $x$  register is 0 (the receive FIFO buffer contains unread messages).

### 19.3.29 RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

**Access:** RSCAN0RFIDx register is a read-only register that can be read in 32-bit units.  
 RSCAN0RFIDxL, RSCAN0RFIDxH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0RFIDxLL, RSCAN0RFIDxLH, RSCAN0RFIDxHL, RSCAN0RFIDxHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RFIDx: <RSCAN0\_base> + 0E00<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFIDxL: <RSCAN0\_base> + 0E00<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFIDxH: <RSCAN0\_base> + 0E02<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFIDxLL: <RSCAN0\_base> + 0E00<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFIDxLH: <RSCAN0\_base> + 0E01<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFIDxHL: <RSCAN0\_base> + 0E02<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFIDxHH: <RSCAN0\_base> + 0E03<sub>H</sub> + (10<sub>H</sub> × x)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	RFIDE	RFRTR	—	RFID[28:16]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	RFID[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 19.44 RSCAN0RFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	When read, the value after reset is returned.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

#### RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

#### RFRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

#### RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

### 19.3.30 RSCAN0RFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

**Access:** RSCAN0RFPTRx register is a read-only register that can be read in 32-bit units.  
 RSCAN0RFPTRxL, RSCAN0RFPTRxH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0RFPTRxLL, RSCAN0RFPTRxLH, RSCAN0RFPTRxHL, RSCAN0RFPTRxHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RFPTRx: <RSCAN0\_base> + 0E04<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFPTRxL: <RSCAN0\_base> + 0E04<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFPTRxH: <RSCAN0\_base> + 0E06<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFPTRxLL: <RSCAN0\_base> + 0E04<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFPTRxLH: <RSCAN0\_base> + 0E05<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFPTRxHL: <RSCAN0\_base> + 0E06<sub>H</sub> + (10<sub>H</sub> × x),  
 RSCAN0RFPTRxHH: <RSCAN0\_base> + 0E07<sub>H</sub> + (10<sub>H</sub> × x)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.45 RSCAN0RFPTRx Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

#### RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

#### RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

#### RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

### 19.3.31 RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7)

**Access:** RSCAN0RFDF0x register is a read-only register that can be read in 32-bit units. RSCAN0RFDF0xL, RSCAN0RFDF0xH registers are the read-only registers that can be read in 16-bit units. RSCAN0RFDF0xLL, RSCAN0RFDF0xLH, RSCAN0RFDF0xHL, RSCAN0RFDF0xHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RFDF0x: <RSCAN0\_base> + 0E08<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFDF0xL: <RSCAN0\_base> + 0E08<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF0xH: <RSCAN0\_base> + 0E0A<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFDF0xLL: <RSCAN0\_base> + 0E08<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF0xLH: <RSCAN0\_base> + 0E09<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF0xHL: <RSCAN0\_base> + 0E0A<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF0xHH: <RSCAN0\_base> + 0E0B<sub>H</sub> + (10<sub>H</sub> × x)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB3[7:0]										RFDB2[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	RFDB1[7:0]										RFDB0[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.46 RSCAN0RFDF0x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB3[7:0]	Receive FIFO Buffer Data Byte 3
23 to 16	RFDB2[7:0]	Receive FIFO Buffer Data Byte 2
15 to 8	RFDB1[7:0]	Receive FIFO Buffer Data Byte 1
7 to 0	RFDB0[7:0]	Receive FIFO Buffer Data Byte 0
		Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCAN0RFPTRx register is smaller than 1000<sub>B</sub>, data bytes for which no data is set are read as 00<sub>H</sub>.

### 19.3.32 RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7)

**Access:** RSCAN0RFDF1x register is a read-only register that can be read in 32-bit units. RSCAN0RFDF1xL, RSCAN0RFDF1xH registers are the read-only registers that can be read in 16-bit units. RSCAN0RFDF1xLL, RSCAN0RFDF1xLH, RSCAN0RFDF1xHL, RSCAN0RFDF1xHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0RFDF1x: <RSCAN0\_base> + 0E0C<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFDF1xL: <RSCAN0\_base> + 0E0C<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF1xH: <RSCAN0\_base> + 0E0E<sub>H</sub> + (10<sub>H</sub> × x)

RSCAN0RFDF1xLL: <RSCAN0\_base> + 0E0C<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF1xLH: <RSCAN0\_base> + 0E0D<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF1xHL: <RSCAN0\_base> + 0E0E<sub>H</sub> + (10<sub>H</sub> × x),  
RSCAN0RFDF1xHH: <RSCAN0\_base> + 0E0F<sub>H</sub> + (10<sub>H</sub> × x)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB7[7:0]										RFDB6[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	RFDB5[7:0]										RFDB4[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.47 RSCAN0RFDF1x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB7[7:0]	Receive FIFO Buffer Data Byte 7
23 to 16	RFDB6[7:0]	Receive FIFO Buffer Data Byte 6
15 to 8	RFDB5[7:0]	Receive FIFO Buffer Data Byte 5
7 to 0	RFDB4[7:0]	Receive FIFO Buffer Data Byte 4
		Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCAN0RFPTRx register is smaller than 1000<sub>B</sub>, data bytes for which no data is set are read as 00<sub>H</sub>.

### 19.3.33 RSCAN0CFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 17)

**Access:** RSCAN0CFCCk register can be read or written in 32-bit units.  
 RSCAN0CFCCkL, RSCAN0CFCCkH registers can be read or written in 16-bit units.  
 RSCAN0CFCCkLL, RSCAN0CFCCkLH, RSCAN0CFCCkHL, RSCAN0CFCCkHH registers can be read or written in 8-bit units.

**Address:** RSCAN0CFCCk: <RSCAN0\_base> + 0118<sub>H</sub> + (04<sub>H</sub> × k)

RSCAN0CFCCkL: <RSCAN0\_base> + 0118<sub>H</sub> + (04<sub>H</sub> × k),  
 RSCAN0CFCCkH: <RSCAN0\_base> + 011A<sub>H</sub> + (04<sub>H</sub> × k)

RSCAN0CFCCkLL: <RSCAN0\_base> + 0118<sub>H</sub> + (04<sub>H</sub> × k),  
 RSCAN0CFCCkLH: <RSCAN0\_base> + 0119<sub>H</sub> + (04<sub>H</sub> × k),  
 RSCAN0CFCCkHL: <RSCAN0\_base> + 011A<sub>H</sub> + (04<sub>H</sub> × k),  
 RSCAN0CFCCkHH: <RSCAN0\_base> + 011B<sub>H</sub> + (04<sub>H</sub> × k)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CFITT[7:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
CFTML[3:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIGCV[2:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
CFIM																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
CFDC[2:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
—																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
—																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
—																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
CFTXIE																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
CFRXIE																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W
CFE																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 19.48 RSCAN0CFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00 <sub>H</sub> to FF <sub>H</sub>
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock dividing pclk/2 by (ITRCP [15:0] bits) 1: Clock dividing pclk/2 by (ITRCP [15:0] bits × 10)
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 19.48 RSCAN0CFCCk Register Contents (2/2)

Bit Position	Bit Name	Function																											
12	CFIM	<p>Transmit/Receive FIFO Interrupt Source Select</p> <p>0:</p> <ul style="list-style-type: none"> <li>• Receive mode/gateway mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated.</li> <li>• Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated.</li> </ul> <p>1:</p> <ul style="list-style-type: none"> <li>• Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received.</li> <li>• Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.</li> </ul>																											
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																											
10 to 8	CFDC[2:0]	<p>Transmit/Receive FIFO Buffer Depth Configuration</p> <table> <tr><td>b10</td><td>b9</td><td>b8</td></tr> <tr><td>0</td><td>0</td><td>0: 0 messages</td></tr> <tr><td>0</td><td>0</td><td>1: 4 messages</td></tr> <tr><td>0</td><td>1</td><td>0: 8 messages</td></tr> <tr><td>0</td><td>1</td><td>1: 16 messages</td></tr> <tr><td>1</td><td>0</td><td>0: 32 messages</td></tr> <tr><td>1</td><td>0</td><td>1: 48 messages</td></tr> <tr><td>1</td><td>1</td><td>0: 64 messages</td></tr> <tr><td>1</td><td>1</td><td>1: 128 messages</td></tr> </table>	b10	b9	b8	0	0	0: 0 messages	0	0	1: 4 messages	0	1	0: 8 messages	0	1	1: 16 messages	1	0	0: 32 messages	1	0	1: 48 messages	1	1	0: 64 messages	1	1	1: 128 messages
b10	b9	b8																											
0	0	0: 0 messages																											
0	0	1: 4 messages																											
0	1	0: 8 messages																											
0	1	1: 16 messages																											
1	0	0: 32 messages																											
1	0	1: 48 messages																											
1	1	0: 64 messages																											
1	1	1: 128 messages																											
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																											
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.																											
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.																											
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.																											

### CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01<sub>B</sub> (transmit mode) or 10<sub>B</sub> (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

### CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01<sub>B</sub> (transmit mode) or 10<sub>B</sub> (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[3:0]).

See **Table 19.13** and **Table 19.14**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to  $001_B$  or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the identical channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

### **CFITR Bit**

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the  $\text{pclk}/2$  clock divided by the value of the ITRCP[15:0] bits in the RSCAN0GCFG register.

When this bit is 1, the interval timer clock source is the  $\text{pclk}/2$  clock divided by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register  $\times 10$ ).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

### **CFITSS Bit**

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel to which the FIFO is linked is the count source of the interval timer.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

### **CFM[1:0] Bits**

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

### **CFIGCV[2:0] Bits**

These bits are used to specify the number of received messages for generating a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to  $00_B$  (receive mode) or  $10_B$  (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to  $001_B$  (4 messages), set the CFIGCV[2:0] bits to  $001_B$ ,  $011_B$ ,  $101_B$ , or  $111_B$ .

Modify these bits only in global reset mode.

### **CFIM Bit**

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

### **CFDC[2:0] Bits**

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to  $000_B$ , do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

### **CFTXIE Bit**

When this bit is set to 1 and the CFTXIF flag in the RSCAN0CFSTS<sub>k</sub> register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

**CFRXIE Bit**

When this bit is set to 1 and the CFRXIF flag in the RSCAN0CFSTS<sub>k</sub> register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

**CFE Bit**

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

### 19.3.34 RSCAN0CFSTS<sub>k</sub> — Transmit/receive FIFO Buffer Status Register ( $k = 0$ to 17)

**Access:** RSCAN0CFSTS<sub>k</sub> register can be read or written in 32-bit units.  
 RSCAN0CFSTS<sub>kL</sub> register can be read or written in 16-bit units.  
 RSCAN0CFSTS<sub>kLL</sub> register can be read or written in 8-bit units.  
 RSCAN0CFSTS<sub>kLH</sub> register is a read-only register that can be read in 8-bit units.

**Address:** RSCAN0CFSTS<sub>k</sub>: <RSCAN0\_base> + 0178<sub>H</sub> + (04<sub>H</sub> × k)

RSCAN0CFSTS<sub>kL</sub>: <RSCAN0\_base> + 0178<sub>H</sub> + (04<sub>H</sub> × k)

RSCAN0CFSTS<sub>kLL</sub>: <RSCAN0\_base> + 0178<sub>H</sub> + (04<sub>H</sub> × k),  
 RSCAN0CFSTS<sub>kLH</sub>: <RSCAN0\_base> + 0179<sub>H</sub> + (04<sub>H</sub> × k)

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W <sup>*1</sup>	R/W <sup>*1</sup>	R/W <sup>*1</sup>	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 19.49 RSCAN0CFSTS<sub>k</sub> Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

#### CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the RSCAN0CFCCk register.

- When CFM[1:0] value is 01<sub>B</sub> (transmit mode): Number of untransmitted messages in the buffer

- When CFM[1:0] value is  $00_B$  (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is  $10_B$  (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is  $00_B$ : In global reset mode
- When CFM[1:0] value is  $01_B$  or  $10_B$ : In channel reset mode

### **CFTXIF Flag**

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to  $01_B$  or  $10_B$ , and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to  $00_B$ : In global reset mode
- When the CFM[1:0] bits are set to  $01_B$  or  $10_B$ : In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **CFRXIF Flag**

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to  $00_B$  or  $10_B$ , and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to  $00_B$ : In global reset mode
- When the CFM[1:0] bits are set to  $01_B$  or  $10_B$ : In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **CFMLT Flag**

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag

- When the CFM[1:0] bits are set to  $00_B$ : In global reset mode
- When the CFM[1:0] bits are set to  $01_B$  or  $10_B$ : In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **CFFLL Flag**

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCAN0CFCCk register.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCAN0CFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to  $00_B$ : In global reset mode
- When the CFM[1:0] bits are set to  $01_B$  or  $10_B$ : In channel reset mode

### **CFEMP Flag**

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to  $00_B$ : All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to  $01_B$  or  $10_B$ : All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to  $00_B$  or  $10_B$ : At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to  $01_B$ : A value of  $FF_H$  has been written to the RSCAN0CFPCTRk register after data was written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers.

#### **NOTE**

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

### 19.3.35 RSCAN0CFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 17)

**Access:** RSCAN0CFPCTRk register is a write-only register that can be written in 32-bit units.  
RSCAN0CFPCTRKL register is a write-only register that can be written in 16-bit units.  
RSCAN0CFPCTRkLL register is a write-only register that can be written in 8-bit units.

**Address:** RSCAN0CFPCTRk: <RSCAN0\_base> + 01D8<sub>H</sub> + (04<sub>H</sub> × k)

RSCAN0CFPCTRKL: <RSCAN0\_base> + 01D8<sub>H</sub> + (04<sub>H</sub> × k)

RSCAN0CFPCTRkLL: <RSCAN0\_base> + 01D8<sub>H</sub> + (04<sub>H</sub> × k)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 19.50 RSCAN0CFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	CFPC[7:0]	<p>Transmit/Receive FIFO Pointer Control</p> <ul style="list-style-type: none"> <li>Receive mode: Writing FF<sub>H</sub> to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer.</li> <li>Transmit mode: Writing FF<sub>H</sub> to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer.</li> <li>Gateway mode: Setting prohibited</li> </ul>

#### CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCAN0CFCCk register is 00<sub>B</sub>):  
Writing FF<sub>H</sub> to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCAN0CFSTS<sub>k</sub> register is decremented. Read the RSCAN0CFID<sub>k</sub>, RSCAN0CFPTR<sub>k</sub>, RSCAN0CFDF0<sub>k</sub>, and RSCAN0CFDF1<sub>k</sub> registers to read messages from the transmit/receive FIFO buffer, and then write FF<sub>H</sub> to the CFPC[7:0] bits.  
When writing FF<sub>H</sub> to these bits, make sure that the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCAN0CFSTS<sub>k</sub> register is 0 (the transmit/receive FIFO buffer contains messages).

- Transmit mode (CFM[1:0] value in the RSCAN0CFCCk register is  $01_B$ ):  
Writing  $FF_H$  to the CFPC[7:0] bits stores the data written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented. Write transmit messages to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers before writing  $FF_H$  to the CFPC[7:0] bits.  
When writing  $FF_H$  to these bits, make sure that the CFE bit in the RSCAN0CFCCk register is set to 1 and the CFFLL flag in the RSCAN0CFSTS<sub>k</sub> register is 0 (the transmit/receive FIFO buffer is not full).
- Gateway mode (CFM[1:0] value in the RSCAN0CFCCk register is  $10_B$ ):  
Setting prohibited

### 19.3.36 RSCAN0CFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 17)

**Access:** RSCAN0CFIDk register can be read or written in 32-bit units.  
 RSCAN0CFIDkL, RSCAN0CFIDkH registers can be read or written in 16-bit units.  
 RSCAN0CFIDkLL, RSCAN0CFIDkLH, RSCAN0CFIDkHL, RSCAN0CFIDkHH registers can be read or written in 8-bit units.

**Address:** RSCAN0CFIDk: <RSCAN0\_base> + 0E80<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFIDkL: <RSCAN0\_base> + 0E80<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFIDkH: <RSCAN0\_base> + 0E82<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFIDkLL: <RSCAN0\_base> + 0E80<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFIDkLH: <RSCAN0\_base> + 0E81<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFIDkHL: <RSCAN0\_base> + 0E82<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFIDkHH: <RSCAN0\_base> + 0E83<sub>H</sub> + (10<sub>H</sub> × k)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.51 RSCAN0CFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01 <sub>B</sub> (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data <ul style="list-style-type: none"> <li>When CFM[1:0] value is 01<sub>B</sub> (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.</li> <li>When CFM[1:0] value is 00<sub>B</sub> (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.</li> </ul>

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01<sub>B</sub> (transmit mode). This register is readable only when the CFM[1:0] value is 00<sub>B</sub> (receive mode). This RSCAN0CFIDk register should not be read or written when the CFM[1:0] value is 10<sub>B</sub> (gateway mode).

**CFIDE Bit**

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is  $00_B$ . When the CFM[1:0] value is  $01_B$ , these bits are used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

**CFRTR Bit**

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is  $00_B$ . When the CFM[1:0] value is  $01_B$ , this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

**THLEN Bit**

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is  $01_B$  (transmit mode).

**CFID[28:0] Bits**

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is  $00_B$ .

When the CFM[1:0] value is  $01_B$ , this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

### 19.3.37 RSCAN0CFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 17)

**Access:** RSCAN0CFPTRk register can be read or written in 32-bit units.  
 RSCAN0CFPTRKL, RSCAN0CFPTRKH registers can be read or written in 16-bit units.  
 RSCAN0CFPTRKLL, RSCAN0CFPTRKLH, RSCAN0CFPTRKHL, RSCAN0CFPTRKHH registers can be read or written in 8-bit units.

**Address:** RSCAN0CFPTRk: <RSCAN0\_base> + 0E84<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFPTRKL: <RSCAN0\_base> + 0E84<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFPTRKH: <RSCAN0\_base> + 0E86<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFPTRKLL: <RSCAN0\_base> + 0E84<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFPTRKLH: <RSCAN0\_base> + 0E85<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFPTRKHL: <RSCAN0\_base> + 0E86<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFPTRKHH: <RSCAN0\_base> + 0E87<sub>H</sub> + (10<sub>H</sub> × k)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.52 RSCAN0CFPTRk Register Contents

Bit Position	Bit Name	Function
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data <ul style="list-style-type: none"> <li>When CFM[1:0] value is 01<sub>B</sub> (transmit mode):            Set the label information to be stored in the transmit history buffer.            Only bits CFPTR[7:0] are valid.</li> <li>When CFM[1:0] value is 00<sub>B</sub> (receive mode):            The label information of the received message can be read.</li> </ul>
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00 <sub>B</sub> (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01<sub>B</sub> (transmit mode). This register is readable only when the CFM[1:0] value is 00<sub>B</sub> (receive mode). This register should not be read or written when the CFM[1:0] value is 10<sub>B</sub> (gateway mode).

**CFDLC[3:0] Bits**

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00<sub>B</sub>. When the CFM[1:0] value is 01<sub>B</sub>, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If the data length is set to 9 bytes or more, the actual transmit data defaults to 8 bytes.

**CFPTR[11:0] Bits**

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00<sub>B</sub>. When the CFM[1:0] value is 01<sub>B</sub>, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

**CFTS[15:0] Bits**

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00<sub>B</sub>.

### 19.3.38 RSCAN0CFDF0k — Transmit/receive FIFO Buffer Access Data Field 0 Register (k = 0 to 17)

**Access:** RSCAN0CFDF0k register can be read or written in 32-bit units.  
 RSCAN0CFDF0kL, RSCAN0CFDF0kH registers can be read or written in 16-bit units.  
 RSCAN0CFDF0kLL, RSCAN0CFDF0kLH, RSCAN0CFDF0kHL, RSCAN0CFDF0kHH registers can be read or written in 8-bit units.

**Address:** RSCAN0CFDF0k: <RSCAN0\_base> + 0E88<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFDF0kL: <RSCAN0\_base> + 0E88<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF0kH: <RSCAN0\_base> + 0E8A<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFDF0kLL: <RSCAN0\_base> + 0E88<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF0kLH: <RSCAN0\_base> + 0E89<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF0kHL: <RSCAN0\_base> + 0E8A<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF0kHH: <RSCAN0\_base> + 0E8B<sub>H</sub> + (10<sub>H</sub> × k)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB3[7:0]								CFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB1[7:0]								CFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.53 RSCAN0CFDF0k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB3[7:0]	Transmit/Receive FIFO Buffer Data Byte 3
23 to 16	CFDB2[7:0]	Transmit/Receive FIFO Buffer Data Byte 2
15 to 8	CFDB1[7:0]	Transmit/Receive FIFO Buffer Data Byte 1
7 to 0	CFDB0[7:0]	<ul style="list-style-type: none"> <li>When CFM[1:0] value is 01<sub>B</sub> (transmit mode): Set the transmit/receive FIFO buffer data.</li> <li>When CFM[1:0] value is 00<sub>B</sub> (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.</li> </ul>

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01<sub>B</sub> (transmit mode).

This register is readable only when the CFM[1:0] value is 00<sub>B</sub> (receive mode). When the CFDLC[3:0] value in the RSCAN0CFPTRk register is smaller than 1000<sub>B</sub>, data bytes for which no data is set are read as 00<sub>H</sub>.

This register should not be read or written when the CFM[1:0] value is 10<sub>B</sub> (gateway mode).

### 19.3.39 RSCAN0CFDF1k — Transmit/receive FIFO Buffer Access Data Field 1 Register (k = 0 to 17)

**Access:** RSCAN0CFDF1k register can be read or written in 32-bit units.  
 RSCAN0CFDF1KL, RSCAN0CFDF1kH registers can be read or written in 16-bit units.  
 RSCAN0CFDF1KLL, RSCAN0CFDF1kLH, RSCAN0CFDF1kHL, RSCAN0CFDF1kHH registers can be read or written in 8-bit units.

**Address:** RSCAN0CFDF1k: <RSCAN0\_base> + 0E8C<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFDF1KL: <RSCAN0\_base> + 0E8C<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF1KH: <RSCAN0\_base> + 0E8E<sub>H</sub> + (10<sub>H</sub> × k)

RSCAN0CFDF1KLL: <RSCAN0\_base> + 0E8C<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF1kLH: <RSCAN0\_base> + 0E8D<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF1kHL: <RSCAN0\_base> + 0E8E<sub>H</sub> + (10<sub>H</sub> × k),  
 RSCAN0CFDF1kHH: <RSCAN0\_base> + 0E8F<sub>H</sub> + (10<sub>H</sub> × k)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB7[7:0]								CFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB5[7:0]								CFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 19.54 RSCAN0CFDF1k Register Contents**

Bit Position	Bit Name	Function
31 to 24	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7
23 to 16	CFDB6[7:0]	Transmit/Receive FIFO Buffer Data Byte 6
15 to 8	CFDB5[7:0]	Transmit/Receive FIFO Buffer Data Byte 5
7 to 0	CFDB4[7:0]	<ul style="list-style-type: none"> <li>When CFM[1:0] value is 01<sub>B</sub> (transmit mode): Set the transmit/receive FIFO buffer data.</li> <li>When CFM[1:0] value is 00<sub>B</sub> (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.</li> </ul>

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01<sub>B</sub> (transmit mode).

This register is readable only when the CFM[1:0] value is 00<sub>B</sub> (receive mode). When the CFDLC[3:0] value in the RSCAN0CFPTRk register is smaller than 1000<sub>B</sub>, data bytes for which no data is set are read as 00<sub>H</sub>.

This register should not be read or written when the CFM[1:0] value is 10<sub>B</sub> (gateway mode).

### 19.3.40 RSCAN0FESTS — FIFO Empty Status Register

**Access:** RSCAN0FESTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0FESTSL, RSCAN0FESTSH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0FESTSLL, RSCAN0FESTSLH, RSCAN0FESTSHL, RSCAN0FESTSHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0FESTS: <RSCAN0\_base> + 0238<sub>H</sub>  
 RSCAN0FESTSL: <RSCAN0\_base> + 0238<sub>H</sub>, RSCAN0FESTSH: <RSCAN0\_base> + 023A<sub>H</sub>

RSCAN0FESTSLL: <RSCAN0\_base> + 0238<sub>H</sub>, RSCAN0FESTSLH: <RSCAN0\_base> + 0239<sub>H</sub>,  
 RSCAN0FESTSHL: <RSCAN0\_base> + 023A<sub>H</sub>, RSCAN0FESTSHH: <RSCAN0\_base> + 023B<sub>H</sub>

**Value after reset:** 03FF FFFF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CF17EMP	CF16EMP	CF15EMP	CF14EMP	CF13EMP	CF12EMP	CF11EMP	CF10EMP	CF9EMP	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.55 RSCAN0FESTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CF17EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 17)
24	CF16EMP	
23	CF15EMP	
22	CF14EMP	
21	CF13EMP	
20	CF12EMP	
19	CF11EMP	
18	CF10EMP	
17	CF9EMP	
16	CF8EMP	
15	CF7EMP	
14	CF6EMP	
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	

**Table 19.55 RSCAN0FESTS Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
7	RF7EMP	Receive FIFO Buffer Empty Status Flag
6	RF6EMP	0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty).
5	RF5EMP	(x = 0 to 7)
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCAN0FESTS register is set to 03FF FFFF<sub>H</sub> in global reset mode.

#### **CFkEMP Flag (k = 0 to 17)**

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCAN0CFSTS<sub>k</sub> register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

#### **RFxEMP Flag (x = 0 to 7)**

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCAN0RFSTS<sub>x</sub> register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

### 19.3.41 RSCAN0FFSTS — FIFO Full Status Register

**Access:** RSCAN0FFSTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0FFSTSL, RSCAN0FFSTSH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0FFSTSLL, RSCAN0FFSTSLH, RSCAN0FFSTSHL, RSCAN0FFSTSHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0FFSTS: <RSCAN0\_base> + 023C<sub>H</sub>  
 RSCAN0FFSTSL: <RSCAN0\_base> + 023C<sub>H</sub>, RSCAN0FFSTSH: <RSCAN0\_base> + 023E<sub>H</sub>

RSCAN0FFSTSLL: <RSCAN0\_base> + 023C<sub>H</sub>, RSCAN0FFSTSLH: <RSCAN0\_base> + 023D<sub>H</sub>,  
 RSCAN0FFSTSHL: <RSCAN0\_base> + 023E<sub>H</sub>, RSCAN0FFSTSHH: <RSCAN0\_base> + 023F<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CF17FL L	CF16FL L	CF15FL L	CF14FL L	CF13FL L	CF12FL L	CF11FL L	CF10FL L	CF9FLL	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.56 RSCAN0FFSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CF17FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full.
24	CF16FLL	(k = 0 to 17)
23	CF15FLL	
22	CF14FLL	
21	CF13FLL	
20	CF12FLL	
19	CF11FLL	
18	CF10FLL	
17	CF9FLL	
16	CF8FLL	
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	

**Table 19.56 RSCAN0FFSTS Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full.
6	RF6FLL	(x = 0 to 7)
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCAN0FFSTS register is cleared to 0000 0000<sub>H</sub> in global reset mode.

#### **CFkFLL Flag (k = 0 to 17)**

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCAN0CFSTS<sub>k</sub> register is set to 1 (the transmit/receive FIFO buffer is full).

When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

#### **RFxFLL Flag (x = 0 to 7)**

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCAN0RFSTS<sub>x</sub> register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

### 19.3.42 RSCAN0FMSTS — FIFO Message Lost Status Register

**Access:** RSCAN0FMSTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0FMSTS<sub>L</sub>, RSCAN0FMSTS<sub>H</sub> registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0FMSTS<sub>SL</sub>, RSCAN0FMSTS<sub>SH</sub>, RSCAN0FMSTS<sub>SLH</sub>, RSCAN0FMSTS<sub>SHH</sub> registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0FMSTS: <RSCAN0\_base> + 0240<sub>H</sub>  
 RSCAN0FMSTS<sub>L</sub>: <RSCAN0\_base> + 0240<sub>H</sub>, RSCAN0FMSTS<sub>H</sub>: <RSCAN0\_base> + 0242<sub>H</sub>  
 RSCAN0FMSTS<sub>SL</sub>: <RSCAN0\_base> + 0240<sub>H</sub>, RSCAN0FMSTS<sub>SH</sub>: <RSCAN0\_base> + 0241<sub>H</sub>,  
 RSCAN0FMSTS<sub>SLH</sub>: <RSCAN0\_base> + 0242<sub>H</sub>, RSCAN0FMSTS<sub>SHH</sub>: <RSCAN0\_base> + 0243<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CF17MLT	CF16MLT	CF15MLT	CF14MLT	CF13MLT	CF12MLT	CF11MLT	CF10MLT	CF9MLT	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.57 RSCAN0FMSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CF17MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 17)
24	CF16MLT	
23	CF15MLT	
22	CF14MLT	
21	CF13MLT	
20	CF12MLT	
19	CF11MLT	
18	CF10MLT	
17	CF9MLT	
16	CF8MLT	
15	CF7MLT	
14	CF6MLT	
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	

**Table 19.57 RSCAN0FMSTS Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost.
6	RF6MLT	(x = 0 to 7)
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCAN0FMSTS register is cleared to 0000 0000<sub>H</sub> in global reset mode.

#### **CFkMLT Flag (k = 0 to 17)**

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCAN0CFSTS<sub>k</sub> register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

#### **RFxMLT Flag (x = 0 to 7)**

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCAN0RFSTS<sub>x</sub> register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

### 19.3.43 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register

**Access:** RSCAN0RFISTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0RFISTSL register is a read-only register that can be read in 16-bit units.  
 RSCAN0RFISTSLL register is a read-only register that can be read in 8-bit units.

**Address:** RSCAN0RFISTS: <RSCAN0\_base> + 0244<sub>H</sub>  
 RSCAN0RFISTSL: <RSCAN0\_base> + 0244<sub>H</sub>  
 RSCAN0RFISTSLL: <RSCAN0\_base> + 0244<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.58 RSCAN0RFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag
6	RF6IF	0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCAN0RFISTS register is cleared to 0000 0000<sub>H</sub> in global reset mode.

#### RF<sub>x</sub>IF Flag (x = 0 to 7)

The RF<sub>x</sub>IF flag is set to 1 when the RFIF flag in the RSCAN0RFSTS<sub>x</sub> register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RF<sub>x</sub>IF flag is cleared to 0.

### 19.3.44 RSCAN0CFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

**Access:** RSCAN0CFRISTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0CFRISTSL, RSCAN0CFRISTSH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0CFRISTSLL, RSCAN0CFRISTSLH, RSCAN0CFRISTSHL registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0CFRISTS: <RSCAN0\_base> + 0248H

RSCAN0CFRISTSL: <RSCAN0\_base> + 0248H, RSCAN0CFRISTSH: <RSCAN0\_base> + 024AH

RSCAN0CFRISTSLL: <RSCAN0\_base> + 0248H, RSCAN0CFRISTSLH: <RSCAN0\_base> + 0249H,  
 RSCAN0CFRISTSHL: <RSCAN0\_base> + 024AH

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF17RXIF	CF16RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15RXIF	CF14RXIF	CF13RXIF	CF12RXIF	CF11RXIF	CF10RXIF	CF9RXIF	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.59 RSCAN0CFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	CF17RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 17)
16	CF16RXIF	
15	CF15RXIF	
14	CF14RXIF	
13	CF13RXIF	
12	CF12RXIF	
11	CF11RXIF	
10	CF10RXIF	
9	CF9RXIF	
8	CF8RXIF	
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCAN0CFRISTS register is cleared to 0000 0000H in global reset mode.

**CFkRXIF Flag (k = 0 to 17)**

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCAN0CFSTS $k$  register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

### 19.3.45 RSCAN0CFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

**Access:** RSCAN0CFTISTS register is a read-only register that can be read in 32-bit units.  
 RSCAN0CFTISTSL, RSCAN0CFTISTSH registers are the read-only registers that can be read in 16-bit units.  
 RSCAN0CFTISTSSL, RSCAN0CFTISTSLH, RSCAN0CFTISTSHL registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0CFTISTS: <RSCAN0\_base> + 024C<sub>H</sub>

RSCAN0CFTISTSL: <RSCAN0\_base> + 024C<sub>H</sub>, RSCAN0CFTISTSH: <RSCAN0\_base> + 024E<sub>H</sub>

RSCAN0CFTISTSSL: <RSCAN0\_base> + 024C<sub>H</sub>, RSCAN0CFTISTSLH: <RSCAN0\_base> + 024D<sub>H</sub>,  
 RSCAN0CFTISTSHL: <RSCAN0\_base> + 024E<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF17TXIF	CF16TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15TXIF	CF14TXIF	CF13TXIF	CF12TXIF	CF11TXIF	CF10TXIF	CF9TXIF	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.60 RSCAN0CFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	CF17TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 17)
16	CF16TXIF	
15	CF15TXIF	
14	CF14TXIF	
13	CF13TXIF	
12	CF12TXIF	
11	CF11TXIF	
10	CF10TXIF	
9	CF9TXIF	
8	CF8TXIF	
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCAN0CFTISTS register is cleared to 0000 0000<sub>H</sub> in global reset mode.

**CFkTXIF Flag (k = 0 to 17)**

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCAN0CFSTS $k$  register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

**19.3.46 RSCAN0TMC $p$  — Transmit Buffer Control Register ( $p$  = 0 to 95)**

**Access:** RSCAN0TMC $p$  register can be read or written in 8-bit units.

**Address:** RSCAN0TMC $p$ : <RSCAN0\_base> + 0250H + (01H × p)

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W* <sup>1</sup>	R/W* <sup>1</sup>

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

**Table 19.61 RSCAN0TMC $p$  Register Contents**

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCAN0TMC $p$  register meets any of the following conditions, set it to 00H.

- The RSCAN0TMC $p$  register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCAN0CFCC $k$  register ( $p$  =  $m \times 16 +$  the value of CFTML[3:0] bits).
- The RSCAN0TMC $p$  register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCAN0TXQCC $m$  ( $m$  = 0 to 5) register ( $p$  =  $(m \times 16 + 15)$  to  $(m \times 16 + 15 -$  the value of TXQDC[3:0] bits)).

Bits in the RSCAN0TMC $p$  register are all cleared to 0 in channel reset mode. Modify the RSCAN0TMC $p$  register in channel communication mode or channel halt mode.

**TMOM Bit**

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCAN0TMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

### TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

### TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] flags in the RSCAN0TMSTSp register is 00<sub>B</sub>.

### 19.3.47 RSCAN0TMSTSp — Transmit Buffer Status Register ( $p = 0$ to 95)

**Access:** RSCAN0TMSTSp register can be read or written in 8-bit units.

**Address:** RSCAN0TMSTSp: <RSCAN0\_base> + 02D0H + (01H × p)

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]	TMSTS	
Value after reset	0	0	0	0	0	0	0	0

R/W	0	R	R	R	R	R/W	R/W	R
-----	---	---	---	---	---	-----	-----	---

Table 19.62 RSCAN0TMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

The RSCAN0TMSTSp register is cleared to all 0 in channel reset mode.

#### TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCAN0TMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCAN0TMCp register is set to 0.

#### TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCAN0TMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCAN0TMCp register is set to 0.

#### TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00<sub>B</sub>: Transmission is in progress or no transmit request is present.

01<sub>B</sub>: Transmission from the transmit buffer was aborted.

10<sub>B</sub>: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 0 (transmit abort is not requested).

11<sub>B</sub>: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 1 (transmit abort is requested).

Write  $00_B$  to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than  $00_B$  to this flag.

### **TMTSTS Flag**

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

### 19.3.48 RSCAN0TMTRSTS $y$ — Transmit Buffer Transmit Request Status Register ( $y = 0$ to $2$ )

**Access:** RSCAN0TMTRSTS $y$  register is a read-only register that can be read in 32-bit units.  
 RSCAN0TMTRSTS $y$ L, RSCAN0TMTRSTS $y$ H registers are the read-only registers that can be read in 16-bit units.

RSCAN0TMTRSTS $y$ LL, RSCAN0TMTRSTS $y$ LH, RSCAN0TMTRSTS $y$ HL, RSCAN0TMTRSTS $y$ HH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0TMTRSTS $y$ : <RSCAN0\_base> + 0350<sub>H</sub> + (04<sub>H</sub> ×  $y$ )

RSCAN0TMTRSTS $y$ L: <RSCAN0\_base> + 0350<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
 RSCAN0TMTRSTS $y$ H: <RSCAN0\_base> + 0352<sub>H</sub> + (04<sub>H</sub> ×  $y$ )

RSCAN0TMTRSTS $y$ LL: <RSCAN0\_base> + 0350<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
 RSCAN0TMTRSTS $y$ LH: <RSCAN0\_base> + 0351<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
 RSCAN0TMTRSTS $y$ HL: <RSCAN0\_base> + 0352<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
 RSCAN0TMTRSTS $y$ HH: <RSCAN0\_base> + 0353<sub>H</sub> + (04<sub>H</sub> ×  $y$ )

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMTRSTS $p$ ( $p = y \times 32 + 31$ to $y \times 32 + 16$ ( $y = 0, 1, 2$ ))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
TMTRSTS $p$ ( $p = y \times 32 + 15$ to $y \times 32 + 0$ ( $y = 0, 1, 2$ ))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.63 RSCAN0TMTRSTS $y$  Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTS $p$	Transmit Buffer Transmit Request Status Flag $p$ ( $p = y \times 32 + 31$ to $y \times 32 + 16$ ) 0: No transmit request is present. 1: A transmit request is present.
15 to 0	TMTRSTS $p$	Transmit Buffer Transmit Request Status Flag $p$ ( $p = y \times 32 + 15$ to $y \times 32 + 0$ ) 0: No transmit request is present. 1: A transmit request is present.

#### TMTRSTS $p$ Flags ( $p = 0$ to $95$ )

These flags indicate the status of the TMTR bit in the RSCAN0TMC $p$  register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTS $p$  flag is set to 1.

The corresponding TMTRSTS $p$  flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 19.64 shows the bit assignment.

**Table 19.64 TMTRSTSp Bit Assignment**

<b>Bit</b>	<b>Channel</b>	<b>Transmit Buffer Number</b>
0	0	0
1	0	1
.	.	.
15	0	15
16	1	0
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
47	2	15
48	3	0
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
78	4	14
79	4	15
80	5	0
81	5	1
.	.	.
94	5	14
95	5	15

### 19.3.49 RSCAN0TMTARSTS<sub>y</sub> — Transmit Buffer Transmit Abort Request Status Register ( $y = 0$ to $2$ )

**Access:** RSCAN0TMTARSTS<sub>y</sub> register is a read-only register that can be read in 32-bit units. RSCAN0TMTARSTS<sub>yL</sub>, RSCAN0TMTARSTS<sub>yH</sub> registers are the read-only registers that can be read in 16-bit units.

RSCAN0TMTARSTS<sub>yLL</sub>, RSCAN0TMTARSTS<sub>yLH</sub>, RSCAN0TMTARSTS<sub>yHL</sub>, RSCAN0TMTARSTS<sub>yHH</sub> registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0TMTARSTS<sub>y</sub>: <RSCAN0\_base> + 0360<sub>H</sub> + (04<sub>H</sub> ×  $y$ )

RSCAN0TMTARSTS<sub>yL</sub>: <RSCAN0\_base> + 0360<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
RSCAN0TMTARSTS<sub>yH</sub>: <RSCAN0\_base> + 0362<sub>H</sub> + (04<sub>H</sub> ×  $y$ )

RSCAN0TMTARSTS<sub>yLL</sub>: <RSCAN0\_base> + 0360<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
RSCAN0TMTARSTS<sub>yLH</sub>: <RSCAN0\_base> + 0361<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
RSCAN0TMTARSTS<sub>yHL</sub>: <RSCAN0\_base> + 0362<sub>H</sub> + (04<sub>H</sub> ×  $y$ ),  
RSCAN0TMTARSTS<sub>yHH</sub>: <RSCAN0\_base> + 0363<sub>H</sub> + (04<sub>H</sub> ×  $y$ )

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMTARSTS <sub>p</sub> ( $p = y \times 32 + 31$ to $y \times 32 + 16$ ( $y = 0, 1, 2$ ))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMTARSTS <sub>p</sub> ( $p = y \times 32 + 15$ to $y \times 32 + 0$ ( $y = 0, 1, 2$ ))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 19.65 RSCAN0TMTARSTS<sub>y</sub> Register Contents**

Bit Position	Bit Name	Function
31 to 16	TMTARSTS <sub>p</sub>	Transmit Buffer Transmit Abort Request Status Flag $p$ ( $p = y \times 32 + 31$ to $y \times 32 + 16$ ) 0: No transmit abort request is present. 1: A transmit abort request is present.
15 to 0	TMTARSTS <sub>p</sub>	Transmit Buffer Transmit Abort Request Status Flag $p$ ( $p = y \times 32 + 15$ to $y \times 32 + 0$ ) 0: No transmit abort request is present. 1: A transmit abort request is present.

#### TMTARSTS<sub>p</sub> Flags ( $p = 0$ to $95$ )

These flags indicate the status of the TMTAR bit in the RSCAN0TMC<sub>p</sub> register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTS<sub>p</sub> flag is set to 1.

The corresponding TMTARSTS<sub>p</sub> flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

**Table 19.66** shows the bit assignment.

**Table 19.66 TMTARSTSp Bit Assignment**

<b>Bit</b>	<b>Channel</b>	<b>Transmit Buffer Number</b>
0	0	0
1	0	1
.	.	.
15	0	15
16	1	0
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
47	2	15
48	3	0
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
78	4	14
79	4	15
80	5	0
81	5	1
.	.	.
94	5	14
95	5	15

### 19.3.50 RSCAN0TMTCSy — Transmit Buffer Transmit Complete Status Register (y = 0 to 2)

**Access:** RSCAN0TMTCSy register is a read-only register that can be read in 32-bit units. RSCAN0TMTCSyL, RSCAN0TMTCSyH registers are the read-only registers that can be read in 16-bit units.

RSCAN0TMTCSyLL, RSCAN0TMTCSyLH, RSCAN0TMTCSyHL, RSCAN0TMTCSyHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0TMTCSy: <RSCAN0\_base> + 0370<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0TMTCSyL: <RSCAN0\_base> + 0370<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTCSyH: <RSCAN0\_base> + 0372<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0TMTCSyLL: <RSCAN0\_base> + 0370<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTCSyLH: <RSCAN0\_base> + 0371<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTCSyHL: <RSCAN0\_base> + 0372<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTCSyHH: <RSCAN0\_base> + 0373<sub>H</sub> + (04<sub>H</sub> × y)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 19.67 RSCAN0TMTCSy Register Contents**

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed.
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

#### TMTCSTSp Flags (p = 0 to 95)

When the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 10<sub>B</sub> (transmission has been completed (without transmit abort request)) or 11<sub>B</sub> (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

A TMTCSTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00<sub>B</sub> or in channel reset mode.

**Table 19.68** shows the bit assignment.

**Table 19.68 TMTCSTSp Bit Assignment**

<b>Bit</b>	<b>Channel</b>	<b>Transmit Buffer Number</b>
0	0	0
1	0	1
.	.	.
15	0	15
16	1	0
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
47	2	15
48	3	0
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
78	4	14
79	4	15
80	5	0
81	5	1
.	.	.
94	5	14
95	5	15

### 19.3.51 RSCAN0TMTASTSy — Transmit Buffer Transmit Abort Status Register (y = 0 to 2)

**Access:** RSCAN0TMTASTSy register is a read-only register that can be read in 32-bit units.  
RSCAN0TMTASTSyL, RSCAN0TMTASTSyH registers are the read-only registers that can be read in 16-bit units.

RSCAN0TMTASTSyLL, RSCAN0TMTASTSyLH, RSCAN0TMTASTSyHL, RSCAN0TMTASTSyHH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0TMTASTSy: <RSCAN0\_base> + 0380<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0TMTASTSyL: <RSCAN0\_base> + 0380<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTASTSyH: <RSCAN0\_base> + 0382<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0TMTASTSyLL: <RSCAN0\_base> + 0380<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTASTSyLH: <RSCAN0\_base> + 0381<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTASTSyHL: <RSCAN0\_base> + 0382<sub>H</sub> + (04<sub>H</sub> × y),  
RSCAN0TMTASTSyHH: <RSCAN0\_base> + 0383<sub>H</sub> + (04<sub>H</sub> × y)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMTASTSp (p = y × 32 + 31 to y × 32 × 16 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 19.69 RSCAN0TMTASTSy Register Contents**

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

#### TMTASTSp Flags (p = 0 to 95)

When the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 01<sub>B</sub> (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

A TMTASTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00<sub>B</sub> or in channel reset mode.

**Table 19.70** shows the bit assignment.

**Table 19.70 TMTASTSp Bit Assignment**

<b>Bit</b>	<b>Channel</b>	<b>Transmit Buffer Number</b>
0	0	0
1	0	1
.	.	.
15	0	15
16	1	0
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
47	2	15
48	3	0
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
78	4	14
79	4	15
80	5	0
81	5	1
.	.	.
94	5	14
95	5	15

### 19.3.52 RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register (y = 0 to 2)

**Access:** RSCAN0TMIECy register can be read or written in 32-bit units.  
 RSCAN0TMIECyl, RSCAN0TMIECyH registers can be read or written in 16-bit units.  
 RSCAN0TMIECyll, RSCAN0TMIECyLH, RSCAN0TMIECyHL, RSCAN0TMIECyHH registers can be read or written in 8-bit units.

**Address:** RSCAN0TMIECy: <RSCAN0\_base> + 0390<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0TMIECyl: <RSCAN0\_base> + 0390<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0TMIECyH: <RSCAN0\_base> + 0392<sub>H</sub> + (04<sub>H</sub> × y)

RSCAN0TMIECyll: <RSCAN0\_base> + 0390<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0TMIECyLH: <RSCAN0\_base> + 0391<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0TMIECyHL: <RSCAN0\_base> + 0392<sub>H</sub> + (04<sub>H</sub> × y),  
 RSCAN0TMIECyHH: <RSCAN0\_base> + 0393<sub>H</sub> + (04<sub>H</sub> × y)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 19.71 RSCAN0TMIECy Register Contents**

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

#### TMIEp Bits (p = 0 to 95)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCAN0TMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

**Table 19.72** shows the bit assignment.

**Table 19.72 TMIEp Bit Assignment**

<b>Bit</b>	<b>Channel</b>	<b>Transmit Buffer Number</b>
0	0	0
1	0	1
.	.	.
15	0	15
16	1	0
.	.	.
30	1	14
31	1	15
32	2	0
33	2	1
.	.	.
47	2	15
48	3	0
.	.	.
62	3	14
63	3	15
64	4	0
65	4	1
.	.	.
78	4	14
79	4	15
80	5	0
81	5	1
.	.	.
94	5	14
95	5	15

### 19.3.53 RSCAN0TMIDp — Transmit Buffer ID Register (p = 0 to 95)

**Access:** RSCAN0TMIDp register can be read or written in 32-bit units.  
 RSCAN0TMIDpL, RSCAN0TMIDpH registers can be read or written in 16-bit units.  
 RSCAN0TMIDpLL, RSCAN0TMIDpLH, RSCAN0TMIDpHL, RSCAN0TMIDpHH registers can be read or written in 8-bit units.

**Address:** RSCAN0TMIDp: <RSCAN0\_base> + 1000<sub>H</sub> + (10<sub>H</sub> × p)

RSCAN0TMIDpL: <RSCAN0\_base> + 1000<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMIDpH: <RSCAN0\_base> + 1002<sub>H</sub> + (10<sub>H</sub> × p)

RSCAN0TMIDpLL: <RSCAN0\_base> + 1000<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMIDpLH: <RSCAN0\_base> + 1001<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMIDpHL: <RSCAN0\_base> + 1002<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMIDpHH: <RSCAN0\_base> + 1003<sub>H</sub> + (10<sub>H</sub> × p)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.73 RSCAN0TMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTS<sub>p</sub> register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

#### TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

#### TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

**THLEN Bit**

With this bit set to 1, the transmit history data of the message transmitted (label information and the number and type) are stored in the transmit history buffer after transmission is completed.

**TMID[28:0] Bits**

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

### 19.3.54 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 95)

**Access:** RSCAN0TMPTRp register can be read or written in 32-bit units.  
 RSCAN0TMPTRpH register can be read or written in 16-bit units.  
 RSCAN0TMPTRpHL, RSCAN0TMPTRpHH registers can be read or written in 8-bit units.

**Address:** RSCAN0TMPTRp: <RSCAN0\_base> + 1004<sub>H</sub> + (10<sub>H</sub> × p)  
 RSCAN0TMPTRpH: <RSCAN0\_base> + 1006<sub>H</sub> + (10<sub>H</sub> × p)  
 RSCAN0TMPTRpHL: <RSCAN0\_base> + 1006<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMPTRpHH: <RSCAN0\_base> + 1007<sub>H</sub> + (10<sub>H</sub> × p)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMDLC[3:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W							
TMPTR[7:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.74 RSCAN0TMPTRp Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

#### TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCAN0TMIDp register is set to 0 (data frame). If the data length is set to 9 bytes or more, the transmit data is 8 bytes long.

When the TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

**TMPTR[7:0] Bits**

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

### 19.3.55 RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register (p = 0 to 95)

**Access:** RSCAN0TMDF0p register can be read or written in 32-bit units.  
 RSCAN0TMDF0pL, RSCAN0TMDF0pH registers can be read or written in 16-bit units.  
 RSCAN0TMDF0pLL, RSCAN0TMDF0pLH, RSCAN0TMDF0pHL, RSCAN0TMDF0pHH registers can be read or written in 8-bit units.

**Address:** RSCAN0TMDF0p: <RSCAN0\_base> + 1008<sub>H</sub> + (10<sub>H</sub> × p)

RSCAN0TMDF0pL: <RSCAN0\_base> + 1008<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF0pH: <RSCAN0\_base> + 100A<sub>H</sub> + (10<sub>H</sub> × p)

RSCAN0TMDF0pLL: <RSCAN0\_base> + 1008<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF0pLH: <RSCAN0\_base> + 1009<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF0pHL: <RSCAN0\_base> + 100A<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF0pHH: <RSCAN0\_base> + 100B<sub>H</sub> + (10<sub>H</sub> × p)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB3[7:0]										TMDB2[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB1[7:0]										TMDB0[7:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.75 RSCAN0TMDF0p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB3[7:0]	Transmit Buffer Data Byte 3
23 to 16	TMDB2[7:0]	Transmit Buffer Data Byte 2
15 to 8	TMDB1[7:0]	Transmit Buffer Data Byte 1
7 to 0	TMDB0[7:0]	Transmit Buffer Data Byte 0
		Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

### 19.3.56 RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register (p = 0 to 95)

**Access:** RSCAN0TMDF1p register can be read or written in 32-bit units.  
 RSCAN0TMDF1pL, RSCAN0TMDF1pH registers can be read or written in 16-bit units.  
 RSCAN0TMDF1pLL, RSCAN0TMDF1pLH, RSCAN0TMDF1pHL, RSCAN0TMDF1pHH registers can be read or written in 8-bit units.

**Address:** RSCAN0TMDF1p: <RSCAN0\_base> + 100C<sub>H</sub> + (10<sub>H</sub> × p)  
 RSCAN0TMDF1pL: <RSCAN0\_base> + 100C<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF1pH: <RSCAN0\_base> + 100E<sub>H</sub> + (10<sub>H</sub> × p)  
 RSCAN0TMDF1pLL: <RSCAN0\_base> + 100C<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF1pLH: <RSCAN0\_base> + 100D<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF1pHL: <RSCAN0\_base> + 100E<sub>H</sub> + (10<sub>H</sub> × p),  
 RSCAN0TMDF1pHH: <RSCAN0\_base> + 100F<sub>H</sub> + (10<sub>H</sub> × p)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB7[7:0]														TMDB6[7:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB5[7:0]														TMDB4[7:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.76 RSCAN0TMDF1p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB7[7:0]	Transmit Buffer Data Byte 7
23 to 16	TMDB6[7:0]	Transmit Buffer Data Byte 6
15 to 8	TMDB5[7:0]	Transmit Buffer Data Byte 5
7 to 0	TMDB4[7:0]	Transmit Buffer Data Byte 4
		Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

### 19.3.57 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 5)

**Access:** RSCAN0TXQCCm register can be read or written in 32-bit units.  
 RSCAN0TXQCCmL register can be read or written in 16-bit units.  
 RSCAN0TXQCCmLL, RSCAN0TXQCCmLH registers can be read or written in 8-bit units.

**Address:** RSCAN0TXQCCm: <RSCAN0\_base> + 03A0<sub>H</sub> + (04<sub>H</sub> × m)  
 RSCAN0TXQCCmL: <RSCAN0\_base> + 03A0<sub>H</sub> + (04<sub>H</sub> × m)  
 RSCAN0TXQCCmLL: <RSCAN0\_base> + 03A0<sub>H</sub> + (04<sub>H</sub> × m),  
 RSCAN0TXQCCmLH: <RSCAN0\_base> + 03A1<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	TXQIM	TXQIE	TXQDC[3:0]				—	—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 19.77 RSCAN0TXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1)-buffer transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

#### TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

#### TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

**TXQDC[3:0] Bits**

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from  $(m \times 16 + 15)$  to  $(m \times 16 + 0)$ . For examples of how buffer allocation is done, see **Figure 19.9**. Modify these bits only in channel reset mode.

**TXQE Bit**

Setting this bit to 1 makes the transmit queue available. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010<sub>B</sub> or more.

### 19.3.58 RSCAN0TXQSTS<sub>m</sub> — Transmit Queue Status Register ( $m = 0$ to 5)

**Access:** RSCAN0TXQSTS<sub>m</sub> register can be read or written in 32-bit units.  
 RSCAN0TXQSTS<sub>mL</sub> register can be read or written in 16-bit units.  
 RSCAN0TXQSTS<sub>mLL</sub> register can be read or written in 8-bit units.

**Address:** RSCAN0TXQSTS<sub>m</sub>: <RSCAN0\_base> + 03C0<sub>H</sub> + (04<sub>H</sub> × m)  
 RSCAN0TXQSTS<sub>mL</sub>: <RSCAN0\_base> + 03C0<sub>H</sub> + (04<sub>H</sub> × m)  
 RSCAN0TXQSTS<sub>mLL</sub>: <RSCAN0\_base> + 03C0<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFLL	TXQEMP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W* <sup>1</sup>	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 19.78 RSCAN0TXQSTS<sub>m</sub> Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	Reserved	When read, the undefined value is returned. When writing, write “0”.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

#### TXQIF Flag

The TXQIF flag is set to 1 when the event specified by the TXQIM bit in the RSCAN0TXQC<sub>Cm</sub> register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCAN0TXQC<sub>Cm</sub> register to 0 (the transmit queue is not used).

#### TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages set for the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCAN0TXQC<sub>Cm</sub> register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

### **TXQEMP Flag**

The TXQEMP flag is cleared to 0 when even a single message is set for the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

### 19.3.59 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 5)

**Access:** RSCAN0TXQPCTRm register is a write-only register that can be written in 32-bit units.  
 RSCAN0TXQPCTRmL register is a write-only register that can be written in 16-bit units.  
 RSCAN0TXQPCTRmLL register is a write-only register that can be written in 8-bit units.

**Address:** RSCAN0TXQPCTRm: <RSCAN0\_base> + 03E0<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0TXQPCTRmL: <RSCAN0\_base> + 03E0<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0TXQPCTRmLL: <RSCAN0\_base> + 03E0<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 19.79 RSCAN0TXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF <sub>H</sub> to these bits moves the write pointer of the transmit queue to the next queue buffer.

#### TXQPC[7:0] Bits

Writing FF<sub>H</sub> to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request of the message. Write transmit messages to the RSCAN0TMIDp, RSCAN0TMRP, RSCAN0TMDF0p, and RSCAN0TMDF1p registers (p = 15, 31, 47, 63, 79, and 95) before writing FF<sub>H</sub> to the TXQPC[7:0] bits.

When writing FF<sub>H</sub> to these bits, make sure that the TXQE bit in the RSCAN0TXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCAN0TXQSTS<sub>m</sub> register is 0 (the transmit queue is not full).

### 19.3.60 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 to 5)

**Access:** RSCAN0THLCCm register can be read or written in 32-bit units.  
 RSCAN0THLCCmL register can be read or written in 16-bit units.  
 RSCAN0THLCCmLL, RSCAN0THLCCmLH registers can be read or written in 8-bit units.

**Address:** RSCAN0THLCCm: <RSCAN0\_base> + 0400<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLCCmL: <RSCAN0\_base> + 0400<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLCCmLL: <RSCAN0\_base> + 0400<sub>H</sub> + (04<sub>H</sub> × m),  
 RSCAN0THLCCmLH: <RSCAN0\_base> + 0401<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 19.80 RSCAN0THLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entry from transmit/receive FIFO buffers and transmit queue 1: Entry from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

#### THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

**THLIM Bit**

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

**THLIE Bit**

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit set to 0.

**THLE Bit**

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

### 19.3.61 RSCAN0THLSTS<sub>m</sub> — Transmit History Status Register ( $m = 0$ to 5)

**Access:** RSCAN0THLSTS<sub>m</sub> register can be read or written in 32-bit units.  
 RSCAN0THLSTS<sub>mL</sub> register can be read or written in 16-bit units.  
 RSCAN0THLSTS<sub>mLL</sub> register can be read or written in 8-bit units.  
 RSCAN0THLSTS<sub>mLH</sub> register is a read-only register that can be read in 8-bit units.

**Address:** RSCAN0THLSTS<sub>m</sub>: <RSCAN0\_base> + 0420<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLSTS<sub>mL</sub>: <RSCAN0\_base> + 0420<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLSTS<sub>mLL</sub>: <RSCAN0\_base> + 0420<sub>H</sub> + (04<sub>H</sub> × m),  
 RSCAN0THLSTS<sub>mLH</sub>: <RSCAN0\_base> + 0421<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0001<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	THLMC[4:0]					—	—	—	—	THLIF	THLELT	THLFLL	THLEM P	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 19.81 RSCAN0THLSTS<sub>m</sub> Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data sets stored in the transmit history buffer.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEM P	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

#### THLMC[4:0] Bits

These bits indicate the number of unread data sets stored in the transmit history buffer.

### **THLIF Flag**

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCAN0THLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **THLELT Flag**

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

### **THLFLL Flag**

The THLFLL flag is set to 1 when 16 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

### **THLEMP Flag**

The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

#### **NOTE**

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

### 19.3.62 RSCAN0THLACCm — Transmit History Access Register (m = 0 to 5)

**Access:** RSCAN0THLACCm register is a read-only register that can be read in 32-bit units.  
 RSCAN0THLACCmL register is a read-only register that can be read in 16-bit units.  
 RSCAN0THLACCmLL, RSCAN0THLACCmLH registers are the read-only registers that can be read in 8-bit units.

**Address:** RSCAN0THLACCm: <RSCAN0\_base> + 1800<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLACCmL: <RSCAN0\_base> + 1800<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLACCmLL: <RSCAN0\_base> + 1800<sub>H</sub> + (04<sub>H</sub> × m),  
 RSCAN0THLACCmLH: <RSCAN0\_base> + 1801<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]								—	BN[3:0]				BT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.82 RSCAN0THLACCm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	Reserved	When read, the value after reset is returned.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

#### TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

#### BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

#### BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

### 19.3.63 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 to 5)

**Access:** RSCAN0THLPCTRm register is a write-only register that can be written in 32-bit units.  
 RSCAN0THLPCTRmL register is a write-only register that can be written in 16-bit units.  
 RSCAN0THLPCTRmLL register is a write-only register that can be written in 8-bit units.

**Address:** RSCAN0THLPCTRm: <RSCAN0\_base> + 0440<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLPCTRmL: <RSCAN0\_base> + 0440<sub>H</sub> + (04<sub>H</sub> × m)

RSCAN0THLPCTRmLL: <RSCAN0\_base> + 0440<sub>H</sub> + (04<sub>H</sub> × m)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 19.83 RSCAN0THLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF <sub>H</sub> to these bits moves the read pointer to the next unread data in the transmit history buffer.

#### THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF<sub>H</sub>, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCAN0THLSTS<sub>m</sub> register is decremented. Write FF<sub>H</sub> to the THLPC[7:0] bits after reading from the RSCAN0THLACC<sub>m</sub> register.

When writing FF<sub>H</sub> to these bits, make sure that the THLE bit in the RSCAN0THLCC<sub>m</sub> register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCAN0THLSTS<sub>m</sub> register is 0.

### 19.3.64 RSCAN0GTSTCFG — Global Test Configuration Register

**Access:** RSCAN0GTSTCFG register can be read or written in 32-bit units.  
 RSCAN0GTSTCFG<sub>L</sub>, RSCAN0GTSTCFG<sub>H</sub> registers can be read or written in 16-bit units.  
 RSCAN0GTSTCFG<sub>LL</sub>, RSCAN0GTSTCFG<sub>HL</sub> registers can be read or written in 8-bit units.

**Address:** RSCAN0GTSTCFG: <RSCAN0\_base> + 0468<sub>H</sub>  
 RSCAN0GTSTCFG<sub>L</sub>: <RSCAN0\_base> + 0468<sub>H</sub>, RSCAN0GTSTCFG<sub>H</sub>: <RSCAN0\_base> + 046A<sub>H</sub>  
 RSCAN0GTSTCFG<sub>LL</sub>: <RSCAN0\_base> + 0468<sub>H</sub>  
 RSCAN0GTSTCFG<sub>HL</sub>: <RSCAN0\_base> + 046A<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTMPS[6:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						
C5ICBCE C4ICBCE C3ICBCE C2ICBCE C1ICBCE C0ICBCE																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						

Table 19.84 RSCAN0GTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	RTMPS[6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00 <sub>H</sub> ) to page 56 (38 <sub>H</sub> ).
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	C5ICBCE	CAN5 Inter-channel Communication Test Enable 0: CAN5 inter-channel communication test is disabled. 1: CAN5 inter-channel communication test is enabled.
4	C4ICBCE	CAN4 Inter-channel Communication Test Enable 0: CAN4 inter-channel communication test is disabled. 1: CAN4 inter-channel communication test is enabled.
3	C3ICBCE	CAN3 Inter-channel Communication Test Enable 0: CAN3 inter-channel communication test is disabled. 1: CAN3 inter-channel communication test is enabled.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled. 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCAN0GTSTCFG register only in global test mode.

#### RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00<sub>H</sub> to 38<sub>H</sub>, inclusive.

**C5ICBCE Bit**

Setting this bit to 1 enables the channel 5 inter-channel communication test.

**C4ICBCE Bit**

Setting this bit to 1 enables the channel 4 inter-channel communication test.

**C3ICBCE Bit**

Setting this bit to 1 enables the channel 3 inter-channel communication test.

**C2ICBCE Bit**

Setting this bit to 1 enables the channel 2 inter-channel communication test.

**C1ICBCE Bit**

Setting this bit to 1 enables the channel 1 inter-channel communication test.

**C0ICBCE Bit**

Setting this bit to 1 enables the channel 0 inter-channel communication test.

### 19.3.65 RSCAN0GTSTCTR — Global Test Control Register

**Access:** RSCAN0GTSTCTR register can be read or written in 32-bit units.  
 RSCAN0GTSTCTRL register can be read or written in 16-bit units.  
 RSCAN0GTSTCTRLL register can be read or written in 8-bit units.

**Address:** RSCAN0GTSTCTR: <RSCAN0\_base> + 046C<sub>H</sub>

RSCAN0GTSTCTRL: <RSCAN0\_base> + 046C<sub>H</sub>

RSCAN0GTSTCTRLL: <RSCAN0\_base> + 046C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCT ME	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 19.85 RSCAN0GTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICBCTME	Communication Test between Channels Enable 0: Communication test between channels disabled 1: Communication test between channels enabled

#### RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

1. Set the GMDC[1:0] bits in the RSCAN0GCTR register to 10<sub>B</sub> (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

#### ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 5) in the RSCAN0GTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

### 19.3.66 RSCAN0GLOCKK — Global Lock Key Register

**Access:** RSCAN0GLOCKK register is a write-only register that can be written in 32-bit units.  
RSCAN0GLOCKKL register is a write-only register that can be written in 16-bit units.

**Address:** RSCAN0GLOCKK: <RSCAN0\_base> + 047C<sub>H</sub>

RSCAN0GLOCKKL: <RSCAN0\_base> + 047C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W <sup>*1</sup>															

Note 1. Writing to these bits is effective only when the RS-CAN module is in global test mode.

Table 19.86 RSCAN0GLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCAN0GLOCKK register releases protection of special test bits and is write only.

For the protection release data, see **Section 19.10.4.2, Procedure for Releasing the Protection**.

#### LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCAN0GTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCAN0\_base> + 0000<sub>H</sub> to <RSCAN0\_base> + 04FF<sub>H</sub>) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

### 19.3.67 RSCAN0RPGACCr — RAM Test Page Access Register (r = 0 to 63)

**Access:** RSCAN0RPGACCr register can be read or written in 32-bit units.  
 RSCAN0RPGACCrL, RSCAN0RPGACCrH registers can be read or written in 16-bit units.  
 RSCAN0RPGACCrLL, RSCAN0RPGACCrLH, RSCAN0RPGACCrHL, RSCAN0RPGACCrHH registers can be read or written in 8-bit units.

**Address:** RSCAN0RPGACCr: <RSCAN0\_base> + 1900<sub>H</sub> + (04<sub>H</sub> × r)  
 RSCAN0RPGACCrL: <RSCAN0\_base> + 1900<sub>H</sub> + (04<sub>H</sub> × r),  
 RSCAN0RPGACCrH: <RSCAN0\_base> + 1902<sub>H</sub> + (04<sub>H</sub> × r)  
 RSCAN0RPGACCrLL: <RSCAN0\_base> + 1900<sub>H</sub> + (04<sub>H</sub> × r),  
 RSCAN0RPGACCrLH: <RSCAN0\_base> + 1901<sub>H</sub> + (04<sub>H</sub> × r),  
 RSCAN0RPGACCrHL: <RSCAN0\_base> + 1902<sub>H</sub> + (04<sub>H</sub> × r),  
 RSCAN0RPGACCrHH: <RSCAN0\_base> + 1903<sub>H</sub> + (04<sub>H</sub> × r)

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDTA[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDTA[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.87 RSCAN0RPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA [31:0]	RAM Data Test Access Data can be read and written in RSCAN RAM.

Modify the RSCAN0RPGACCr register in global test mode with the RTME bit in the RSCAN0GTSTCTR register set to 1 (RAM test is enabled).

The RSCAN0RPGACCr register is readable and writable when the RTME bit is set to 1.

## 19.4 Interrupt Sources

The RS-CAN module has 20 interrupts that are grouped into global interrupts and channel interrupts.

- Global interrupts (2 sources):  
Receive FIFO interrupt  
Global error interrupt
- Channel interrupts (3 sources/channel):  
CANm transmit interrupt ( $m = 0$  to 5)
  - CANm transmit complete interrupt
  - CANm transmit abort interrupt
  - CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
  - CANm transmit history interrupt
  - CANm transmit queue InterruptCANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode)  
CANm error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RS-CAN module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

**Table 19.88** lists the CAN interrupt sources. **Figure 19.2** shows the CAN global interrupt block diagram. **Figure 19.3** shows the CAN channel interrupt block diagram.

**Table 19.88 List of CAN Interrupt Sources**

Interrupt Source		Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit	
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF in the RSCAN0RFSTS0 register	
		Receive FIFO 1	RFIF in the RSCAN0RFSTS1 register	
		Receive FIFO 2	RFIF in the RSCAN0RFSTS2 register	
		Receive FIFO 3	RFIF in the RSCAN0RFSTS3 register	
		Receive FIFO 4	RFIF in the RSCAN0RFSTS4 register	
		Receive FIFO 5	RFIF in the RSCAN0RFSTS5 register	
		Receive FIFO 6	RFIF in the RSCAN0RFSTS6 register	
		Receive FIFO 7	RFIF in the RSCAN0RFSTS7 register	
Global error			<ul style="list-style-type: none"> <li>• DEF in the RSCAN0GERFL register</li> <li>• MES in the RSCAN0GERFL register</li> <li>• THLES in the RSCAN0GERFL register</li> </ul>	
			<ul style="list-style-type: none"> <li>• DEIE in the RSCAN0GCTR register</li> <li>• MEIE in the RSCAN0GCTR register</li> <li>• THLEIE in the RSCAN0GCTR register</li> </ul>	
Channel interrupts (m = 0 to 5)	CANm transmit	CANm transmit complete	TMTRF[1:0] in the RSCAN0TMSTSp register	
		CANm transmit abort	TMTRF[1:0] in the RSCAN0TMSTSp register	
		CANm transmit/receive FIFO transmit complete	CFTXIF in the RSCAN0CFSTS <sub>k</sub> register	
		CANm transmit queue	TXQIF in the RSCAN0TXQSTS <sub>m</sub> register	
		CANm transmit history	THLIF in the RSCAN0THLSTS <sub>m</sub> register	
		CANm transmit/receive FIFO receive complete	CFRXIF in the RSCAN0CFSTS <sub>k</sub> register	
		CANm error	<ul style="list-style-type: none"> <li>• BEF in the RSCAN0CmERFL register</li> <li>• ALF in the RSCAN0CmERFL register</li> <li>• BLF in the RSCAN0CmERFL register</li> <li>• OVLF in the RSCAN0CmERFL register</li> <li>• BORF in the RSCAN0CmERFL register</li> <li>• BOEF in the RSCAN0CmERFL register</li> <li>• EPF in the RSCAN0CmERFL register</li> <li>• EWF in the RSCAN0CmERFL register</li> <li>• BEIE in the RSCAN0CmCTR register</li> <li>• ALIE in the RSCAN0CmCTR register</li> <li>• BLIE in the RSCAN0CmCTR register</li> <li>• OLIE in the RSCAN0CmCTR register</li> <li>• BORIE in the RSCAN0CmCTR register</li> <li>• BOEIE in the RSCAN0CmCTR register</li> <li>• EPIE in the RSCAN0CmCTR register</li> <li>• EWIE in the RSCAN0CmCTR register</li> </ul>	

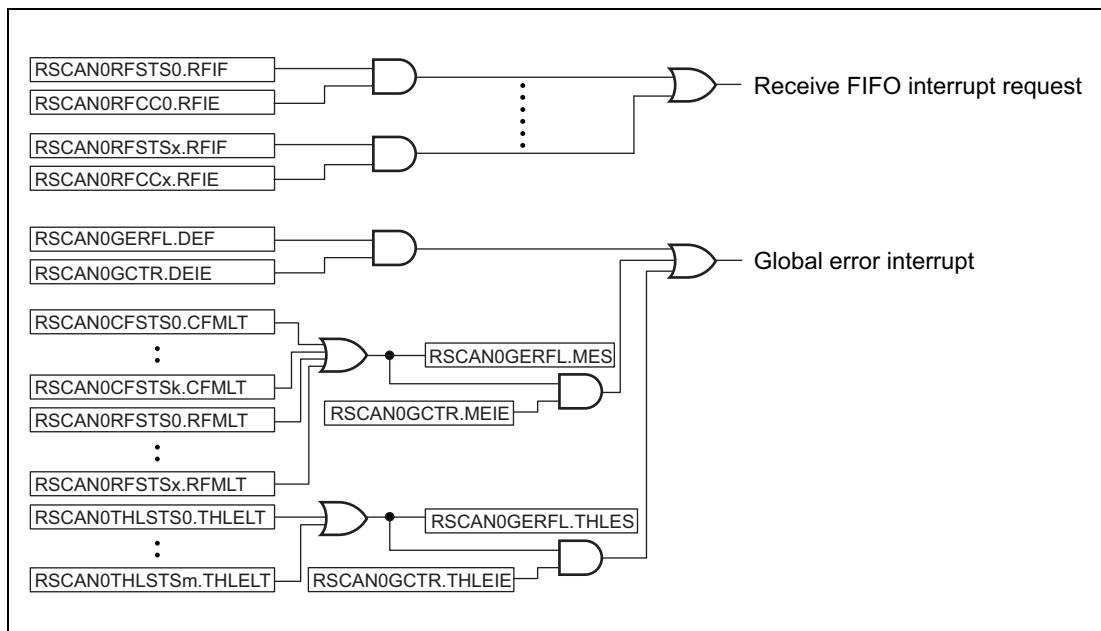


Figure 19.2 CAN Global Interrupt Block Diagram

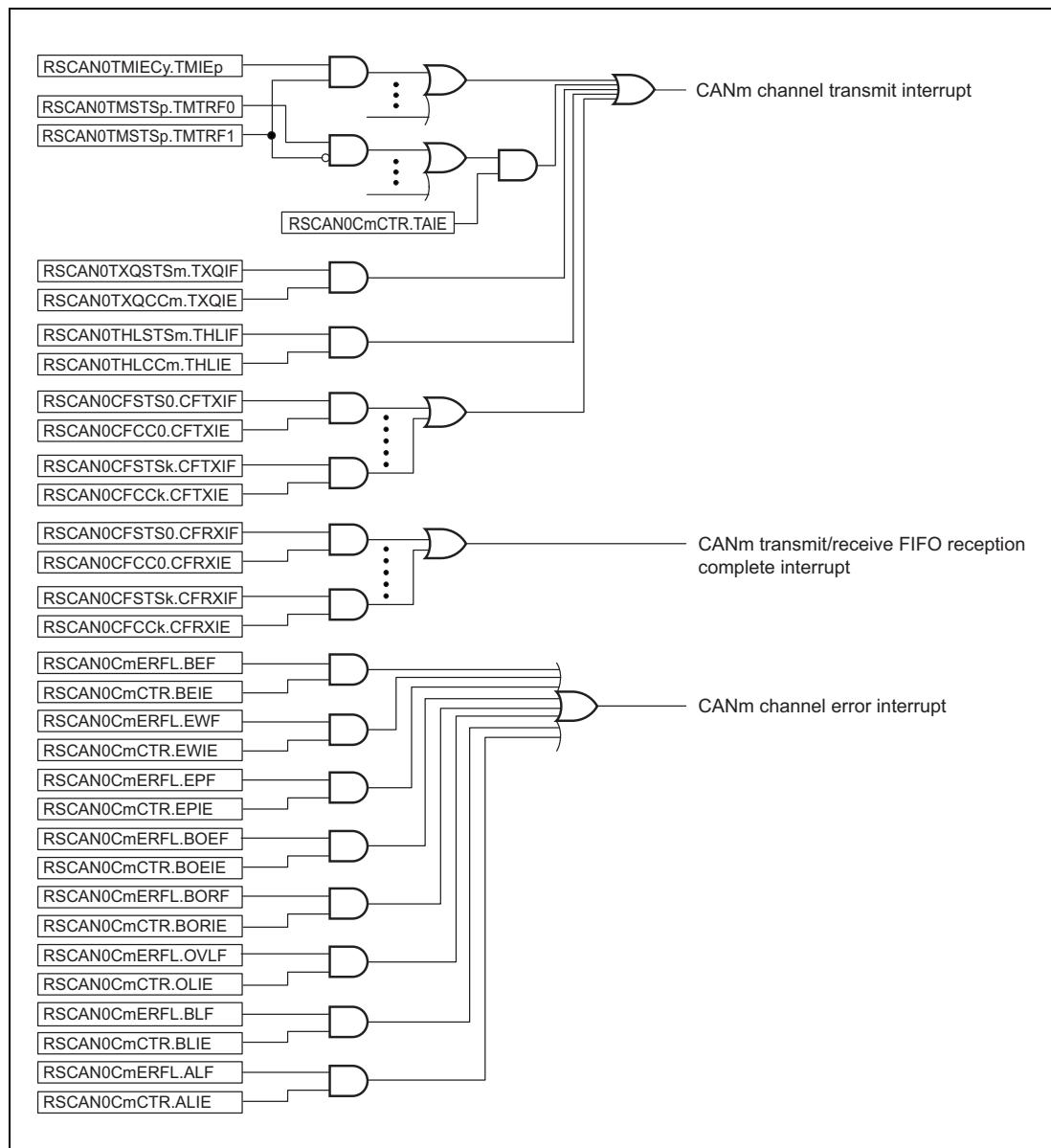


Figure 19.3 CAN Channel Interrupt Block Diagram

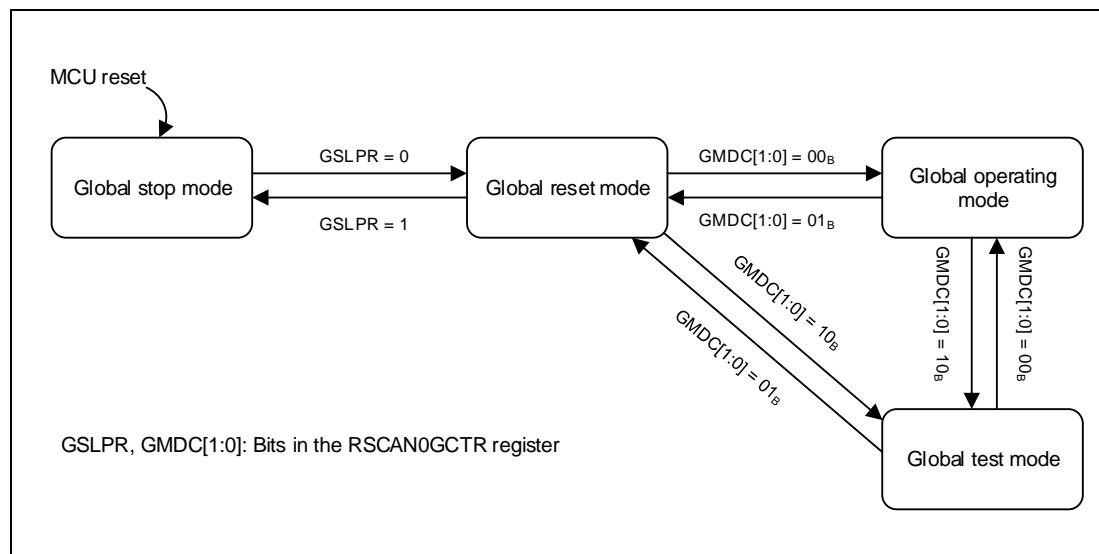
## 19.5 CAN Modes

The RS-CAN module has four global modes to control the entire RS-CAN module status and four channel modes to control individual channel status. Details of global modes are described in **Section 19.5.1, Global Modes**, and details of channel modes are described in **Section 19.5.2, Channel Modes**.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs the RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

### 19.5.1 Global Modes

**Figure 19.4** shows the transitions of global modes.



**Figure 19.4** Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 19.89** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

**Table 19.89 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)**

Channel Mode before Setting	Channel Mode after Setting			
	GMDC[1:0] = 00 <sub>B</sub> GSLPR = 0 (Global Operation)	GMDC[1:0] = 10 <sub>B</sub> GSLPR = 0 (Global Test)	GMDC[1:0] = 01 <sub>B</sub> GSLPR = 0 (Global Reset)	GMDC[1:0] = 01 <sub>B</sub> GSLPR = 1 (Global Stop)
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

**Note:** GMDC[1:0], GSLPR: Bits in the RSCAN0GCTR register

**Table 19.90** shows the global mode transition time.

**Table 19.90 Global Mode Transition Time**

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	Three pclk cycles
Global reset	Global stop	Three pclk cycles
Global reset	Global test	Ten pclk cycles
Global reset	Global operating	Ten pclk cycles
Global test	Global reset	Three pclk cycles
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Three pclk cycles
Global operating	Global test	Two CAN frames <sup>*1</sup>

Note 1. CAN frame time of the lowest communication speed of the channels in use

#### 19.5.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After the MCU is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RSCAN0GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each of the RSCAN0CmCTR register to 1 (channel stop mode). Afterwards, if all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

#### 19.5.1.2 Global Reset Mode

In global reset mode, RS-CAN module settings are performed. When the RS-CAN module transitions to global reset mode, some registers are initialized. **Table 19.93** and **Table 19.94** list the registers to be initialized.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to  $01_B$  sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR registers ( $m = 0$  to  $5$ ) to  $01_B$  (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to  $01_B$ ).

#### 19.5.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to  $10_B$  sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR register to  $10_B$  (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

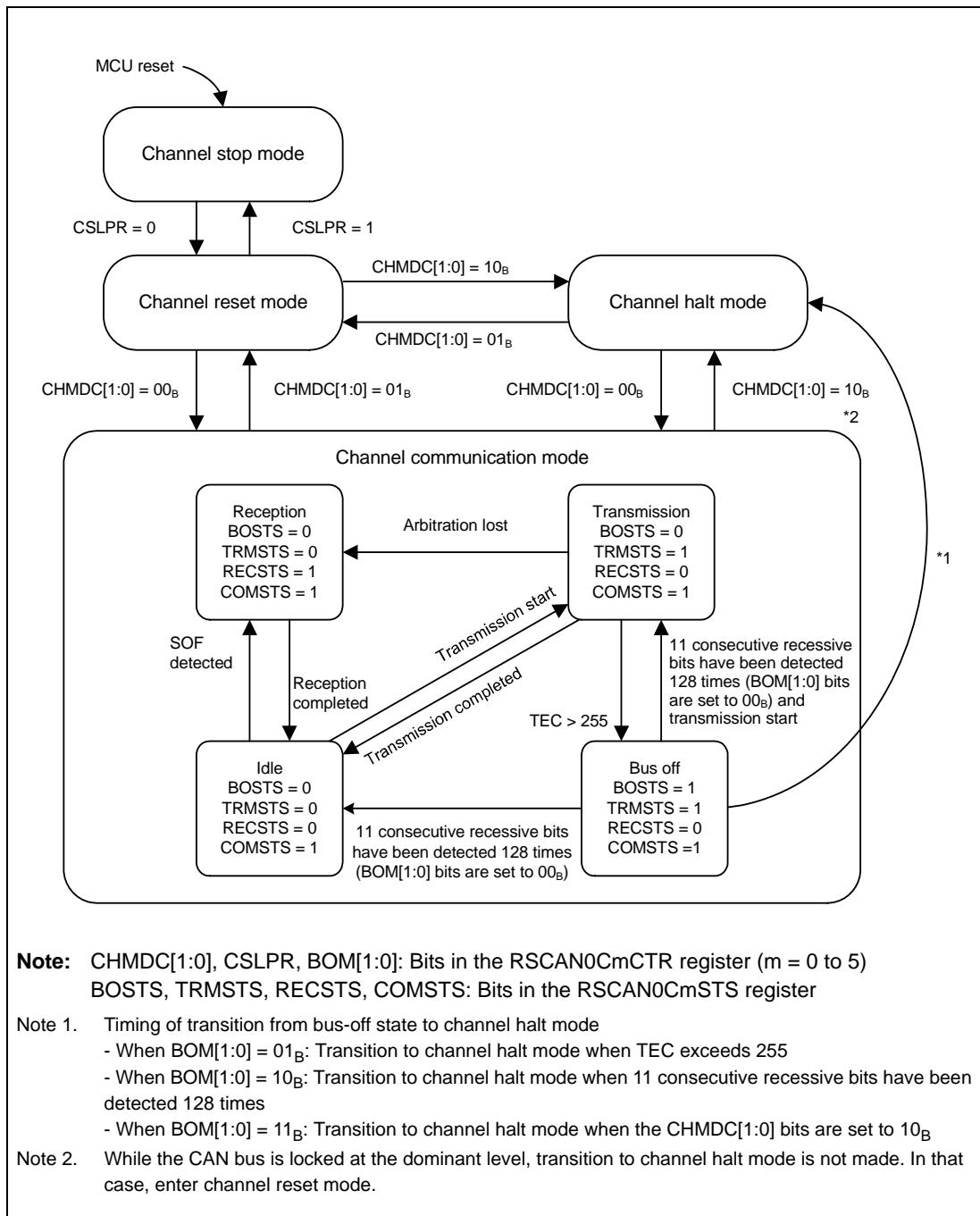
#### 19.5.1.4 Global Operating Mode

The RS-CAN module operates in global operating mode.

When the GMDC[1:0] bits in the RSCAN0GCTR register are set to  $00_B$ , the RS-CAN module transitions to global operating mode.

### 19.5.2 Channel Modes

**Figure 19.5** shows a channel mode state transition chart. **Table 19.91** shows the channel mode transition time.



**Figure 19.5** Channel Mode State Transition Chart

**Table 19.91** Channel Mode Transition Time (1/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	Three pclk cycles
Channel reset	Channel stop	Three pclk cycles
Channel reset	Channel halt	Three CANm bit times
Channel reset	Channel communication	Two CANm bit times

**Table 19.91 Channel Mode Transition Time (2/2)**

<b>Mode before Transition</b>	<b>Mode after Transition</b>	<b>Maximum Transition Time</b>
Channel halt	Channel reset	Three pclk cycles
Channel halt	Channel communication	Three CANm bit times
Channel communication	Channel reset	Three pclk cycles
Channel communication	Channel halt	Two CANm frames

### 19.5.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

Each channel enters channel stop mode after the MCU is reset. Channels also transition to channel stop mode when the CSLPR bit in the RSCAN0CmCTR register ( $m = 0$  to  $5$ ) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

### 19.5.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. **Table 19.93** lists the registers to be initialized.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to  $01_B$  (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. **Table 19.92** shows the operation when the CHMDC[1:0] bits are set to  $01_B$  (channel reset mode) during CAN communication.

### 19.5.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

**Table 19.92** shows operation when the CHMDC[1:0] bits are set to  $10_B$  (channel halt mode) during CAN communication.

**Table 19.92 Operation a Channel Transitions to Channel Reset Mode/Channel Halt Mode**

<b>Mode</b>	<b>During Reception</b>	<b>During Transmission</b>	<b>Bus-off State</b>
Channel reset (CHMDC[1:0] = $01_B$ )	Transitions to channel reset mode before reception is completed.* <sup>1</sup>	Transitions to channel reset mode before transmission is completed.* <sup>1</sup>	Transitions to channel reset mode before bus-off recovery.
Channel halt* <sup>3</sup> (CHMDC[1:0] = $10_B$ )	Transitions to channel halt mode after reception is completed.* <sup>2</sup>	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] = $00_B$ ] Transitions to channel halt mode (CHMDC[1:0] = $10_B$ ) only after bus-off recovery. [When BOM[1:0] = $01_B$ ] Transitions to channel halt mode automatically when the condition for transition to bus-off state is met. [When BOM[1:0] = $10_B$ ] Transitions to channel halt mode automatically after bus-off recovery. [When BOM[1:0] = $11_B$ ] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to $10_B$ before bus-off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to

$10_B$  and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to  $01_B$ .

- Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCAN0CmERFL register that becomes 1 when dominant lock is detected.
- Note 3. When the transition from channel reset mode to channel halt mode is to be made, set the RSCAN0CmCFG register in channel reset mode and then shift to channel halt mode.

#### 19.5.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception: Receiving a message sent from another node.
- Transmission: Transmitting a message.
- Bus-off: Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to  $00_B$ , the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCAN0CmSTS register ( $m = 0$  to 5) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

#### 19.5.2.5 Bus-off State

A channel transitions to the bus-off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

The conditions for returning from the bus-off state are determined by the BOM[1:0] bits in the RSCAN0CmCTR register.

- When  $BOM[1:0] = 00_B$ :  
Bus-off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus-off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are initialized to  $00_H$ , the BORF flag in the RSCAN0CmERFL register is set to 1 (bus-off recovery is detected), and a bus-off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to  $10_B$  (channel halt mode) in the bus-off state, the channel transitions to channel halt mode after bus-off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When  $BOM[1:0] = 01_B$ :  
When a channel transitions to the bus-off state, the CHMDC[1:0] bits are set to  $10_B$  and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to  $00_H$ . The BORF flag is not set to 1, and bus-off recovery interrupt request is not generated.
- When  $BOM[1:0] = 10_B$ :  
When a channel has transitioned to the bus-off state, the CHMDC[1:0] bits are set to  $10_B$ . After bus-off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to  $00_H$ , the BORF flag is set to 1, and a bus-off recovery interrupt request is generated.

- When  $BOM[1:0] = 11_B$ :

When the CHMDC[1:0] bits are set to  $10_B$  in the bus-off state, the channel transitions to channel halt mode before bus-off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to  $00_H$ , but the BORF flag is not set to 1. Also, a bus-off recovery interrupt is not generated.

However, the BORF flag becomes 1 and a bus-off recovery interrupt request is generated if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before CHMDC[1:0] bits are set to  $10_B$ .

If the RS-CAN module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to  $01_B$  or  $10_B$  is made only when the CHMDC[1:0] bits are  $00_B$  (channel communication mode).

Furthermore, setting the RTBO bit in the RSCAN0CmCTR register to 1 allows a forced recovery from the bus-off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to  $00_H$ . Write 1 to the RTBO bit only when the BOM[1:0] value is  $00_B$ . Writing the RTBO bit to 1 in a state other than the bus-off state is ignored, and the RTBO bit is immediately set to 0.

**Table 19.93 Registers Initialized in Global Reset Mode or Channel Reset Mode**

Register	Bit / Flag
RSCAN0CmCTR register	CTMS[1:0], CTME, CHMDC[1:0]
RSCAN0CmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, REC[7:0], TEC[7:0]
RSCAN0CmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RSCAN0CFSTSTS register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RSCAN0CFTISTS register	CFkTXIF
RSCAN0TMCP register	TMOM, TMTAR, TMTR
RSCAN0TMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RSCAN0TMTRSTS register	TMTRSTS (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTARSTS register	TMTARSTS (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMCSTS register	TMTCSTS (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTASTS register	TMTASTS (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TXQCCM register	TXQE
RSCAN0TXQSTS register	TXQIF, TXQFLL, TXQEMP
RSCAN0THLCCM register	THLE
RSCAN0THLSTS register	THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP
RSCAN0GTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 to 3)
RSCAN0GTINTSTS1 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 4, 5)

**Table 19.94 Registers Initialized Only in Global Reset Mode**

<b>Register</b>	<b>Bit / Flag</b>
RSCAN0GSTS register	GHLSTS
RSCAN0GERFL register	THLES, MES, DEF
RSCAN0GTSC register	TS[15:0]
RSCAN0RMNDy register	RMNSq
RSCAN0RFCCx register	RFE
RSCAN0RFSTSx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RSCAN0FESTS register	CFkEMP, RFxEMP
RSCAN0FFSTS register	CFkFLL, RFxFLL
RSCAN0FMSTS register	CFkMLT, RFxMLT
RSCAN0RFISTS register	RFxIF
RSCAN0CFRISTS register	CFkRXIF
RSCAN0GTSTCFG register	RTMPS[6:0], C0ICBCE, C1ICBCE, C2ICBCE, C3ICBCE, C4ICBCE, C5ICBCE
RSCAN0GTSTCTR register	RTME, ICBCTME

## 19.6 Reception Function

There are two reception types.

- Reception by receive buffers:  
Zero to 96 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):  
Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

### 19.6.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to ( $64 \times$  number of channels) total receive rules can be registered in the entire module. (Up to 384 receive rules can be registered in this module that has six channels.) Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. **Figure 19.6** illustrates how receive rules are registered.

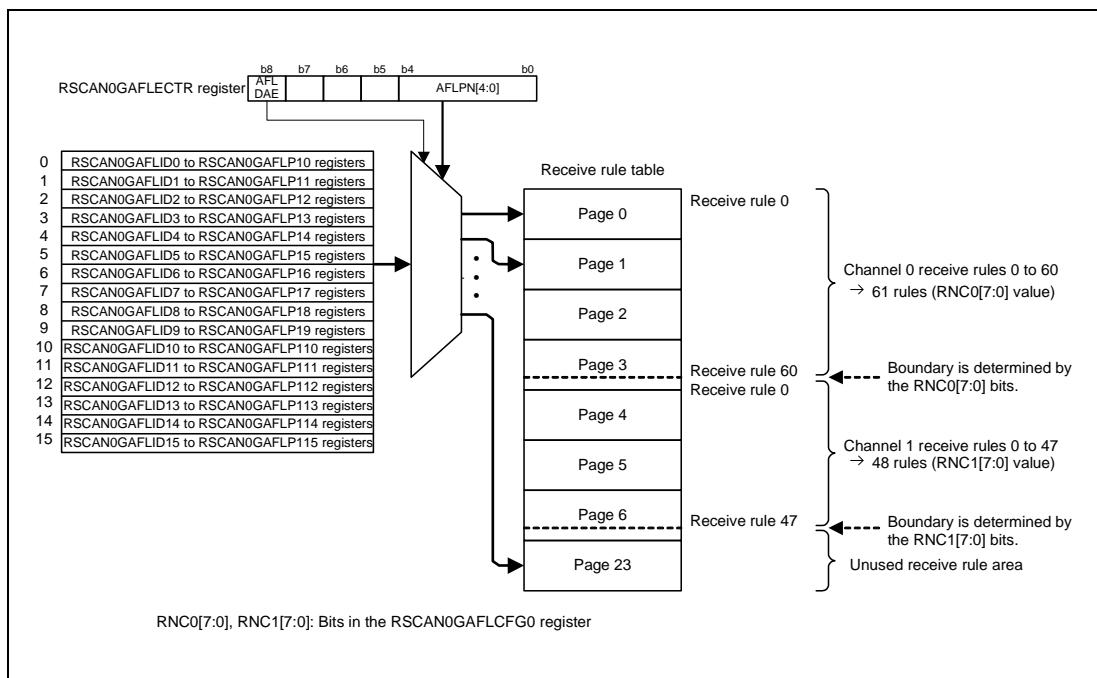


Figure 19.6 Entry of Receive Rules (for Setting Channel 0 and 1)

#### CAUTION

Receive rules for each channel must be set in contiguous blocks.

Channel 1 rules and channel 0 rules must be set separately.

Each receive rule consists of 16 bytes in the RSCAN0GAFLID $j$ , RSCAN0GAFLM $j$ , RSCAN0GAFLP0 $j$ , and RSCAN0GAFLP1 $j$  registers ( $j = 0$  to 15). The RSCAN0GAFLID $j$  register is used to set GAFLID, GAFLIDE bit, GAFLRTR bit, and the mirror function, the RSCAN0GAFLM $j$  register is used to set mask, the RSCAN0GAFLP0 $j$  register is used to set label information to be added, DLC value, and storage receive buffer, and the RSCAN0GAFLP1 $j$  register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

### 19.6.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RSCAN0GAFLM $j$  register are not compared and are regarded as matched.

Check begins with the receive rule of the minimum number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

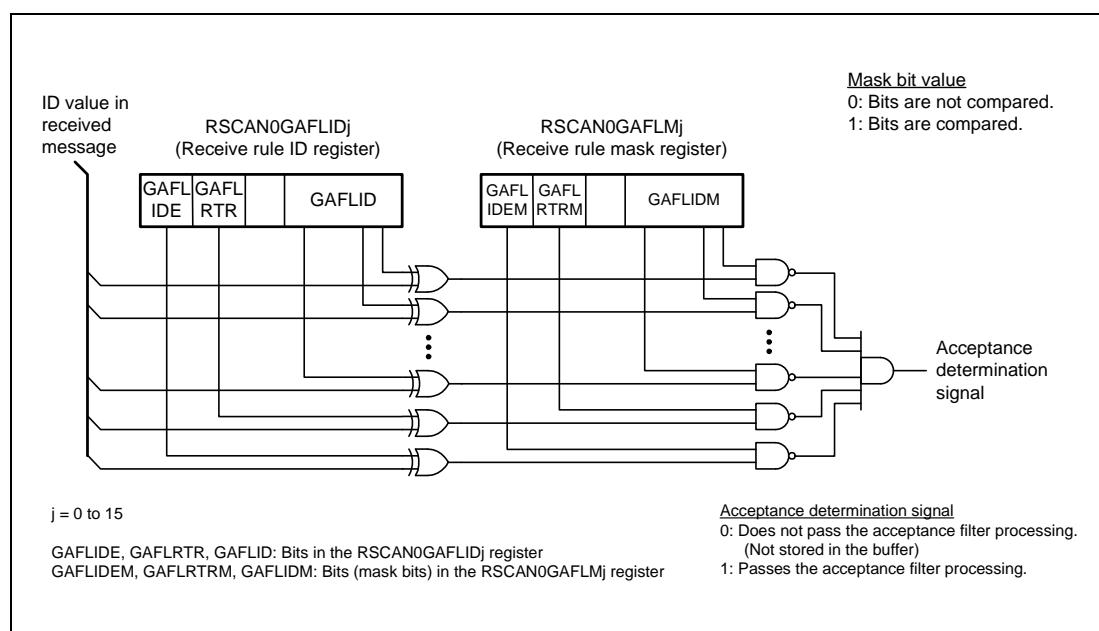


Figure 19.7 Acceptance Filter Function

### 19.6.1.2 DLC Filter Processing

When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of  $00_H$  is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCAN0GERFL register is set to 1 (a DLC error is present).

### 19.6.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCAN0GAFLP0j register ( $j = 0$  to 15) and by the RSCAN0GAFLP1j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

### 19.6.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the RSCAN0GAFLP0j register.

### 19.6.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is made available by setting the MME bit in the RSCAN0GCFG register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCAN0GAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

### 19.6.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. Either  $\text{pclk}/2$  or the CANm bit time clock ( $m = 0$  to 5) may be selected as a timestamp counter clock source using the TSBTCS[2:0] and TSSS bits in the RSCAN0GCFG register. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] value in the RSCAN0GCFG register.

When the CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the  $\text{pclk}/2$  is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to  $0000_H$  by setting the TSRST bit in the RSCAN0GCTR register to 1.

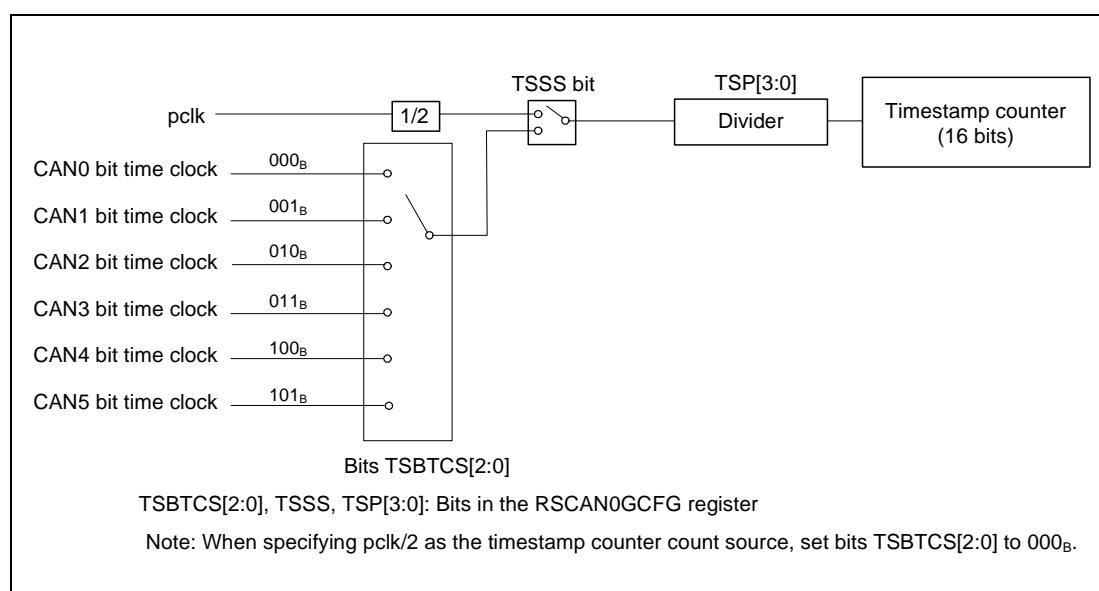


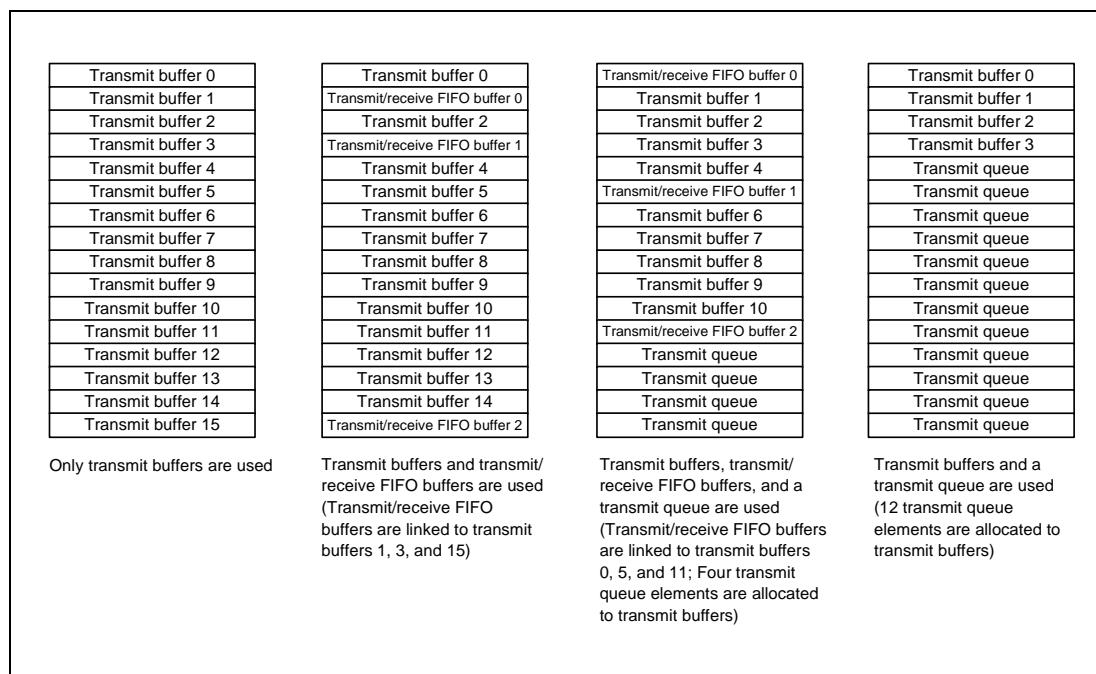
Figure 19.8    Timestamp Function Block Diagram

## 19.7 Transmission Functions

There are three types of transmission.

- Transmission using transmit buffers:  
Each channel has 16 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):  
Each channel has three FIFO buffers. Up to 128 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.
- Transmission using transmit queues:  
Up to 16 transmit buffers per channel can be allocated to the transmit queues. Transmit buffer  $((16 \times m) + 15)$  is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

**Figure 19.9** shows the allocation of transmit queues and transmit/receive FIFO buffer link.



**Figure 19.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links**

### 19.7.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

The priority is determined by using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RSCAN0GCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the minimum buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again according to the TPRI bit.

### 19.7.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCAN0TMCP register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register ( $p = 0$  to 95). When transmit completes successfully, the TMTRF[1:0] flag is set to  $10_B$  (transmission has been completed (without transmit abort request)) or  $11_B$  (transmission has been completed (with transmit abort request)).

#### 19.7.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RSCAN0TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RSCAN0TMCP register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to  $01_B$  (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

### 19.7.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RSCAN0TMCP register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10<sub>B</sub> or 11<sub>B</sub>. When an arbitration-lost or an error occurs, the TMTRF[1:0] flag is set to 01<sub>B</sub> (transmit abort has been completed).

### 19.7.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffers, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RSCAN0CFCCk register (k = 0 to 17). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[3:0] bits in the RSCAN0CFCCk register. When the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

### 19.7.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RSCAN0CFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCAN0CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00<sub>H</sub>.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCAN0CFCCk register. When the CFITR and CFITSS bits are set to 00<sub>B</sub>, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10<sub>B</sub>, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10). When the CFITR and CFITSS bits are set to x1<sub>B</sub>, the CANm bit time clock is used as a count source.

The interval time is calculated by the following equations where M is the value of ITRCP[15:0] and N is the value of CFITT[7:0].

- When CFITR and CFITSS = 00<sub>B</sub> (fPBA is the frequency of pclk):

$$\frac{1}{f_{PBA}} \times 2 \times M \times N$$

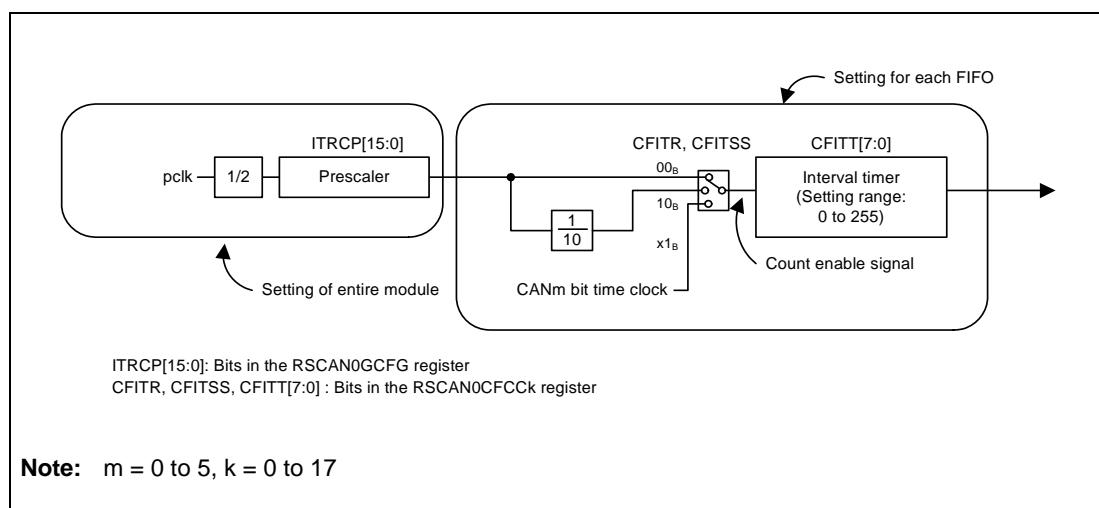
- When CFITR and CFITSS = 10<sub>B</sub>:

$$\frac{1}{f_{PBA}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = x1<sub>B</sub> (fCANBIT is the frequency of CANm bit time clock):

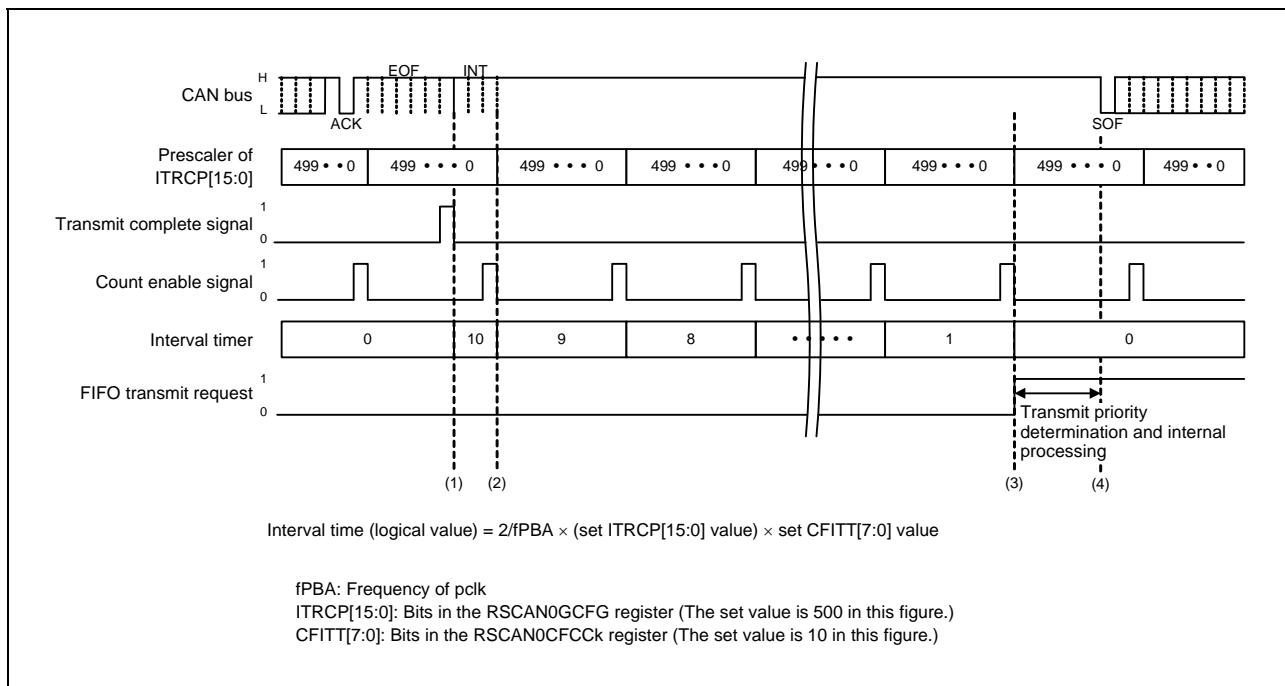
$$\frac{1}{f_{CANBIT}} \times N$$

**Figure 19.10** shows the interval timer block diagram.



**Figure 19.10 Interval Timer Block Diagram**

**Figure 19.11** shows the interval timer timing diagram.



**Figure 19.11 Interval Timer Timing Chart**

- (1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.
- (2) The interval timer is decremented by the next count enable signal.
- (3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.
- (4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 582 cycles of the pclk may be generated.

#### 19.7.4 Transmission Using Transmit Queues

Three to sixteen buffers are allocated to a transmit queue for each channel, and transmit buffer ( $(16 \times m) + 15$ ) is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RSCAN0TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCAN0TXQSTS<sub>m</sub> register is set to 1 (the transmit queue contains no messages (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

#### 19.7.5 Transmit History Function

Information about transmission-completed messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCAN0THLCCm register. The THLEN bit in the RSCAN0CFID<sub>k</sub> register ( $k = 0$  to 17) determines whether transmit history data is stored for each message.

The following information on a transmitted message will be stored in the transmission history buffer after the successful completion of transmission.

Storage of the transmission history data after the successful completion of transmission may take up to 150 cycles of pclk.

- |                 |  |
|-----------------|--|
| • Buffer type   | 001 <sub>B</sub> : Transmit buffer<br>010 <sub>B</sub> : Transmit/receive FIFO buffer<br>100 <sub>B</sub> : Transmit queue                       |
| • Buffer number | Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See <b>Table 19.95</b> . |
| • Label data    | Label information of the transmit message  |

Table 19.95 Transmit History Data Buffer Numbers

Buffer No. Buffer type	<b>001<sub>B</sub></b>	<b>010<sub>B</sub></b>	<b>100<sub>B</sub></b>
0000 <sub>B</sub>	Transmit buffer 16 × m + 0		
0001 <sub>B</sub>	Transmit buffer 16 × m + 1		
0010 <sub>B</sub>	Transmit buffer 16 × m + 2		
0011 <sub>B</sub>	Transmit buffer 16 × m + 3		
0100 <sub>B</sub>	Transmit buffer 16 × m + 4		
0101 <sub>B</sub>	Transmit buffer 16 × m + 5		
0110 <sub>B</sub>	Transmit buffer 16 × m + 6		
0111 <sub>B</sub>	Transmit buffer 16 × m + 7		
1000 <sub>B</sub>	Transmit buffer 16 × m + 8		
1001 <sub>B</sub>	Transmit buffer 16 × m + 9		
1010 <sub>B</sub>	Transmit buffer 16 × m + 10		
1011 <sub>B</sub>	Transmit buffer 16 × m + 11		
1100 <sub>B</sub>	Transmit buffer 16 × m + 12		
1101 <sub>B</sub>	Transmit buffer 16 × m + 13		
1110 <sub>B</sub>	Transmit buffer 16 × m + 14		
1111 <sub>B</sub>	Transmit buffer 16 × m + 15		

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

Transmit history data can be read from the RSCAN0THLACCM register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

## 19.8 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive messages can be transmitted from an arbitrary channel without CPU intervention.

When the CFM[1:0] bits in the RSCAN0CFCCk register are set to 10<sub>B</sub> (gateway mode) and the transmit/receive FIFO buffer of a channel transmitting a message is selected in RSCAN0GAFLP1j register, messages that pass through filter processing according to the reception rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCAN0CFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted and will not be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

## 19.9 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
  - Standard test mode
  - Listen-only mode
  - Self-test mode 0 (external loopback mode)
  - Self-test mode 1 (internal loopback mode)
- Global tests: Performed for the entire module
  - RAM test (read/write test)
  - Inter-channel communication test

### 19.9.1 Standard Test Mode

Standard test mode allows CRC test.

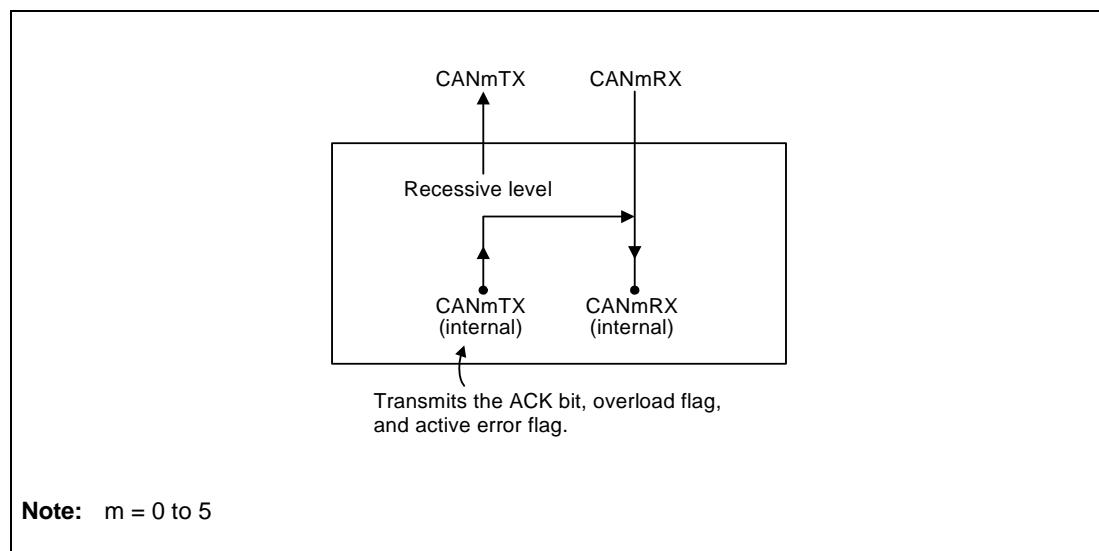
### 19.9.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

**Figure 19.12** shows the connection when listen-only mode is selected.



**Figure 19.12** Connection when Listen-Only Mode is Selected

### 19.9.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCAN0GAFLIDj register ( $j = 0$  to 15) is set to 0 (when a message transmitted from another CAN node is received).

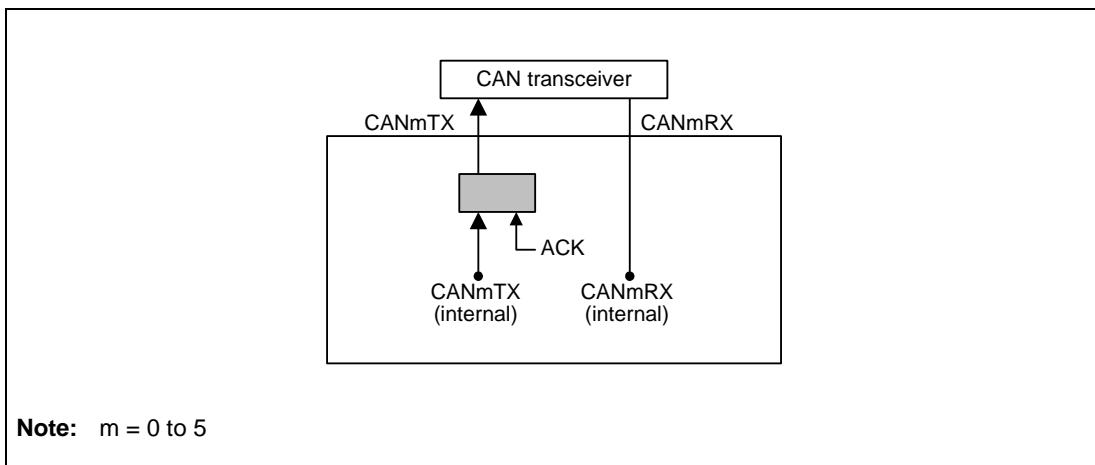
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

#### 19.9.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

**Figure 19.13** shows the connection when self-test mode 0 is selected.



**Figure 19.13** Connection when Self-Test Mode 0 is Selected

### 19.9.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin ( $m = 0$  to 5) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

**Figure 19.14** shows the connection when self-test mode 1 is selected.

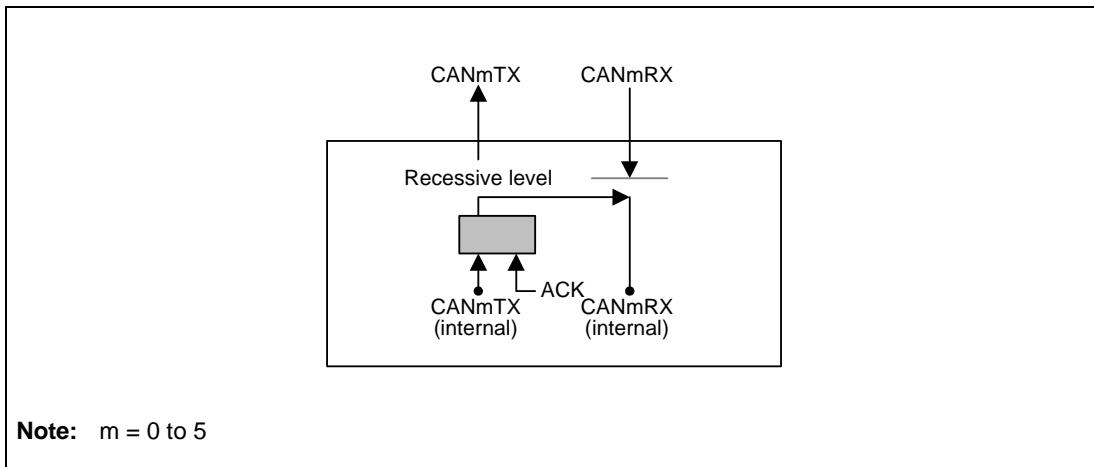


Figure 19.14 Connection when Self-Test Mode 1 is Selected

#### 19.9.4 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the RTMPS[6:0] bits in the RSCAN0GTSTCFG register. Data in the set page can be read from and written to the RSCAN0RPGACCr register ( $r = 0$  to 63). The available total RAM size is 14592 bytes ( $3900_{H}$ ).

#### 19.9.5 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel.

**Figure 19.15** shows the connection for inter-channel communication test.

##### CAUTION

Inter-channel communication test is not supported in 48 pins product.

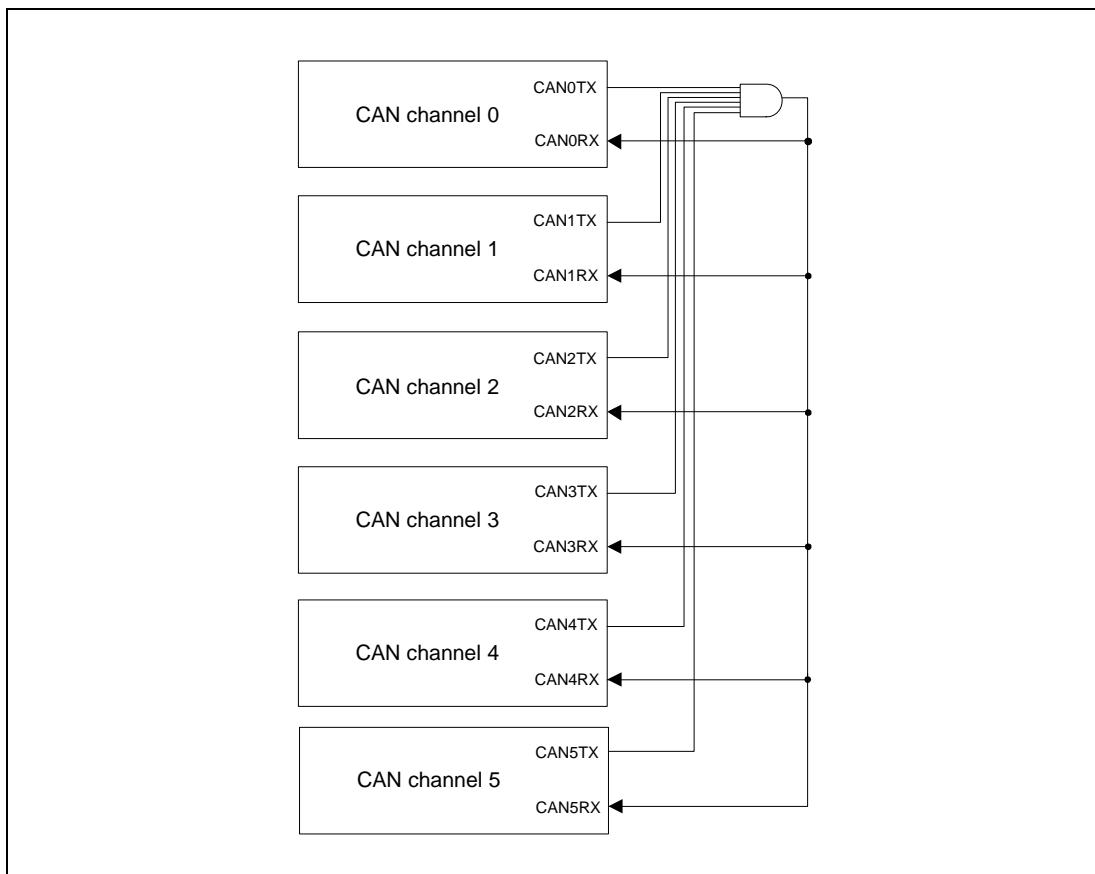


Figure 19.15 Connection for Inter-Channel Communication Test

## 19.10 RS-CAN Setting Procedure

### 19.10.1 Initial Settings

The RS-CAN module initializes the CAN RAM after the MCU is reset. The RAM initialization time is 7298 cycles of the pclk. The GRAMINIT flag in the RSCAN0GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. **Figure 19.16** shows the CAN setting procedure after the MCU is reset.

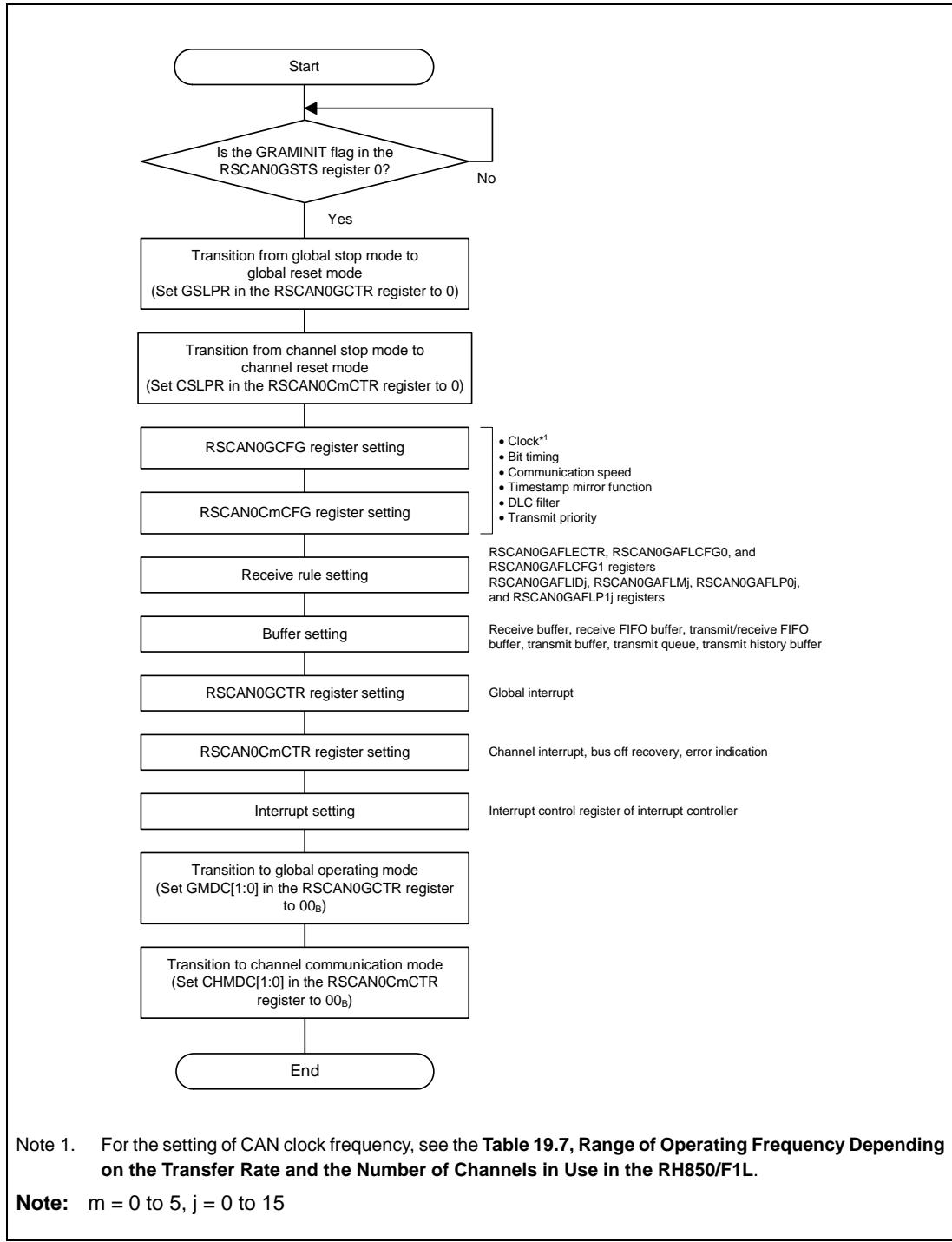


Figure 19.16 CAN Setting Procedure after the MCU is Reset

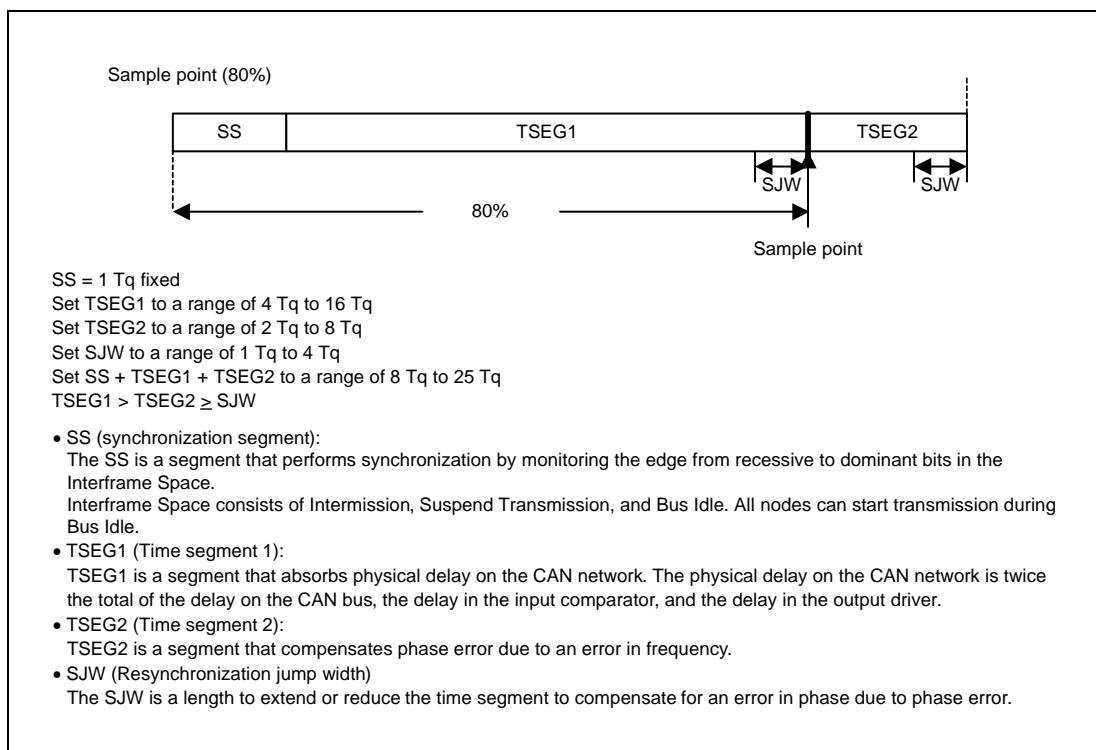
### 19.10.1.1 Clock Setting

Set the CAN clock (fCAN) as a clock source of the RS-CAN module. Select the clk\_xincan or clkc using the DCS bit in the RSCAN0GCFG register.

### 19.10.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the RSCAN0CmCFG register for each channel. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as Tq). 1 Tq is equal to one CANmTq clock cycle. The CANmTq clock is obtained by selecting the clock source with the DCS bit in the RSCAN0GCFG register and selecting the clock division ratio with the BRP[9:0] bits in the RSCAN0CmCFG register.

**Figure 19.17** shows the bit timing chart. **Table 19.96** shows an example of bit timing setting.



**Figure 19.17 Bit Timing Chart**

**Table 19.96 Example of Bit Timing Setting**

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 19.17.
	SS	TSEG1	TSEG2	SJW	
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00

### 19.10.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value (BRP[9:0] bits in the RSCAN0CmCFG register), and Tq count per bit time.

**Figure 19.18** shows the CAN clock control block diagram, and **Table 19.97** shows an example of the communication speed setting.

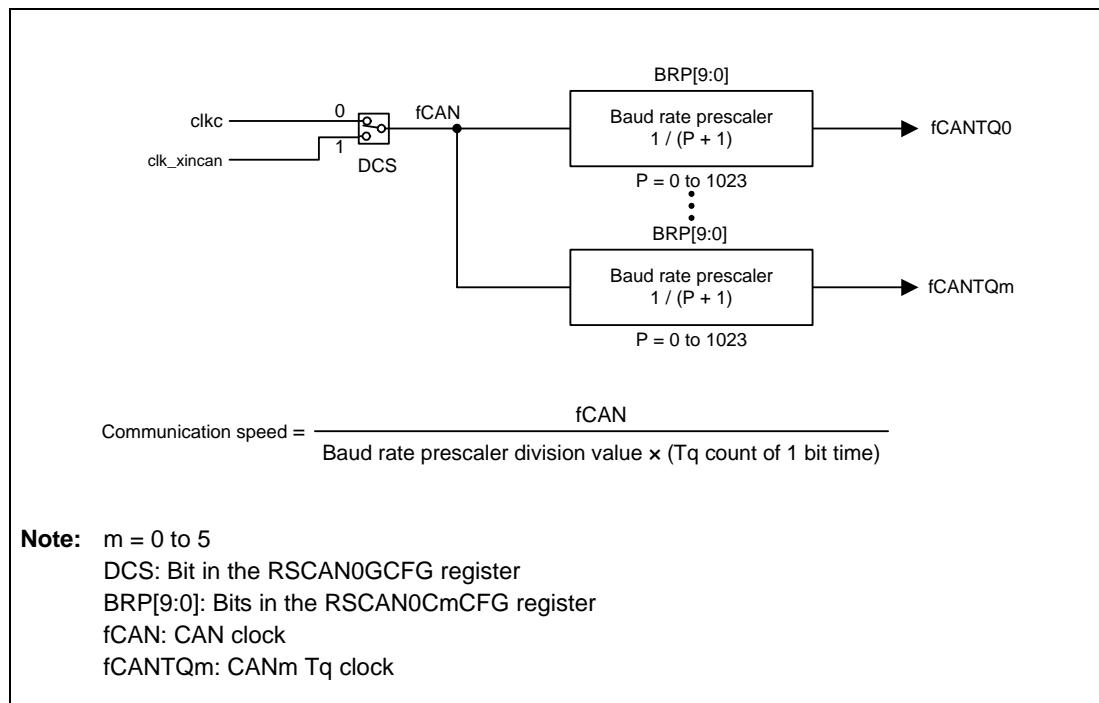


Figure 19.18 CAN Clock Control Block Diagram

Table 19.97 Example of Communication Speed Setting

Communication speed \ fCAN	40 MHz	32 MHz	24 MHz	16 MHz	8 MHz
1 Mbps	8 Tq (5) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (3) 12 Tq (2) 24 Tq (1)	8 Tq (2) 16 Tq (1)	8 Tq (1)
500 Kbps	8 Tq (10) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (6) 12 Tq (4) 24 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (2) 16 Tq (1)
250 Kbps	8 Tq (20) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (12) 12 Tq (8) 24 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (4) 16 Tq (2)
125 Kbps	8 Tq (40) 20 Tq (16)	8 Tq (32) 16 Tq (16)	8 Tq (24) 12 Tq (16) 24 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (8) 16 Tq (4)

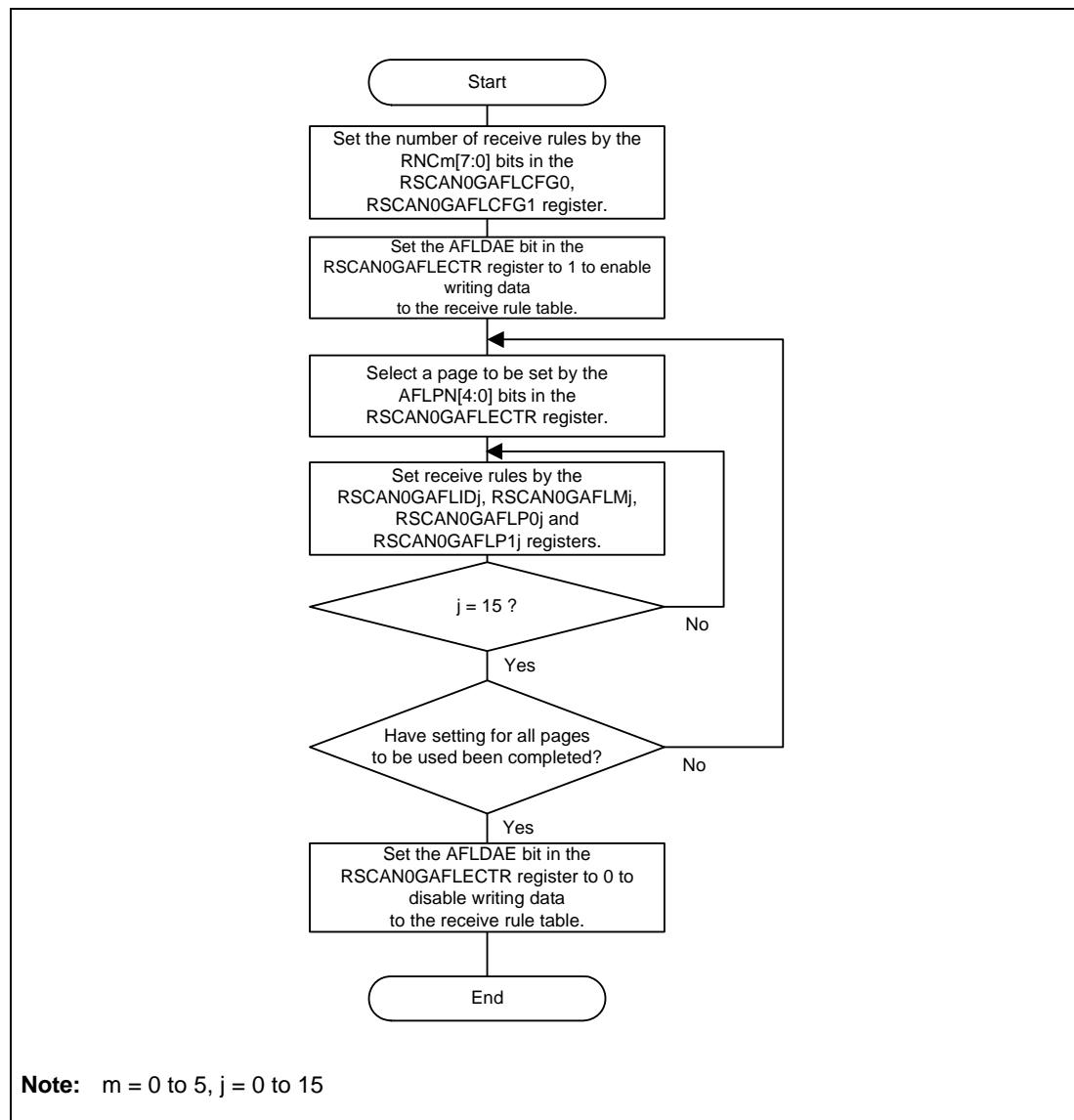
**Note:** Values in ( ) are baud rate prescaler division values.

#### 19.10.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 23 by the AFLPN[4:0] bits in the RSCAN0GAFLECTR register. Set receive rule table write enable/disable using the AFLDAE bit.

**Figure 19.19** shows the receive rule setting procedure.

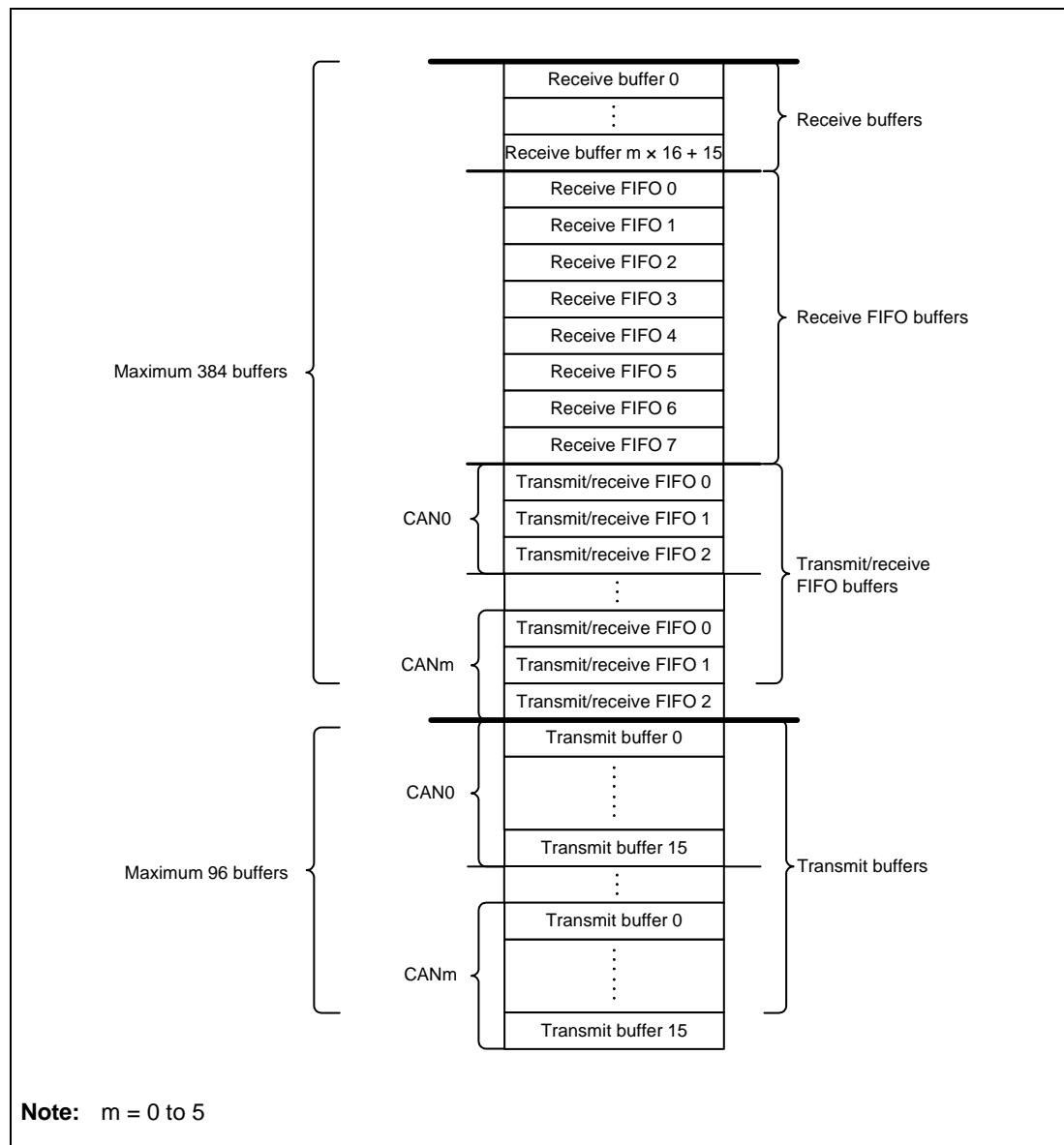


**Figure 19.19** Receive Rule Setting Procedure

### 19.10.1.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

**Figure 19.20** shows the buffer configuration. **Figure 19.21** shows the buffer setting procedure.



**Figure 19.20** Buffer Configuration

#### CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.

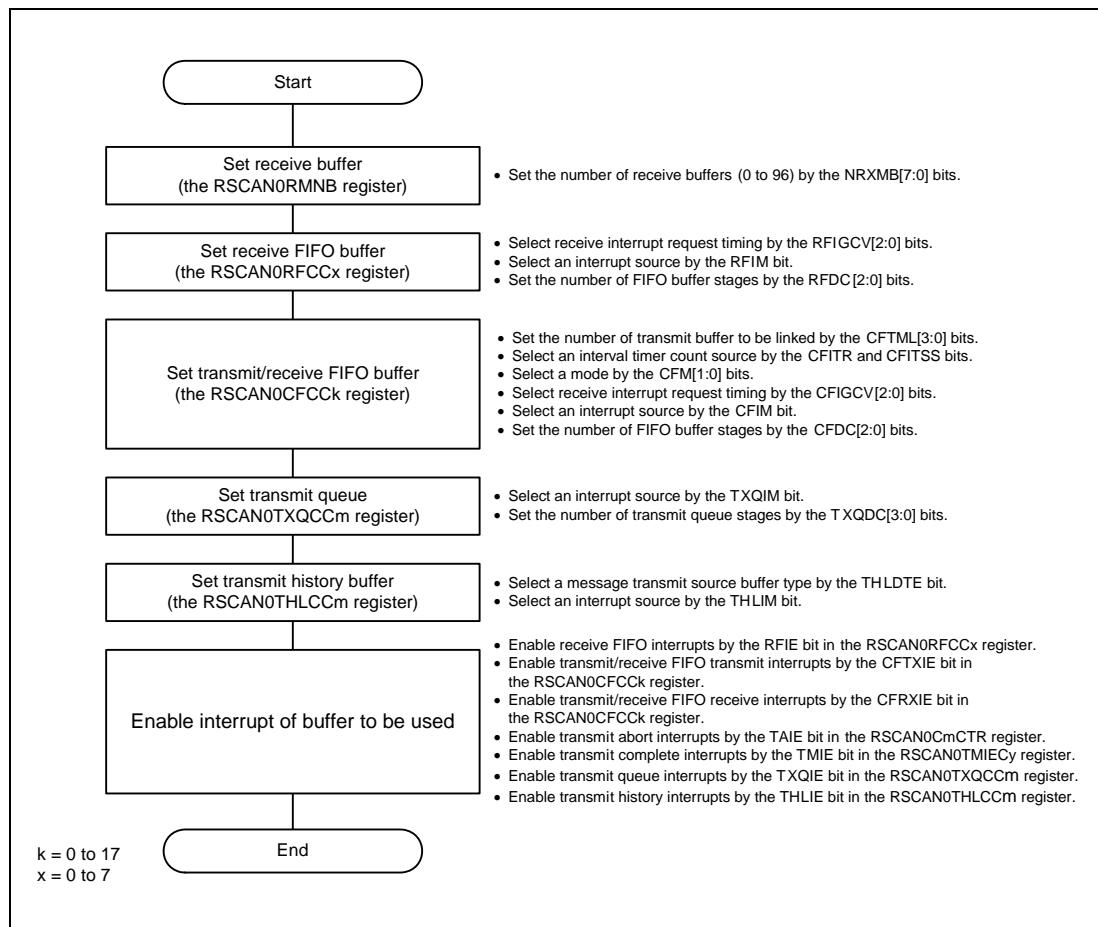


Figure 19.21 Buffer Setting Procedure

## 19.10.2 Reception Procedure

### 19.10.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCAN0RMNDy register ( $y = 0$  to  $2$ ,  $q = 0$  to  $95$ ) is set to 1 (receive buffer  $q$  contains a new message). Messages can be read from the RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. **Figure 19.22** shows the receive buffer reading procedure.

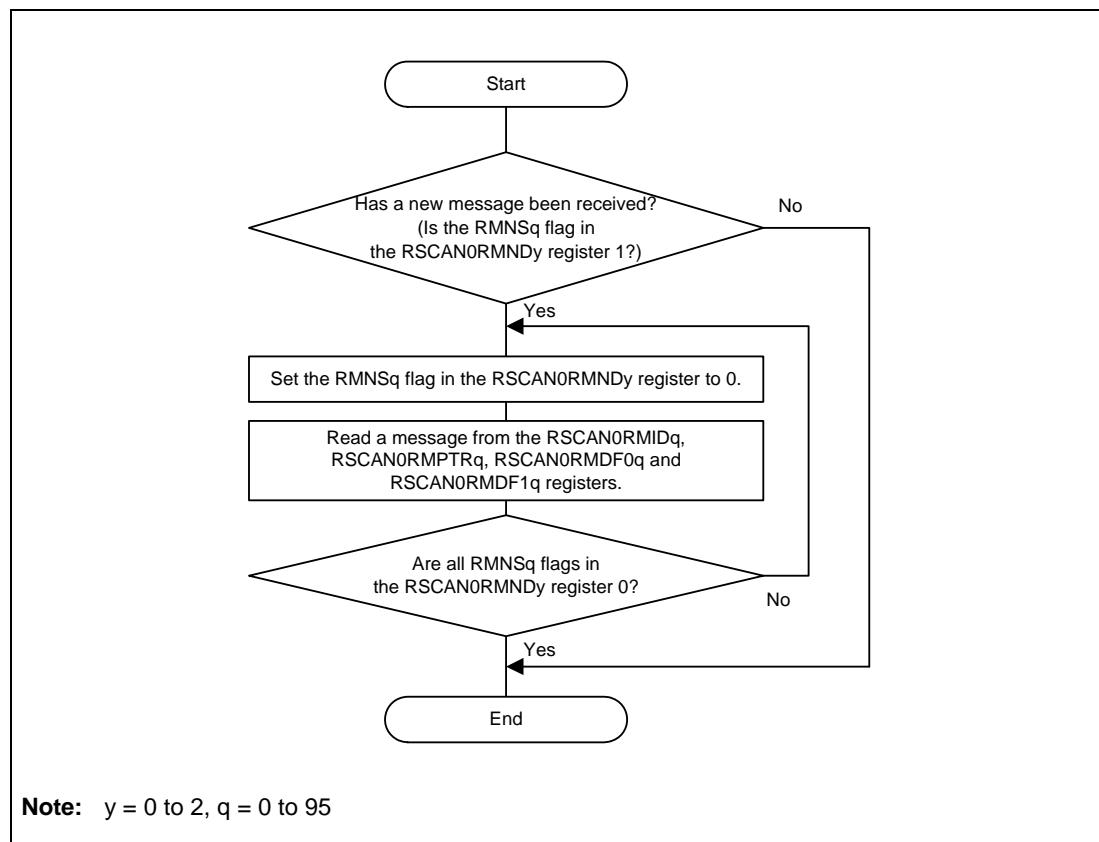


Figure 19.22 Receive Buffer Reading Procedure

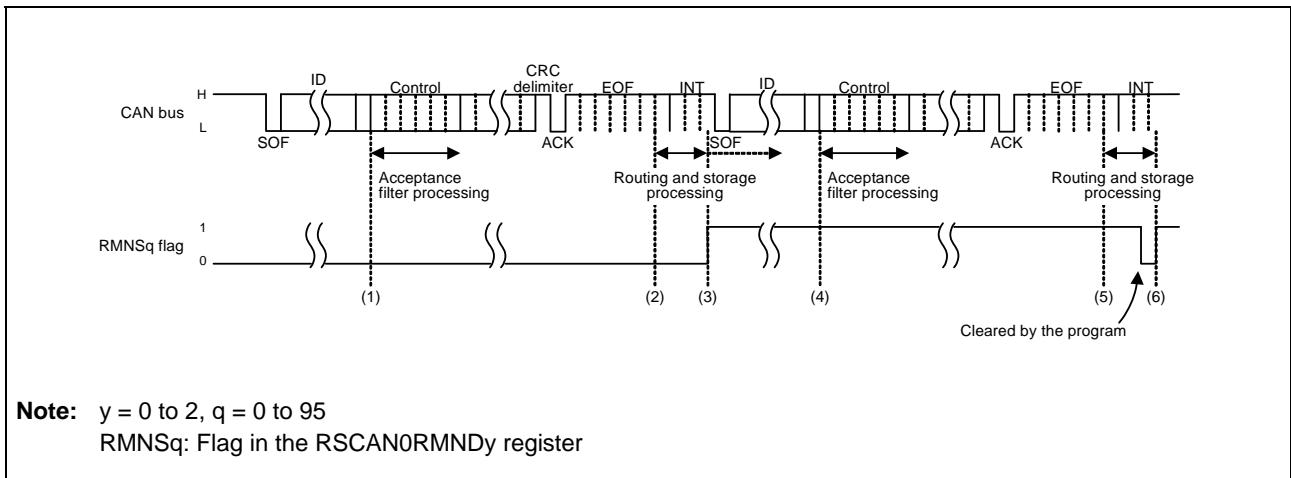


Figure 19.23 Receive Buffer Reception Timing Chart

- (1) When the ID field in a message has been received, the acceptance filter processing starts.
- (2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.  
When the message storage processing starts, the RMNSq flag in the corresponding RSCAN0RMNDy register is set to 1 (the receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.
- (4) When the ID field of the next message has been received, the acceptance filter processing starts.
- (5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (6) When the corresponding RMNSq flag is cleared to 0 (the receive buffer contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag should not be cleared to 0 during storage of messages.

### 19.10.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message count display counter (RFMC[7:0] bits in the RSCAN0RFSTS $x$  register ( $x = 0$  to 7) or CFMC[7:0] bits in the RSCAN0CFSTS $k$  register ( $k = 0$  to 17)) is incremented. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RSCAN0RFCC $x$  register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCAN0CFCC $k$  register is set to 1, an interrupt request is generated. Received messages can be read from the RSCAN0RFID $x$ , RSCAN0RFPTR $x$ , RSCAN0RFDF0 $x$ , and RSCAN0RFDF1 $x$  registers for receive FIFO buffers, or from the RSCAN0CFID $k$ , RSCAN0CFPTR $k$ , RSCAN0CFDF0 $k$ , and RSCAN0CFDF1 $k$  registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message count display counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCAN0RFCC $x$  register or the CFDC[2:0] bits in the RSCAN0CFCC $k$  register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCAN0RFSTS $x$  register or the CFEMP flag in the RSCAN0CFSTS $k$  register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCAN0RFSTS $x$  register or CFRXIF flag in the RSCAN0CFSTS $k$  register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.

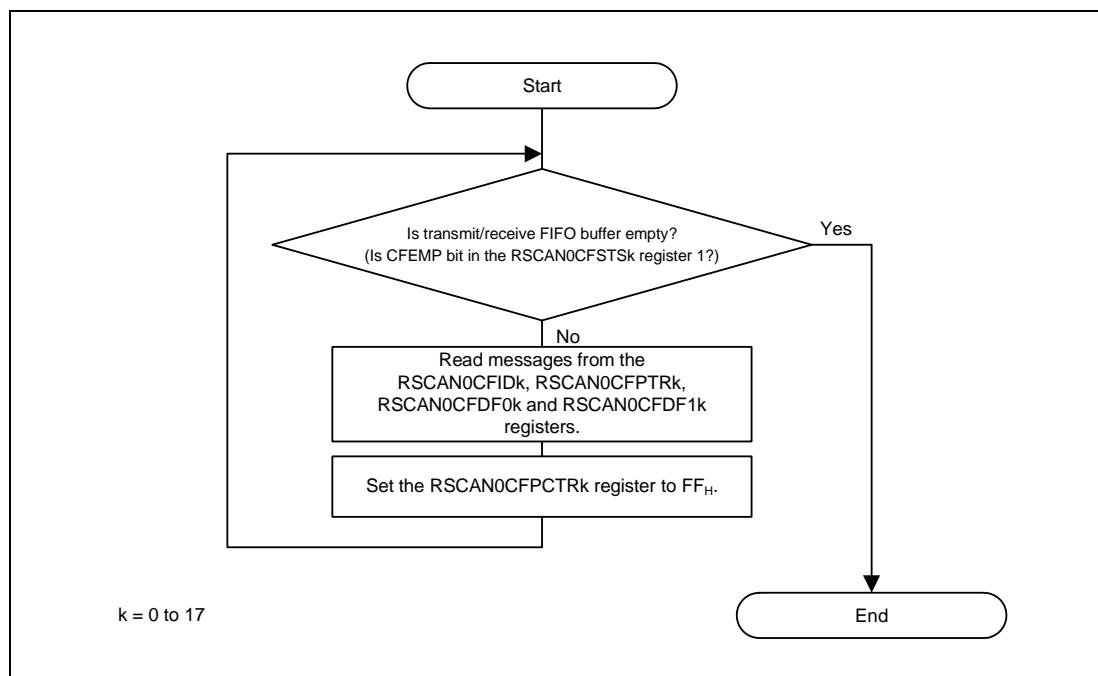


Figure 19.24 Transmit/Receive FIFO Buffer Reading Procedure

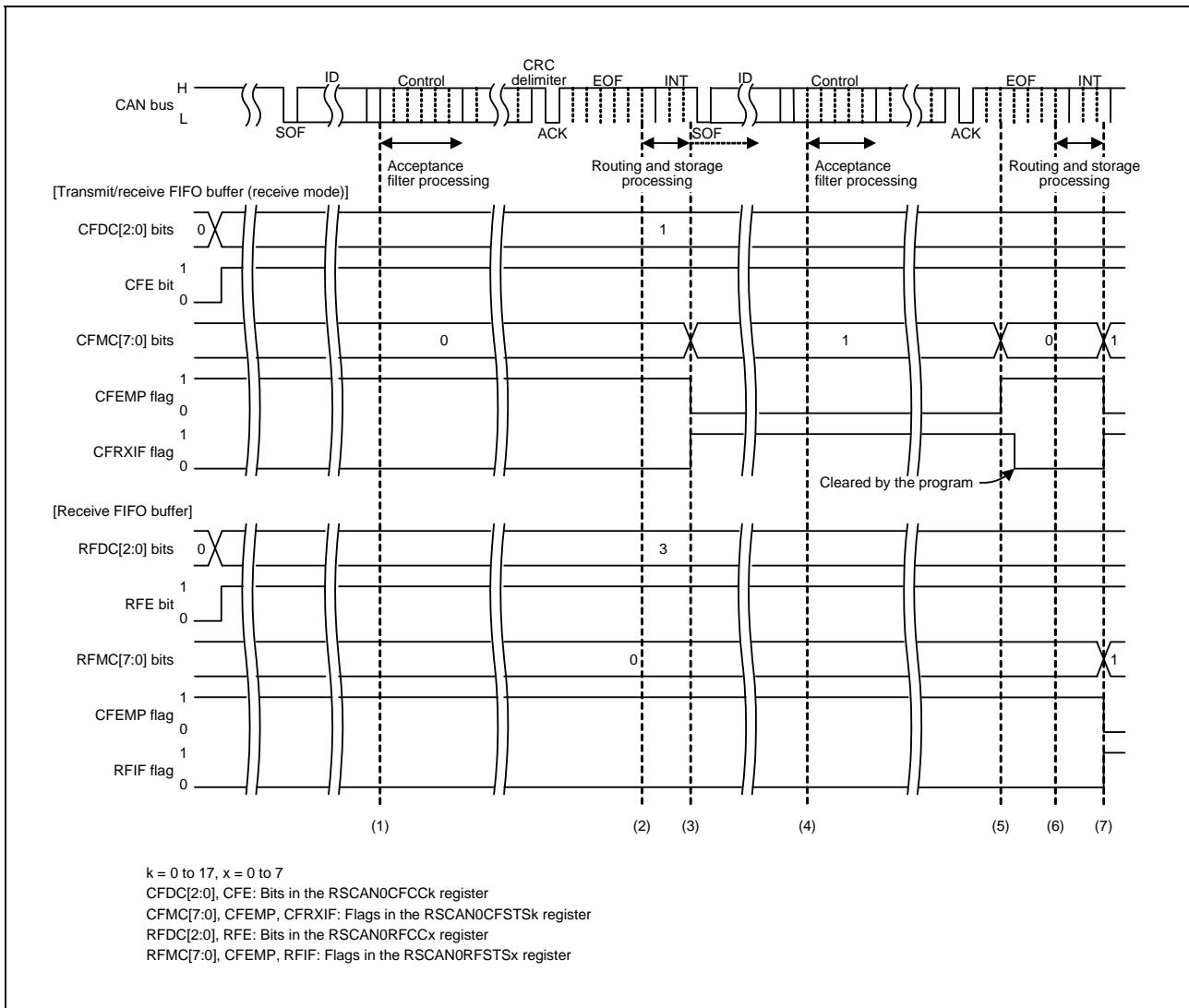


Figure 19.25 FIFO Buffer Reception Timing Chart

- (1) When the ID field in a message has been received, the acceptance filter processing starts.
- (2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (3) When the message has passed through the DLC filter processing and the CFE bit in the RSCAN0CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCk register is 001<sub>B</sub> or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the RSCAN0CFSTS $k$  register is incremented and becomes 01<sub>H</sub>. When the CFIM bit in the RSCAN0CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCAN0CFSTS $k$  register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.
- (4) When the ID field of the next message has been received, the acceptance filter processing starts.
- (5) Read received messages from the RSCAN0CFID $k$ , RSCAN0CFPTR $k$ , RSCAN0CFDF0 $k$ , and RSCAN0CFDF1 $k$  registers and write FF<sub>H</sub> to the RSCAN0CFPCTR $k$  register. This causes the

CFMC[7:0] bits in the RSCAN0CFSTS<sub>k</sub> register to be decremented. When CFMC[7:0] becomes 00<sub>H</sub>, the CFEMP flag in the RSCAN0CFSTS<sub>k</sub> register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

- (6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.
- (7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), and the CFDC[2:0] bits are set to 001<sub>B</sub> or more. The CFMC[7:0] bit value is incremented by 1 to be 01<sub>H</sub>. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).

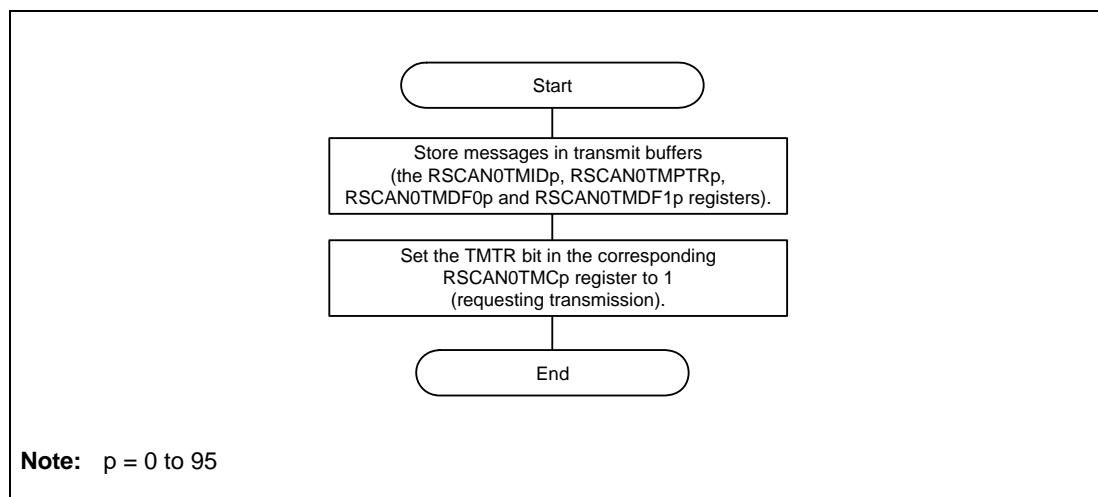
The message is stored in the receive FIFO buffer if the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bits in the RSCAN0RFCCx register are set to 001<sub>B</sub> or more. The RFMC[7:0] bits in the RSCAN0RFSTS<sub>x</sub> register are set to 01<sub>H</sub> by being incremented by 1. When the RFIM bit in the RSCAN0RFCCx register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RSCAN0RFSTS<sub>x</sub> register is set to 1 (a receive FIFO interrupt request is present).

### 19.10.3 Transmission Procedure

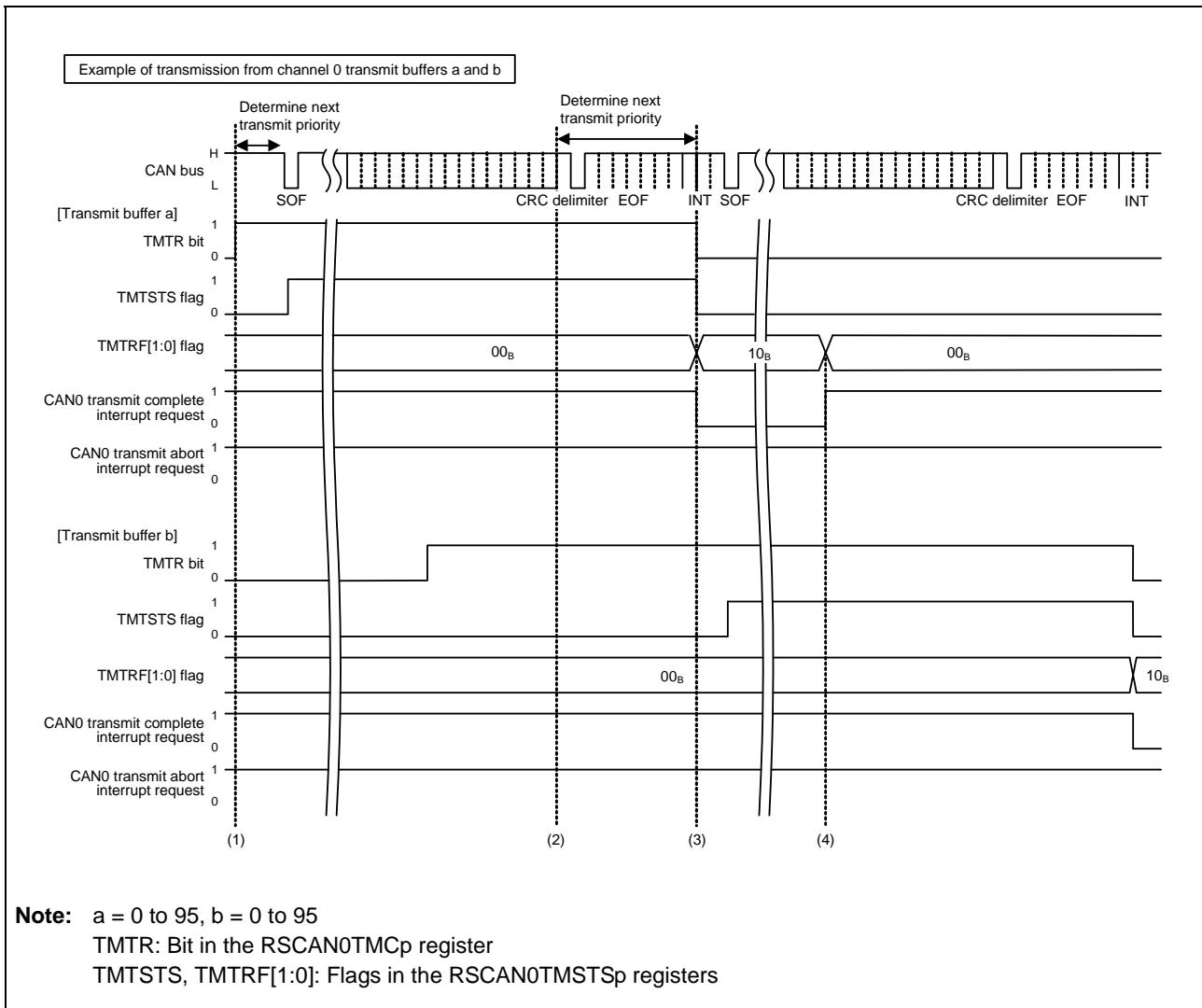
#### 19.10.3.1 Procedure for Transmission from Transmit Buffers

**Figure 19.26** shows the procedure for transmission from transmit buffers.

**Figure 19.27** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 19.28** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.



**Figure 19.26** Procedure for Transmission from Transmit Buffers



**Figure 19.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)**

- (1) When the TMTR bit in the RSCAN0TMCA register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
- (2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.
- (3) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMSTSa register is set to 10B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCA register are cleared to 0. When the TMIEa bit in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00B (transmission is in progress or no transmit request is present).

- (4) Before starting the next transmission, set the TMTRF[1:0] flag to  $00_B$ . Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is  $00_B$ .

If an arbitration-lost has occurred after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

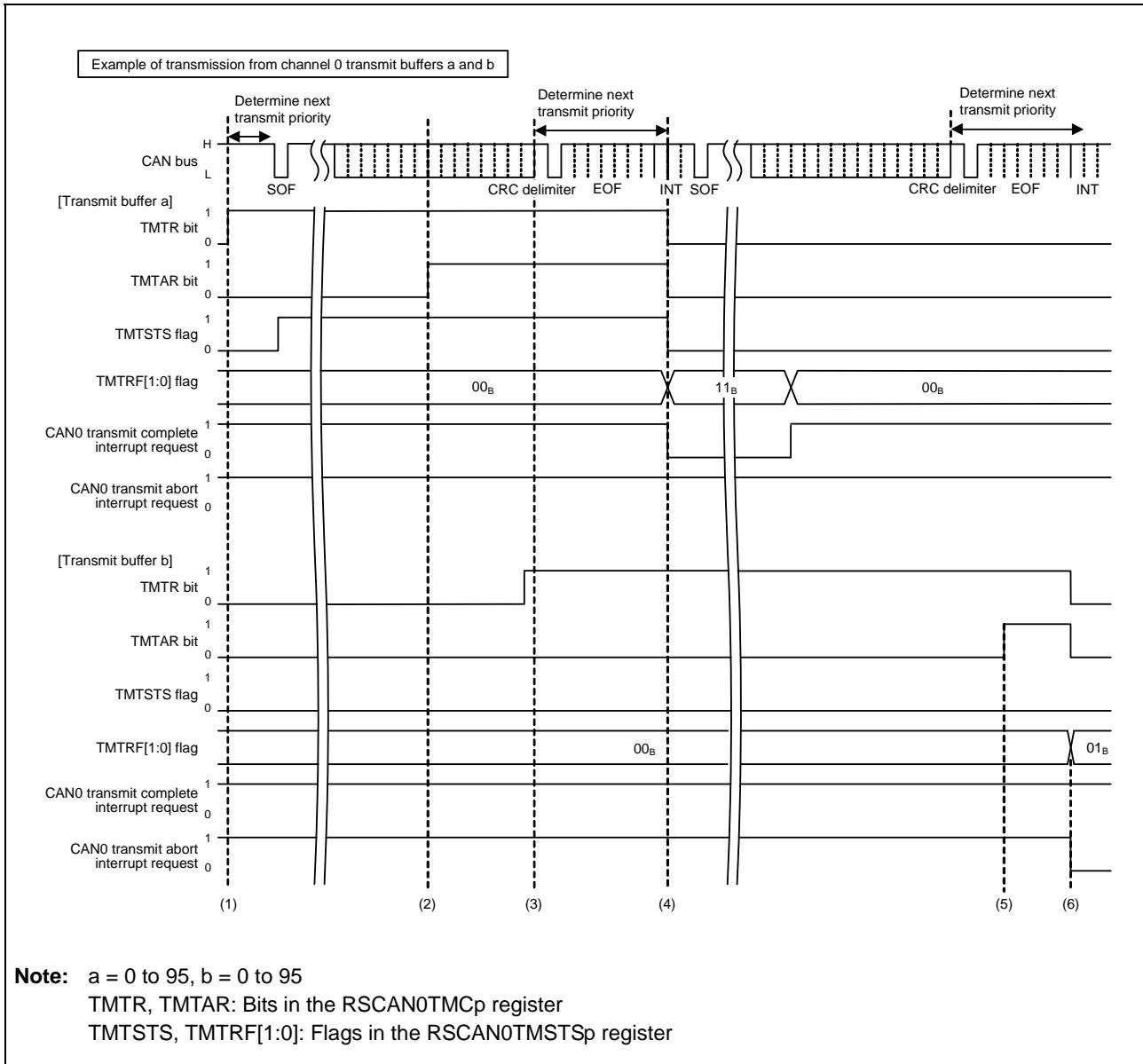


Figure 19.28 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

- (1) When the TMTR bit in the RSCAN0TMCA register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

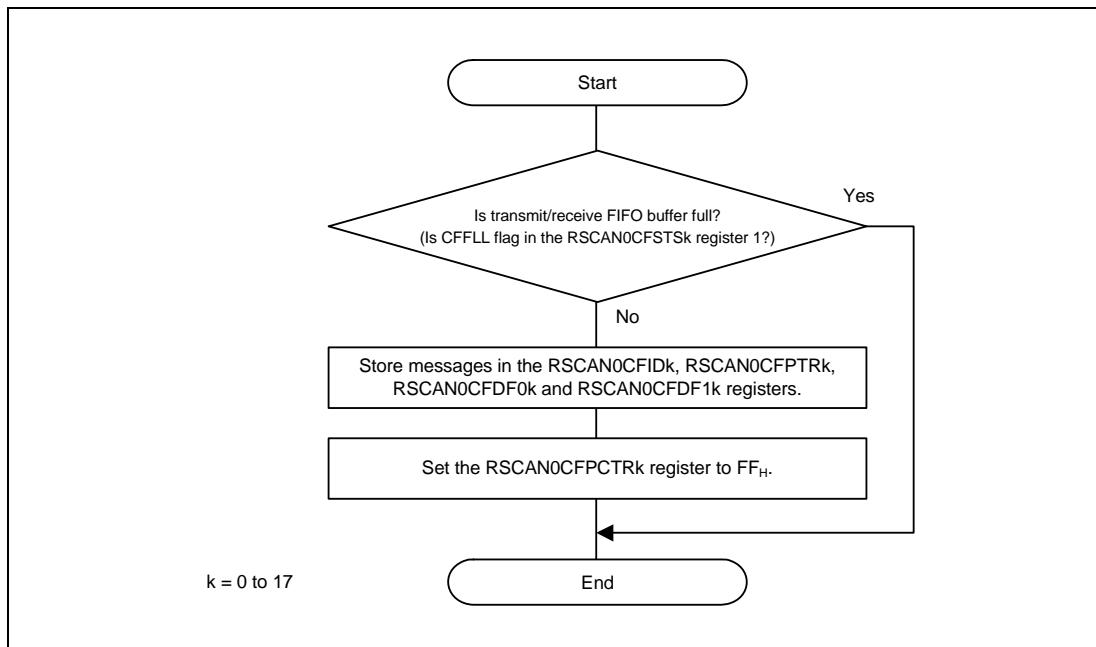
- (2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).
- (3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.
- (4) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMSTS<sub>a</sub> register is set to 11<sub>B</sub> (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMC<sub>a</sub> register are cleared to 0. When the TMIE<sub>a</sub> value in the RSCAN0TMIEC<sub>0</sub> register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00<sub>B</sub> (transmission is in progress or no transmit request is present).
- (5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.
- (6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01<sub>B</sub>. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01<sub>B</sub>. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCAN0CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00<sub>B</sub>.

If an arbitration loss has occurred after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

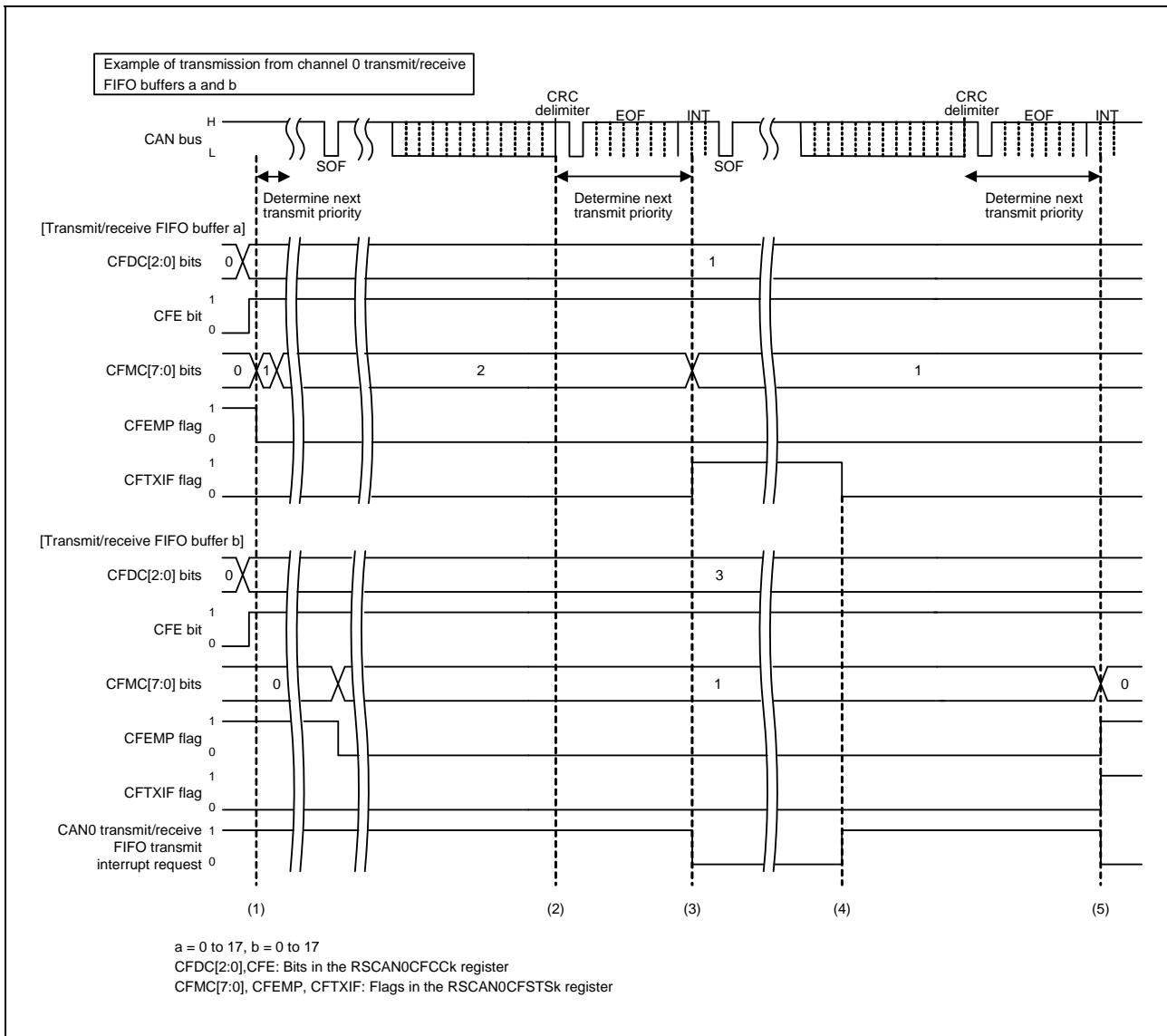
### 19.10.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

**Figure 19.29** shows the procedure for transmission from transmit/receive FIFO buffers.

**Figure 19.30** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. **Figure 19.31** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.



**Figure 19.29** Procedure for Transmission from Transmit/Receive FIFO Buffers



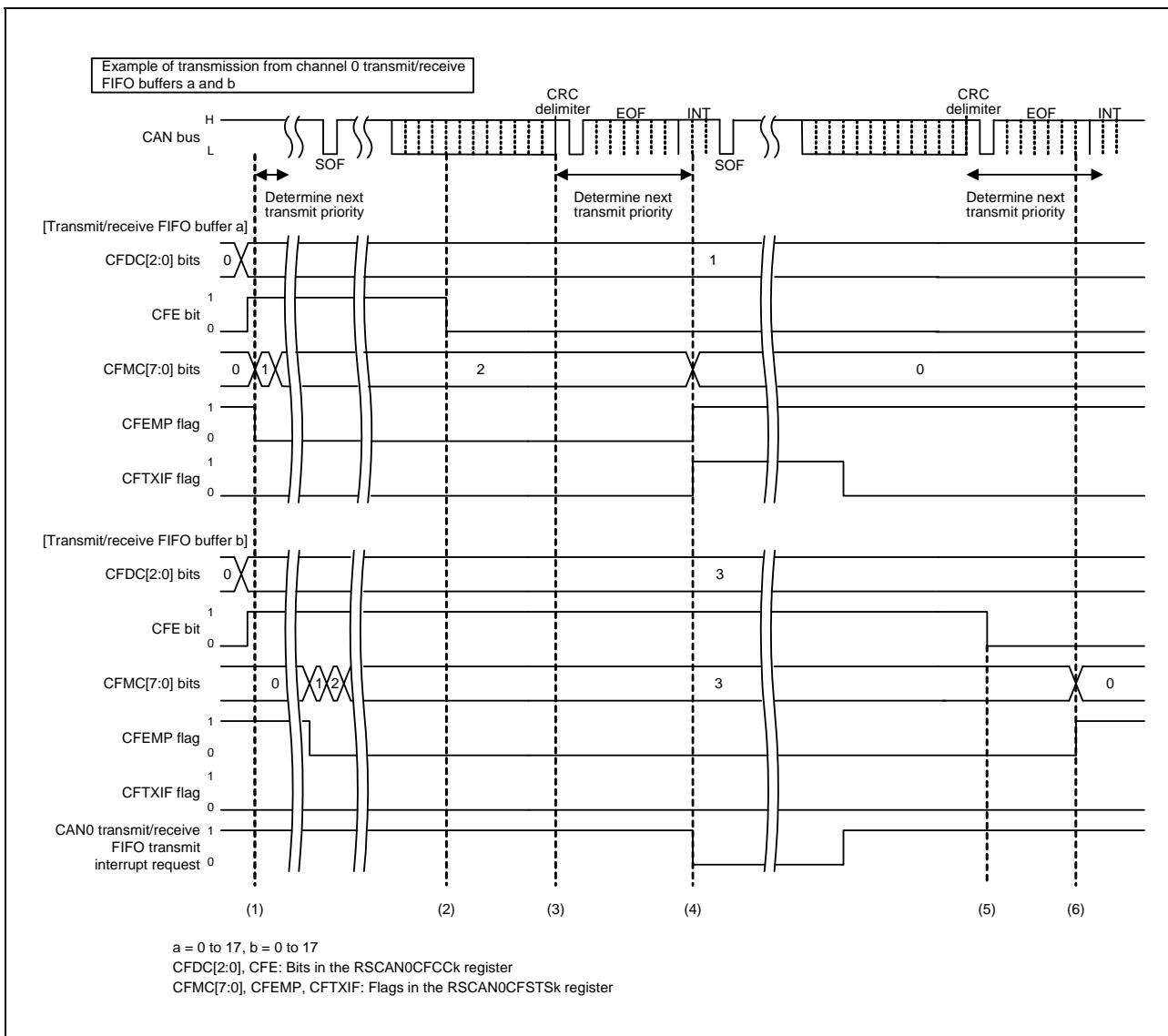
**Figure 19.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)**

- (1) While the CAN bus is idle, when the CFE bit in the RSCAN0CFCCA register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCA register is 001<sub>B</sub> (4 messages) or more and the CFMC[7:0] value in the RSCAN0CFSTS register is 01<sub>H</sub> or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.
- (2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.
- (3) When transmission completes successfully, the CFMC[7:0] value in the RSCAN0CFSTS register is decremented. Setting the CFIM bit in the RSCAN0CFCCA register to 1 (a FIFO transmit

interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTS<sub>k</sub> register to 1 (a transmit/receive FIFO transmit interrupt request is present).

- (4) The program can clear the CFTXIF flag.
- (5) Message transmission from transmit/receive FIFO buffer b of channel 0 has been completed and the CFMC[7:0] value in the RSCAN0CFSTS<sub>b</sub> register is decremented. The CFMC[7:0] bits are cleared to 00<sub>H</sub> and therefore the CFEMP flag in the RSCAN0CFSTS<sub>k</sub> register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCAN0CFSTS<sub>a</sub> and RSCAN0CFSTS<sub>b</sub> register is set to 1 (the transmit/receive FIFO buffer is full).



**Figure 19.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)**

- (1) While the CAN bus is idle, when the CFE bit in the RSCAN0CFCCa register ( $a = 0$  to 17) is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCa register is 001<sub>B</sub> (4 messages) or more and the CFMC[7:0] value in the RSCAN0CFSTS<sub>a</sub> register is 01<sub>H</sub> or more, the priority determination processing starts to determine the highest-priority transmit

message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

- (2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).
- (3) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. In this figure, transmit/receive FIFO buffer b is not selected as a buffer for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.
- (4) When transmit completes successfully, the CFMC[7:0] value is cleared to  $00_H$ . Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTS<sub>a</sub> register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.
- (5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer b), transmit/receive FIFO buffers cannot be disabled immediately even if the CFE bit in the RSCAN0CFCC<sub>b</sub> register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCAN0CFSTS<sub>b</sub> register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)
- (6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFMC[7:0] bits in the RSCAN0CFSTS<sub>b</sub> register are cleared to  $00_H$  and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer is immediately disabled. (The CFMC[7:0] bits are cleared to  $00_H$  and the CFEMP flag is set to 1.)

### 19.10.3.3 Procedure for Transmission from the Transmit Queue

Figure 19.32 shows the procedure for transmission from the transmit queue.

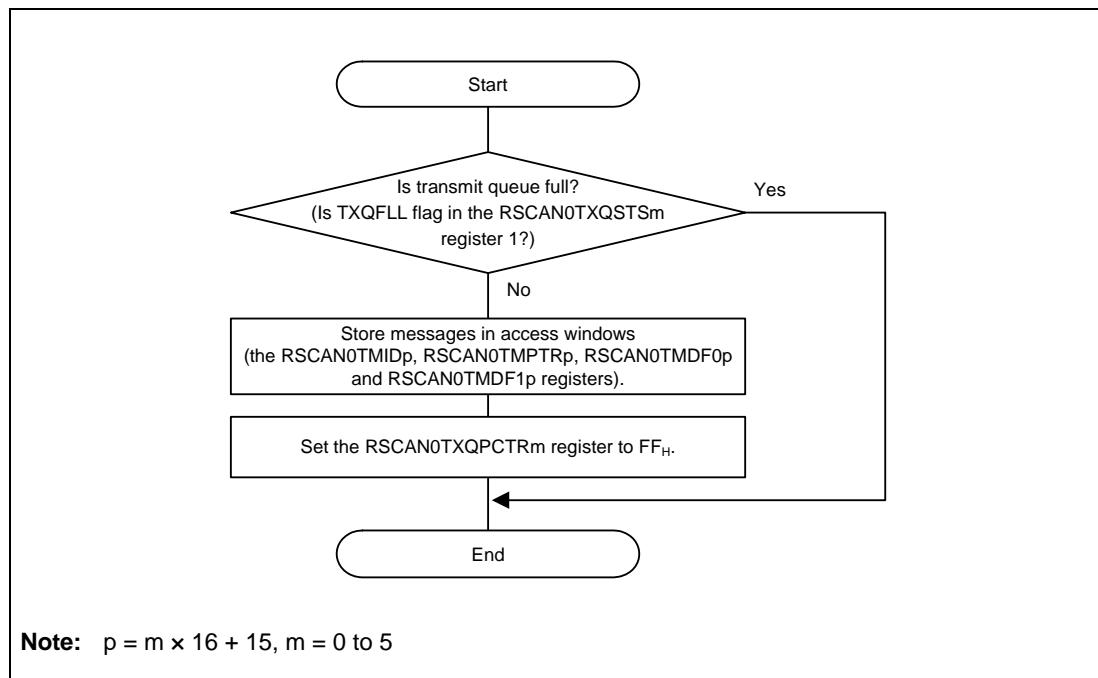


Figure 19.32 Procedure for Transmission from the Transmit Queue

#### 19.10.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCAN0THLACCM register. The next data can be accessed by writing FF<sub>H</sub> to the corresponding RSCAN0THLPCTRm register (m = 0 to 5) after reading a set of data. **Figure 19.33** shows the transmit history buffer reading procedure.

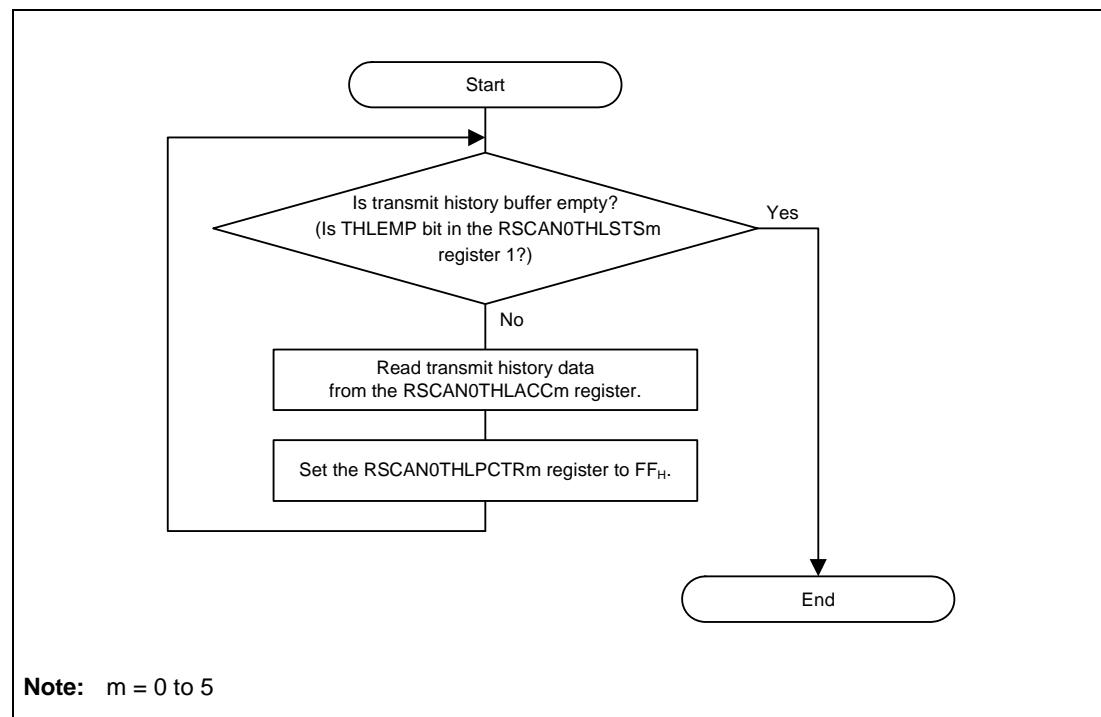


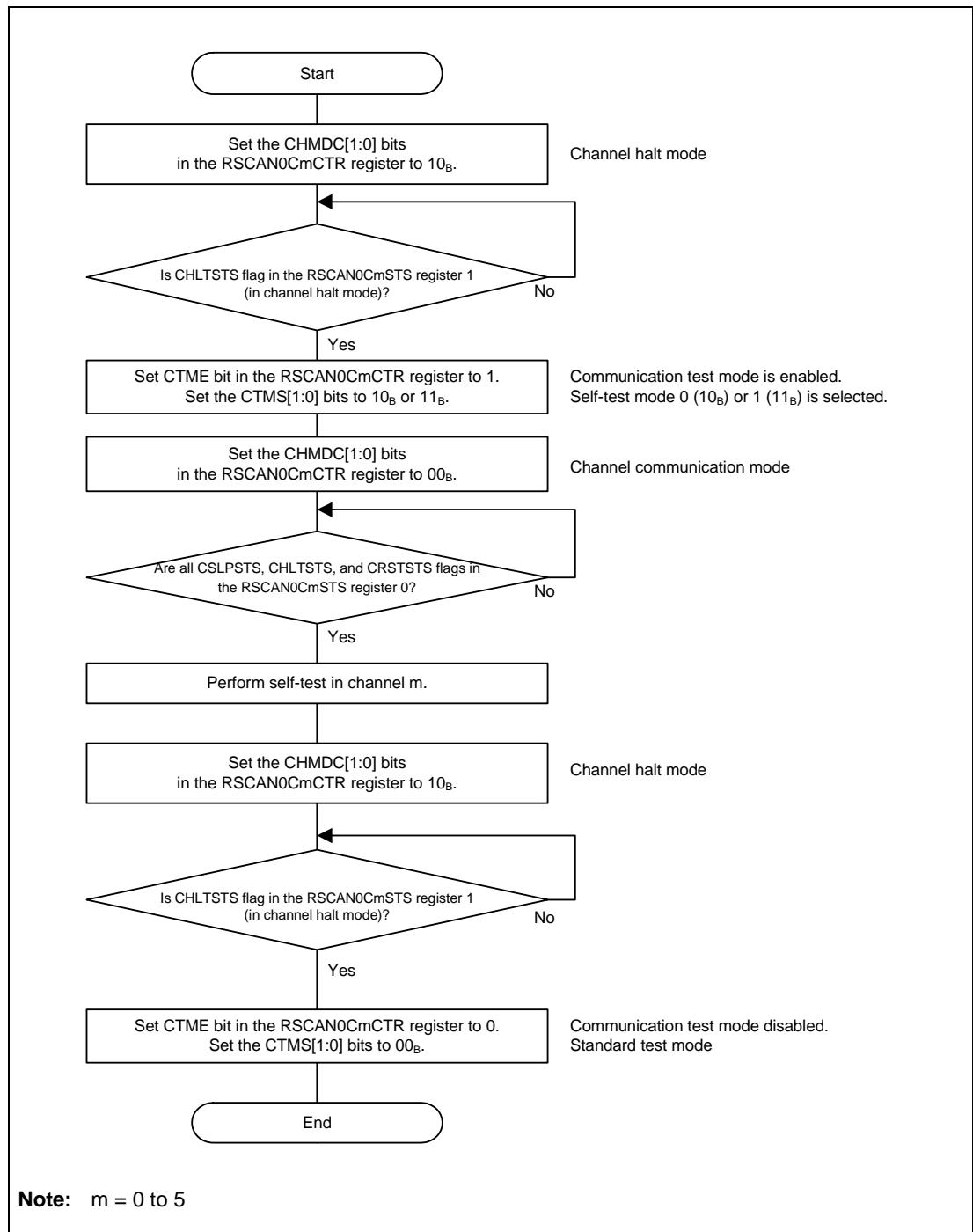
Figure 19.33 Transmit History Buffer Reading Procedure

## 19.10.4 Test Settings

### 19.10.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

**Figure 19.34** shows the self-test mode setting procedure.



**Figure 19.34** Self-Test Mode Setting Procedure

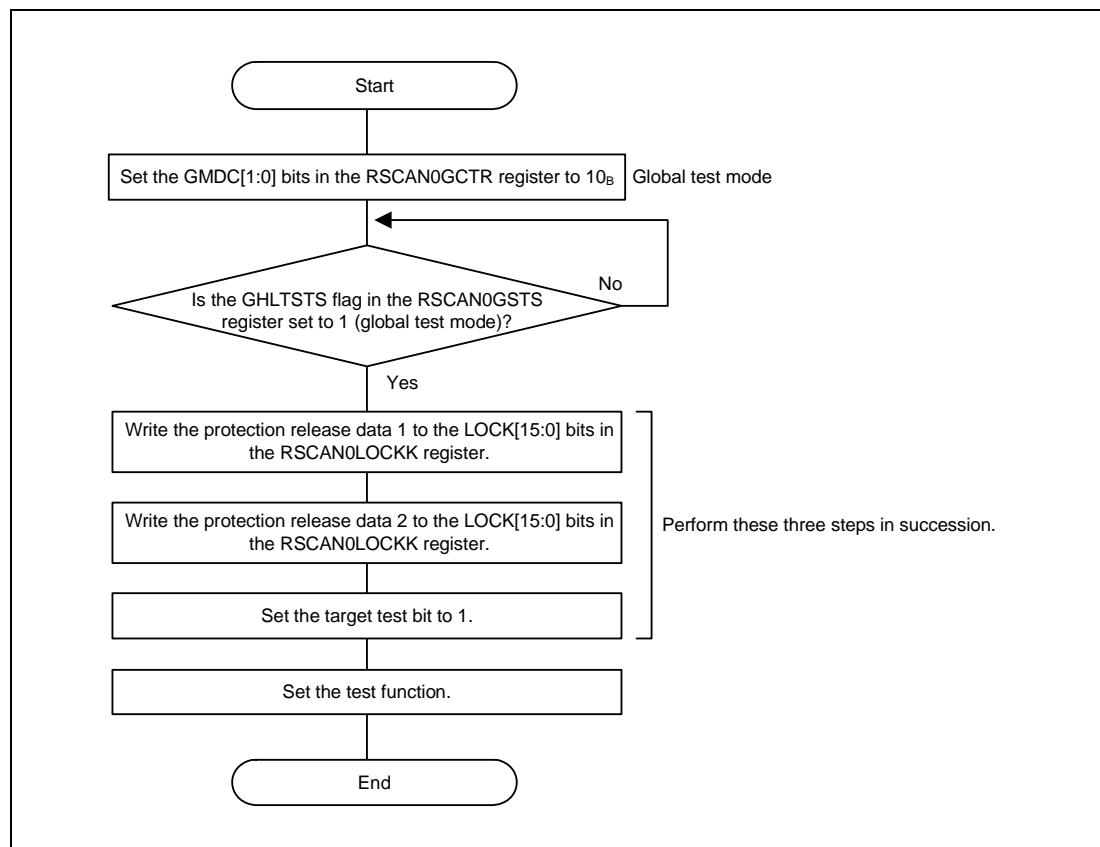
### 19.10.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 19.98** is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RSCAN0GLOCKK register, then set the target test bit to 1.

**Table 19.98 Protection Release Data for Test Function**

Test Function	Protection Release Data 1	Protection Release Data 2	Target Bit
RAM test	7575 <sub>H</sub>	8A8A <sub>H</sub>	RTME bit in the RSCAN0GTSTCTR register

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data 1. **Figure 19.35** shows the procedure for releasing the protection.

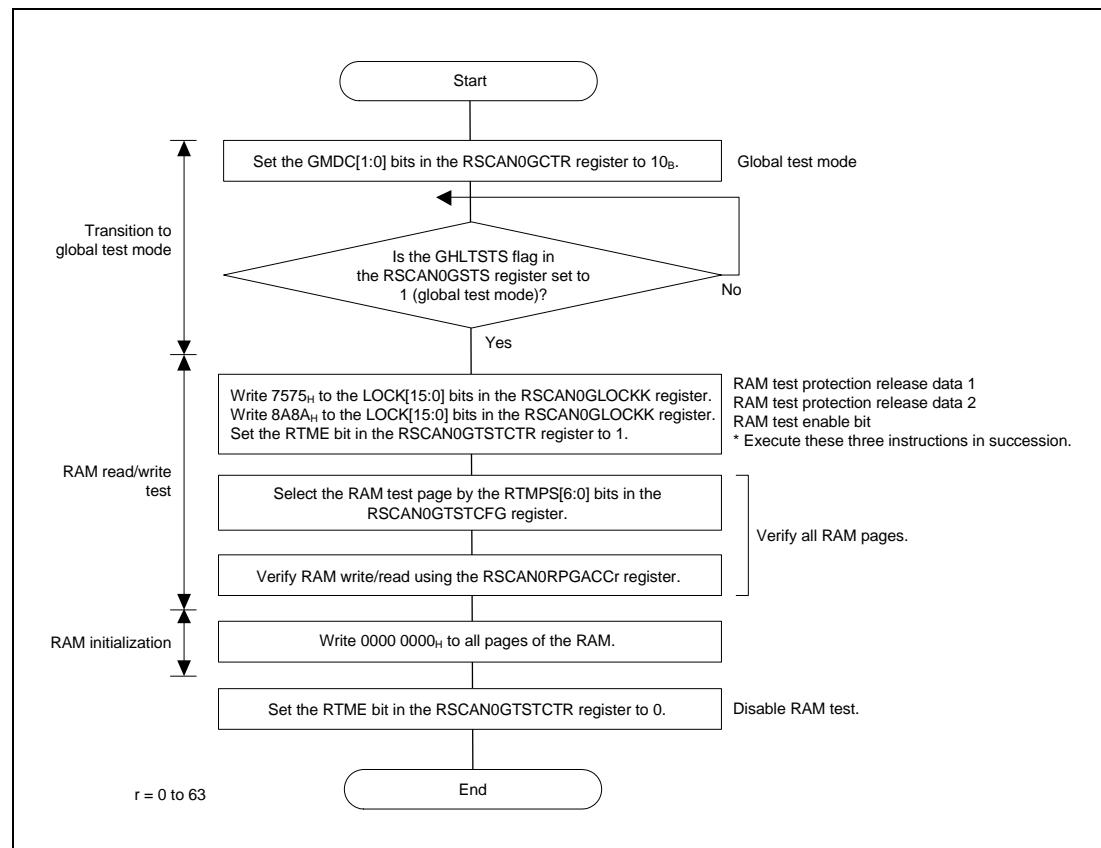


**Figure 19.35 Protection Release Procedure**

### 19.10.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write  $0000\ 0000_H$  to all pages of the CAN RAM.

**Figure 19.36** shows the RAM test setting procedure.

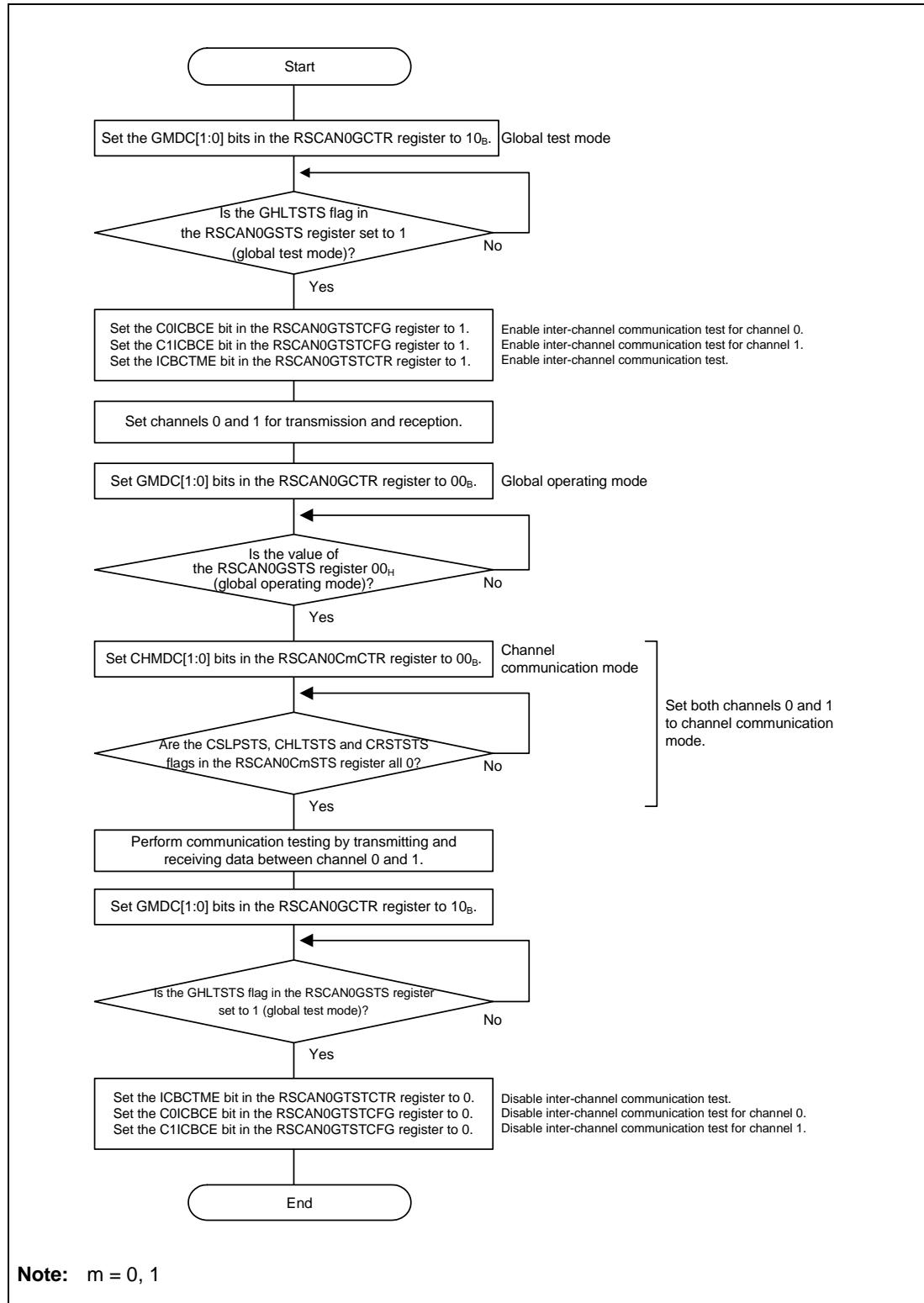


**Figure 19.36** RAM Test Setting Procedure

#### 19.10.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

**Figure 19.37** shows the inter-channel communication test setting procedure.



**Figure 19.37** Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

## 19.11 Detection and Correction of Errors in RS-CAN RAM

### 19.11.1 ECC for the RSCAN0 RAM

**Table 19.99** gives an outline of the ECC functions for the RSCAN0 RAM.

**Table 19.99 List of the ECC Functions for the RSCAN0 RAM**

Item	Outline of Functions
ECC error detection/correction	The RAM is checked for ECC errors. The following options are selectable. <ul style="list-style-type: none"> <li>• 2-bit error detection and 1-bit error detection/correction</li> <li>• 2-bit error detection and 1-bit error detection</li> </ul> The ECC error detection/correction can be disabled by using through mode. With the initial settings, error detection/correction is enabled.
Error notification	When an ECC 2-bit error is generated, the error is notified. <ul style="list-style-type: none"> <li>• Error notification can be enabled or disabled when an ECC 2-bit error is detected.</li> </ul> In the initial setting, 2-bit error notification is enabled. However, when the interrupt is masked by the FEINTFMSK register, interrupt processing is not performed.
Error status	Monitoring for the detection of two-bit ECC errors and for the detection of one-bit ECC errors is available. A register for clearing the error status is provided.

#### CAUTION

When ECC error detection/correction is performed, confirm initialization of the RSCAN0 RAM (RSCAN0GSTS.GRAMINIT = 0) before it is used.

### 19.11.2 Interrupt Request

**Table 19.100** lists the ECC interrupt request of RSCAN0 RAM.

**Table 19.100 RS-CAN ECC Interrupt Request (FE-Level Maskable Interrupt)**

Unit Interrupt Signal Name	Outline	Name	DMA Trigger Number
—	RSCAN ECC 2 bit error interrupt	INTECCDCNRAM	—

### 19.11.3 ECCRCAN0CTL — RSCAN0 ECC Control Register

The ECCRCAN0CTL register controls the mode of the ECC and the status for RSCAN0.

Bits 7, 5 and 4 should be set (written) while the RSCAN0 operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be  $01_B$ .

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC7 1000<sub>H</sub>

**Value after reset:** Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDI	—	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W*	R/W*	R	R	R	R/W*	R/W*	R	R/W	R	R/W	R/W	R	R	R	R
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—

Note 1. These bits are always read as 0.

Table 19.101 ECCRCAN0CTL Register Contents (1/2)

Bit position	Bit Name	Function
15	EMCA1	Access control bits 1 and 0 to ECC mode selection bit
14	EMCA0	These bits specify whether modifying the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are $01_B$ , writing to bit 7 is enabled.
13 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECER2C	2-bit ECC error detection flag clear bit This bit is used to clear the 2 bit error detection flag of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-bit ECC error correction accumulation flag clear bit This bit is used to clear the 1 bit error detection/correction flag of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC function through mode selection bit This bit is used to set enabling and disabling of ECC.  Setting this bit to 1 disables ECC function. When writing to this bit, (0, 1) must be written to (EMCA1, EMCA0) at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-bit error correction enable bit This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.

**Table 19.101 ECCRCAN0CTL Register Contents (2/2)**

<b>Bit position</b>	<b>Bit Name</b>	<b>Function</b>
4	EC2EDIC	<p>2-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a INTECCDCNRAM interrupt will not be generated. 1: When 2-bit error is detected, a INTECCDCNRAM interrupt will be generated. (initial value)</p>
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ECER2F	<p>2-bit error detection flag bit This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDCNRAM) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt signal will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred.</p>
1	ECER1F	<p>1-bit error detection/correction flag bit This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred.</p>
0	ECEMF	<p>ECC error message flag This flag indicates whether an error exists in the current read data. This bit is updated whenever the RAM reads data. This bit might be set if it is read before initialization of the RAM. This bit is cleared at the time of through mode enable selection (ECTHM = 1) and when there is no 1 bit error in the decode circuit input data. 0: The currently-read RAM data does not have bit errors. 1: The currently-read RAM data have bit errors.</p>

**CAUTION**

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

### 19.11.4 ECCRCAN0TMC — RSCAN0 ECC Test Mode Control Register

The ECCRCAN0TMC register switches to and controls the test mode.

This register can be used when RS-CAN is not accessed to RAM.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC7 1004<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W* <sup>1</sup>	R/W* <sup>1</sup>	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

**Table 19.102 ECCRCAN0TMC Register Contents (1/2)**

Bit position	Bit Name	Function
15	ETMA1	Access control bits 1 and 0 to ECC test mode bit
14	ETMA0	These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10 <sub>B</sub> , writing to bit 7 is enabled.
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: ECCRCAN0TED, ECCRCAN0TRC, ECCRCAN0SYND, ECCRCAN0HORD, ECCRCAN0ECRD, ECCRCAN0ERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading ECCRCAN0TED register and reading destination when reading ECCRCAN0ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: The read value of the ECCRCAN0TED register is the write value of the ECCRCAN0TED register. The read value of the ECCRCAN0ERDB register is the write value of the ECCRCAN0ERDB register. 1: The read value of the ECCRCAN0TED register can read RAM data. The read value of the ECCRCAN0ERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identify and select the ECC data to be stored in RAM as ECC data generated for write data or the value of the ECCRCAN0ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of ECCRCAN0ERDB register to RAM.

**Table 19.102 ECCRCAN0TMC Register Contents (2/2)**

<b>Bit position</b>	<b>Bit Name</b>	<b>Function</b>
2	ECENS	<p>ECC encoder input selection bit</p> <p>This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the ECCRCAN0TED register.</p> <p>Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously).</p> <p>This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the ECCRCAN0TED register.</p>
1	ECDCS	<p>ECC decoder input selection bit</p> <p>This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCAN0TED register.</p> <p>Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the ECCRCAN0TED register and detect errors.</p>
0	ECREIS	<p>ECC redundant bit input data selection bit</p> <p>This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCAN0ERDB register.</p> <p>Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously).</p> <p>0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the ECCRCAN0ERDB register and detect errors.</p>

### 19.11.5 ECCRCAN0TED — RSCAN0 ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

The value of the register can be used to generate ECC data or syndrome code.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), it is accessible. When ECCRCAN0TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000<sub>H</sub> is read.

This register can be used when RS-CAN is not accessed to RAM.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFC7 100C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECEDB[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ECEDB[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 19.103 ECCRCAN0TED Register Contents**

Bit position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCRCAN0TMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When ECCRCAN0TMC.ECDCS = 1, the value of this register is used to generate syndrome code and the value of this register is stored in ECC decode syndrome data register (ECCRCAN0SYND). In addition, when ECCRCAN0TMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

### 19.11.6 ECCRCAN0TRC — RSCAN0 ECC Redundant Bit Data Control Test Register

In ECC test mode, this test register, for ECC data, consists of four 8-bit registers, ECCRCAN0SYND, ECCRCAN0HORD, ECCRCAN0ECRD, and ECCRCAN0ERDB.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed.

When ECCRCAN0TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000<sub>H</sub> is read.

This register can be used when RS-CAN is not accessed to RAM.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFC7 1008<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCRCAN0SYND (See <a href="#">Section 19.11.7</a> )								ECCRCAN0HORD (See <a href="#">Section 19.11.8</a> )							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCRCAN0ECRD (See <a href="#">Section 19.11.9</a> )								ECCRCAN0ERDB (See <a href="#">Section 19.11.10</a> )							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W						

### 19.11.7 ECCRCAN0SYND — RSCAN0 ECC Decode Syndrome Data Register

In ECC test mode, this is a read-only register for storing generated syndrome code.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed.

When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC7 100B<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 19.104 ECCRCAN0SYND Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	The generated syndrome code is stored as needed.

### 19.11.8 ECCRCAN0HORD — RSCAN0 ECC 7-Bit Redundant Bit Data Hold Test Register

In ECC test mode, this register is used to store ECC data for read RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed.

When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC7 100A<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 19.105 ECCRCAN0HORD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	ECC code for read RAM data is stored as needed. When ECCRCAN0TMC.ECTRSS = 1 and ECCRCAN0TED register is read, ECC code is stored.

### 19.11.9 ECCRCAN0ECRD — RSCAN0 ECC Encode Test Register

In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC7 1009<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 19.106 ECCRCAN0ECRD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the ECCRCAN0TED register when ECCRCAN0TMC.ECENS = 1.

### 19.11.10 ECCRCAN0ERDB — RSCAN0 ECC Redundant Bit Input/Output Replacement Buffer Register

In ECC test mode, this register handles ECC data.

The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00<sub>H</sub> is read.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFC7 1008<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0

Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W						

**Table 19.107 ECCRCAN0ERDB Register Contents**

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	When ECCRCAN0TMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When ECCRCAN0TMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When ECCRCAN0TMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

### 19.11.11 SELB\_READTEST — ECCREAD Test Select Register

SELB\_READTEST is used to check read/write access to the RSCAN0 ECC registers. For detail, see **Section 15.7.11, SELB\_READTEST — ECCREAD Test Select Register**.

## 19.12 Notes on the RS-CAN Module

- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCAN0GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCAN0CmSTS register ( $m = 0$  to 5) for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCAN0TMCP) of the corresponding transmit buffer to 00H. The status register (RSCAN0TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RSCAN0TMTRSTS0 to RSCAN0TMTRSTS2, RSCAN0TMTARSTS0 to RSCAN0TMTARSTS2, RSCAN0TMTARSTS0 to RSCAN0TMTARSTS2, RSCAN0TMTARSTS0 to RSCAN0TMTARSTS2, and RSCAN0TMTASTS0 to RSCAN0TMTASTS2), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (registers RSCAN0TMIEC0 to RSCAN0TMIEC2) to 0 (transmit buffer interrupt is disabled).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not be allocated to transmit queues.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- The values of unused receive buffers (RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers), receive FIFO buffer access registers (RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers), and transmit/receive FIFO buffer access registers (RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers) are undefined when the RS-CAN module transitions to global operation mode or global test mode after exiting from global reset mode.

## Section 20 Window Watchdog Timer (WDTA)

This section contains a generic description of the Window Watchdog Timer (WDTA).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of WDTA.

### 20.1 Features of RH850/F1L WDTA

#### 20.1.1 Number of Units and Channels

This microcontroller has the following number of WDTA units.

**Table 20.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units				2		
Name				WDTAn (n = 0, 1)		

**Table 20.2 Index**

Index	Description
n	Throughout this section, the individual window watchdog timer units are identified by the index "n" : for example, WDTAnWDTE (n = 0, 1) is the WDTAn enable register.

#### 20.1.2 Register Base Address

WDTAn base addresses are listed in the following table.

WDTAn register addresses are given as offsets from the base addresses.

**Table 20.3 Register Base Addresses**

Base Address Name	Base Address
<WDTA0_base>	FFED 0000 <sub>H</sub>
<WDTA1_base>	FFED 1000 <sub>H</sub>

#### 20.1.3 Clock Supply

The WDTAn clock supply is shown in the following table.

**Table 20.4 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
WDTA0	WDTATCKI	CKSCLK_AWDTA
	Register access clock	CPUCLK2
WDTA1	WDTATCKI	LS IntOSC
	Register access clock	CPUCLK2

### 20.1.4 Interrupt Requests

WDTAn interrupt requests are listed in the following table.

**Table 20.5 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>WDTA0</b>			
INTWDTAn	WDTA0 75% interrupt	32	—
<b>WDTA1</b>			
INTWDTAn	WDTA1 75% interrupt	33	—

**Table 20.6 Interrupt Request (FE Level Non-Maskable Interrupt)**

Unit Interrupt Signal	Outline	Interrupt Name	DMA Trigger Number
<b>WDTA0</b>			
WDTAnTNMI	WDTA0 FENMI interrupt (in WDTA error detection mode with an NMI request)	WDTA0NMI	—
<b>WDTA1</b>			
WDTAnTNMI	WDTA1 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA1NMI	—

### 20.1.5 Reset Sources

WDTAn reset sources are listed in the following table. WDTAn is initialized by these reset sources.

**Table 20.7 Reset Sources**

Unit Name	Reset Source
WDTA0	Reset sources (AWORES)
WDTA1	All reset sources (ISORES)

**Note:** WDTA1 is stopped in STOP mode.

## 20.2 Overview

### 20.2.1 Functional Overview

WDTA has the following functions:

- Selection of the operation mode after reset, by using the option bytes

Enabling/disabling of WDTA, starting/stopping of the counter after reset, setting of the counter overflow time, and enabling/disabling of the VAC function can be selected. WDTA startup options to be set by the option bytes are described in **Table 20.8**.

- WDTA trigger function

Writing an activation code to the WDTA trigger register starts WDTA and restarts the counter.

Activation codes include fixed activation codes and variable activation codes (VAC function). In a variable activation code, a different value from the previous time (variable value) is written to the WDTA trigger register, which causes the counter to be restarted.

- 75% interrupt request signal

An interrupt request signal can be generated when the WDTA counter reaches 75% of the overflow interval time (this function can be enabled or disabled by the setting of WDTAnMD.WDTAnWIE).

- Window function

The period during which writing to the WDTA trigger register is valid (window-open period) can be set. Writing to the WDTA trigger register during a period other than the window-open period causes an error.

- WDTA error detection function

When an error is detected, a non-maskable interrupt request or an internal reset is generated.

For details about the error sources, see **Section 20.5.3, WDTA Error Detection**.

**Table 20.8 WDTA Start-Up Options**

Start-Up Option	Function	Description	Option Byte
OPWDEN	WDTA setting	Enables/disables the WDTA. 0: WDTA is disabled 1: WDTA is enabled	<ul style="list-style-type: none"> <li>• WDTA0: OPBT0.OPBT0[19]</li> <li>• WDTA1: OPBT0.OPBT0[23]</li> </ul>
OPWDVOF[2:0]	Overflow interval time reset value setting	Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0].	• WDTA0/WDTA1: OPBT0.OPBT0[18:16]
OPWDRUN	Start mode setting	Specifies the start mode. 0: Software trigger start mode 1: Default start mode  For details, see <b>Section 20.5.1, WDTA after Reset Release.</b>	<ul style="list-style-type: none"> <li>• WDTA0: OPBT0.OPBT0[20]</li> <li>• WDTA1: OPBT0.OPBT0[24]</li> </ul>
OPWDVAC	Variable activation code selection	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable)  When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (ACH). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, see <b>Section 20.5.2, WDTA Trigger</b> and <b>20.5.2.1, Calculating an Activation Code when the VAC Function is Used.</b>	<ul style="list-style-type: none"> <li>• WDTA0: OPBT0.OPBT0[22]</li> <li>• WDTA1: OPBT0.OPBT0[26]</li> </ul>

**NOTE**

For the option byte settings, see **Section 35.11, Option Bytes.**

## 20.2.2 Block Diagram

Figure 20.1 shows the main components of the WDTA.

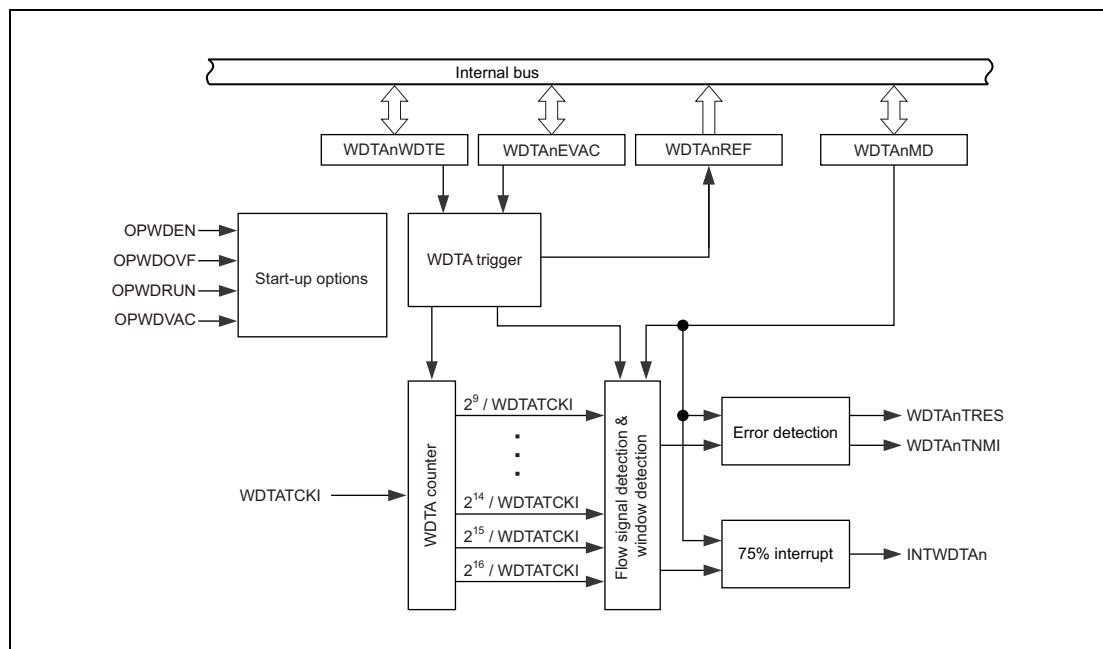


Figure 20.1 Block Diagram of the Window Watchdog Timer A

## 20.3 Registers

### 20.3.1 List of Registers

WDTA registers are listed in the following table.

For details about <WDTAn\_base>, see **Section 20.1.2, Register Base Address**.

Table 20.9 List of Registers

Module	Register	Symbol	Address
WDTAn	WDTA enable register	WDTAnWDTE	<WDTAn_base> + 0000 <sub>H</sub>
WDTAn	WDTA enable VAC register	WDTAnEVAC	<WDTAn_base> + 0004 <sub>H</sub>
WDTAn	WDTA reference value register	WDTAnREF	<WDTAn_base> + 0008 <sub>H</sub>
WDTAn	WDTA mode register	WDTAnMD	<WDTAn_base> + 000C <sub>H</sub>

### 20.3.2 WDTAnWDTE — WDTA Enable Register

This register is the WDTA trigger register when the VAC function is not used (start-up option OPWDVAC = 0).

Writing AC<sub>H</sub> to this register generates a WDTA trigger and starts or restarts the WDTA counter. See **Section 20.5.2, WDTA Trigger**, for details.

The behavior of this register depends on the setting of the start-up option (OPWDVAC), see **Table 20.12, WDTAnWDTE Behavior**.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

**Access:** This register can be read or written in 8-bit units.

**Address:** <WDTAn\_base> + 0000<sub>H</sub>

**Value after reset:** The initial value depends on the start-up options (OPWDEN, OPWDRUN and OPWDVAC). See **Table 20.11, Values of WDTAnRUN7 after Reset**.

Bit	7	6	5	4	3	2	1	0
WDTAnRUN[7:0]								
Value after reset	0/1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 20.10 WDTAnWDTE Register Contents**

Bit Position	Bit Name	Function
7 to 0	WDTAnRUN [7:0]	Writing the fixed activation code (AC <sub>H</sub> ) generates the WDTA trigger and starts/restarts the WDTAn counter. Writing a value other than AC <sub>H</sub> generates an error. The WDTAn cannot be stopped once it is started. See <b>Table 20.12, WDTAnWDTE Behavior</b> , when reading from or writing to these bits.

The WDTAnRUN7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is disabled (OPWDVAC = 0). **Table 20.11** lists the values of the WDTAnRUN7 bit after reset according to the start-up options.

**Table 20.11 Values of WDTAnRUN7 after Reset**

Start-Up Options			Start Mode	Value of WDTAnRUN7 after Reset
OPWDEN	OPWDVAC	OPWDRUN		
1	0	1	Default start	1
		0	Software trigger start	0

The behavior of WDTAnWDTE during read/write accesses depends on the OPWDVAC setting, as shown in **Table 20.12, WDTAnWDTE Behavior**.

Table 20.12 WDTAnWDTE Behavior

OPWDVAC	Description	WDTAnWDTE	
		Read	Write
0	The VAC function is disabled. WDTAnWDTE is enabled.	$2C_H$ is returned (in software trigger start mode, before the activation of WDTAn). $AC_H$ is returned (after the activation of WDTAn).	WDTA trigger Write $AC_H^{>1}$ .
1	The VAC function is enabled. WDTAnWDTE is disabled.	$2C_H$ is returned.	Writing is ignored.

Note 1. Any other write value will cause an error.

### 20.3.3 WDTAnEVAC — WDTA Enable VAC Register

This register is the WDTA trigger register when the VAC function is used (start-up option OPWDVAC = 1).

Writing a correct activation code to this register generates a WDTA trigger and starts or restarts the WDTA counter. For details, see **Section 20.5.2, WDTA Trigger**. For details about the activation codes when the VAC function is used, see **Section 20.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

The behavior of this register depends on the setting of the start-up option (OPWDVAC). See **Table 20.15, WDTAnEVAC Behavior**.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

**Access:** This register can be read or written in 8-bit units.

**Address:** <WDTAn\_base> + 0004<sub>H</sub>

**Value after reset:** The initial value depends on the start-up options (OPWDEN, OPWDRUN and OPWDVAC). See **Table 20.14, Values of WDTAnEVAC7 after Reset**.

Bit	7	6	5	4	3	2	1	0
WDTAnEVAC[7:0]								
Value after reset	0/1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 20.13 WDTAnEVAC Register Contents**

Bit Position	Bit Name	Function
7 to 0	WDTAnEVAC [7:0]	Writing a variable activation code generates the WDTA trigger and starts/restarts the WDTAn counter. Writing an incorrect activation code generates an error. The WDTAn cannot be stopped once it is started. See <b>Table 20.15, WDTAnEVAC Behavior</b> , when reading from or writing to these bits.

The WDTAnEVAC7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is enabled (OPWDVAC = 1). **Table 20.14** lists the values of the WDTAnEVAC7 bit after reset according to the start-up options.

**Table 20.14 Values of WDTAnEVAC7 after Reset**

Start-Up Options			Start Mode	Value of WDTAnEVAC7 after Reset
OPWDEN	OPWDVAC	OPWDRUN		
1	1	1	Default start	1
		0	Software trigger start	0

The behavior of WDTAnEVAC during read/write accesses depends on the OPWDVAC setting, as shown in **Table 20.15**.

**Table 20.15 WDTAnEVAC Behavior**

OPWDVAC	Description	WDTAnEVAC	
		Read	Write
0	The VAC function is disabled. WDTAnEVAC is disabled.	$2C_H$ is returned.	Writing is ignored.
1	The VAC function is enabled. WDTAnEVAC is enabled.	$2C_H$ is returned (in software trigger start mode, before the activation of WDTAn). The variable activation code written last is read (after the activation of WDTAn).	Write the variable activation code* <sup>1</sup> For details, see <b>Section 20.5.2.1, Calculating an Activation Code when the VAC Function is Used</b> .

Note 1. Any other write value will cause an error.

#### 20.3.4 WDTAnREF — WDTA Reference Value Register

This register contains the reference value for calculating the activation code of the VAC function. It is automatically updated after every trigger operation. See **Section 20.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

If the VAC function is disabled (OPWDVAC = 0), reading this register returns  $00_H$ .

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <WDTAn\_base> +  $0008_H$

**Value after reset:**  $00_H$

Bit	7	6	5	4	3	2	1	0
WDTAnREF[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 20.16 WDTAnREF Register Contents**

Bit Position	Bit Name	Function
7 to 0	WDTAnREF[7:0]	Reference value for activation code calculation for the VAC function

### 20.3.5 WDTAnMD — WDTA Mode Register

This register specifies the overflow interval time, 75% interrupt enable/disable, the error mode, and the window-open period.

The value of this register can be updated only once after reset release and before the first trigger is generated. The updated value will be effective after the WDTA trigger register is written to next.

Updating this register after the first WDTA trigger is generated generates an error, but an error does not occur if the same value has been written to it.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

**Access:** This register can be read or written in 8-bit units.

**Address:** <WDTAn\_base> + 000C<sub>H</sub>

**Value after reset:** The initial value depends on the start-up options (OPWDVOF[2:0]). See **Table 20.8, WDTA Start-Up Options**.

Bit	7	6	5	4	3	2	1	0
	—	WDTAnOVF[2:0]			WDTAnWIE	WDTAnERM	WDTAnWS[1:0]	
Value after reset	0	*1	*1	*1	0	1	1	1

R/W      R      R/W      R/W      R/W      R/W      R/W      R/W      R/W

Note 1. The WDTAnOVF[2:0] value after reset can be set by the start-up options OPWDVOF[2:0].

**Table 20.17 WDTAnMD Register Contents**

Bit Position	Bit Name	Function																																				
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																				
6 to 4	WDTAnOVF[2:0]	Selects the overflow interval time																																				
		<table border="1"> <thead> <tr> <th>WDTAn OVF2</th> <th>WDTAn OVF1</th> <th>WDTAn OVF0</th> <th>Overflow Interval Time</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td><math>2^9 / \text{WDTATCKI}</math></td></tr> <tr><td>0</td><td>0</td><td>1</td><td><math>2^{10} / \text{WDTATCKI}</math></td></tr> <tr><td>0</td><td>1</td><td>0</td><td><math>2^{11} / \text{WDTATCKI}</math></td></tr> <tr><td>0</td><td>1</td><td>1</td><td><math>2^{12} / \text{WDTATCKI}</math></td></tr> <tr><td>1</td><td>0</td><td>0</td><td><math>2^{13} / \text{WDTATCKI}</math></td></tr> <tr><td>1</td><td>0</td><td>1</td><td><math>2^{14} / \text{WDTATCKI}</math></td></tr> <tr><td>1</td><td>1</td><td>0</td><td><math>2^{15} / \text{WDTATCKI}</math></td></tr> <tr><td>1</td><td>1</td><td>1</td><td><math>2^{16} / \text{WDTATCKI}</math></td></tr> </tbody> </table>	WDTAn OVF2	WDTAn OVF1	WDTAn OVF0	Overflow Interval Time	0	0	0	$2^9 / \text{WDTATCKI}$	0	0	1	$2^{10} / \text{WDTATCKI}$	0	1	0	$2^{11} / \text{WDTATCKI}$	0	1	1	$2^{12} / \text{WDTATCKI}$	1	0	0	$2^{13} / \text{WDTATCKI}$	1	0	1	$2^{14} / \text{WDTATCKI}$	1	1	0	$2^{15} / \text{WDTATCKI}$	1	1	1	$2^{16} / \text{WDTATCKI}$
WDTAn OVF2	WDTAn OVF1	WDTAn OVF0	Overflow Interval Time																																			
0	0	0	$2^9 / \text{WDTATCKI}$																																			
0	0	1	$2^{10} / \text{WDTATCKI}$																																			
0	1	0	$2^{11} / \text{WDTATCKI}$																																			
0	1	1	$2^{12} / \text{WDTATCKI}$																																			
1	0	0	$2^{13} / \text{WDTATCKI}$																																			
1	0	1	$2^{14} / \text{WDTATCKI}$																																			
1	1	0	$2^{15} / \text{WDTATCKI}$																																			
1	1	1	$2^{16} / \text{WDTATCKI}$																																			
3	WDTAnWIE	Enables/disables the 75% interrupt request INTWDTAn. 0: INTWDTAn is disabled. 1: INTWDTAn is enabled.																																				
2	WDTAnERM	Specifies the error mode. 0: NMI request mode 1: Reset mode																																				
1, 0	WDTAnWS[1:0]	Selects the window-open period.																																				
		<table border="1"> <thead> <tr> <th>WDTAn WS1</th> <th>WDTAn WS0</th> <th>Window-Open Period</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>25%</td></tr> <tr><td>0</td><td>1</td><td>50%</td></tr> <tr><td>1</td><td>0</td><td>75%</td></tr> <tr><td>1</td><td>1</td><td>100%</td></tr> </tbody> </table>	WDTAn WS1	WDTAn WS0	Window-Open Period	0	0	25%	0	1	50%	1	0	75%	1	1	100%																					
WDTAn WS1	WDTAn WS0	Window-Open Period																																				
0	0	25%																																				
0	1	50%																																				
1	0	75%																																				
1	1	100%																																				

## 20.4 Interrupt Sources

WDTA checks the status of the WDTA counter value, detects illegal accesses to the WDTA-related registers, and generates an interrupt request. The following are WDTA interrupt requests:

- (1) INTWDTAn (WDTA timer count 75% interrupt request)  
An interrupt request signal is generated at 75% of the counter overflow time of the WDTA timer.  
An interrupt request signal can be enabled or disabled by using the WDTA mode register (WDTAnMD).
- (2) WDTAnTNMI (WDTA error detection interrupt)  
Detection of a WDTA error to generation of an NMI interrupt request. The WDTA mode register (WDTAnMD) can be used to switch between an NMI interrupt and a reset. For details about WDTA errors, see **Section 20.5.3, WDTA Error Detection**.

## 20.5 Functions

### 20.5.1 WDTA after Reset Release

#### 20.5.1.1 Start Modes

There are two start modes (software start mode and default start mode) when WDTAn starts after reset release. The start mode can be selected by the start-up option.

The start mode selection is listed in **Table 20.18**.

**Table 20.18 Start Modes**

Start-Up Option OPWDRUN	Start Mode	Description
0	Software trigger	<ul style="list-style-type: none"> <li>The WDTA counter stops (0000<sub>H</sub>) after reset release.</li> <li>Writing an activation code to the WDTA trigger register starts WDTA.</li> </ul>
1	Default	The WDTA counter starts after reset release.

#### 20.5.1.2 WDTA Settings after Reset Release

(1) **Table 20.19** shows the WDTA settings after reset release.

**Table 20.19 WDTA Settings after Reset Release**

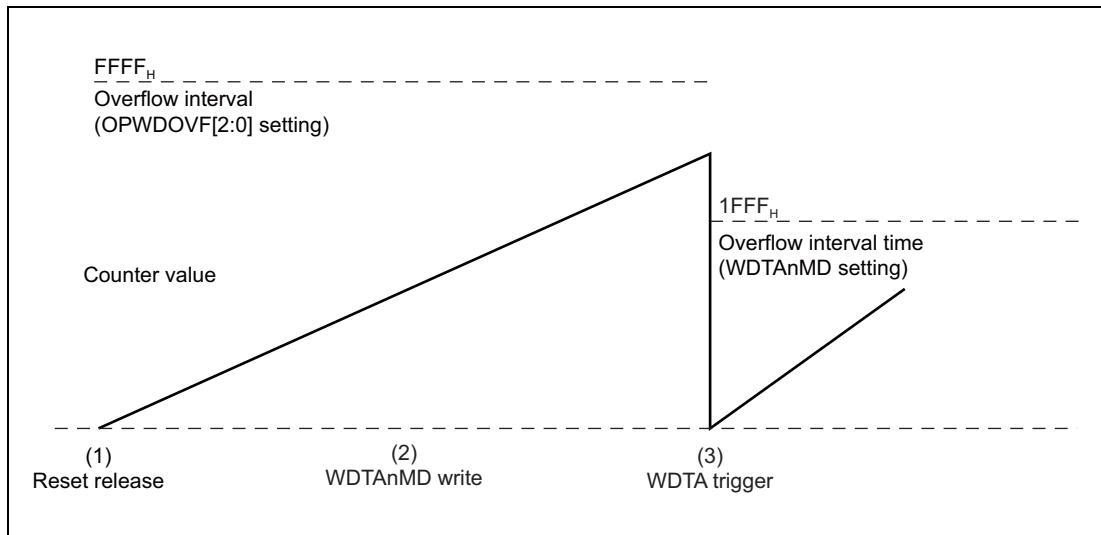
Function	Setting	Remark
WDTA enable/disable	Specified by start-up options	
Start mode		
VAC function		
WDTA overflow interval time	Specified by start-up options	Modification is possible only once by the setting of the WDTA mode register (WDTAnMD).
75% interrupt mode	75% interrupt disabled	
Behavior on error detection	Reset generation	
Window-open period	100%	

The setting of the WDTA mode register (WDTAnMD) is enabled when the first WDTA trigger is generated (writing an activation code to WDTAnWDTE and WDTAnEVAC). Perform the WDTAnMD register setting before a WDTA trigger is generated.

Setting of WDTA by using WDTAnMD is possible only once. If the value set to WDTAnMD is changed after a WDTA trigger is generated, an error occurs. However, an error does not occur if the same value is set.

### 20.5.1.3 Default Start Mode Timing

The default start mode timing and the changes to the WDTA settings are illustrated in **Figure 20.2**.



**Figure 20.2** Timing Diagram of WDTA Start in Default Start Mode

The timing diagram shown in **Figure 20.2** shows the following behaviors:

- (1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

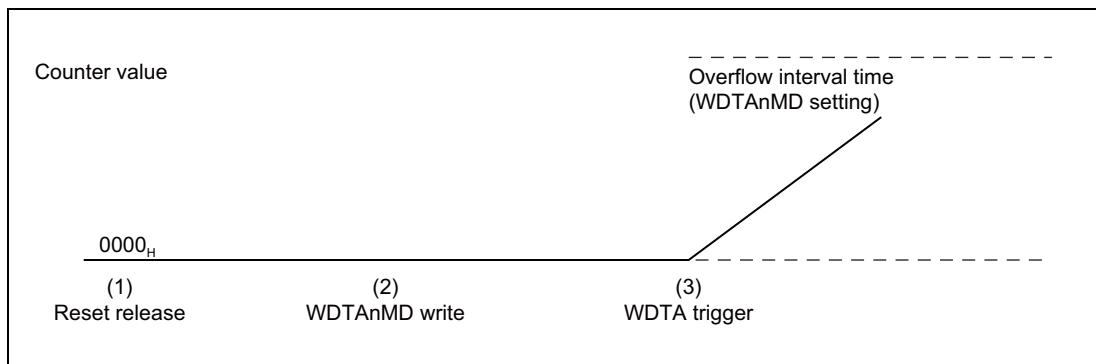
Example: Overflow interval time after reset release  
 $= 2^{16}/\text{WDTATCKI}$  ( $\text{OPWDOVF}[2:0] = 111_B$ )

- (2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.
- (3) Write to the WDTA trigger register before the WDTA counter overflows.  
 The WDTAnMD setting is applied by the generation of a WDTA trigger.

Example: Overflow interval time after a WDTA trigger is generated  
 $= 2^{13}/\text{WDTATCKI}$

#### 20.5.1.4 Software Trigger Start Mode Timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in **Figure 20.3**.



**Figure 20.3 Timing Diagram of WDTA Start in Software Trigger Start Mode**

The timing diagram shown in **Figure 20.3** shows the following behaviors:

- (1) The WDTA counter value remains 0000<sub>H</sub> until the first trigger is generated after reset release. The overflow interval time is set by start-up options, but this setting has no effect because the counter is not operating.
- (2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.
- (3) The WDTA counter starts by the generation of a WDTA trigger. The overflow interval time specified in WDTAnMD and other settings are applied.

## 20.5.2 WDTA Trigger

Writing a special value called an activation code to the WDTA enable register (WDTAnWDTE) and the WDTA enable VAC register (WDTAnEVAC) leads to generation of a WDTA trigger.

The WDTA trigger has the following functions:

- Starting the WDTA counter in software trigger start mode
- Restarting the WDTA counter
- Setting the WDTA mode specified by the WDTAnMD register (only for the first WDTA trigger after reset release)

The WDTA trigger register, which generates a WDTA trigger, is specified by the start-up option OPWDVAC.

**Table 20.20** lists the WDTA trigger registers and activation codes.

**Table 20.20 WDTA Trigger Registers and Activation Codes**

Type of Activation Code	Trigger Register	Activation Code
Fixed (OPWDVAC = 0)	WDTAnWDTE	AC <sub>H</sub>
Variable (OPWDVAC = 1)	WDTAnEVAC	For details, see <b>20.5.2.1, Calculating an Activation Code when the VAC Function is Used.</b>

### CAUTION

- When writing the processing to clear WDTA successively, the clear processing below is not acknowledged for the following period:  
“12 × CPU clock<sup>\*1</sup> cycles + 6 × WDT clock<sup>\*2</sup> cycles”
- After writing the processing to clear WDTA and then changing to standby mode, the clear processing below is not acknowledged for the following period after return from stand-by mode:  
“6 × CPU clock<sup>\*1</sup> cycles + 3 × WDT clock<sup>\*2</sup> cycles”

Note 1. CPU clock: Clock selected by CKSC\_CPUCLKS\_CTL and CKSC\_CPUCLKD\_CTL

Note 2. WDT clock: Clock selected by CKSC\_AWDTAD\_CTL in WDTA0.

LS IntOSC in WDTA1.

### 20.5.2.1 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (ExpectWDTE) to be set in the WDTA trigger register (WDTAnEVAC) when the VAC function is used, by using the WDTA reference value register (WDTAnREF):

$$\text{ExpectWDTE} = \text{AC}_H - \text{WDTAnREF} \text{ (previous)}$$

Note that the value in the WDTAnREF register is updated every time a start-code is written to the trigger register WDTAnEVAC. Use the following expression to calculate the updated value of the WDTAnREF register:

$$\text{WDTAnREF (following)} = (\text{rotate the value of ExpectWDTE to the left by 1 bit})$$

**Table 20.21** lists the variable activation codes according to the number of WDTA triggers.

**Table 20.21 Expected Variable Activation Code Development**

No <sup>*1</sup>	WDTAnREF (Previous)		ExpectWDT <sub>E</sub> (AC <sub>H</sub> - WDTAnREF)		WDTAnREF (Following)	
0	0000 0000	00 <sub>H</sub>	1010 1100	AC <sub>H</sub>	0101 1001	59 <sub>H</sub>
1	0101 1001	59 <sub>H</sub>	0101 0011	53 <sub>H</sub>	1010 0110	A6 <sub>H</sub>
2	1010 0110	A6 <sub>H</sub>	0000 0110	06 <sub>H</sub>	0000 1100	0C <sub>H</sub>
...	...	...	...	...	...	...

Note 1. Number of triggers after reset

#### NOTE

Writing an incorrect activation code generates an error.

### 20.5.3 WDTA Error Detection

WDTA detects an error, including generation of the WDTA count overflow or illegal operations.

The following events are detected as errors:

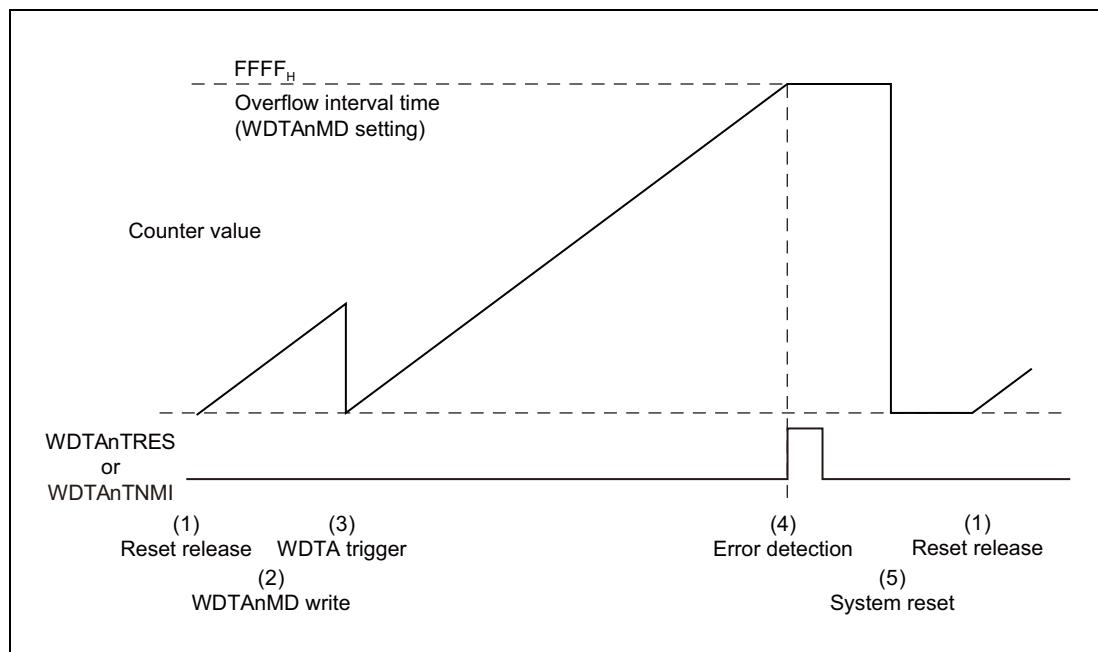
- WDTA counter overflow
- Incorrect activation code is written to the WDTA trigger register
- Writing to the trigger register at a time outside the window-open period
- When the setting value in the WDTA mode register (WDTAnMD) is changed after the first WDTA trigger is generated
- When the value of the WDTA mode register (WDTAnMD) is changed twice before the first WDTA trigger is generated

#### 20.5.3.1 WDTA Error Mode

When a WDTA error is detected, either an NMI interrupt or a reset is generated according to the setting of the WDTA error mode bit (WDTAnMD.WDTAnERM). The error mode bit after reset release is set to the reset mode.

- WDTAnMD.WDTAnERM = 0: NMI mode
- WDTAnMD.WDTAnERM = 1: Reset mode

**Figure 20.4** shows the reset or NMI request generation when the counter overflows and default start mode is selected.



**Figure 20.4 Timing Diagram of WDTA NMI Request or Reset Generation**

The timing diagram shown in **Figure 20.4** shows the following behaviors:

- (1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.
- (2) WDTAnMD is set before a WDTA trigger is generated.  
In this case,  $2^{16}/\text{WDTATCKI}$  is set for the overflow interval time.
- (3) A WDTA trigger is generated, and the WDTAnMD setting is applied.
- (4) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated.  
The counter value remains until a system reset is performed.
- (5) When the system is reset, the counter is cleared and stops until reset release.

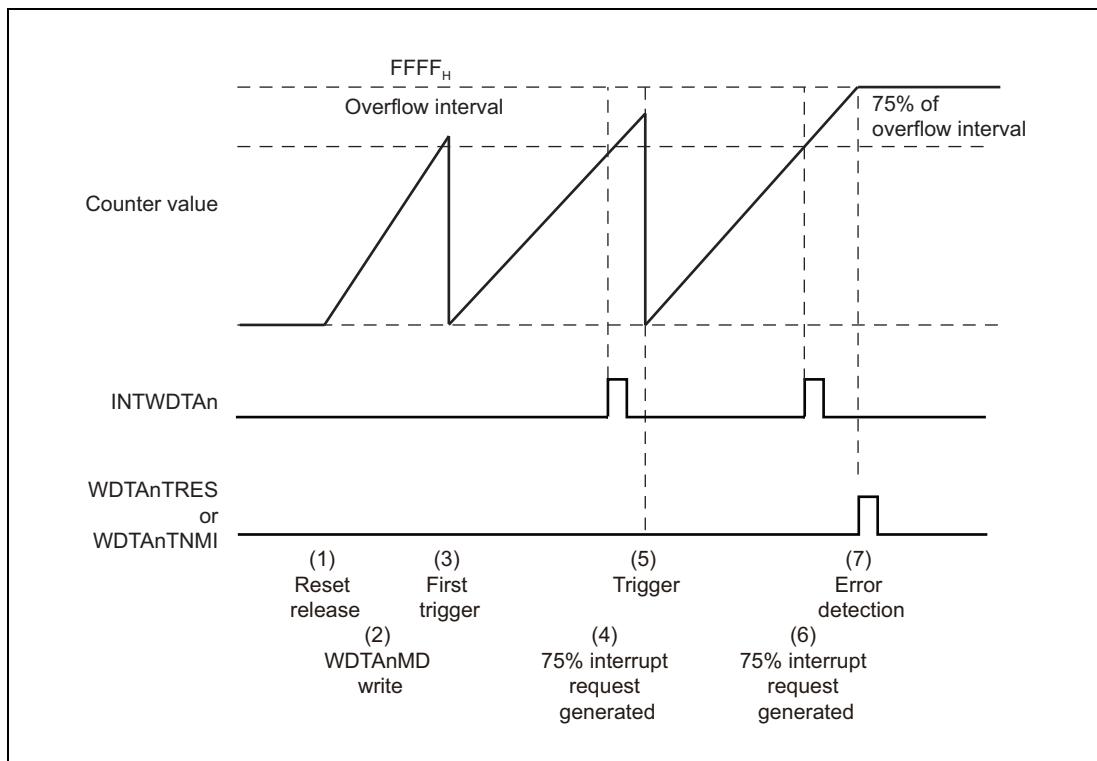
### 20.5.4 75% Interrupt Request Signals

When the WDTA counter reaches 75% of the time set for the overflow interval, the interrupt request INTWDTAn is generated.

This function can be enabled or disabled by the WDTAnMD.WDTAnWIE register.

**Figure 20.5** shows the 75% interrupt request generated under following conditions:

- Default start mode is selected.
- 75% interrupt request is enabled after the first WDTA trigger is generated
- WDTA overflow interval time is  $2^{16}/\text{WDTATCKI}$



**Figure 20.5 Timing Diagram of WDTA 75% Interrupt Request Signals**

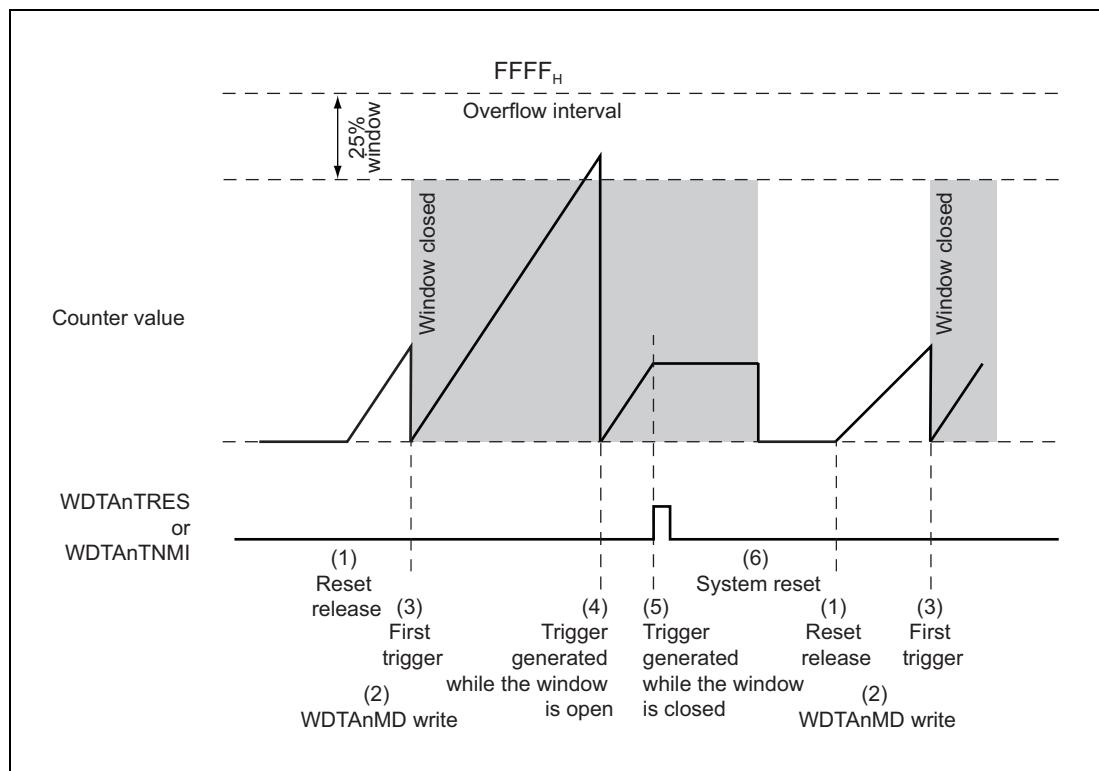
- (1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.
- (2) WDTAnMD is set before a WDTA trigger is generated. In this case,  $2^{16}/\text{WDTATCKI}$  is set for the overflow interval time.
- (3) A WDTA trigger is generated, and the WDTAnMD setting is applied.
- (4) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.
- (5) The WDTA trigger restarts counting.
- (6) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.
- (7) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains until a system reset is performed.

### 20.5.5 Window Function

The period when a WDTA trigger is valid (window-open period) can be set. If the window-open period is set to the value less than 100%, the generation of a WDTA trigger not in the window-open period causes an error. The window-open period after reset release is 100%. The period is set to the value by the WDTAnMD.WDTAnWS[1:0] setting after the first WDTA trigger is generated.

**Figure 20.6** shows the behavior of the window function under the following conditions.

- Default start mode is selected.
- 25% window-open period is enabled after the first WDTA trigger is generated (WDTAnWS[1:0] = 00<sub>B</sub>)
- WDTA overflow interval time is  $2^{16}/\text{WDTATCKI}$



**Figure 20.6 Timing Diagram of WDTA Window Function**

- (1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.
- (2) WDTAnMD is set before a WDTA trigger is generated. In this case,  $2^{16}/\text{WDTATCKI}$  is set for the overflow interval time.
- (3) A WDTA trigger is generated, and the WDTAnMD setting is applied.
- (4) During the window-open period, the WDTA trigger restarts counting.
- (5) During the window-closed period, an error is detected by the WDTA trigger. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains until a system reset is performed.
- (6) When the system is reset, the counter is cleared and stops until reset release.

## Section 21 OS Timer (OSTM)

This section contains a generic description of the OS Timer (OSTM).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the OSTM.

### 21.1 Features of RH850/F1L OSTM

#### 21.1.1 Number of Units

This microcontroller has the following number of units of the OSTM.

**Table 21.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	OSTMn (n=0)					

**Table 21.2 Index**

Index	Description
n	Throughout this section, the individual OSTM units are identified by the index "n"; for example, <u>OSTMnCNT</u> is the OSTM counter register.

#### 21.1.2 Register Base Address

OSTM base address is listed in the following table.

OSTM register addresses are given as offsets from the base address.

**Table 21.3 Register Base Address**

Base Address Name	Base Address
<OSTM0_base>	FFEC 0000 <sub>H</sub>

### 21.1.3 Clock Supply

The OSTM clock supply is shown in the following table.

**Table 21.4 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
OSTM0	PCLK	CPUCLK2
	Register access clock	CPUCLK2

### 21.1.4 Interrupt Request

OSTM interrupt requests are listed in the following table.

**Table 21.5 Interrupt Requests EI INT**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>OSTM0</b>			
OSTMTINT	OSTM0 interrupt	76	—

**Table 21.6 Interrupt Request (FE Level Maskable Interrupt Request)**

Unit Interrupt Signal	Outline	Interrupt Name	DMA Trigger Number
<b>OSTM0</b>			
OSTMTINT	OSTM0 interrupt	INTOSTM0_FE	—

### 21.1.5 Reset Sources

OSTM reset sources are listed in the following table. OSTM is initialized by these reset sources.

**Table 21.7 Reset Sources**

Unit Name	Reset Source
OSTM0	All reset sources (ISORES)

## 21.2 Overview

OSTM is a 32-bit timer/counter.

It can be used in interval timer mode or in free-run compare mode. The settings for operating mode specify the direction of counting (up or down) to control the generation of interrupt requests.

### 21.2.1 Functional Overview

OSTM has the following features.

- Two operating modes **两种操作模式**
  - Interval timer mode
  - Free-run compare mode
- OSTMTINT interrupt

### 21.2.2 Block Diagram

The following block diagram shows the main components of OSTM.

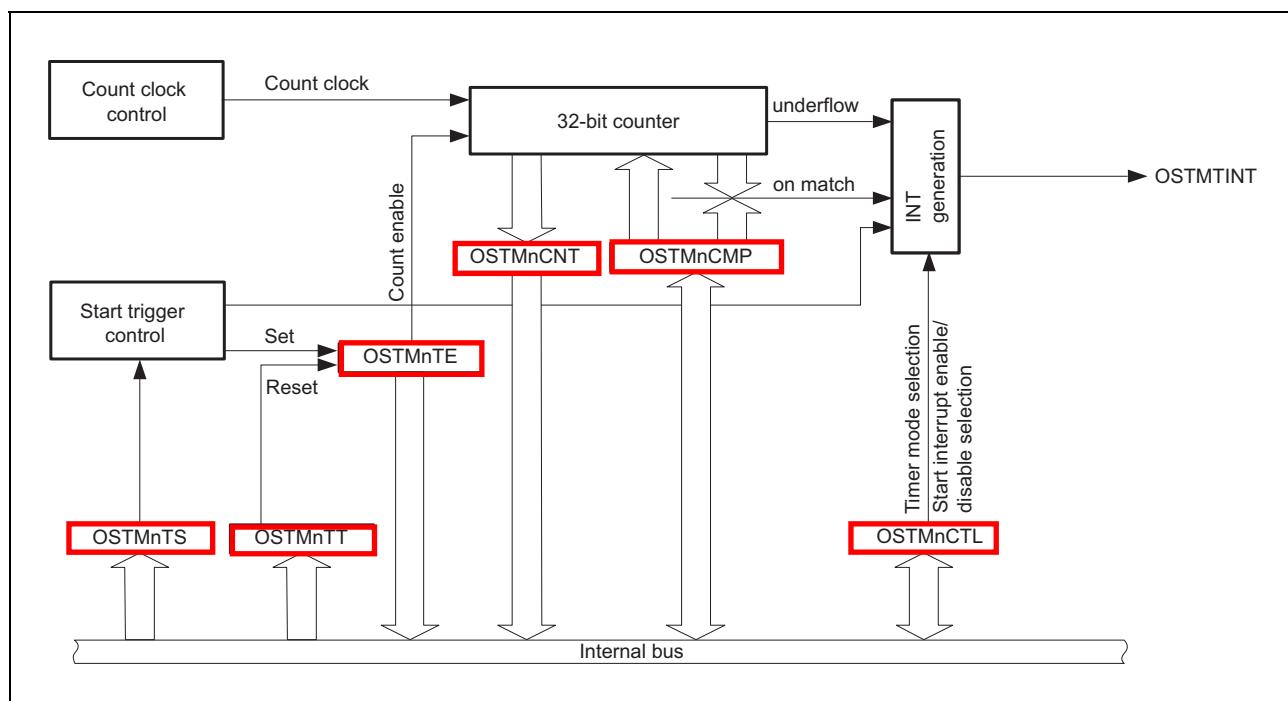


Figure 21.1 Block Diagram of OSTM

### 21.2.3 Count Clock

The count clock used by OSTM is PCLK.

### 21.2.4 Interrupt Sources (OSTMTINT)

#### 1. OSTMINT interrupt request产生的条件

An OSTMINT interrupt request is generated on counter underflow (interval timer mode) or when the counter matches the compare value (free-run compare mode).

**启动或者重启计数器也会产生interrupt请求，这由OSTMnCTL.OSTMnMD0 bit控制**  
An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

This is illustrated in the following figure.

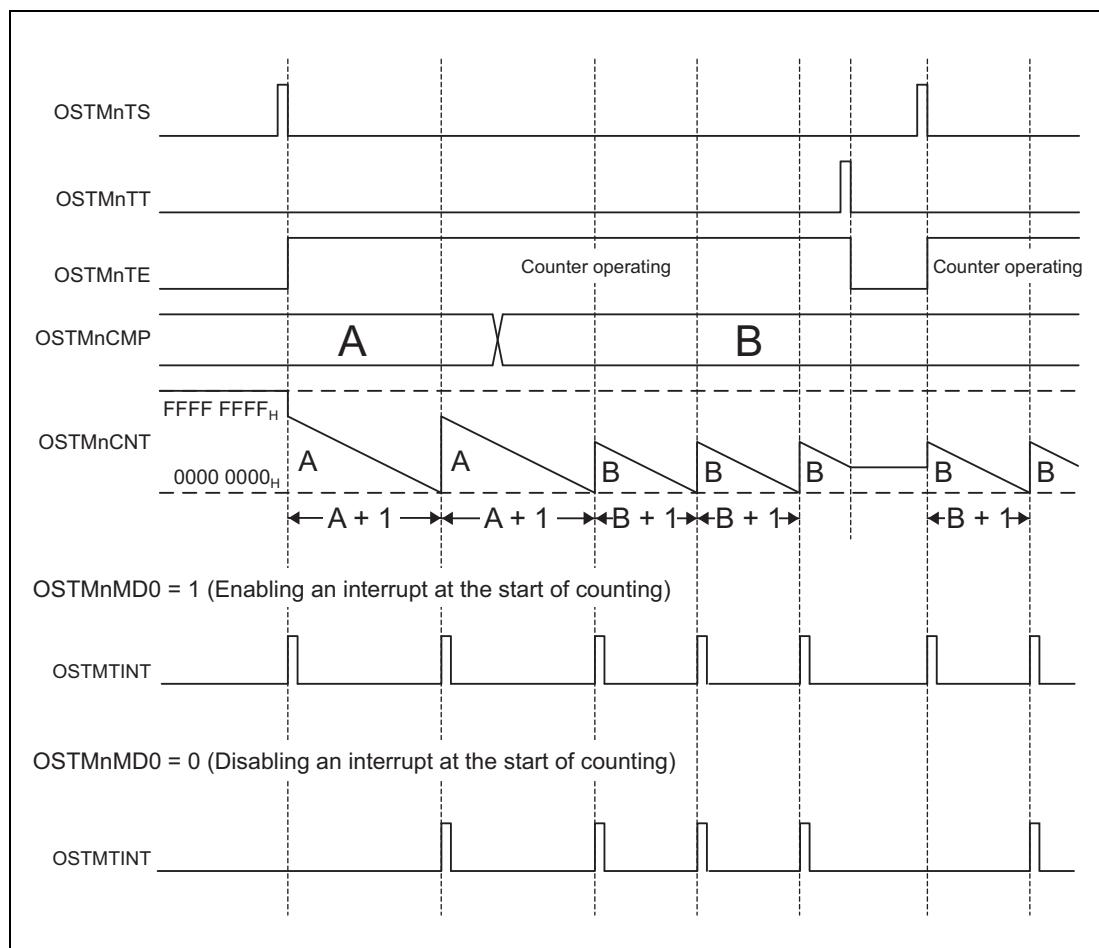


Figure 21.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

## 21.3 Registers

### 21.3.1 List of Registers

OSTM registers are listed in the following table.

For details about <OSTMn\_base>, see **Section 21.1.2, Register Base Address**.

Table 21.8 Registers

Module	Register	Symbol	Address
OSTMn	OSTMn compare register	OSTMnCMP	<OSTMn_base> + 00 <sub>H</sub>
OSTMn	OSTMn counter register	OSTMnCNT	<OSTMn_base> + 04 <sub>H</sub>
OSTMn	OSTMn count enable status register	OSTMnTE	<OSTMn_base> + 10 <sub>H</sub>
OSTMn	OSTMn count start trigger register	OSTMnTS	<OSTMn_base> + 14 <sub>H</sub>
OSTMn	OSTMn count stop trigger register	OSTMnTT	<OSTMn_base> + 18 <sub>H</sub>
OSTMn	OSTMn control register	OSTMnCTL	<OSTMn_base> + 20 <sub>H</sub>
OSTMn	OSTMn emulation register	OSTMnEMU	<OSTMn_base> + 24 <sub>H</sub>

### 21.3.2 OSTMN<sub>n</sub>CMP — OSTMN<sub>n</sub> Compare Register

此寄存器保存了在interval timer mode下down-counter的起始值，或者在比较模式下计数器要比较的值

This register stores the start value of the down-counter or the value with which the counter is compared, depending on the operation mode.

**Access:** This register can be read or written in 32-bit units.

**Address:** <OSTMN<sub>n</sub>\_base> + 00H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTMN <sub>n</sub> CMP[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTMN <sub>n</sub> CMP[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.9 OSTMN<sub>n</sub>CMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMN <sub>n</sub> CMP [31:0]	<ul style="list-style-type: none"> <li>In interval timer mode: start value of the down-counter</li> <li>In free-run compare mode: compare value</li> </ul>

### 21.3.3 OSTMN<sub>n</sub>CNT — OSTMN<sub>n</sub> Counter Register

**此寄存器指示了系统时钟的计数值，只读**

This register indicates the counter value of the timer.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <OSTMN<sub>n</sub>\_base> + 04H

**Value after reset:** FFFF FFFF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
OSTMN <sub>n</sub> CNT[31:16]																
OSTMN <sub>n</sub> CNT[15:0]																

**Table 21.10 OSTMN<sub>n</sub>CNT Register Contents**

Bit Position	Bit Name	Function
31 to 0	OSTMN <sub>n</sub> CNT [31:0]	Timer counter value

**Table 21.11** lists the correspondence among the OSTM operating mode, counting direction, and start value. The start value indicates the value to be read after the operating mode is changed.

**Table 21.11 Correspondence among Operating Mode, Counting Direction, and Start Value**

Timer Operating Mode	OSTMN <sub>n</sub> CTL.OSTMN <sub>n</sub> MD1	Counting Direction	Start Value
Interval timer mode	0 <sup>*1</sup>	Down	FFFF FFFF <sub>H</sub>
Free-run compare mode	1	Up	0000 0000 <sub>H</sub>

Note 1. Value after reset.

### 21.3.4 OSTMNTE — OSTMN Count Enable Status Register

此寄存器指示计数器是否使能或者禁止的状态，只读寄存器，注意这是状态寄存器

This register indicates whether the counter is enabled or disabled.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <OSTMN\_base> + 10H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMNTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.12 OSTMNTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OSTMNTE	Indicates whether the counter is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMNTS.OSTMNTS being set to 1. Setting OSTMNTT.OSTMNTT to 1 resets this bit to 0.
NOTE	如果OSTMNTE.OSTMNTE 被置为1，则此位标识为1 如果OSTMNTT.OSTMNTT 被置为1，则此位标志为0	

If the counter is disabled, the counter value retains its value.

If the counter is restarted, it

- restarts counting down from the value in the OSTMNCOMP register if it is in interval timer mode or
- restarts counting up from the counter value 0000 0000H if it is in free-run compare mode.

**如果计数器被禁止，则保持其当前的值**

**如果计数器被重新启动，在interval mode下，以OSTNMNCMP中的值重新下减，如果在free-run compare模式，由(uint32\_t)0x00开始计数**

**重要**

### 21.3.5 OSTMnTS — OSTMn Count Start Trigger Register

启动计时器，只能写

This register starts the counter.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <OSTMn\_base> + 14H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 21.13 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTS	<p>Starts the counter:</p> <p>0: This setting is invalid.</p> <p>1: Starts the counter and sets OSTMnTE.OSTMnTE = 1.</p> <ul style="list-style-type: none"> <li>In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. 在interval mode下，当OSTMnTE为1，当此bit被置1，会强制重启</li> <li>In free-run compare mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1. 在free-run 比较模式下，只要OSTMnTE为1，设置此位无效</li> </ul>

### 21.3.6 OSTMnTT — OSTMn Count Stop Trigger Register

停止计数器，只读

This register stops the counter.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <OSTMn\_base> + 18H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 21.14 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTT	<p>Stops the counter:</p> <p>0: This setting is invalid.</p> <p>1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.</p> <p style="color: red;">停止计数器的计数并且清除OSTMnTE位</p>

### 21.3.7 OSTMNCTL — OSTMN Control Register

**此寄存器指定计数器的工作模式，并且控制在计数开始时候是否产生OSTMTINT中断请求**

This register specifies the operating mode for the counter and controls the generation of OSTMTINT interrupt requests when counting starts.

Although this register is readable and writable, writing to it is only possible when OSTMNTE.OSTMNTE = 0; that is, the register becomes read only when OSTMNTE.OSTMNTE = 1.

**Access:** This register can be read or written in 8-bit units.

**Address:** <OSTMN\_base> + 20H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OSTMNMD1	OSTMNMD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 21.15 OSTMNCTL Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OSTMNMD1	Specifies the operating mode for the counter: 0: Interval timer mode 1: Free-run compare mode
0	OSTMNMD0	Controls the generation of OSTMTINT interrupt requests at the start of counting: 0: Interrupts when counting starts are disabled. 1: Interrupts when counting starts are enabled.

### 21.3.8 OSTMNEMU — OSTMN Emulation Register

This register controls operation in combination with SVSTOP.

**Access:** This register can be read or written in 8-bit units.  
Only proceed with writing while the counter is stopped (OSTMNTE.OSTMNTE = 0 and EPC.SVSTOP = 0).

**Address:** <OSTMN\_base> + 24H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	OSTMNsvsdis	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 21.16 OSTMNEMU Register Contents

Bit Position	Bit Name	Function
7	OSTMNsvsdis	When EPC.SVSTOP = 0 在调试器接管MCU的控制权(设置断点等)计数时钟会继续 Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on) regardless of the value of this bit (0/1). When EPC.SVSTOP = 1 在调试器接管MCU的控制权(设置断点等)计数时钟会停止 0: The count clock is stopped when the debugger acquires control of the microcontroller (at breakpoints and so on). 1: Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on). 在调试器接管MCU的控制权(设置断点等)计数时钟会继续
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 21.4 Operation

### 21.4.1 Starting and Stopping OSTM

OSTM is started and stopped as follows:

#### Starting the timer 启动时钟

OSTM is started by the following setting.

- Setting the OSTMnTS.OSTMnTS bit to 1

The OSTMnTE.OSTMnTE status bit is set to 1.

The counter starts to count up or down in accordance with the settings for operating mode. For details, see **Section 21.4.2, Interval Timer Mode** and **Section 21.4.3, Free-Run Compare Mode**.

#### Stopping the timer 停止时钟

Setting the OSTMnTT.OSTMnTT bit to 1 stops OSTM.

This also clears the OSTMnTE.OSTMnTE status bit.

### 21.4.2 Interval Timer Mode

In interval timer mode, OSTM can be used as a reference timer generating interrupt requests at fixed intervals. **在此模式下，OSTM作为一个靠时钟，在固定的间隔后产生一个中断请求**

#### 21.4.2.1 Basic Operation in Interval Timer Mode

**在此模式下，时钟根据OSTMnCMP寄存器的值进行下减，当计数器达到0x00的时候OSTMTINT产生中断请求**  
In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMTINT interrupt request is generated when the counter underflows (reaches 0000 0000<sub>H</sub>).

To select interval timer mode, set OSTMnCTL.OSTMnMD1 = 0.

**新的值可以随时写入OSTMnCMP寄存器，在计数器工作的时候重新写入计数值，那么计数器会在这次达到0x00之后使用新的计数值**  
New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next 0000 0000<sub>H</sub> is reached. Then the counter continues with the new value.

#### OSTMTINT period

The periods of OSTMTINT is:

- OSTMTINT generation period = count clock period × (OSTMnCMP + 1)

The following figure shows the basic operation of OSTM when counter-start interrupts is enabled in interval timer mode.

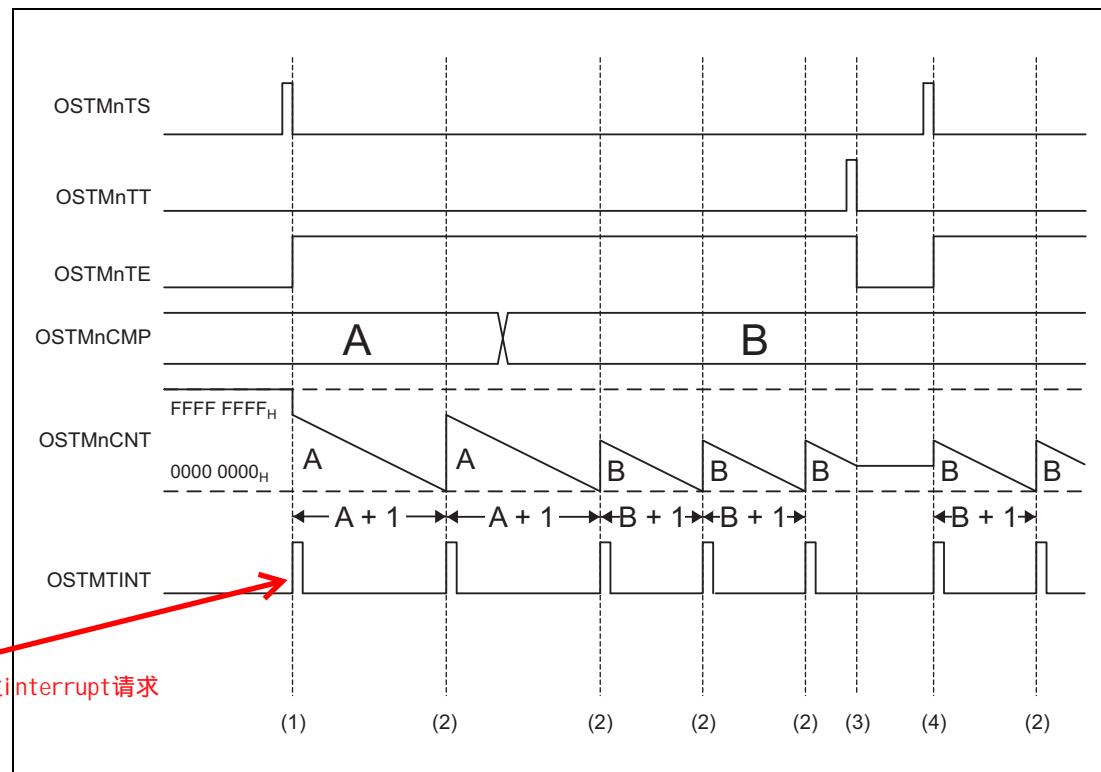


Figure 21.3 Timing Diagram of OSTM in Interval Timer Mode

The timing diagram above shows the following:

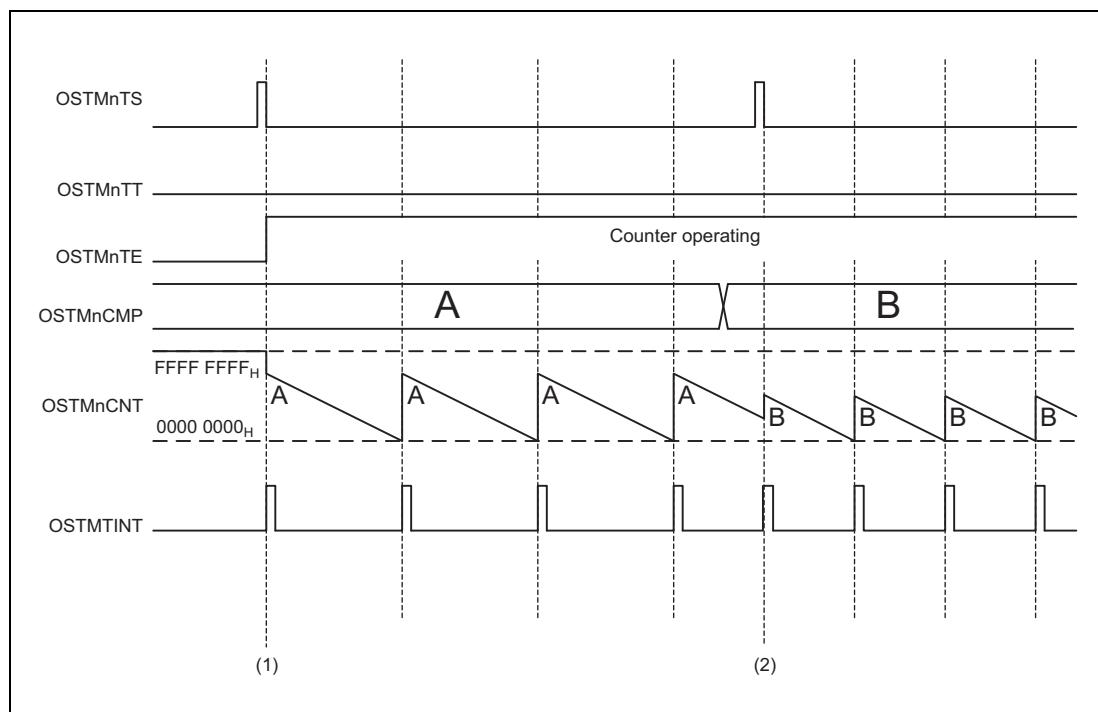
- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.  
The counter starts counting down from the value of OSTMnCMP.  
If OSTMnCTL.OSTMnMD0 is 1, OSTMTINT interrupt requests are generated at the start of counting. The OSTMnCNT register indicates the counter value.
- (2) When the counter reaches  $0000\ 0000_H$ , an OSTMTINT interrupt request is generated. The counter loads the new start value from OSTMnCMP and continues counting down.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.
- (4) When counting is restarted (OSTMnTS.OSTMnTS = 1), the counter loads the new start value from OSTMnCMP and starts counting down. **重启时重新载入计数值进行启动**

### Forced restart

The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 during counting.

The counter loads the start value from the OSTMnCMP register and continues to count down.

The following figure shows the timing in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1).



**Figure 21.4 Timing Diagram of Forced Restart in Interval Timer Mode**

The timing diagram above shows the following operations:

- (1) The counter is started and stopped as described under **Figure 21.3, Timing Diagram of OSTM in Interval Timer Mode.**
- (2) Setting OSTMnTS.OSTMnTS = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).

The counter immediately restarts counting down, starting with the current value of OSTMnCMP.

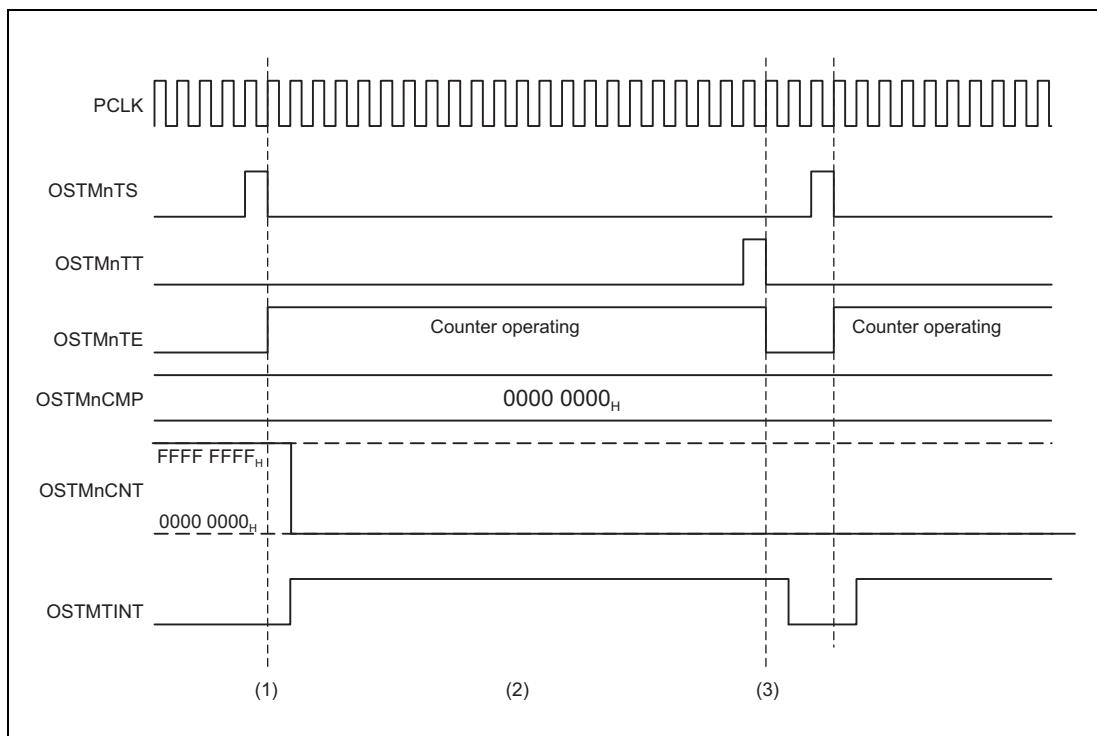
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated when counting starts.

### 21.4.2.2 Operation when OSTMnCMP = 0000 0000<sub>H</sub>

When OSTMnCMP = 0000 0000<sub>H</sub>, OSTM behaves as follows.

- When the counter is enabled, the OSTMTINT interrupt request is always set to 1.

The following figure shows operations of OSTM when OSTMnCMP = 0000 0000<sub>H</sub>, and counter-start interrupts are enabled.



**Figure 21.5 Timing Diagram when OSTMnCMP = 0000 0000<sub>H</sub> in Interval Timer Mode**

The timing diagram above shows the following operations:

- (1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000<sub>H</sub> is retained in OSTMnCMP.
- (2) The OSTMTINT interrupt request is continuously asserted. **中断请求持续产生**
- (3) After the counter stops, the OSTMTINT interrupt request signal is deasserted.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

### 21.4.2.3 Setting Procedure for Interval Timer Mode

The setting procedure in interval timer mode after reset release is described below:

#### Setting procedure

- (1) Set the start value of the counter in the OSTMnCMP register.
- (2) Select interval timer mode by setting the OSTMnCTL.OSTMnMD1 bit to 0.
- (3) Enable or disable interrupts when counting starts (OSTMnCTL.OSTMnMD0).

### 21.4.3 Free-Run Compare Mode

#### 21.4.3.1 Basic Operation in Free-Run Compare Mode

**在此模式下，计数器从0x00开始到0xFFFFFFFF**

In free-run compare mode, the counter counts up from  $0000\ 0000_H$  to  $FFFF\ FFFF_H$ . When the value of the OSTMnCMP register matches the current counter value, an OSTMTINT interrupt request is output.

In free-run compare mode, set OSTMnCTL.OSTMnMD1 = 1.

New values can be written to the OSTMnCMP register at any time. **新的计数值可以被随时写入OSTMnCMP**

The following figure shows the basic operation of OSTM in free-run compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1).

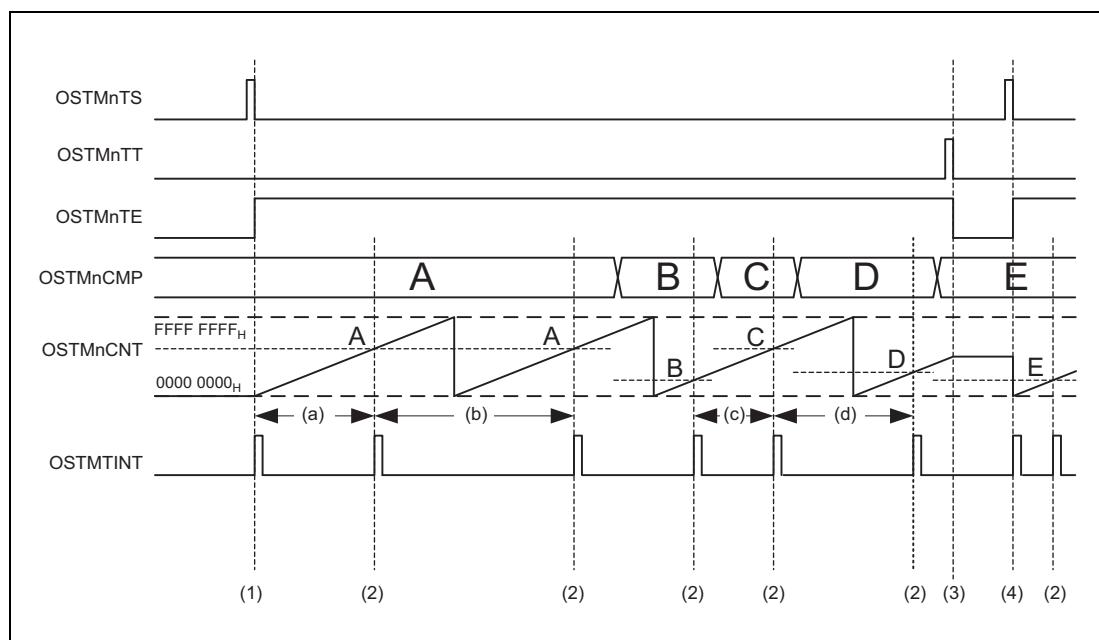


Figure 21.6 Timing Diagram of OSTM in Free-Run Compare Mode

The timing diagram above shows the following:

- (1) The counter starts counting when OSTMnTS.OSTMnTS = 1.  
The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from  $0000\ 0000_H$  to  $FFFF\ FFFF_H$ . The OSTMnCNT register indicates the counter value.  
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated at the start of counting.
- (2) **当当前的计数器与OSTMnCMP寄存器中的值相等时，中断请求就会产生**  
When the current counter value matches the value in the OSTMnCMP register, an OSTMTINT interrupt request is generated.
- (3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.  
**计数器会保持当前的数值直到重启**  
当重启以后，计数值从0x00开始
- (4) Counting by the counter restarts from  $0000\ 0000_H$  when OSTMnTS.OSTMnTS = 1.

### OSTMTINT period

The OSTMTINT generation period differs depending on when the counting starts and varies depending on the magnitude relation of the old and new comparison values if OSTMnCMP is rewritten during operation.

**Table 21.17 OSTMTINT Generation Timing**

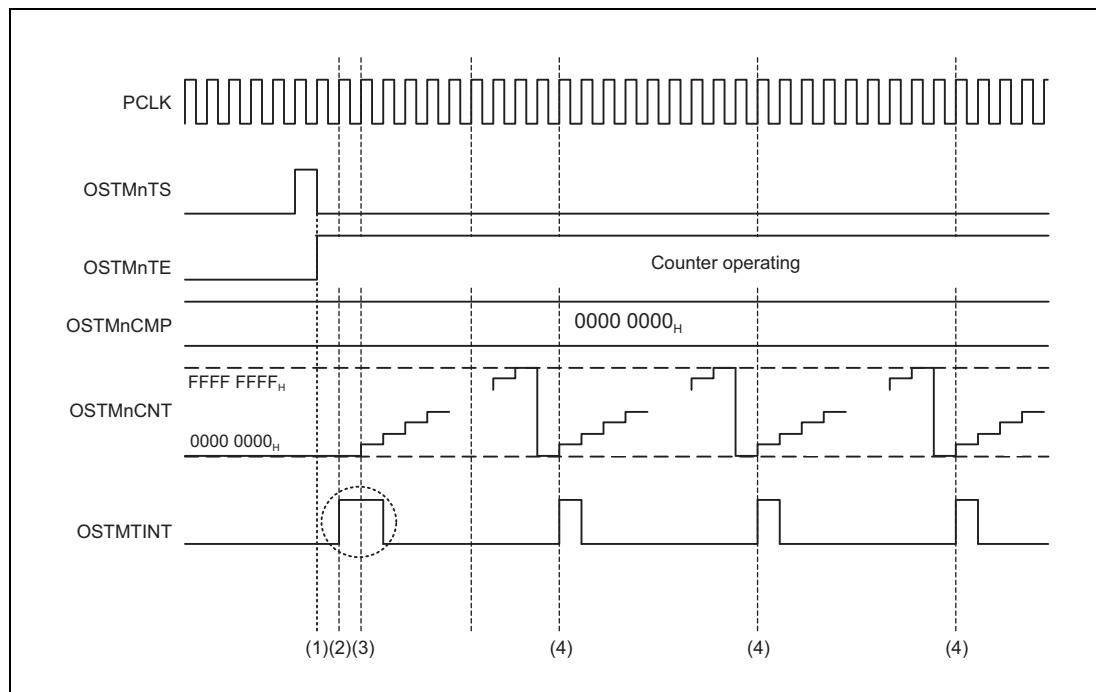
Old Value for Comparison	New Value for Comparison	Counter Value at Time of Rewriting	Period of OSTMTINT Generation	Label in Timing Diagram
Counter starts			(A + 1) × count clock period	(a)
A	A	No rewriting	(FFFF FFFF <sub>H</sub> + 1) × count clock period	(b)
B	C > B	B < counter value < C	(C - B) × count clock period	(c)
C	D < C	Counter value > D,C	(FFFF FFFF <sub>H</sub> - C + D + 1) × count clock period	(d)

### Forced restart

Forced restarting of the counter does not proceed if the OSTMnTS.OSTMnTS bit is set during counting. The counter ignores the attempted setting and continues counting.

#### 21.4.3.2 Operation when OSTMnCMP = 0000 0000<sub>H</sub>

The following figure shows the operation of OSTM when OSTMnCMP = 0000 0000<sub>H</sub>, and counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1).



**Figure 21.7 Timing Diagram when OSTMnCMP = 0000 0000<sub>H</sub> in Free-Run Compare Mode**

The timing diagram above shows the following operations.

- (1) Once the counter starts, it counts up from 0000 0000<sub>H</sub> to FFFF FFFF<sub>H</sub>.
- (2) An OSTMTINT interrupt request is generated when counting starts.

- (3) If the current counter value matches OSTMnCMP, an OSTMTINT interrupt request is generated.  
If OSTMnCMP = 0000 0000<sub>H</sub> as shown above, OSTMTINT is generated over two clock cycles.
  - (4) AN OSTMTINT interrupt request is generated for each clock cycle (FFFF FFFF<sub>H</sub> +1).
- When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

#### 21.4.3.3 Setting Procedure for Free-Run Compare Mode

The setting procedure in free-run compare mode after reset release is described below:

##### Setting procedure

- (1) Set the compare value in the OSTMnCMP register.
- (2) Select free-run compare mode by setting the OSTMnCTL.OSTMnMD1 bit to 1.
- (3) Enable or disable interrupts when counting starts by the OSTMnCTL.OSTMnMD0 bit.

## Section 22 Timer Array Unit B (TAUB)

This section contains a generic description of the timer array unit B (TAUB).

The first part of this section describes the RH850/F1L specific features such as the number of units and the register base addresses. The remainder of the section describes the functions and registers of the TAUB.

### 22.1 Features of RH850/F1L TAUB

#### 22.1.1 Number of Units and Channels

This microcontroller has the following number of TAUB units.

**Table 22.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	—	—	—	1	1	2
Name	—	—	—	TAUBn (n = 0)	TAUBn (n = 0)	TAUBn (n = 0, 1)

TAUBn has the following number of channels of timers.

**Table 22.2 TAUBn Unit Configurations and Channels**

Unit Name (Channel Name) TAUBn	Channels per Unit	RH850/F1L 48 pins (—)	RH850/F1L 64 pins (—)	RH850/F1L 80 pins (—)	RH850/F1L 100 pins (16 ch)	RH850/F1L 144 pins (16 ch)	RH850/F1L 176 pins (32 ch)
TAUB0	16	—	—	—	✓	✓	✓
TAUB1	16	—	—	—	—	—	✓

**Table 22.3 Indices**

Index	Description
n	Throughout this section, the individual TAUB units are identified by the index "n"; for example, TAUBnTOM is the TAUBn channel output mode register.
m	The TAUB has 16 channels. Throughout this section, the individual channels are identified by the index "m" (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

#### 22.1.2 Register Base Address

TAUBn base addresses are listed in the following table.

TAUBn register addresses are given as offsets from the base addresses in general.

**Table 22.4 Register Base Addresses**

Name	Base Address
<TAUB0_base>	FFE3 0000 <sub>H</sub>
<TAUB1_base>	FFE3 1000 <sub>H</sub>

### 22.1.3 Clock Supply

The TAUBn clock supply is shown in the following table.

**Table 22.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
TAUBn	PCLK	CKSCLK_IPERI2
	Register access clock	CKSCLK_IPERI2

### 22.1.4 Interrupt Requests

TAUBn interrupt requests are listed in the following table.

**Table 22.6 Interrupt Requests (1/2)**

Unit Interrupt Name	Outline	Interrupt Number	DMA Trigger Number
<b>TAUB0</b>			
INTTAUB0I0	Channel 0 interrupt	134	33 (Channels 0 to 7)
INTTAUB0I1	Channel 1 interrupt	135	28 (Channels 8 to 15)
INTTAUB0I2	Channel 2 interrupt	136	34 (Channels 0 to 7)
INTTAUB0I3	Channel 3 interrupt	137	29 (Channels 8 to 15)
INTTAUB0I4	Channel 4 interrupt	138	35 (Channels 0 to 7)
INTTAUB0I5	Channel 5 interrupt	139	30 (Channels 8 to 15)
INTTAUB0I6	Channel 6 interrupt	140	36 (Channels 0 to 7)
INTTAUB0I7	Channel 7 interrupt	141	31 (Channels 8 to 15)
INTTAUB0I8	Channel 8 interrupt	142	32 (Channels 8 to 15)
INTTAUB0I9	Channel 9 interrupt	143	37 (Channels 0 to 7)
INTTAUB0I10	Channel 10 interrupt	144	33 (Channels 8 to 15)
INTTAUB0I11	Channel 11 interrupt	145	38 (Channels 0 to 7)
INTTAUB0I12	Channel 12 interrupt	146	34 (Channels 8 to 15)
INTTAUB0I13	Channel 13 interrupt	147	39 (Channels 0 to 7)
INTTAUB0I14	Channel 14 interrupt	148	35 (Channels 8 to 15)
INTTAUB0I15	Channel 15 interrupt	149	40 (Channels 0 to 7)
<b>TAUB1</b>			
INTTAUB1I0	Channel 0 interrupt	248	52 (Channels 0 to 7)
INTTAUB1I1	Channel 1 interrupt	249	51 (Channels 8 to 15)
INTTAUB1I2	Channel 2 interrupt	250	53 (Channels 0 to 7)
INTTAUB1I3	Channel 3 interrupt	251	52 (Channels 8 to 15)
INTTAUB1I4	Channel 4 interrupt	252	54 (Channels 0 to 7)
INTTAUB1I5	Channel 5 interrupt	253	53 (Channels 8 to 15)
INTTAUB1I6	Channel 6 interrupt	254	55 (Channels 0 to 7)
INTTAUB1I7	Channel 7 interrupt	255	54 (Channels 8 to 15)
INTTAUB1I8	Channel 8 interrupt	256	55 (Channels 8 to 15)
INTTAUB1I9	Channel 9 interrupt	257	56 (Channels 0 to 7)
INTTAUB1I10	Channel 10 interrupt	258	56 (Channels 8 to 15)
INTTAUB1I11	Channel 11 interrupt	259	57 (Channels 0 to 7)
INTTAUB1I12	Channel 12 interrupt	260	57 (Channels 8 to 15)

Table 22.6 Interrupt Requests (2/2)

Unit Interrupt Name	Outline	Interrupt Number	DMA Trigger Number
INTTAUB1I13	Channel 13 interrupt	261	58 (Channels 0 to 7)
INTTAUB1I14	Channel 14 interrupt	262	58 (Channels 8 to 15)
INTTAUB1I15	Channel 15 interrupt	263	59 (Channels 0 to 7)

### 22.1.5 Reset Sources

TAUBn reset sources are listed in the following table. TAUBn is initialized by these reset sources.

Table 22.7 Reset Sources

Unit Name	Reset Source
TAUBn	All reset sources (ISORES)

### 22.1.6 External Input/Output Signals

External input/output signals of TAUBn are listed below.

Table 22.8 External Input/Output Signals (1/2)

TAUB Signal	Description	Alternative Port Pin Signal
<b>TAUB0</b>		
TAUBTTIN0	Channel 0 input* <sup>1</sup>	TAUB0I0
TAUBTTIN1	Channel 1 input* <sup>1</sup>	TAUB0I1
TAUBTTIN2	Channel 2 input* <sup>1</sup>	TAUB0I2
TAUBTTIN3	Channel 3 input* <sup>1</sup>	TAUB0I3
TAUBTTIN4	Channel 4 input* <sup>1</sup>	TAUB0I4
TAUBTTIN5	Channel 5 input* <sup>1</sup>	TAUB0I5
TAUBTTIN6	Channel 6 input* <sup>1</sup>	TAUB0I6
TAUBTTIN7	Channel 7 input* <sup>1</sup>	TAUB0I7
TAUBTTIN8	Channel 8 input* <sup>1</sup>	TAUB0I8
TAUBTTIN9	Channel 9 input* <sup>1</sup>	TAUB0I9
TAUBTTIN10	Channel 10 input* <sup>1</sup>	TAUB0I10
TAUBTTIN11	Channel 11 input* <sup>1</sup>	TAUB0I11
TAUBTTIN12	Channel 12 input* <sup>1</sup>	TAUB0I12
TAUBTTIN13	Channel 13 input* <sup>1</sup>	TAUB0I13
TAUBTTIN14	Channel 14 input* <sup>1</sup>	TAUB0I14
TAUBTTIN15	Channel 15 input* <sup>1</sup>	TAUB0I15
TAUBTTOUT0	Channel 0 output	TAUB0O0
TAUBTTOUT1	Channel 1 output	TAUB0O1
TAUBTTOUT2	Channel 2 output	TAUB0O2
TAUBTTOUT3	Channel 3 output	TAUB0O3
TAUBTTOUT4	Channel 4 output	TAUB0O4
TAUBTTOUT5	Channel 5 output	TAUB0O5
TAUBTTOUT6	Channel 6 output	TAUB0O6

Table 22.8 External Input/Output Signals (2/2)

TAUB Signal	Description	Alternative Port Pin Signal
TAUBTTOUT7	Channel 7 output	TAUB0O7
TAUBTTOUT8	Channel 8 output	TAUB0O8
TAUBTTOUT9	Channel 9 output	TAUB0O9
TAUBTTOUT10	Channel 10 output	TAUB0O10
TAUBTTOUT11	Channel 11 output	TAUB0O11
TAUBTTOUT12	Channel 12 output	TAUB0O12
TAUBTTOUT13	Channel 13 output	TAUB0O13
TAUBTTOUT14	Channel 14 output	TAUB0O14
TAUBTTOUT15	Channel 15 output	TAUB0O15
<b>TAUB1</b>		
TAUBTTIN0	Channel 0 input <sup>*1</sup>	TAUB1I0
TAUBTTIN1	Channel 1 input <sup>*1</sup>	TAUB1I1
TAUBTTIN2	Channel 2 input <sup>*1</sup>	TAUB1I2
TAUBTTIN3	Channel 3 input <sup>*1</sup>	TAUB1I3
TAUBTTIN4	Channel 4 input <sup>*1</sup>	TAUB1I4
TAUBTTIN5	Channel 5 input <sup>*1</sup>	TAUB1I5
TAUBTTIN6	Channel 6 input <sup>*1</sup>	TAUB1I6
TAUBTTIN7	Channel 7 input <sup>*1</sup>	TAUB1I7
TAUBTTIN8	Channel 8 input <sup>*1</sup>	TAUB1I8
TAUBTTIN9	Channel 9 input <sup>*1</sup>	TAUB1I9
TAUBTTIN10	Channel 10 input <sup>*1</sup>	TAUB1I10
TAUBTTIN11	Channel 11 input <sup>*1</sup>	TAUB1I11
TAUBTTIN12	Channel 12 input <sup>*1</sup>	TAUB1I12
TAUBTTIN13	Channel 13 input <sup>*1</sup>	TAUB1I13
TAUBTTIN14	Channel 14 input <sup>*1</sup>	TAUB1I14
TAUBTTIN15	Channel 15 input <sup>*1</sup>	TAUB1I15
TAUBTTOUT0	Channel 0 output	TAUB1O0
TAUBTTOUT1	Channel 1 output	TAUB1O1
TAUBTTOUT2	Channel 2 output	TAUB1O2
TAUBTTOUT3	Channel 3 output	TAUB1O3
TAUBTTOUT4	Channel 4 output	TAUB1O4
TAUBTTOUT5	Channel 5 output	TAUB1O5
TAUBTTOUT6	Channel 6 output	TAUB1O6
TAUBTTOUT7	Channel 7 output	TAUB1O7
TAUBTTOUT8	Channel 8 output	TAUB1O8
TAUBTTOUT9	Channel 9 output	TAUB1O9
TAUBTTOUT10	Channel 10 output	TAUB1O10
TAUBTTOUT11	Channel 11 output	TAUB1O11
TAUBTTOUT12	Channel 12 output	TAUB1O12
TAUBTTOUT13	Channel 13 output	TAUB1O13
TAUBTTOUT14	Channel 14 output	TAUB1O14
TAUBTTOUT15	Channel 15 output	TAUB1O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions.  
For details, see **Section 2.12, Noise Filter & Edge/Level Detector**.

## 22.2 Overview

### 22.2.1 Functional Overview

The TAUB has the following functions:

- 独立的channel 操作功能  
Independent channel operation function (operated using a single channel)  
同步channel操作功能 (使用一个master channel和多个slave channels)  
Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUB is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUBnCNTm and a 16-bit data register TAUBnCDRm to hold the count start value and compare value. 一个是保存起始值，一个保存比较值

It also contains several control and status registers.

#### Independent and synchronous operation

一个channel 可以独立操作，也可以跟其他的channel 一起

Every channel can operate either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel. 如果master channel 和 slave channel 一起，那么 slave channel 依赖于 master channel

如果一个channel 独立的操作，那么他就独立于其他所有的channels

When a channel is operated independently, it can be operated independent of all other channels. The synchronous operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

## 22.2.2 Terms

In this section, the following terms are used:

- Independent channel operation function/synchronous channel operation function  
16个channel  
 TAUB has 16 channels, and provides an independent channel operation function whereby individual channels operate independently and a synchronous channel operation function whereby multiple channels operate in combination.
- The independent channel operation function can be used by any channel independently of all other channels.
- The synchronous channel operation function is implemented using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

- Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

一个master 和至少一个slave  
A channel group has one master channel and one or more slave channels.

- Upper/lower channel

依赖于channel 号

Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- channel NO比较小的channel
- Upper channel: Channel with a smaller channel number
  - Channel NO 比较大的channel
  - Lower channel: Channel with a larger channel number

Example:

For channel 5, channel 3 is an upper channel and channel 9 is a lower channel. Channel 0 is the highest channel and channel 15 is the lowest channel.  
0最高，15最低

The following describes the functional blocks:

### Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of  $2^0$  to  $2^{15}$ .  
根据可配置的分频因子(2^0 ~ 2^15, CK0 到CK3 源于PCLK)

### Clock and count clock selection

对于每一个channel , 通过clock selector选取以下作为时钟源

For every channel, the count clock selector selects which of the following is used as the clock source:  
由时钟选择器来选择

可以作为时钟源

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

## Controller 控制器

此控制器控制计数器的主要操作  
The controller controls the main operations of the counter:

### 选择操作模式

- Operating mode (selected with the TAUBnCMORm.TAUBnMD[4:0] bits)  
使能和停止计数器
- Counter start enable (TAUBnTS, TAUBnTSm) and counter stop (TAUBnTT, TAUBnTTm)  
When counter start is enabled, status flag TAUBnTE, TAUBnTEm is set.  
计数的方向 可以由master channel来控制
- Count direction (up/down) (can be controlled by the master channel)

计数器的主要操作

## Trigger selector 触发选择器

当TAUBnTEm = 1时候，计数器会自动运行，或者的等待一个外部的触发信号，以下均可以作为起始触发  
The counter starts automatically when it is enabled (TAUBnTE, TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTOUTm generation unit.

可以做起始触发的信号

## Simultaneous rewrite controller 同时复写控制器

同时复写控制是一个特殊的功能，用于同步操作功能。  
Simultaneous rewrite control is a special function that can be used in synchronous operation functions.  
The data registers (TAUBnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time 同一channel group下的所有的channel的数据寄存器(TAUBnCDR)在任何时间可以被复写操作  
同时复写控制器保证所有channel的数据寄存器的值在同一时刻生效。

## TAUBnTO Controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves

每个channel的输出控制使能各种输出信号形式(PWM或者三角波)的产生

### 22.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels. **此定时器提供如下的功能，通过独立地操作每个channel 或者通过结合多个channels**

**Table 22.9 Functional List of TAUB Operations**

Operation Function	Setting Example
<b>Independent Channel Operation Functions</b>	<b>Section 22.12</b>
Interval Timer Function	<b>Section 22.12.1</b>
TAUBTTINm Input Interval Timer Function	<b>Section 22.12.2</b>
Clock Divide Function	<b>Section 22.12.3</b>
External Event Count Function	<b>Section 22.12.4</b>
One-Pulse Output Function	<b>Section 22.12.5</b>
TAUBTTINm Input Pulse Interval Measurement Function	<b>Section 22.12.6</b>
TAUBTTINm Input Signal Width Measurement Function	<b>Section 22.12.7</b>
TAUBTTINm Input Position Detection Function	<b>Section 22.12.8</b>
TAUBTTINm Input Period Count Detection Function	<b>Section 22.12.9</b>
TAUBTTINm Input Pulse Interval Judgment Function	<b>Section 22.12.10</b>
TAUBTTINm Input Signal Width Judgment Function	<b>Section 22.12.11</b>
Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)	<b>Section 22.12.12</b>
Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)	<b>Section 22.12.13</b>
<b>Independent Channel Simultaneous Rewrite Functions</b>	<b>Section 22.13</b>
Simultaneous Rewrite Trigger Generation Function Type 1	<b>Section 22.13.1</b>
<b>Synchronous Channel Operation Functions</b>	<b>Section 22.14</b>
PWM Output Function	<b>Section 22.14.1</b>
One-Shot Pulse Output Function	<b>Section 22.14.2</b>
Delay Pulse Output Function	<b>Section 22.14.3</b>
AD Conversion Trigger Output Function Type 1	<b>Section 22.14.4</b>
Triangle PWM Output Function	<b>Section 22.14.5</b>
Triangle PWM Output Function with Dead Time	<b>Section 22.14.6</b>
AD Conversion Trigger Output Function Type 2	<b>Section 22.14.7</b>

## 22.2.4 Input/Output Interrupt Request Signals

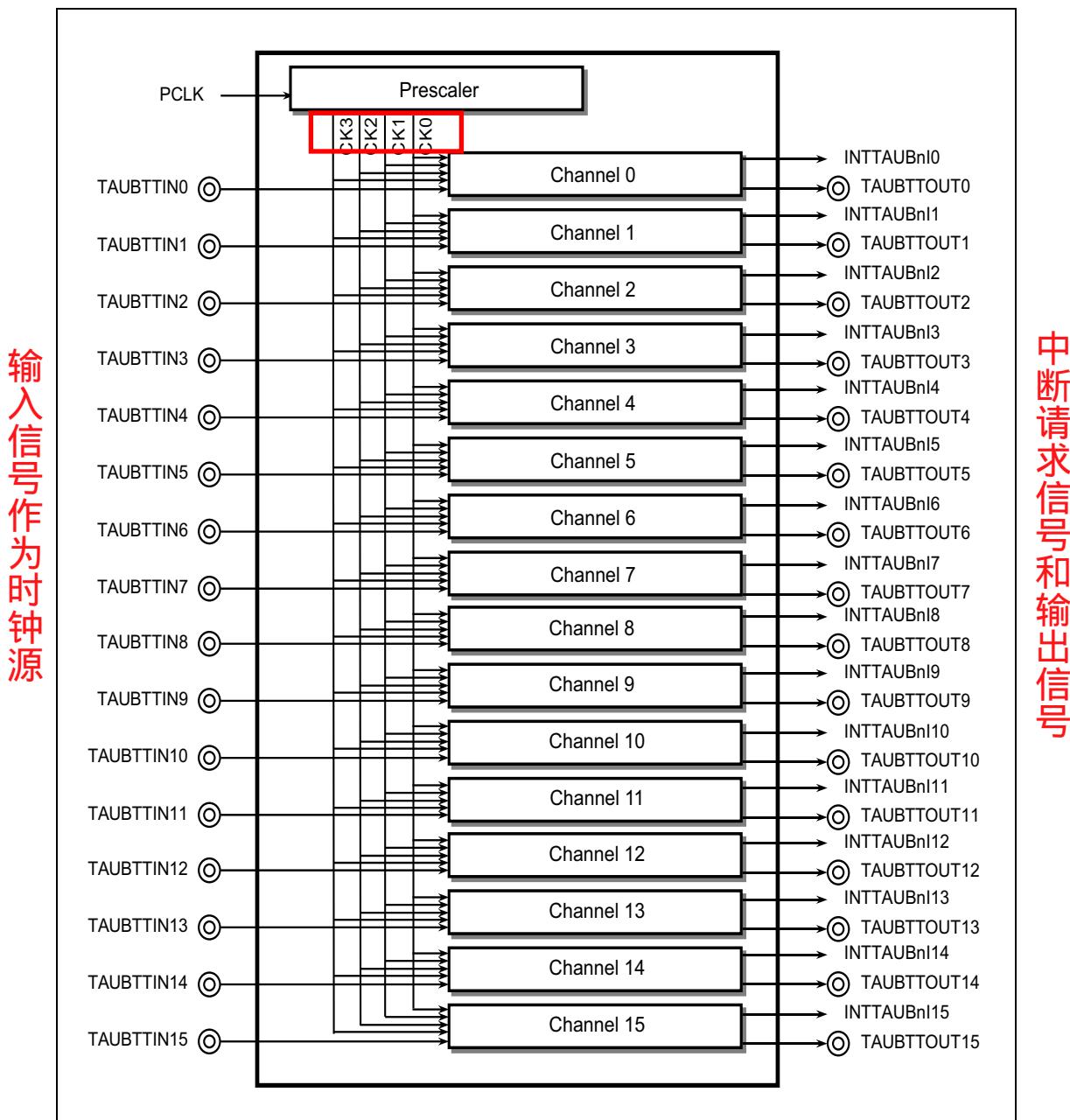


Figure 22.1 TAUB Input/Output and Interrupt Request Signals

## 22.2.5 Block Diagram

The following figure shows the main components of the TAUB.

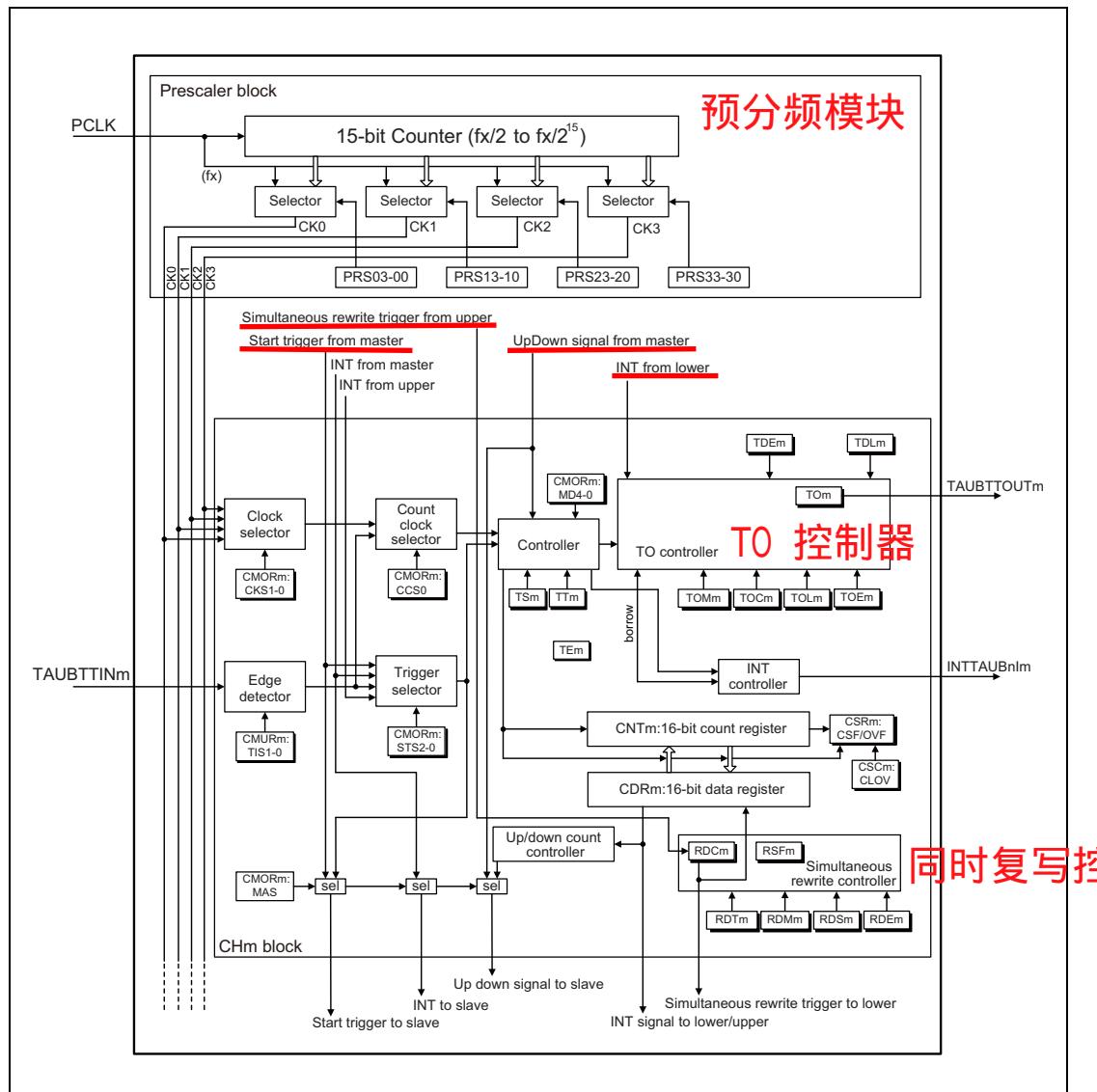


Figure 22.2 Block Diagram of the TAUB

The prefix “TAUBn” has been omitted from the register names for the sake of clarity in the above figure.

## 22.2.6 Description of Blocks

The following describes the functional blocks:

### Prescaler block 预分频区域

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.  
提供4路时钟信号

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of  $2^0$  to  $2^{15}$ . 代表 $2^0$  到 $2^{15}$

### Clock and count clock selection 时钟和计数时钟选择

对于每一个channel，计数时钟选择器选择以下作为时钟源

For every channel, the count clock selector selects which of the following is used as the clock source:

CK0到CK3中的一个

- One of the clocks CK0 to CK3 (selected by the clock selector)  
输入信号有效沿信号
- TAUBTTINm input signal valid edge

时钟源



### Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUBnCMORm.TAUBnMD[4:0])
- Counter start enable (TAUBnTS.TAUBnTSm) and counter stop (TAUBnTT.TAUBnTTm)  
When counter start is enabled, status flag TAUBnTE.TAUBnTEM is set.
- Count direction (up/down) (can be controlled by the master channel)

### Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUBnTE.TAUBnTEM = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- Input of TAUBTTINm valid edge
- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit

与1291页重复

### Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation functions. The data registers (TAUBnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

### TAUBnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

## 22.3 Registers

### 22.3.1 List of Registers

TAUB registers are listed in the following table.

For details about <TAUBn\_base>, see **Section 22.1.2, Register Base Address**.

Table 22.10 List of Registers

Module Name	Register Name	Symbol	Address
<b>TAUBn prescaler registers 预分频寄存器</b>			
TAUBn	TAUBn prescaler clock select register	TAUBnTPS	<TAUBn_base> + 240 <sub>H</sub>
<b>TAUBn control registers 控制寄存器</b>			
TAUBn	TAUBn channel data register m	TAUBnCDRm	<TAUBn_base> + 0 <sub>H</sub> + m × 4 <sub>H</sub>
TAUBn	TAUBn channel counter register m	TAUBnCNTm	<TAUBn_base> + 80 <sub>H</sub> + m × 4 <sub>H</sub>
TAUBn	TAUBn channel mode OS register m	TAUBnCMORm	<TAUBn_base> + 200 <sub>H</sub> + m × 4 <sub>H</sub>
TAUBn	TAUBn channel mode user register m	TAUBnCMURm	<TAUBn_base> + C0 <sub>H</sub> + m × 4 <sub>H</sub>
TAUBn	TAUBn channel status register m	TAUBnCSRm	<TAUBn_base> + 140 <sub>H</sub> + m × 4 <sub>H</sub>
TAUBn	TAUBn channel status clear trigger register m	TAUBnCSCm	<TAUBn_base> + 180 <sub>H</sub> + m × 4 <sub>H</sub>
TAUBn	TAUBn channel start trigger register	TAUBnTS	<TAUBn_base> + 1C4 <sub>H</sub>
TAUBn	TAUBn channel enable status register	TAUBnTE	<TAUBn_base> + 1C0 <sub>H</sub>
TAUBn	TAUBn channel stop trigger register	TAUBnTT	<TAUBn_base> + 1C8 <sub>H</sub>
<b>TAUBn output registers 输出寄存器</b>			
TAUBn	TAUBn channel output enable register	TAUBnTOE	<TAUBn_base> + 5C <sub>H</sub>
TAUBn	TAUBn channel output register	TAUBnTO	<TAUBn_base> + 58 <sub>H</sub>
TAUBn	TAUBn channel output mode register	TAUBnTOM	<TAUBn_base> + 248 <sub>H</sub>
TAUBn	TAUBn channel output configuration register	TAUBnTOC	<TAUBn_base> + 24C <sub>H</sub>
TAUBn	TAUBn channel output active level register	TAUBnTOL	<TAUBn_base> + 040 <sub>H</sub>
TAUBn	TAUBn channel dead time output enable register	TAUBnTDE	<TAUBn_base> + 250 <sub>H</sub>
TAUBn	TAUBn channel dead time output level register	TAUBnTDL	<TAUBn_base> + 54 <sub>H</sub>
<b>TAUBn reload data registers 重载数据寄存器</b>			
TAUBn	TAUBn channel reload data enable register	TAUBnRDE	<TAUBn_base> + 260 <sub>H</sub>
TAUBn	TAUBn channel reload data mode register	TAUBnRDM	<TAUBn_base> + 264 <sub>H</sub>
TAUBn	TAUBn channel reload data control CH select register	TAUBnRDS	<TAUBn_base> + 268 <sub>H</sub>
TAUBn	TAUBn channel reload data control register	TAUBnRDC	<TAUBn_base> + 26C <sub>H</sub>
TAUBn	TAUBn channel reload data trigger register	TAUBnRDT	<TAUBn_base> + 44 <sub>H</sub>
TAUBn	TAUBn channel reload status register	TAUBnRSF	<TAUBn_base> + 48 <sub>H</sub>
<b>TAUBn emulation register</b>			
TAUBn	TAUBn emulation register	TAUBnEMU	<TAUBn_base> + 290 <sub>H</sub>

## 22.3.2 Details of TAUBn Prescaler Registers

### 22.3.2.1 TAUBnTPS — TAUBn Prescaler Clock Select Register

This register specifies the clocks CK0, CK1, CK2, and CK3 for all channels of the PCLK prescaler.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUBn\_base> + 240<sub>H</sub>

**Value after reset:** FFFF<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnPRS3[3:0]				TAUBnPRS2[3:0]				TAUBnPRS1[3:0]				TAUBnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.11 TAUBnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUBnPRS3 [3:0]	Specifies the CK3 clock.
	TAUBnPRS3[3:0]	CK3 clock
	0000 <sub>B</sub>	PCLK/2 <sup>0</sup>
	0001 <sub>B</sub>	PCLK/2 <sup>1</sup>
	0010 <sub>B</sub>	PCLK/2 <sup>2</sup>
	0011 <sub>B</sub>	PCLK/2 <sup>3</sup>
	0100 <sub>B</sub>	PCLK/2 <sup>4</sup>
	0101 <sub>B</sub>	PCLK/2 <sup>5</sup>
	0110 <sub>B</sub>	PCLK/2 <sup>6</sup>
	0111 <sub>B</sub>	PCLK/2 <sup>7</sup>
	1000 <sub>B</sub>	PCLK/2 <sup>8</sup>
	1001 <sub>B</sub>	PCLK/2 <sup>9</sup>
	1010 <sub>B</sub>	PCLK/2 <sup>10</sup>
	1011 <sub>B</sub>	PCLK/2 <sup>11</sup>
	1100 <sub>B</sub>	PCLK/2 <sup>12</sup>
	1101 <sub>B</sub>	PCLK/2 <sup>13</sup>
	1110 <sub>B</sub>	PCLK/2 <sup>14</sup>
	1111 <sub>B</sub>	PCLK/2 <sup>15</sup>

只有在使用CK3的所有计数器停止的情况下才可以重新写入  
TAUBnTE.TAUBnTEm = 0

These bits can only be rewritten when all counters using CK3 are stopped  
(TAUBnTE.TAUBnTEm = 0).

Table 22.11 TAUBnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUBnPRS2 [3:0]	Specifies the CK2 clock.
		<b>TAUBnPRS2[3:0] CK2 Clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>
只有在使用CK2的所有计数器停止的情况下才可以重新写入 TAUBnTE.TAUBnTEm = 0		
These bits can only be rewritten when all counters using CK2 are stopped (TAUBnTE.TAUBnTEm = 0).		
7 to 4	TAUBnPRS1 [3:0]	Specifies the CK1 clock.
		<b>TAUBnPRS1[3:0] CK1 Clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>
只有在使用CK1的所有计数器停止的情况下才可以重新写入 TAUBnTE.TAUBnTEm = 0		
These bits can only be rewritten when all counters using CK1 are stopped (TAUBnTE.TAUBnTEm = 0).		

Table 22.11 TAUBnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUBnPRS0 [3:0]	Specifies the CK0 clock.
<b>TAUBnPRS0[3:0]</b>		<b>CK0 Clock</b>
0000 <sub>B</sub>		PCLK/2 <sup>0</sup>
0001 <sub>B</sub>		PCLK/2 <sup>1</sup>
0010 <sub>B</sub>		PCLK/2 <sup>2</sup>
0011 <sub>B</sub>		PCLK/2 <sup>3</sup>
0100 <sub>B</sub>		PCLK/2 <sup>4</sup>
0101 <sub>B</sub>		PCLK/2 <sup>5</sup>
0110 <sub>B</sub>		PCLK/2 <sup>6</sup>
0111 <sub>B</sub>		PCLK/2 <sup>7</sup>
1000 <sub>B</sub>		PCLK/2 <sup>8</sup>
1001 <sub>B</sub>		PCLK/2 <sup>9</sup>
1010 <sub>B</sub>		PCLK/2 <sup>10</sup>
1011 <sub>B</sub>		PCLK/2 <sup>11</sup>
1100 <sub>B</sub>		PCLK/2 <sup>12</sup>
1101 <sub>B</sub>		PCLK/2 <sup>13</sup>
1110 <sub>B</sub>		PCLK/2 <sup>14</sup>
1111 <sub>B</sub>		PCLK/2 <sup>15</sup>

只有在使用CK0的所有计数器停止的情况下才可以重新写入  
TAUBnTE.TAUBnTEm = 0

These bits can only be rewritten when all counters using CK0 are stopped  
(TAUBnTE.TAUBnTEm = 0).

#### NOTE

The TAUBn clock input PCLK is specified in the first part of this section, **Section 22.1.3, Clock Supply**.

### 22.3.3 Details of TAUBn Control Registers

channel 数据寄存器

#### 22.3.3.1 TAUBnCDRm — TAUBn Channel Data Register

既是一个比较寄存器也是一个捕获寄存器，取决于操作模式 (TAUBnCMORm.TAUBnMD[4:1])

This register functions either as a compare register or as a capture register, depending on the operation mode specified in TAUBnCMORm.TAUBnMD[4:1].

**Access:** This register can be read or written in 16-bit units.

- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

**Address:** <TAUBn\_base> + m × 4<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUBnCDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.12 TAUBnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCDR [15:0]	Data register for the capture/compare value.

### 22.3.3.2 TAUBnCNTm — TAUBn Channel Counter Register

This register is the channel m counter register.

**只读**

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAUBn\_base> + 80H + m × 4H

**Value after reset:** FFFF<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUBnCNT[15:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.13 TAUBnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCNT [15:0]	16-bit counter value.

读取的数值取决于计数器，操作模式，TAUBnTSm 和TAUBnTTm的bits。  
The read value depends on the counter, the operation mode change, and the values of the TAUBnTS.TAUBnTSm and TAUBnTT.TAUBnTTm bits.

初始化后的计数器的读取值取决于操作模式，和计数器停止的方式  
The initial counter read value depends on the operation mode and how the counter was stopped:

- by a reset
- by a counter stop trigger (TAUBnTT.TAUBnTTm = 1)

以下列出了计数器停止和重新使能后，初始化的计数器读取值

The following table lists the initial counter read values after the counter has stopped (TAUBnTE.TAUBnTEm = 0) and re-enabled (TAUBnTS.TAUBnTSm = 1).

The table also contains the counter read value one count after the counter is enabled

TAUBnTS.TAUBnTSm = 1) for modes where the counter waits for a start trigger.

Table 22.14 TAUBnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUBnCNTm		
		Start Value <sup>*1</sup>	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF <sub>H</sub>	Stop value	—
Judge mode	Count down	FFFF <sub>H</sub>	Stop value	—
Capture mode	Count up	0000 <sub>H</sub>	Stop value	—
Event count mode	Count down	FFFF <sub>H</sub>	Stop value	—
One-count mode	Count down	FFFF <sub>H</sub>	Stop value	Stop value
Capture and one-count mode	Count up	0000 <sub>H</sub>	Stop value	Capture value + 1 (TAUBnCDRm)
Judge and one-count mode	Count down	FFFF <sub>H</sub>	Stop value	TAUBnCNTm value – 1
Count-up/-down mode	Count up/down	FFFF <sub>H</sub>	Stop value	—
Pulse one-count mode	Count down	FFFF <sub>H</sub>	Stop value	0000 <sub>H</sub>
Count capture mode	Count up	0000 <sub>H</sub>	Stop value	—
Gate count mode	Count down	FFFF <sub>H</sub>	Stop value	Stop value
Capture and gate count mode	Count up	0000 <sub>H</sub>	Stop value	Stop value

Note 1. The value set for TAUBnCNTm when operation mode is changed after reset release

### 22.3.3.3 TAUBnCMORm — TAUBn Channel Mode OS Register

This register controls channel m operation.

**Access:** This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

**Address:** <TAUBn\_base> + 200<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.15 TAUBnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function															
15, 14	TAUBnCKS [1:0]	Selects the operation clock. <b>选择工作时钟，用于TAUBTTINm输入沿探测电路</b> The operation clock is used for the TAUBTTINm input edge detection circuit. TAUBnCNTm can also be used as the count clock depending on the setting of the TAUBnCMORm.TAUBnCCS0 bit <b>根据TAUBCCObit的设置，TAUBnCNT也可以用于计数时钟</b>															
		<table border="1"> <thead> <tr> <th>TAUBn CKS1</th> <th>TAUBn CKS0</th> <th>Selection of Count Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CK0</td> </tr> <tr> <td>0</td> <td>1</td> <td>CK1</td> </tr> <tr> <td>1</td> <td>0</td> <td>CK2</td> </tr> <tr> <td>1</td> <td>1</td> <td>CK3</td> </tr> </tbody> </table>	TAUBn CKS1	TAUBn CKS0	Selection of Count Clock	0	0	CK0	0	1	CK1	1	0	CK2	1	1	CK3
TAUBn CKS1	TAUBn CKS0	Selection of Count Clock															
0	0	CK0															
0	1	CK1															
1	0	CK2															
1	1	CK3															
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
12	TAUBnCCS0	Selects the count clock for the TAUBnCNTm counter <b>为TAUBnCNTm计数器选择计数时钟</b>															
		<table border="1"> <thead> <tr> <th>TAUBnCCS0</th> <th>Selected Count Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td><b>在TAUBnCKS[1:0]中指定的工作时钟</b> Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0].</td> </tr> <tr> <td>1</td> <td><b>TAUBTTIN 输入信号的有效沿</b> Valid edge of TAUBTTINm input signal</td> </tr> </tbody> </table>	TAUBnCCS0	Selected Count Clock	0	<b>在TAUBnCKS[1:0]中指定的工作时钟</b> Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0].	1	<b>TAUBTTIN 输入信号的有效沿</b> Valid edge of TAUBTTINm input signal									
TAUBnCCS0	Selected Count Clock																
0	<b>在TAUBnCKS[1:0]中指定的工作时钟</b> Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0].																
1	<b>TAUBTTIN 输入信号的有效沿</b> Valid edge of TAUBTTINm input signal																
11	TAUBnMAS	Specifies the channel as master or slave channel during synchronous channel operation: <b>在同步channel操作中选择信道座位master还是slave</b> 0: Slave 1: Master This bit is only valid for even channels (CHm_even). For odd channels (CHm_odd), it is fixed to 0.															

**注意：对于偶数channel，此bit是有效的，对于奇数channels，必须固定为0**

Table 22.15 TAUBnCMORm Register Contents (2/3)

Bit Position	Bit Name	Function																																				
10 to 8	TAUBnSTS [2:0]	Selects the external start trigger: 选择外部起始触发																																				
		<table border="1"> <thead> <tr> <th>TAUBn STS2</th> <th>TAUBn STS1</th> <th>TAUBn STS0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>Software trigger 软件触发</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>Valid edge of the TAUBTTINm input signal. TAUBnCSRm.TAUBnTIS[1:0] specifies the valid edge. 有效沿触发</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>Valid edge of the TAUBTTINm input signal is used as the start trigger and the reverse edge is used as the stop trigger. 有效沿触发作为起始触发，相反的边沿座位停止触发</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>Setting prohibited. INTTAUBnIm作为master channel的起始触发</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>INTTAUBnIm is the start trigger of the master channel</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>INTTAUBnIm of upper channel (m – 1) is the start trigger regardless of master setting</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>Dead-time output signal of the TAUBTOUTm generation unit</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of the master channel.</td></tr> </tbody> </table>	TAUBn STS2	TAUBn STS1	TAUBn STS0	Description	0	0	0	Software trigger 软件触发	0	0	1	Valid edge of the TAUBTTINm input signal. TAUBnCSRm.TAUBnTIS[1:0] specifies the valid edge. 有效沿触发	0	1	0	Valid edge of the TAUBTTINm input signal is used as the start trigger and the reverse edge is used as the stop trigger. 有效沿触发作为起始触发，相反的边沿座位停止触发	0	1	1	Setting prohibited. INTTAUBnIm作为master channel的起始触发	1	0	0	INTTAUBnIm is the start trigger of the master channel	1	0	1	INTTAUBnIm of upper channel (m – 1) is the start trigger regardless of master setting	1	1	0	Dead-time output signal of the TAUBTOUTm generation unit	1	1	1	Up/down output trigger signal of the master channel.
TAUBn STS2	TAUBn STS1	TAUBn STS0	Description																																			
0	0	0	Software trigger 软件触发																																			
0	0	1	Valid edge of the TAUBTTINm input signal. TAUBnCSRm.TAUBnTIS[1:0] specifies the valid edge. 有效沿触发																																			
0	1	0	Valid edge of the TAUBTTINm input signal is used as the start trigger and the reverse edge is used as the stop trigger. 有效沿触发作为起始触发，相反的边沿座位停止触发																																			
0	1	1	Setting prohibited. INTTAUBnIm作为master channel的起始触发																																			
1	0	0	INTTAUBnIm is the start trigger of the master channel																																			
1	0	1	INTTAUBnIm of upper channel (m – 1) is the start trigger regardless of master setting																																			
1	1	0	Dead-time output signal of the TAUBTOUTm generation unit																																			
1	1	1	Up/down output trigger signal of the master channel.																																			
7, 6	TAUBnCOS [1:0]	<p>Specifies when the capture register TAUBnCDRm and the overflow flag TAUBnCSRm.TAUBnOVF of channel m are updated.</p> <p>These bits are only valid if channel m is in capture function (capture mode and capture &amp; one-count mode).</p> <table border="1"> <thead> <tr> <th>TAUBn COS1</th> <th>TAUBn COS0</th> <th>TAUBnCDRm</th> <th>TAUBnCSRm.TAUBnOVF</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Updated upon detection of a TAUBTTINm input valid edge.</td><td>           Updated (cleared or set) upon detection of a TAUBTTINm input valid edge:           <ul style="list-style-type: none"> <li>If a counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is set.</li> <li>If no counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is cleared.</li> </ul> </td></tr> <tr> <td>0</td><td>1</td><td></td><td>Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.</td></tr> <tr> <td>1</td><td>0</td><td>Updated upon detection of a TAUBTTINm input valid edge and upon counter overflow:</td><td>Not set.</td></tr> <tr> <td>1</td><td>1</td><td> <ul style="list-style-type: none"> <li>TAUBTTINm input valid edge: Counter value is written to TAUBnCDRm</li> <li>Overflow: FFFF<sub>H</sub> is written to TAUBnCDRm. The next TAUBTTINm input valid edge detection is ignored.</li> </ul> </td><td>Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.</td></tr> </tbody> </table>	TAUBn COS1	TAUBn COS0	TAUBnCDRm	TAUBnCSRm.TAUBnOVF	0	0	Updated upon detection of a TAUBTTINm input valid edge.	Updated (cleared or set) upon detection of a TAUBTTINm input valid edge: <ul style="list-style-type: none"> <li>If a counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is set.</li> <li>If no counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is cleared.</li> </ul>	0	1		Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.	1	0	Updated upon detection of a TAUBTTINm input valid edge and upon counter overflow:	Not set.	1	1	<ul style="list-style-type: none"> <li>TAUBTTINm input valid edge: Counter value is written to TAUBnCDRm</li> <li>Overflow: FFFF<sub>H</sub> is written to TAUBnCDRm. The next TAUBTTINm input valid edge detection is ignored.</li> </ul>	Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.																
TAUBn COS1	TAUBn COS0	TAUBnCDRm	TAUBnCSRm.TAUBnOVF																																			
0	0	Updated upon detection of a TAUBTTINm input valid edge.	Updated (cleared or set) upon detection of a TAUBTTINm input valid edge: <ul style="list-style-type: none"> <li>If a counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is set.</li> <li>If no counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is cleared.</li> </ul>																																			
0	1		Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.																																			
1	0	Updated upon detection of a TAUBTTINm input valid edge and upon counter overflow:	Not set.																																			
1	1	<ul style="list-style-type: none"> <li>TAUBTTINm input valid edge: Counter value is written to TAUBnCDRm</li> <li>Overflow: FFFF<sub>H</sub> is written to TAUBnCDRm. The next TAUBTTINm input valid edge detection is ignored.</li> </ul>	Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.																																			
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																				

Table 22.15 TAUBnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function												
4 to 0	TAUBnMD [4:0]	Specifies the operation mode. For details, refer to the settings for individual functions. 指定工作模式												
TAUBn MD4	TAUBn MD3	TAUBn MD2	TAUBn MD1	TAUBn MD0	Description									
0	0	0	0	1/0	Interval timer mode									
0	0	0	1	1/0	Judge mode									
0	0	1	0	1/0	Capture mode									
0	0	1	1	0	Event count mode									
0	1	0	0	1/0	One-count mode									
0	1	0	1	1/0	Setting prohibited									
0	1	1	0	0	Capture and one-count mode									
0	1	1	1	1/0	Judge and one-count mode									
1	0	0	0	0	Setting prohibited									
1	0	0	1	0	Count-up/-down mode									
1	0	1	0	1/0	Pulse one-count mode									
1	0	1	1	1/0	Count capture mode									
1	1	0	0	0	Gate count mode									
1	1	0	1	0	Capture and gate count mode									
Mode	Role of TAUBnMD0 Bit													
Interval timer mode	指定INTTAUBnIm。Specifies whether INTTAUBnIm is output at the beginning of count operation (when a start trigger is entered) or not.													
Capture mode	在计数操作的开始是否输出。0: INTTAUBnIm is not output. 1: INTTAUBnIm is output.													
Count capture mode	(当start触发进入trigger is entered) INTTAUBnIm不输出													
Event count mode	此位应该指定为0。This bit should be set to 0 (INTTAUBnIm signal is not output at the beginning of count operation).													
Count-up/-down mode	在计数过程中使能/未能start触发检测。0: Disables detection. 1: Enables detection.													
One-count mode	Enables/disables start trigger detection during counting.													
Pulse one-count mode	在计数过程中使能/未能start触发检测。0: Disables detection. 1: Enables detection.													
CAUTION														
• In one-count mode, INTTAUBnIm signal is not output at the beginning of count operation. 在one-count mode下，在计数操作的开始 INTTAUBnIm不输出														
• In pulse one-count mode, INTTAUBnIm signal is output at the beginning of count operation. 在pulse one-count mode下，在计数操作的开始 INTTAUBnIm输出														
Gate count mode	应该为设置为0。This bit should be set to 0 (start trigger detection during counting is disabled).													
Capture and one-count mode	This bit should be set to 0. 要被设置为0													
Capture and gate count mode	INTTAUBnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.													
Judge mode	指定INTTAUBnIm。Specifies INTTAUBnIm output timing.													
Judge and one-count mode	0: When TAUBnCNTm ≤ TAUBnCDRm 1: When TAUBnCNTm > TAUBnCDRm													

### 22.3.3.4 TAUBnCMURm — TAUBn Channel Mode User Register

**指定用于TAUBTTINm输入的有效边沿的类型**

This register specifies the type of valid edge detection used for the TAUBTTINm input.

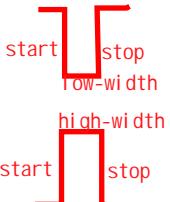
**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUBn\_base> + C0H + m × 4H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.16 TAUBnCMURm Register Contents

Bit Position	Bit Name	Function															
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
1, 0	TAUBnTIS [1:0]	Specifies the valid edge of the TAUBTTINm input signal:															
		<table border="1"> <thead> <tr> <th>TAUBn TIS1</th> <th>TAUBn TIS0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Falling edge</td> </tr> <tr> <td>0</td> <td>1</td> <td>Rising edge</td> </tr> <tr> <td>1</td> <td>0</td> <td>Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge</td> </tr> <tr> <td>1</td> <td>1</td> <td>Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge</td> </tr> </tbody> </table>	TAUBn TIS1	TAUBn TIS0	Description	0	0	Falling edge	0	1	Rising edge	1	0	Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge	1	1	Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge
TAUBn TIS1	TAUBn TIS0	Description															
0	0	Falling edge															
0	1	Rising edge															
1	0	Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge															
1	1	Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge															
		 <p>边沿检测基于由TAUBnCKS[1:0]选择的操作时钟</p> <ul style="list-style-type: none"> <li>Edge detection for TAUBTTINm input signals is performed based on the operation clock selected by <u>TAUBnCMORm.TAUBnCKS[1:0]</u>.</li> </ul>															

### 22.3.3.5 TAUBnCSRm — TAUBn Channel Status Register

This register indicates the count direction and the overflow status of the counter for channel m.

**只读**

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <TAUBn\_base> + 140<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnCSF	TAUBnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.17 TAUBnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUBnCSF	<p>Indicates the count direction: 0: Counts up 1: Counts down</p> <p>The read value of this bit is only valid in the following mode: • Up/Down Count mode <b>只有在此模式下，此位才有效</b></p>
0	TAUBnOVF	<p>Indicates the counter overflow status: 0: No overflow occurred <b>没有溢出</b> 1: Overflow occurred <b>溢出</b></p> <p>This bit is used only in the following modes: <b>只有在下面的模式下才有效</b> • Capture mode • Capture and one-count mode</p> <p><b>此位的功能取决于TAUBnCOS[1:0]的设置</b> The function of this bit depends on the setting of control bits TAUBnCMORm.TAUBnCOS[1:0].</p>

### 22.3.3.6 TAUBnCSCm — TAUBn Channel Status Clear Register

This register is a trigger register for clearing the overflow flag TAUBnCSRm.TAUBnOVF of channel m.

**只写**

**Access:** This register is a write-only register that can be written in 8-bit units. It is always read as 00<sub>H</sub>.

**Address:** <TAUBn\_base> + 180<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 22.18 TAUBnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUBnCLOV	<p>0: No function</p> <p>1: Clears the overflow flag TAUBnCSRm.TAUBnOVF <b>清除溢出的标志</b></p>

### 22.3.3.7 TAUBnTS — TAUBn Channel Start Trigger Register

This register enables the counter for each channel.

**只写**

**Access:** This register is a write-only register that can be written in 16-bit units. It is always read as 0000<sub>H</sub>

**Address:** <TAUBn\_base> + 1C4H

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TS15	TAUBn TS14	TAUBn TS13	TAUBn TS12	TAUBn TS11	TAUBn TS10	TAUBn TS09	TAUBn TS08	TAUBn TS07	TAUBn TS06	TAUBn TS05	TAUBn TS04	TAUBn TS03	TAUBn TS02	TAUBn TS01	TAUBn TS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 22.19 TAUBnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTSm	Enables the counter for channel m: 0: No function 1: Enables the counter and sets TAUBnTE.TAUBnTEm = 1. <b>使能计数器并且设置TE为1</b> TAUBnTE.TAUBnTEm = 1 only enables the counter. Whether the counter starts depends on the selected operation mode.

TE=1只是使能计数器，计数器是否启动决定于所选的模式

**使能状态寄存器**

### 22.3.3.8 TAUBnTE — TAUBn Channel Enable Status Register

This register indicates whether counter is enabled or disabled.

**只读**

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAUBn\_base> + 1C0H

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TE15	TAUBn TE14	TAUBn TE13	TAUBn TE12	TAUBn TE11	TAUBn TE10	TAUBn TE09	TAUBn TE08	TAUBn TE07	TAUBn TE06	TAUBn TE05	TAUBn TE04	TAUBn TE03	TAUBn TE02	TAUBn TE01	TAUBn TE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.20 TAUBnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTEm	Indicates whether counter for channel m is enabled or disabled: 0: Counter disabled 1: Counter enabled Setting TAUBnTS.TAUBnTSm to 1 sets this bit to 1. Setting TAUBnTT.TAUBnTTm to 1 resets this bit to 0.

这一位受到TS和TT的控制

Channel 停止触发寄存器

**22.3.3.9 TAUBnTT — TAUBn Channel Stop Trigger Register**

This register stops the counter for each channel.

**只写**

**Access:** This register is a write-only register that can be written in 16-bit units. It is always read as 0000<sub>H</sub>.

**Address:** <TAUBn\_base> + 1C8H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TT15	TAUBn TT14	TAUBn TT13	TAUBn TT12	TAUBn TT11	TAUBn TT10	TAUBn TT09	TAUBn TT08	TAUBn TT07	TAUBn TT06	TAUBn TT05	TAUBn TT04	TAUBn TT03	TAUBn TT02	TAUBn TT01	TAUBn TT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

**Table 22.21 TAUBnTT Register Contents**

Bit Position	Bit Name	Function
15 to 0	TAUBnTTm	<p>Stops the counter of channel m:</p> <p>0: No function</p> <p>1: Stops the counter and resets TAUBnTE.TAUBnTEm.  <u>TAUBnCNTm, TAUBnTO, TAUBnTOM, and TAUBTTOUTm all retain the values they had before the counter was stopped.</u></p>

### 同时复写寄存器

#### 22.3.4 Details of TAUBn Simultaneous Rewrite Registers

##### 重载数据使能寄存器

##### 22.3.4.1 TAUBnRDE — TAUBn Channel Reload Data Enable Register

使能或者失能对数据寄存器 TAUBnCDRm/TAUBnTOLm的同时复写

This register enables and disables simultaneous rewrite of the data register TAUBnCDRm/TAUBnTOLm.

在TAUBnTEm=0的情况下只能写入

**Access:** This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

**Address:** <TAUBn\_base> + 260H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDE15	TAUBn RDE14	TAUBn RDE13	TAUBn RDE12	TAUBn RDE11	TAUBn RDE10	TAUBn RDE09	TAUBn RDE08	TAUBn RDE07	TAUBn RDE06	TAUBn RDE05	TAUBn RDE04	TAUBn RDE03	TAUBn RDE02	TAUBn RDE01	TAUBn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 22.22 TAUBnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 禁止 1: Enabled simultaneous rewrite 使能

#### 22.3.4.2 TAUBnRDS — TAUBn Channel Reload Data Control Channel Select Register

选择同时复写下的控制channel

This register selects the control channel for simultaneous rewrite.

**Access:** This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

**Address:** <TAUBn\_base> + 268H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDS15	TAUBn RDS14	TAUBn RDS13	TAUBn RDS12	TAUBn RDS11	TAUBn RDS10	TAUBn RDS09	TAUBn RDS08	TAUBn RDS07	TAUBn RDS06	TAUBn RDS05	TAUBn RDS04	TAUBn RDS03	TAUBn RDS02	TAUBn RDS01	TAUBn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 22.23 TAUBnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDSm	Specifies which channel is controlled for the simultaneous rewrite trigger: 0: Master channel 1: Another upper channel

### 22.3.4.3 TAUBnRDM — TAUBn Channel Reload Data Mode Register

**此寄存器选择控制同时复写的信号什么时候产生**

This register selects when the signal that controls simultaneous rewrite is generated.

**Access:** This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

**Address:** <TAUBn\_base> + 264H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDM15	TAUBn RDM14	TAUBn RDM13	TAUBn RDM12	TAUBn RDM11	TAUBn RDM10	TAUBn RDM09	TAUBn RDM08	TAUBn RDM07	TAUBn RDM06	TAUBn RDM05	TAUBn RDM04	TAUBn RDM03	TAUBn RDM02	TAUBn RDM01	TAUBn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.24 TAUBnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDMm	Selects when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting <b>当master开始计数的时候</b> 1: At the top of a triangle wave cycle <b>在三角波cycle的顶端</b> These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 0.

**在TAUBnRDE = 1和TAUBnRDSm=0的情况下，这些bits才会应用**

### 22.3.4.4 TAUBnRDC — TAUBn Channel Reload Data Control Register

**此寄存器指定产生INTTAUBnIm信号的channel，此信号用于触发同时复写**  
This register specifies the channel that generates the INTTAUBnIm signal that triggers simultaneous rewrite.

**Access:** This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0

**Address:** <TAUBn\_base> + 26CH

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDC15	TAUBn RDC14	TAUBn RDC13	TAUBn RDC12	TAUBn RDC11	TAUBn RDC10	TAUBn RDC09	TAUBn RDC08	TAUBn RDC07	TAUBn RDC06	TAUBn RDC05	TAUBn RDC04	TAUBn RDC03	TAUBn RDC02	TAUBn RDC01	TAUBn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.25 TAUBnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. <b>不作为触发channel</b> 1: Operates as a simultaneous rewrite trigger channel. <b>作为触发channel</b> These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 1.

**在TAUBnRDE = 1和TAUBnRDSm=1的情况下，这些bits才会应用**

### 22.3.4.5 TAUBnRDT — TAUBn Channel Reload Data Trigger Register

触发同时复写的使能状态

This register triggers the simultaneous rewrite enabling state.

只写

**Access:** This register is a write-only register that can be written in 16-bit units. It is always read as 0000<sub>H</sub>.

**Address:** <TAUBn\_base> + 044H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDT15	TAUBn RDT14	TAUBn RDT13	TAUBn RDT12	TAUBn RDT11	TAUBn RDT10	TAUBn RDT09	TAUBn RDT08	TAUBn RDT07	TAUBn RDT06	TAUBn RDT05	TAUBn RDT04	TAUBn RDT03	TAUBn RDT02	TAUBn RDT01	TAUBn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 22.26 TAUBnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDTm	<p>Triggers the simultaneous rewrite enabling state:            0: No function. Writing 0 is ignored (the operation is not affected).            1: The simultaneous rewrite enabling flag (TAUBnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger.            These bits only apply when: 1: 同时复写使能标志 (TAUBnRSFm) 被设置为1，系统等待同时复写触发            • TAUBnRDE.TAUBnRDEM = 1            只有在TAUBnRDE = 1的情况下这些bits才能应用</p>

### 22.3.4.6 TAUBnRSF — TAUBn Channel Reload Status Register

指示同时复写的状态

This flag register indicates the simultaneous rewrite status.

只读

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAUBn\_base> + 048H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RSF15	TAUBn RSF14	TAUBn RSF13	TAUBn RSF12	TAUBn RSF11	TAUBn RSF10	TAUBn RSF09	TAUBn RSF08	TAUBn RSF07	TAUBn RSF06	TAUBn RSF05	TAUBn RSF04	TAUBn RSF03	TAUBn RSF02	TAUBn RSF01	TAUBn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.27 TAUBnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRSFm	<p>Indicates the simultaneous rewrite status:            0: Indicates simultaneous rewrite is completed due to the generation of the simultaneous rewrite trigger. 由于同时复写触发的产生，同时复写功能已经完成            1: Indicates the simultaneous rewrite trigger waiting state when simultaneous rewrite is enabled (TAUBnRDTm = 1).</p>

在同时复写被使能 (TAUBnRDTm = 1) 后，同时复写触发处于等待的状态

### 22.3.5 Details of TAUBn Output Registers

#### 22.3.5.1 TAUBnTOE — TAUBn Channel Output Enable Register

使能或者失能由软件控制的独立的channel输出模式

This register enables and disables independent channel output mode controlled by software.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUBn\_base> + 5C<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOE15	TAUBn TOE14	TAUBn TOE13	TAUBn TOE12	TAUBn TOE11	TAUBn TOE10	TAUBn TOE09	TAUBn TOE08	TAUBn TOE07	TAUBn TOE06	TAUBn TOE05	TAUBn TOE04	TAUBn TOE03	TAUBn TOE02	TAUBn TOE01	TAUBn TOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.28 TAUBnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOEm	Enables/disables independent channel output mode: 0: Disables independent channel output mode (controlled by software) 1: Enables independent channel output mode

#### 22.3.5.2 TAUBnTO — TAUBn Channel Output Register

指定或者读取 TAUBTTOUTm的电平

This register specifies and reads the level of TAUBTTOUTm.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUBn\_base> + 58<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TO15	TAUBn TO14	TAUBn TO13	TAUBn TO12	TAUBn TO11	TAUBn TO10	TAUBn TO09	TAUBn TO08	TAUBn TO07	TAUBn TO06	TAUBn TO05	TAUBn TO04	TAUBn TO03	TAUBn TO02	TAUBn TO01	TAUBn TO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.29 TAUBnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOm	Specifies/reads the level of TAUBTTOUTm: 0: Low 1: High Only TAUBnTOm bits for which Independent Channel Output function is disabled (TAUBnTOEm = 0) can be written.

对于某个独立的channel 输出功能，只有在TAUBnTOEm=0的情况下此位才能被写入

### 22.3.5.3 TAUBnTOM — TAUBn Channel Output Mode Register

指定某个channel的输出模式

This register specifies the output mode of each channel.

只有在计数器停止的情况下才能被写入

**Access:** This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEM = 0).

**Address:** <TAUBn\_base> + 248H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOM15	TAUBn TOM14	TAUBn TOM13	TAUBn TOM12	TAUBn TOM11	TAUBn TOM10	TAUBn TOM09	TAUBn TOM08	TAUBn TOM07	TAUBn TOM06	TAUBn TOM05	TAUBn TOM04	TAUBn TOM03	TAUBn TOM02	TAUBn TOM01	TAUBn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 22.30 TAUBnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode

### 22.3.5.4 TAUBnTOC — TAUBn Channel Output Configuration Register

跟TAUBnTOMm一起指定某个channel的输出模式

This register specifies the output mode of each channel in combination with TAUBnTOMm.

只有在计数器停止的情况下才能被写入

**Access:** This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEM = 0).

**Address:** <TAUBn\_base> + 24CH

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOC15	TAUBn TOC14	TAUBn TOC13	TAUBn TOC12	TAUBn TOC11	TAUBn TOC10	TAUBn TOC09	TAUBn TOC08	TAUBn TOC07	TAUBn TOC06	TAUBn TOC05	TAUBn TOC04	TAUBn TOC03	TAUBn TOC02	TAUBn TOC01	TAUBn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 22.31 TAUBnTOC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOCm	Specifies the output mode: 0: Operation mode 1 1: Operation mode 2  The output mode also depends on TAUBnTOM.TAUBnTOMm, as can be seen in the following table.

TOMm	TOCm	Description
0	0	Toggle mode: Toggles when INTTAUBnIm occurs.
1		Set/reset mode: Set when INTTAUBnIm occurs upon count start and reset when INTTAUBnIm occurs due to detection of a match between TAUBnCNTm and TAUBnCDRm.
1	0	Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.
1	1	Synchronous channel operation mode 2: Set when INTTAUBnIm occurs while the slave channel is counting down and reset when INTTAUBnIm occurs while the slave channel is counting up.

### 22.3.5.5 TAUBnTOL — TAUBn Channel Output Level Register

指定某个channel output bit (TAUBnTO.TAUBnTOm)的输出逻辑  
This register specifies the output logic of the channel output bit (TAUBnTO.TAUBnTOm).

解释：如果此值为低电平，那么TAUBTOUT在有效期间为低电平  
有效期间是指TAUBnCNT在下降过程中。

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUBn\_base> + 040H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOL15	TAUBn TOL14	TAUBn TOL13	TAUBn TOL12	TAUBn TOL11	TAUBn TOL10	TAUBn TOL09	TAUBn TOL08	TAUBn TOL07	TAUBn TOL06	TAUBn TOL05	TAUBn TOL04	TAUBn TOL03	TAUBn TOL02	TAUBn TOL01	TAUBn TOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 22.32 TAUBnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOLm	<p>Specifies the output logic of the channel m output bit (TAUBnTO.TAUBnTOm):</p> <p>0: Positive logic (active high) 高电平 1: Negative logic (active low) 低电平</p> <p>The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software.</p> <p>这些bits的设置除了independent channel 输出模式以外均可以通过软件应用于各种模式</p>

### 22.3.6 Details of TAUBn Dead Time Output Registers

#### 22.3.6.1 TAUBnTDE — TAUBn Channel Dead Time Output Enable Register

This register enables/disables dead time operation for each channel.

只有在计时器停止的情况下才能写入

**Access:** This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

**Address:** <TAUBn\_base> + 250H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDE15	TAUBn TDE14	TAUBn TDE13	TAUBn TDE12	TAUBn TDE11	TAUBn TDE10	TAUBn TDE09	TAUBn TDE08	TAUBn TDE07	TAUBn TDE06	TAUBn TDE05	TAUBn TDE04	TAUBn TDE03	TAUBn TDE02	TAUBn TDE01	TAUBn TDE00

Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.33 TAUBnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDEm	<p>Enables/disables dead time control operation of channel m:            0: Disables dead time operation 禁止            1: Enables dead time operation 使能</p> <p>The same settings must be set for the even and the odd slave channel that comprise a set. 必须为同一集合下的 even odd slave channel 配置相同的设置</p> <p>These bits only apply when: 只有在下面条件下才能应用</p> <ul style="list-style-type: none"> <li>• TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMM,            TAUBnTOC.TAUBnTOCm = 1</li> </ul>

#### 22.3.6.2 TAUBnTDL — TAUBn Channel Dead Time Output Level Register

选择添加到dead time的phase period

This register selects the phase period to which dead time is added.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUBn\_base> + 54H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDL15	TAUBn TDL14	TAUBn TDL13	TAUBn TDL12	TAUBn TDL11	TAUBn TDL10	TAUBn TDL09	TAUBn TDL08	TAUBn TDL07	TAUBn TDL06	TAUBn TDL05	TAUBn TDL04	TAUBn TDL03	TAUBn TDL02	TAUBn TDL01	TAUBn TDL00

Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.34 TAUBnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDLm	<p>Selects the phase period to which dead time is added:            0: Positive phase period            1: Negative phase period</p> <p>These bits only apply when:</p> <ul style="list-style-type: none"> <li>• TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMM,            TAUBnTOC.TAUBnTOCm, TAUBnTDE.TAUBnTDEM = 1</li> </ul>

## 22.3.7 TAUBn Emulation Register

### 22.3.7.1 TAUBnEMU — TAUBn Emulation Register

This register controls SVSTOP operations.

**Access:** This register can be read or written in 8-bit units.  
Write to this register only when the counter is stopped (TAUBnTE.TAUBnTEm = 0) and when EPC.SVSTOP = 0.

**Address:** <TAUBn\_base> + 290H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	TAUBnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 22.35 TAUBnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUBnSVSDIS	(When EPC.SVSTOP = 0) Regardless of the value of this bit (1/0), the count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint).  (When EPC.SVSTOP = 1) 0: The count clock stops when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). 1: The count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

同OSTM的寄存器

## 22.4 General Operating Procedure

The following describes the general operation procedure for the TAUBn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUBTOUTm is also initialized and outputs a low level.

设置TAUBnTPS来指定CK0到CK3的时钟频率

1. Set the TAUBnTPS register to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUBn function:
  - Set the operation mode 设置工作模式
  - Set the channel output mode 设置channel输出模式
  - Set any other control bits 设置其他的control bits
3. Enable the counter by setting the TAUBnTS.TAUBnTSm bit to 1. 启动计数器  
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings. 计数器立即启动，或者等待相应的触发，根据bit的设置来定  
可以操作对应的寄存器来停止或者强制重启计数器
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUBnTT.TAUBnTTm bit to 1. The counter can be forcibly restarted by setting the TAUBnTS.TAUBnTSm bit to 1.  
停止计数器的功能
5. Stop the function by setting the TAUBnTT.TAUBnTTm bit to 1.

### NOTE

- A detailed description of the required control bits and the operation of the individual functions is given below.
  - **Section 22.12, Independent Channel Operation Functions**
  - **Section 22.14, Synchronous Channel Operation Functions**
- The function can be changed while the counter is stopped (TAUBnTE.TAUBnTEM=0).  
在计数器停止的情况下，计数器的功能可以被改变

## 22.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels). 使用channel 组 (包含master和slave ) 的形式实现

Several rules apply to the settings of channels.

These rules are detailed in **Section 22.5.1, Rules of Synchronous Channel Operation Function.**

Two special features for synchronous channel operation are detailed in the following:

- **Section 22.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 22.6, Simultaneous Rewrite**

### 规则

#### 22.5.1 Rules of Synchronous Channel Operation Function

##### Number of master and slave channels

只有偶数的channel可以被作为master channels

- Only even channels (CH0, CH2, CH4, ...) can be set as master channels. Any channel apart from CH0 can be set as a slave channel. CH0之外的channel 可以被设置为 slave channel
- 只有低于Master channel 的channel 才可以作为 slave channels , 可以为一个主通道设置多个slave通道
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.

Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, ...) can be set as slave channels.

如果使用了多个master channels , 包含另一个主通道的一系列的slave通道不能设置为指定的master channel

- If multiple master channels are used, a range of slave channels that includes another master channel cannot be set for a given master channel.

Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

##### Operation clock

必须为master channel 和同步的slave channel 设置相同的工作时钟

- The same operation clock must be set for the master channel and the synchronized slave channel.

This is achieved by setting the same value to the TAUBnCMORm.TAUBnCKS[1:0] bits of the master and slave channels. 在master和slave的TAUBnCKS[1:0]设置可以实现上面的功能

The basic concepts of master/slave channel usage and operation clocks are illustrated in the following figure.

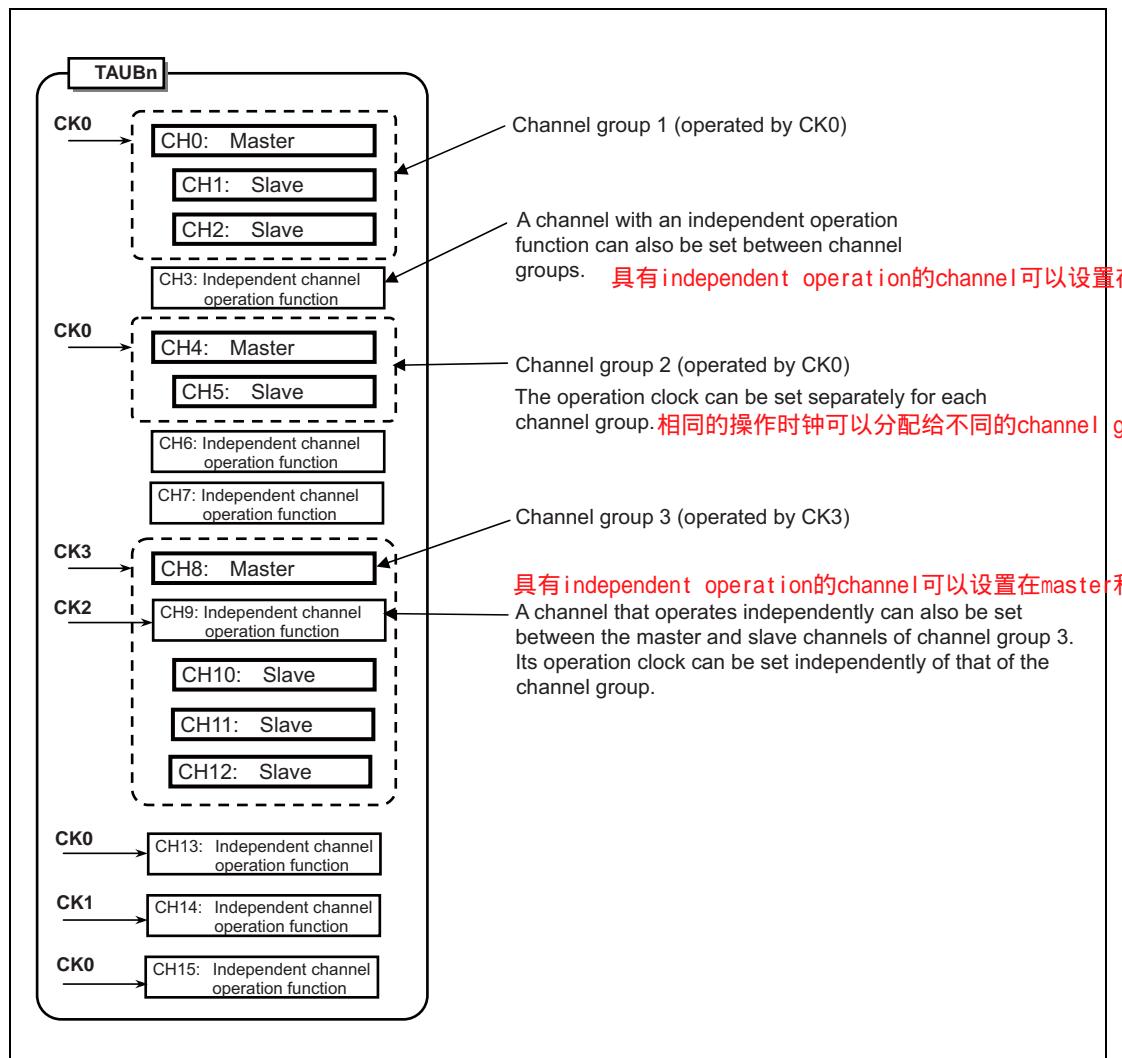


Figure 22.3 Grouping of the Channels and Assignment of Operation Clocks

## 22.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

相同Unit下的同时启动和停止

### 22.5.2.1 Simultaneous Start and Stop within the Same Unit

同时启动同步的channel，需要在同一时刻设置TAUBnTSm bit

- To simultaneously start synchronized channels, the TAUBnTS.TAUBnTSm bits of the channels must be set at the same time.  
同时停止同步的channel，需要在同一时刻设置TAUBnTTm bit
- To simultaneously stop synchronized channels, the TAUBnTT.TAUBnTTm bits of the channels must be set at the same time.

Setting the TAUBnTS.TAUBnTSm bits to 1 sets the corresponding TAUBnTE.TAUBnTEm bits to 1, enabling counting. The exact time that it starts depends on the operation mode.

准确的启动时间依赖于操作模式

## 22.6 同时复写 Simultaneous Rewrite

### 22.6.1 Introduction

**同时复写可以在同一时刻改变多个channel的compare/start value和输出逻辑**

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

**对应的数据和控制寄存器可以在任何时间写入，直到同时复写被触发，新的值不会影响计数器的操作**  
The corresponding data and control registers (TAUBnCDRm and TAUBnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

**同时复写由下列条件触发**

Simultaneous rewrite can be triggered by:

- master或者更高的channel的计数器达到某个值  
The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- 由TAUBnRDCm指定的upper channel上产生INTTAUBnIm中断  
INTTAUBnIm being issued on the upper channel specified by TAUBnRDC.TAUBnRDCm

There are three methods for simultaneous rewrite. These are listed in the following table, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 22.36 Simultaneous Rewrite Methods and when They are Triggered

Method	Trigger	TAUBn	TAUBn	TAUBn
		RDE.	RDS.	RDM.
		TAUBn	TAUBn	TAUBn
		RDEm	RDSm	RDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting master启动计数	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular wave of the corresponding slave channel. master已经启动计数，master在对应slave的三角波的定点开始向下计数	1	0	1
C1	INTTAUBnIm is generated on an upper channel specified by TAUBnRDC.TAUBnRDCm 由TAUBnRDCm指定的upper channel上产生INTTAUBnIm中断	1	1	0/1

The following table lists which of these three methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 22.12, Independent Channel Operation Functions** and **Section 22.14, Synchronous Channel Operation Functions**.

Table 22.37 Channel Functions and the Methods They Use for Simultaneous Rewrite

Descriptions	A	B	C1	TAUBnTOL. TAUBnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			√	
PWM Output Function	√		√	√
One-Shot Pulse Output Function	√			
Delay Pulse Output Function	√			
Triangle PWM Output Function		√	√	√
Triangle PWM Output Function with Dead Time		√	√	
AD Conversion Trigger Output Function Type 1	√		√	
AD Conversion Trigger Output Function Type 2		√	√	

**Note:** √: Available, (Blank): Unavailable

## 控制同时复写的流程

### 22.6.2 How to Control Simultaneous Rewrite

具体流程见下一页

The following figure shows the general procedure for simultaneous rewrite.

The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

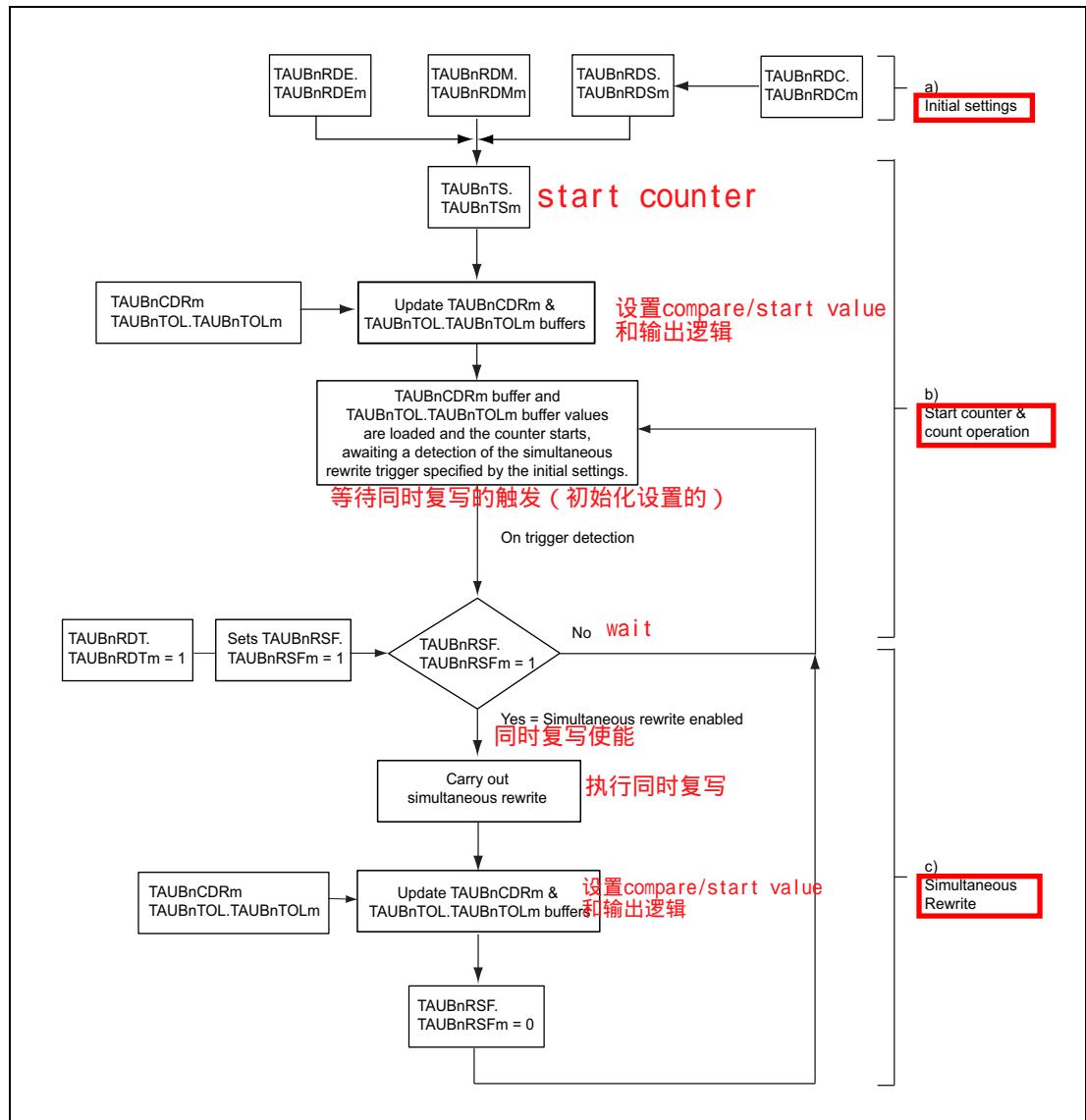


Figure 22.4 General Procedure for Simultaneous Rewrite

### 22.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set 使能同时复写 TAUBnRDE.TAUBnRDEm = 1.
- To select the type of simultaneous rewrite, set 选择同时复写的类型 TAUBnRDM.TAUBnRDMm and TAUBnRDS.TAUBnRDSm according to the values in **Table 22.36, Simultaneous Rewrite Methods and when They are Triggered.**
- Specify a simultaneous rewrite trigger channel by using 指定同时复写的触发channel TAUBnRDC.TAUBnRDCm.  
(Prerequisite: TAUBnRDS.TAUBnRDSm has been set to the upper channel.)  
先决条件：TAUBnRDSm 已经被设置为the upper channel，也就是1

### 22.6.2.2 Start Counter and Count Operation

- To start all the TAUBnCNTm counters in the channel group, set the corresponding 启动channel group下的所有的counters TAUBnTS.TAUBnTSm bits to 1. TAUBnTOL.TAUBnTOLm and the values in the data registers (TAUBnCDRm) are written to the corresponding TAUBnTOL.TAUBnTOLm buffer (TAUBnTOL.TAUBnTOLm buf) and data buffer registers (TAUBnCDRm buf) and the counters start.  
设置重载数据触发bit为1，这样就可以将reload flag设置为1，使能同时复写
- Setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1 sets the reload flag (TAUBnRSF.TAUBnRSFm) to 1, enabling simultaneous rewrite. TAUBnRSF.TAUBnRSFm remains at 1 until simultaneous rewrite has taken place. 在同时复写发生之前，TAUBnRSFm保持1的状态
- If the specified trigger for simultaneous rewrite is detected, the TAUBnRSF.TAUBnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1). If it is, simultaneous rewrite is carried out.  
Otherwise, simultaneous rewrite is not carried out, and the system awaits the next simultaneous rewrite trigger detection. 不然只能等待检测下次的同时复写触发

### 22.6.2.3 Simultaneous Rewrite 同时复写的执行过程

- When simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then written to the corresponding counters and the values are applied the next time the counter starts or restarts. 这些值会被写入响应的计数器并且在下次计数器开始或重启时生效
- When simultaneous rewrite is finished, the TAUBnRSF.TAUBnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.  
当重写完成后，RSF bit会被设置为0，然后等待检测下次的同时复写触发

### 22.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUBnRDE.TAUBnRDEM, TAUBnRDS.TAUBnRDSM, TAUBnRDM.TAUBnRDMm, and TAUBnRDC.TAUBnRDCM cannot be changed while the counter is in operation (TAUBnTE.TAUBnTEM = 1). 在计数器工作的时候，以上寄存器不能被改变  
TAUBnTOLm 只有在 PWM 输出功能或者三角 PWM 输出功能下才能被重新写入
- TAUBnTOL.TAUBnTOLM can only be rewritten during operation in PWM output function or triangle PWM output function. For all other output functions, TAUBnTOL.TAUBnTOLM must be written before the counter starts. If it is rewritten in another function mode, TAUBTTOUTM outputs an invalid wave.  
对于其他的输出功能，TAUBnTOLm 必须在计数器启动前被写入，否则 TAUBTTOUTM 会输出无效的波形
- 如果在upper channel上，同时复写被触发，则TAUBnRDCM bit控制所有低级的channel  
When a simultaneous rewrite trigger is issued on an upper channel (TAUBnRDS.TAUBnRDSM = 1), the TAUBnRDC.TAUBnRDCM bit controls all the lower channels. This means that if the TAUBnRDC.TAUBnRDCM bits of CH2 and CH7 are set to 1 and the TAUBnRDC.TAUBnRDCM bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.  
也就是说如果CH2和CH7的RDCm bit设置为1，并且其他channels的RDCm设置为0，CH2和CH7作为同时复写触发的产生信道，CH2控制CH3到CH6，CH7控制CH8到CH15
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger generation channel (TAUBnRDE.TAUBnRDEM and TAUBnRDS.TAUBnRDSM = 1) but no upper channel is set (TAUBnRDC.TAUBnRDC[15:0] = 0), simultaneous rewrite cannot take place.  
那么同时复写不能被执行

## 22.6.4 Types of Simultaneous Rewrite

In the following section, the three simultaneous rewrite methods are explained using timing diagrams.

### 22.6.4.1 Simultaneous Rewrite when the Master Channel (Re)Starts Counting (Method A)

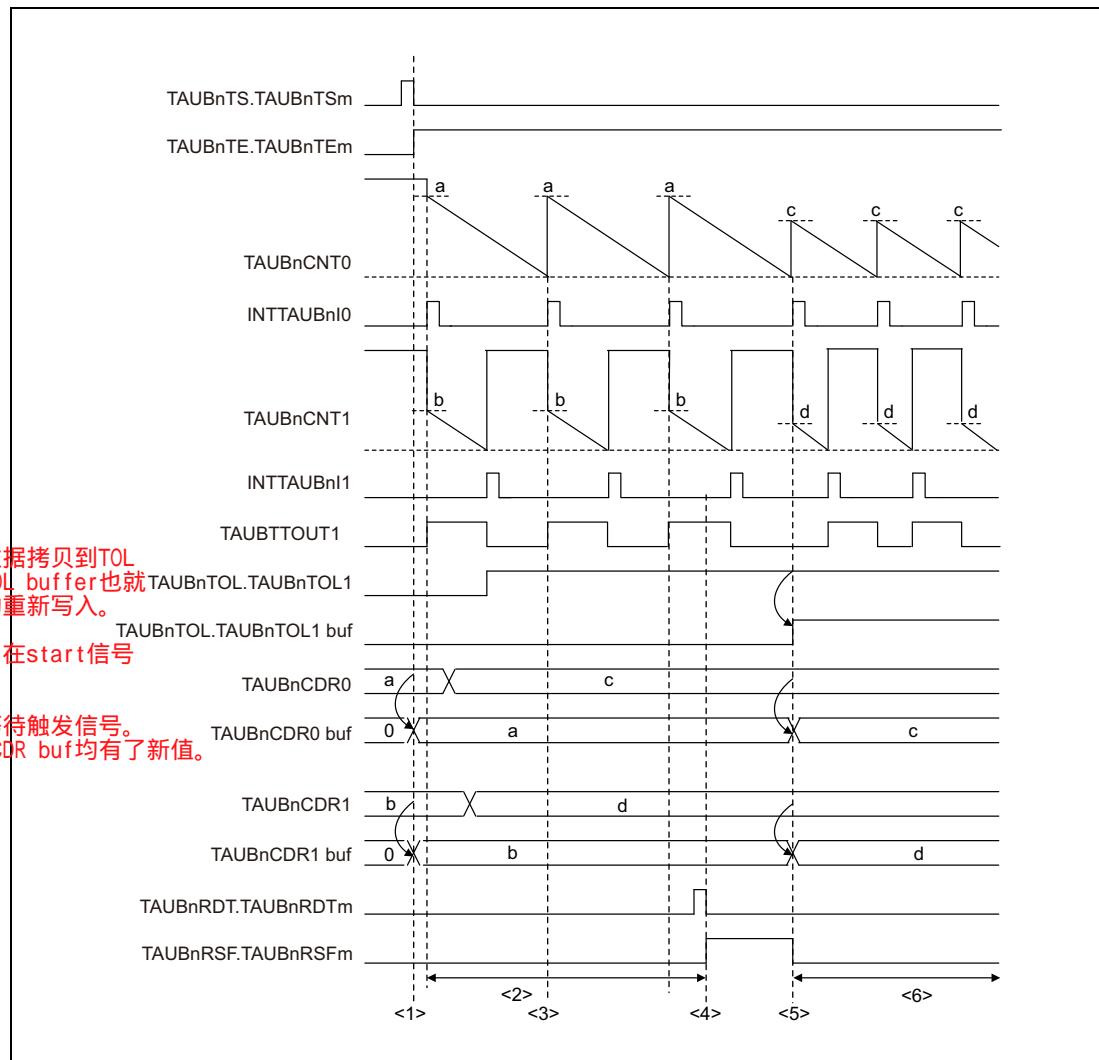


Figure 22.5 Simultaneous Rewrite when the Master Channel (Re)Starts Counting

#### Setting:

CH0是master channel , CH1待变slave channel

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel.

The simultaneous rewrite method A is applied.

应用同时复写A方法

#### Description:

- (1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer and the value of TAUBnTOL.TAUBnTOLm is copied to the TAUBnTOL.TAUBnTOLm buffer.  
CDR和TOL可以在任何时间写入
- (2) The TAUBnCDRm and TAUBnTOL.TAUBnTOLm registers can be written at any time.
- (3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0)

- (4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.
- 由于同时复写功能被使能，当CH0开始计数下降的时候开始触发
- (5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.
- (6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

三角波的顶点

#### 22.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel (Method B)

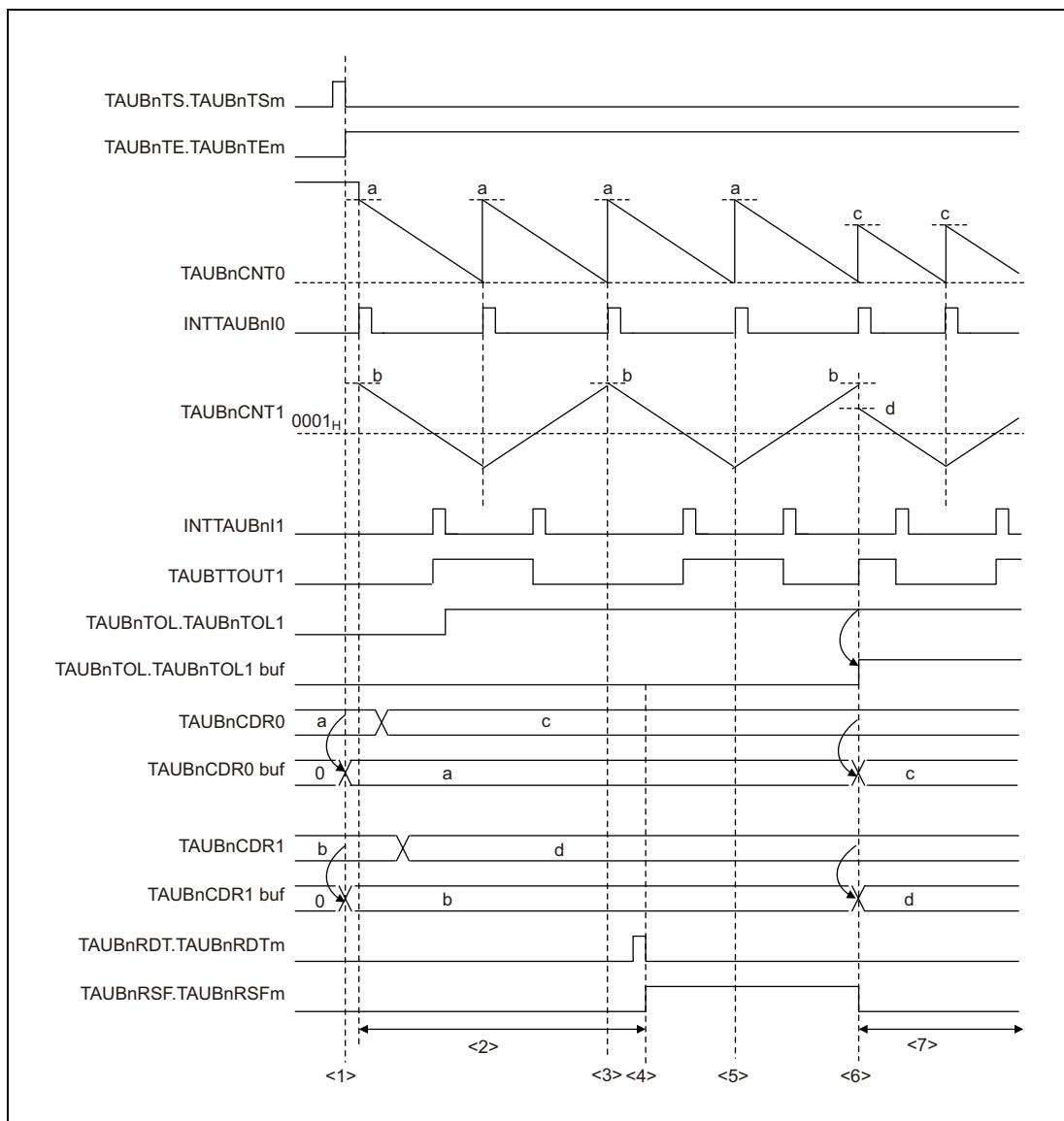


Figure 22.6 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel

#### Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

**Description:**

- (1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer.
- (2) The TAUBnCDRm and TAUBnTOL registers can be written at any time.
- (3) Simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0).
- (4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.
- (5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.
- (6) Simultaneous rewrite takes place at the start timing of the top of the triangular cycle. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.
- (7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

### 22.6.4.3 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm (Method C1)

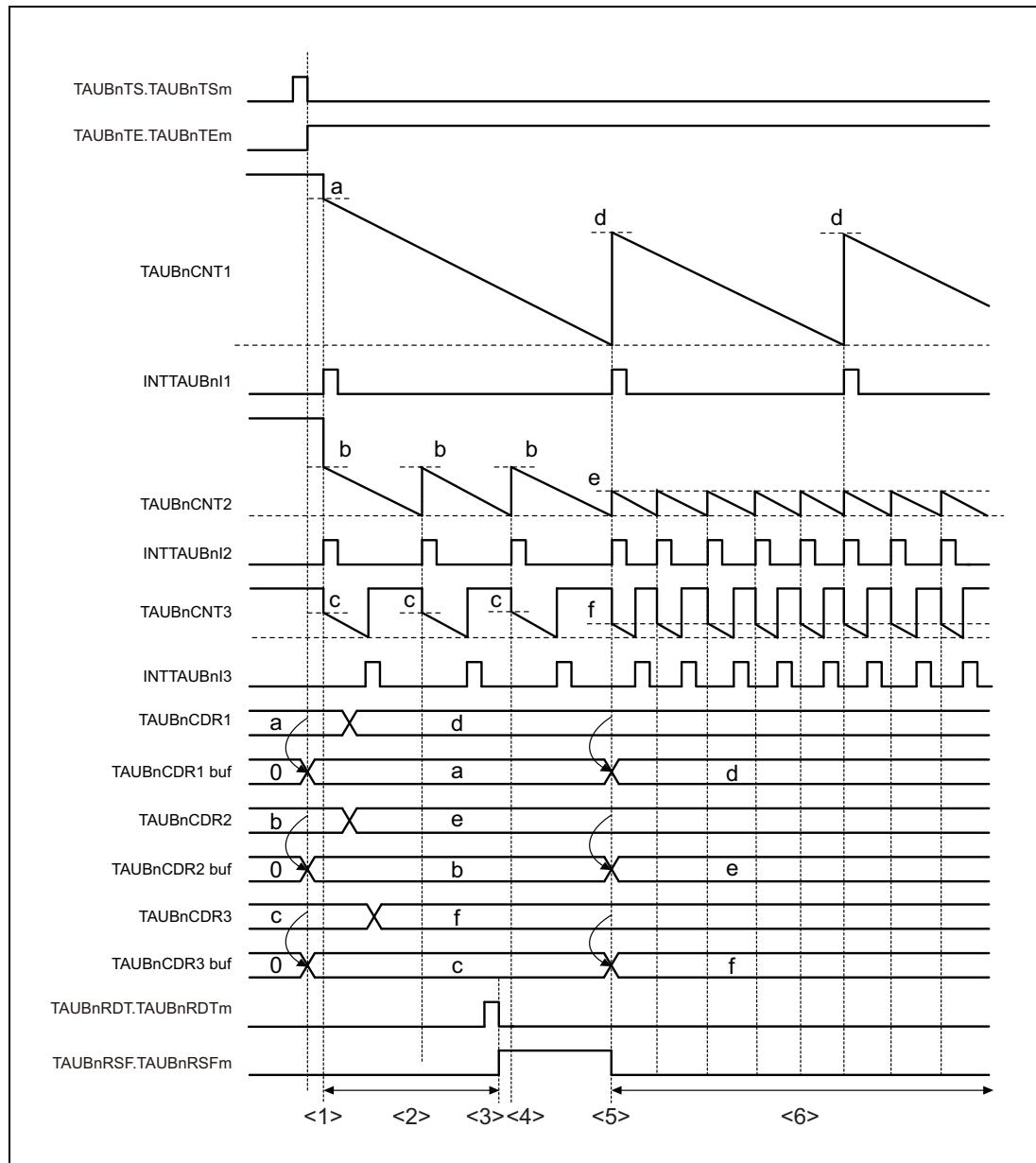


Figure 22.7 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm

**Setting:**

CH1作为高阶channel , CH2作为masterchannel , CH3作为slave channel

CH1 is an upper channel used counting down, CH2 is a master channel, and CH3 is the slave channel.

The simultaneous rewrite method C1 is applied. The TAUBnRDC register specifies a channel which generates simultaneous rewrite triggers.

这就解释了为什么CH1可以触发

**Description:**

- (1) When TAUBnTS.TAUBnTSm is set to 1, the TAUBnCDRm value is copied to the TAUBnCDRm buffer.
- (2) The TAUBnCDRm register is always ready to write.
- (3) By setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1, the status flag is set (TAUBnRSF.TAUBnRSFm = 1) to enable simultaneous rewrite.
- (4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.
- (5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000<sub>H</sub>. The TAUBnCDRm values are loaded into the corresponding TAUBnCDRm buffers.
- (6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUBnCDRm register can be recharged.

## 22.7 Channel Output Modes

**TOUT pin的输出可以用两种方式控制，后一种模式可以分为几个单独的模式**

The output of the TAUBTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

**由软件控制 (TOE = 0)**

- By software (TAUBnTOE.TAUBnTOEm = 0)

When controlled by software, the value written in the output register bit (TAUBnTO.TAUBnTOm) is sent to the output pin (TAUBTTOUTm).

**由TAUB信号控制 (TOE=1)**

- By TAUB signals (TAUBnTOE.TAUBnTOEm = 1)

当由TAUB信号控制的时候，TOUT的输出 由内部信号set, reset或者 toggle。 When controlled by TAUB signals, the output level of TAUBTTOUTm is set or reset or toggled by internal signals. The value of TAUBnTO.TAUBnTOm is updated accordingly to reflect the value of TAUBTTOUTm. **由TOM来读取TOUTm的值**

- Independently (TAUBnTOM.TAUBnTOMm = 0) **独立的**

在独立操作模式下，TOUTm的输出只是被channel m自身的设置所影响 In case of independent operation, the output of the TAUBTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUBnTOM.TAUBnTOMm = 0).

- Synchronously (TAUBnTOM.TAUBnTOMm = 1) **同步的**

在同步操作模式下，TOUTm的输出受到channel m的设置和其他channel的影响 因此同步channel 操作模式应该被所有的channel选择 be selected for all synchronized channels (TAUBnTOM.TAUBnTOMm = 1).

**由TOMbit来读取TOUTm的值，不管这个pin是什么来控制**

The TAUBnTO.TAUBnTOm bit can always be read to determine the current value of TAUBTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

### Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 22.38, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 22.7.2, Channel Output Modes Controlled Independently by TAUBn Signals**
- **Section 22.7.3, Channel Output Modes Controlled Synchronously by TAUBn Signals**

### Batch operation of TAUBnTOm bit

Whether a set value is reflected to the TAUBnTOm bit or not is controlled by the TAUBnTOE.TAUBnTOEm bit. **设置的value是否反映到TOM bit 由TOEm bit 来控制**

**只有在TOEm bit = 0的情况下 TOM才能被写入**  
The TAUBnTOm setting is written only to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 0 when a write to the TAUBnTO register is attempted. No TAUBnTOm setting is reflected to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 1.

### NOTE

The TAUBnTO.TAUBnTOm bit is placed so that its bit number corresponds to a channel number.

### Output logic

Positive logic or negative logic of the output is specified by control bit TAUBnTOL.TAUBnTOLm.

**TOLm必须在计数器启动前被设置**  
 The value of TAUBnTOL.TAUBnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. If TAUBnTOL.TAUBnTOLm is changed after the counter starts, the output of TAUBTTOUTm is undefined.

See **Section 22.6, Simultaneous Rewrite.**

The various channel output modes and the channel output control bits are listed in **Table 22.38.**

**Table 22.38 Channel Output Modes**

Channel Output Mode	TAUBnTOE. TAUBnTOEm	TAUBnTOM. TAUBnTOMm	TAUBnTOC. TAUBnTOCm	TAUBnTDE. TAUBnTDEM
<b>By software</b>				
Independent channel output mode controlled by software	0		x	
<b>By TAUB signals, independently</b>				
Independent channel output mode 1	1	0	0	0
Independent channel output mode 2			1	
<b>By TAUB signals, synchronously</b>				
Synchronous channel output mode 1	1	1	0	0
Synchronous channel output mode 2			1	0
Synchronous channel output mode 2 with dead time output				1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

#### NOTE

The following bits cannot be changed during count operation (TAUBnTE.TAUBnTEm = 1):

- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEM

**这些寄存器在计数器正常工作的情况下是不能被改变的**

### 22.7.1 General Procedures for Specifying a Channel Output Mode

**此节介绍用于指定TO<sub>UTm</sub> channel 输出模式的一般流程**

This section describes the general procedures for specifying a TAUBTOUT<sub>m</sub> channel output mode.

前提是定时器的输出操作被禁了  
The prerequisite is that timer output operation is disabled (TAUBnTOE.TAUBnTOEm = 0).

**指定TO<sub>UTm</sub>的输出的初始电平**

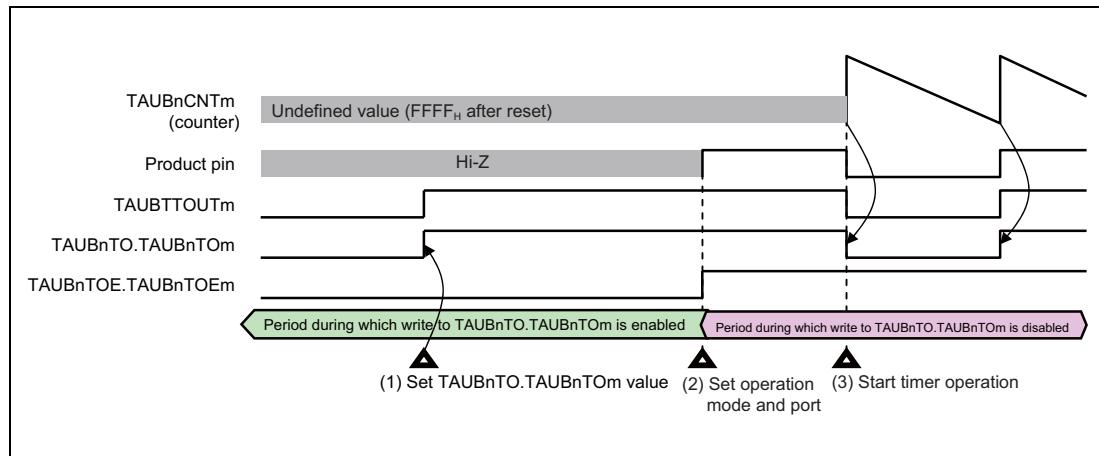
(1) Set TAUBnTO.TAUBnTOm to specify the initial level of the TAUBTOUT<sub>m</sub> output.

**设置输出模式，使用TOL<sub>m</sub> bit设置输出逻辑电平**

(2) Set channel output mode according to **Table 22.38, Channel Output Modes**, and the output logic using the TAUBnTOL.TAUBnTOLm bit.

**启动计数器**

(3) Start the counter (TAUBnTS.TAUBnTSm = 1).



**Figure 22.8 General Procedure for Specifying a TAUBTOUT<sub>m</sub> Channel Output Mode**

## 22.7.2 Channel Output Modes Controlled Independently by TAUBn Signals

This section lists the channel output modes that are controlled independently by TAUBn signals. The control bits used to specify a mode are listed in **Table 22.38, Channel Output Modes**.

### 22.7.2.1 独立channel输出模式1 Independent Channel Output Mode 1

#### Set/reset conditions

**在此输出模式下，当INTTAUBnIm被检测到，TAUBTOUTm会toggle。TOLm的值会被忽略。**  
In this output mode, TAUBTOUTm toggles when INTTAUBnIm is detected. The value of TAUBnTOL.TAUBnTOLm is ignored.

#### Prerequisites

除了Table 22.38中的意外没有其他的前提条件。  
There are no prerequisites other than those shown in **Table 22.38, Channel Output Modes**.

#### 独立channel 输出模式 2

### 22.7.2.2 Independent Channel Output Mode 2

#### Set/reset conditions

**在此模式下，INTTAUBnIm发生的时候，TAUBTOUT被置位，一旦CNTm和CDRm相匹配也会导致INTTAUBnIm中断**  
In this output mode, TAUBTOUTm is set when INTTAUBnIm occurs at the time of count start, and reset when INTTAUBnIm occurs due to a match between TAUBnCNTm and TAUBnCDRm.  
**从而使得TOUT reset**

#### Prerequisites

There are no prerequisites other than those shown in **Table 22.38, Channel Output Modes**.

## 22.7.3 Channel Output Modes Controlled Synchronously by TAUBn Signals

This section lists the channel output modes that are controlled synchronously by TAUBn signals. The control bits used to specify a mode are listed in **Table 22.38, Channel Output Modes**.

### 22.7.3.1 Synchronous Channel Output Mode 1

#### Set/reset conditions

**在此模式下，master的INTTAUBnIm作为set 信号，slave的作为reset信号**  
In this output mode, INTTAUBnIm of master channel serves as a set signal and INTTAUBnIm of the slave channel as a reset signal. If INTTAUBnIm of the master channel and INTTAUBnIm of the slave channel are generated at the same time, INTTAUBnIm of the slave channel (reset signal) has priority over INTTAUBnIm (set signal) of the master channel, i.e., the master channel is ignored.



#### Prerequisites

There are no prerequisites other than those shown in **Table 22.38, Channel Output Modes**.

### 22.7.3.2 Synchronous Channel Output Mode 2

**在这种输出模式，工作模式应该被设置为count-up/down mode**

**结果导致TAUBTOUTm的三角波PWM输出**

In this output mode, the operating mode should be set to count-up/-down mode. The result is triangular wave PWM output at TAUBTOUTm. For details, see **Section 22.14.5, Triangle PWM Output Function**.

#### Set/reset conditions

**slave channel的CNT count up 和都玩腻士可选的，当其达到0001就会产生中断，导致TOUT toggle**  
TAUBnCNTm of the slave channel counts down and up alternatively. When it passes 0001H it generates an interrupt, causing TAUBTOUTm to toggle.

### Prerequisites

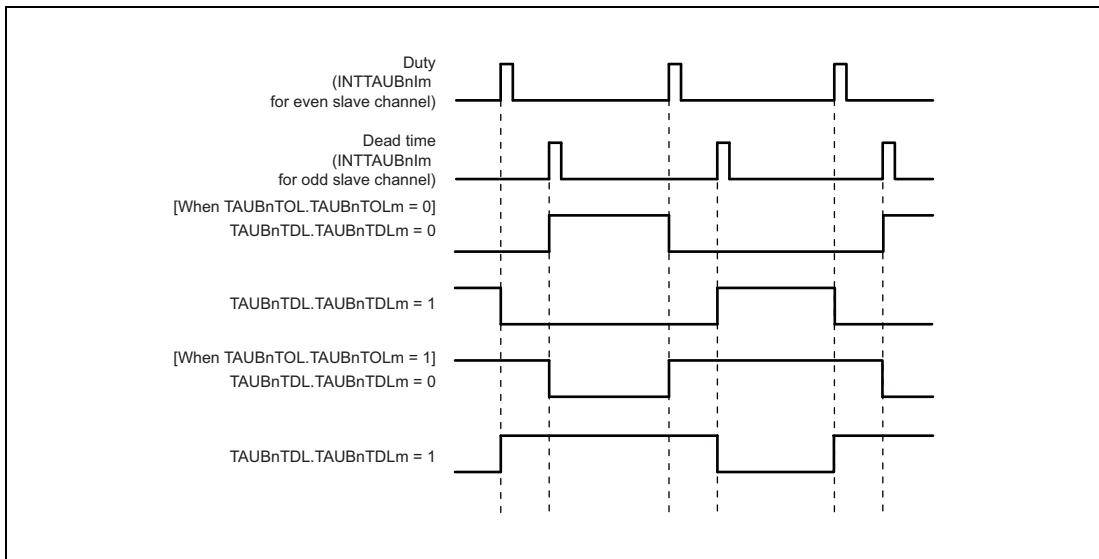
A set of two channels is required to generate the triangle PWM output. TAUBTOUTm should be set to 0 before the function starts.

#### 22.7.3.3 Synchronous Channel Output Mode 2 with Dead Time Output

*在此模式下，Dead time 延迟会被添加到TAUBTOUTm。*

In this output mode, a dead time delay is added to TAUBTOUTm. The set/reset conditions are shown in **Figure 22.9**.

#### Set/reset conditions



**Figure 22.9 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output**

With regard to the edge to which dead time is added, set TAUBnTDL.TAUBnTDLm = 0 for rising edges and TAUBnTDL.TAUBnTDLm = 1 for falling edges.

### Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel  
The master channel should be set to interval timer mode.
- One even slave channel  
The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)  
The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUBnTOE.TAUBnTOEm
- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEM

每种模式下的Start Timing

## 22.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUBnTS.TAUBnTSm is set to 1 in each operating mode.

**在所有模式中，data register的数据和中断是否产生取决于mode和register的设置**  
In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

### CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

### 22.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating at the next count clock after TAUBnTS.TAUBnTSm is set to 1. The value of data register is also loaded when the counter starts.

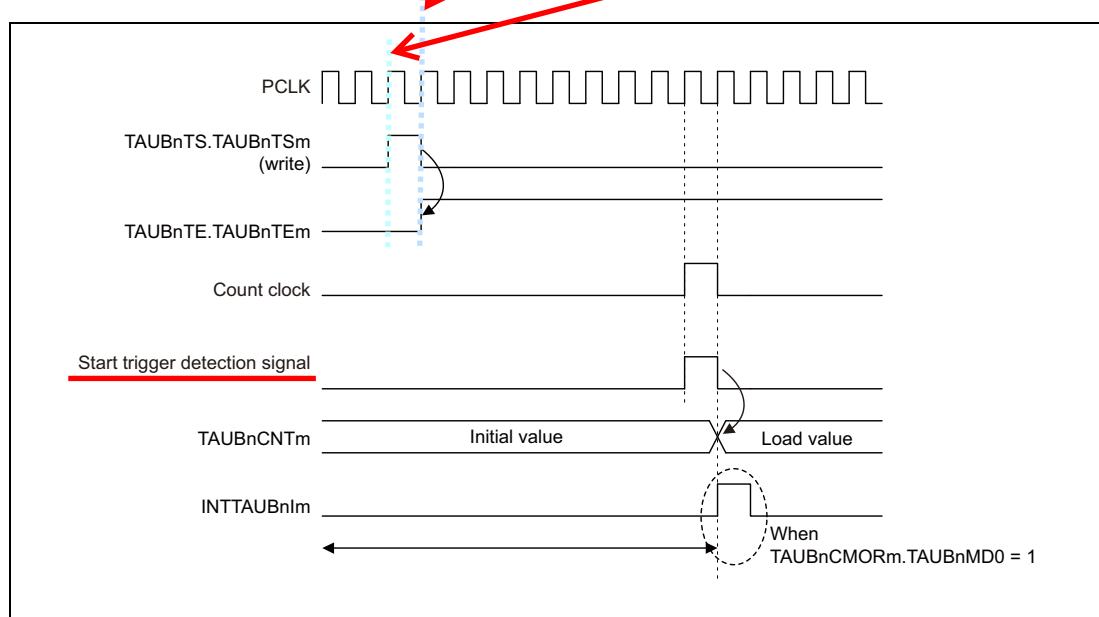


Figure 22.10 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Up/Down Count Mode, and Count Capture Mode

### NOTE

Make sure to set TAUBnCMORm.TAUBnMD0 to 0 when using the count-up/-down mode.

## 22.8.2 Event Count Mode

The value of data register is loaded as soon as TAUBnTS.TAUBnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

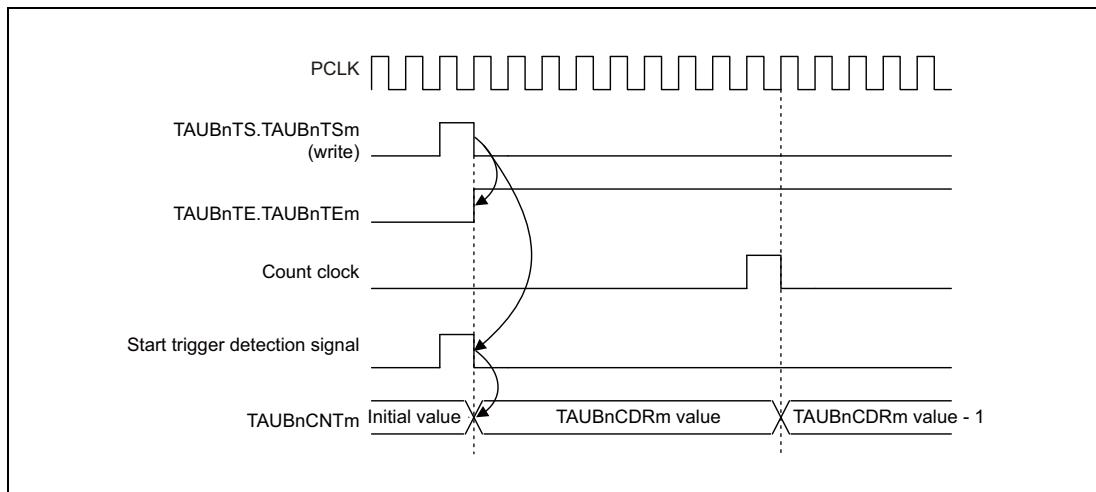


Figure 22.11 Start Timing in Event Count Mode

## 22.8.3 Other Operating Modes

**在其他模式下，计数器操作的开始时机只依赖于当TAUBTTINm有效沿信号检测到的时候。**  
 In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUBTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place. **计数时钟周期，与开始计数器操作无关，确定所有操作发生的频率。**

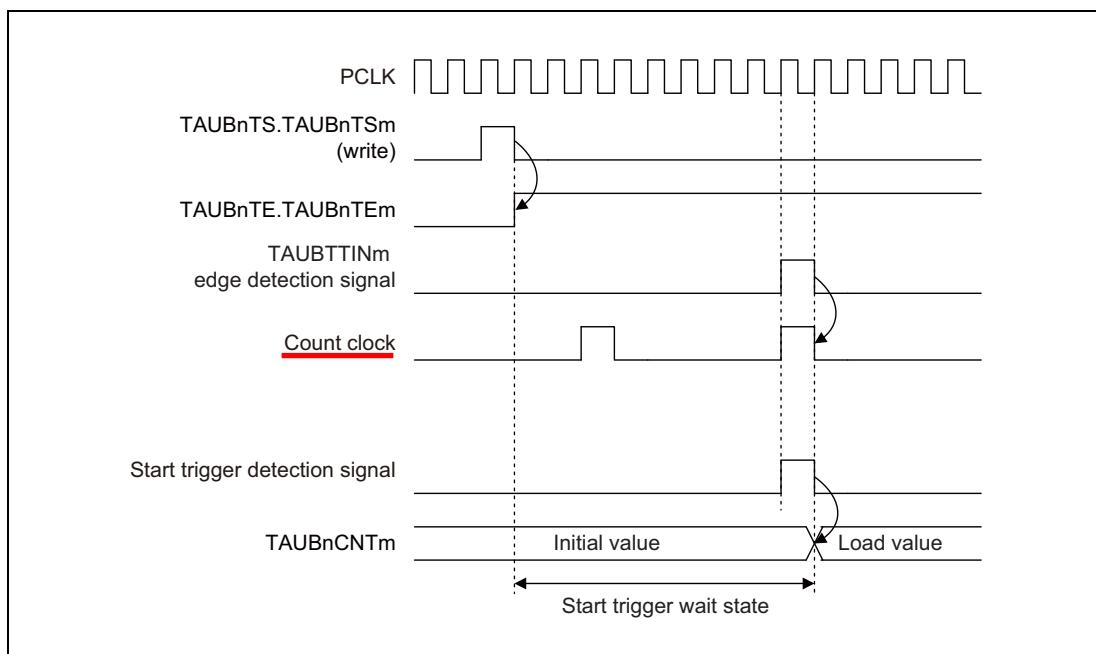


Figure 22.12 Count Start Timing in Other Operating Modes

## 22.9 TAUBTTOUTm Output and INTTaubnIm Generation when Counter Starts or Restarts

当计数器启动的时候，使用MD0bit 可以指定INTTaubnIm是否产生中断。

When the counter starts, it is possible to specify whether an INTTaubnIm is generated using the INTTaubnIm的产生和对TOUT的影响取决于所选的功能。

TAUBnCMORm.TAUBnMD0 bit starts counting and the effect to TAUBTTOUTm depend on the selected function. For details, refer to the description of TAUBnCMORm.TAUBnMD0 of each function.

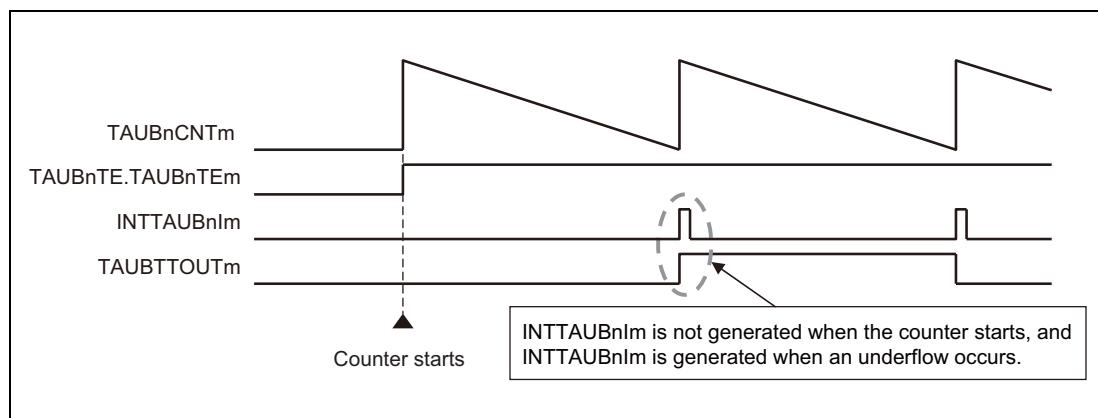


Figure 22.13 INTTaubnIm Generation Timing (TAUBnCMORm.TAUBnMD0=0)

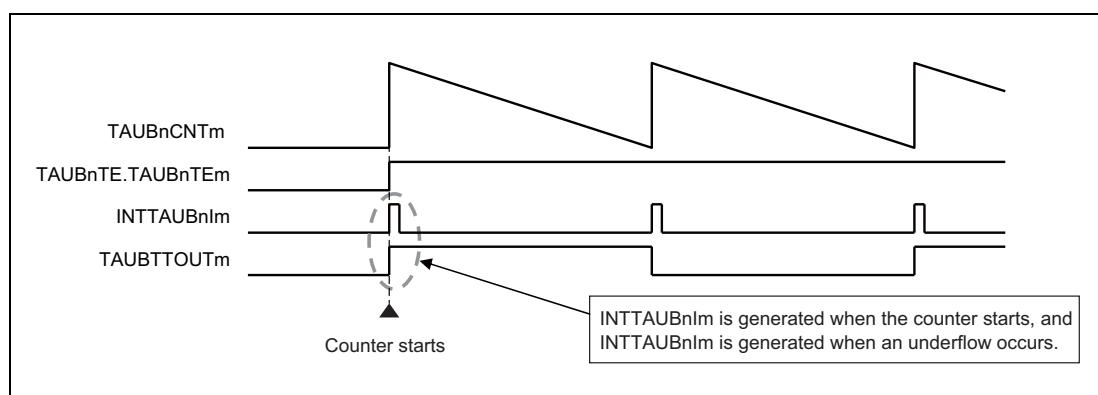


Figure 22.14 INTTaubnIm Generation Timing (TAUBnCMORm.TAUBnMD0=1)

## 22.10 Interrupt Generation upon Overflow

**在一些独立的功能中，当计数器达到0xFFFF并且增加到溢出时，中断不会产生。**

In certain independent functions, an interrupt is not generated when the counter value reaches  $\text{FFFF}_{\text{H}}$  and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

**对于第二个channel合适的操作模式依赖于第一个channel的工作模式，这条原则适用于所有的组合**  
The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches  $0000_{\text{H}}$  at the same time as the first channel overflows ( $\text{TAUBnCNTm} = \text{FFFF}_{\text{H}}$ ).
- Set  $\text{TAUBnCDRm}$  of the second channel to  $\text{FFFF}_{\text{H}}$ .
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same  $\text{TAUBTTINm}$  input.
- The trigger detection settings ( $\text{TAUBnCMORm.TAUBnSTS[2:0]}$  and  $\text{TAUBnCMURm.TAUBnTIS[1:0]}$ ) must be identical for both channels.

**Result: 结果**

**第二个channel的down-counter 达到0000H，此时第一个channel的up-counter也溢出了。因此第二个channel产生预期的中断**  
The down-counter of the second channel reaches  $0000_{\text{H}}$  at exactly the same time as the up-counter of the first channel overflows ( $\text{TAUBnCNTm} = \text{FFFF}_{\text{H}}$ ). Thus the second channel generates the desired interrupt.

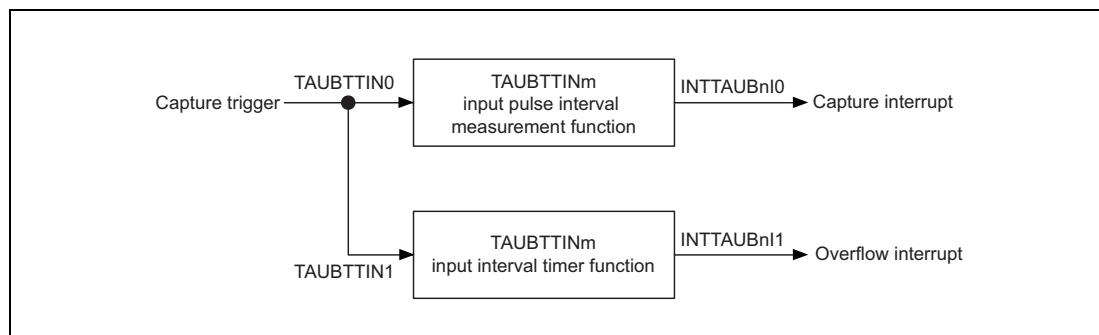
**以下部分列举了具有count down的操作模式需要匹配具有count-up的模式**  
The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

### 22.10.1 Example of Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

当capture trigger 被同时输入两个channel TAUBTTINm , TINm input timer 功能的INTTAUBnIm  
When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of  
the TAUBTTINm input interval timer function can detect the overflow when TAUBnCNTm of the

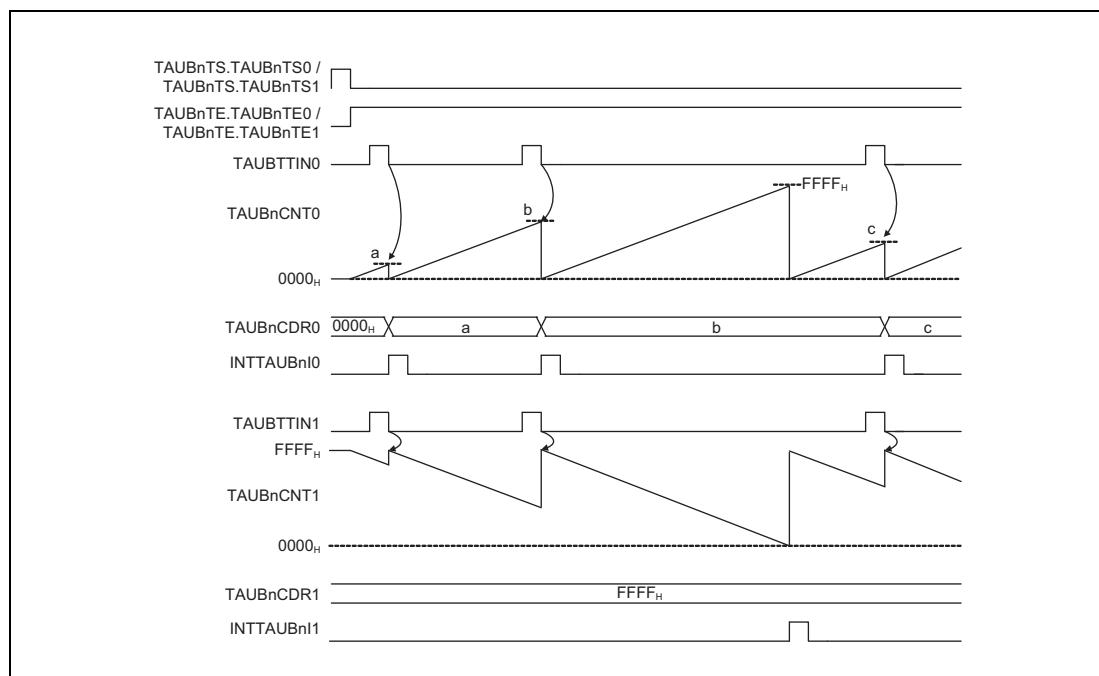
TAUBTTINm input pulse interval measurement function exceeds FFFF<sub>H</sub>.

可以探测到overflow, 此时的TAUBTTINm input pulse interval measurement 功能的CNTm超过FFFF



**Figure 22.15 Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function**

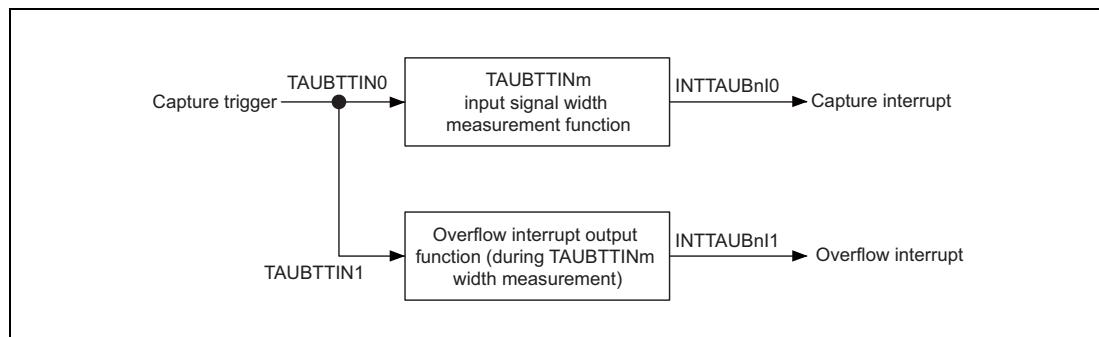
#### Timing diagram



**Figure 22.16 Interrupt Generation by Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function**

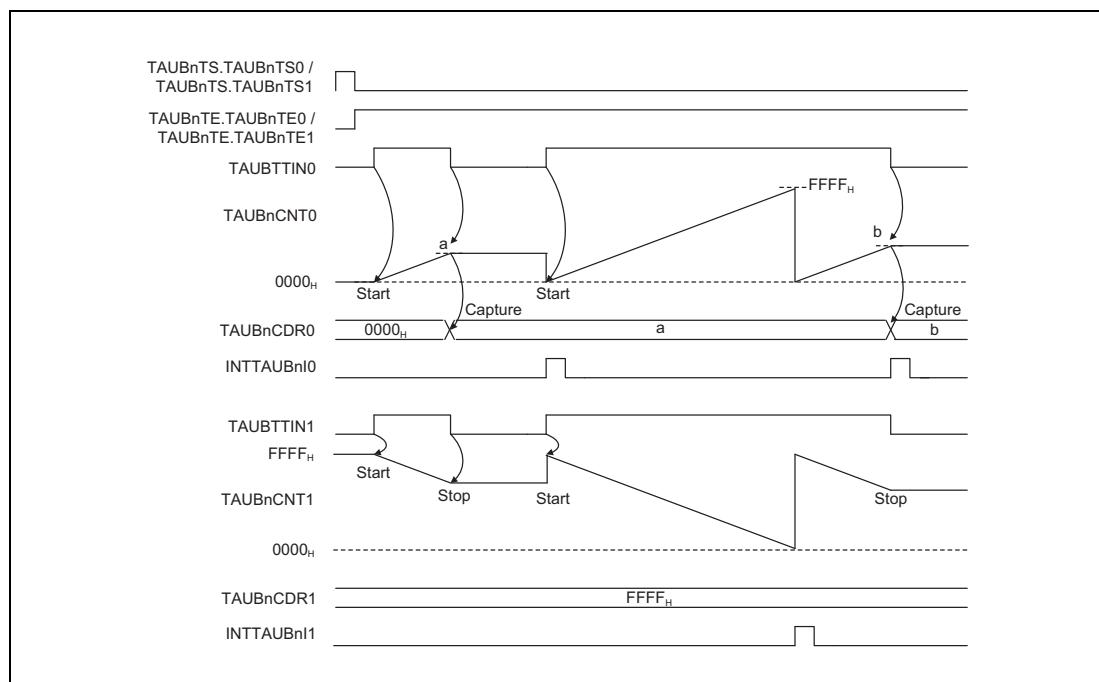
## 22.10.2 Example of Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm width measurement) can detect the overflow when TAUBnCNTm of the TAUBTTINm input signal width measurement function exceeds  $FFFF_H$ .



**Figure 22.17 Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)**

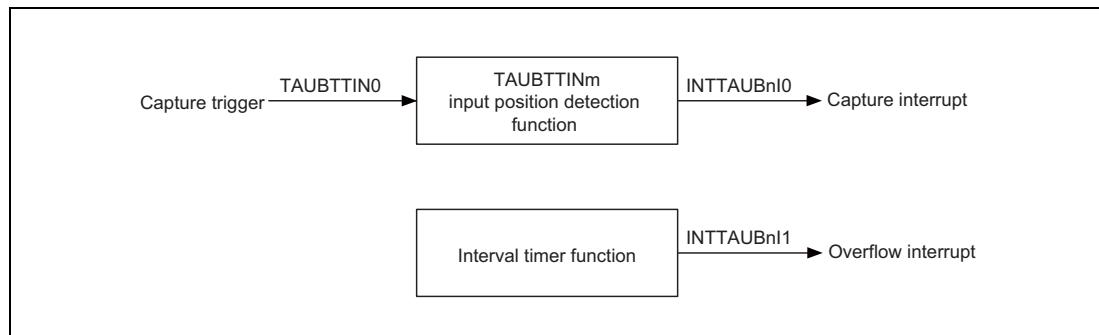
### Timing diagram



**Figure 22.18 Interrupt Generation by Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)**

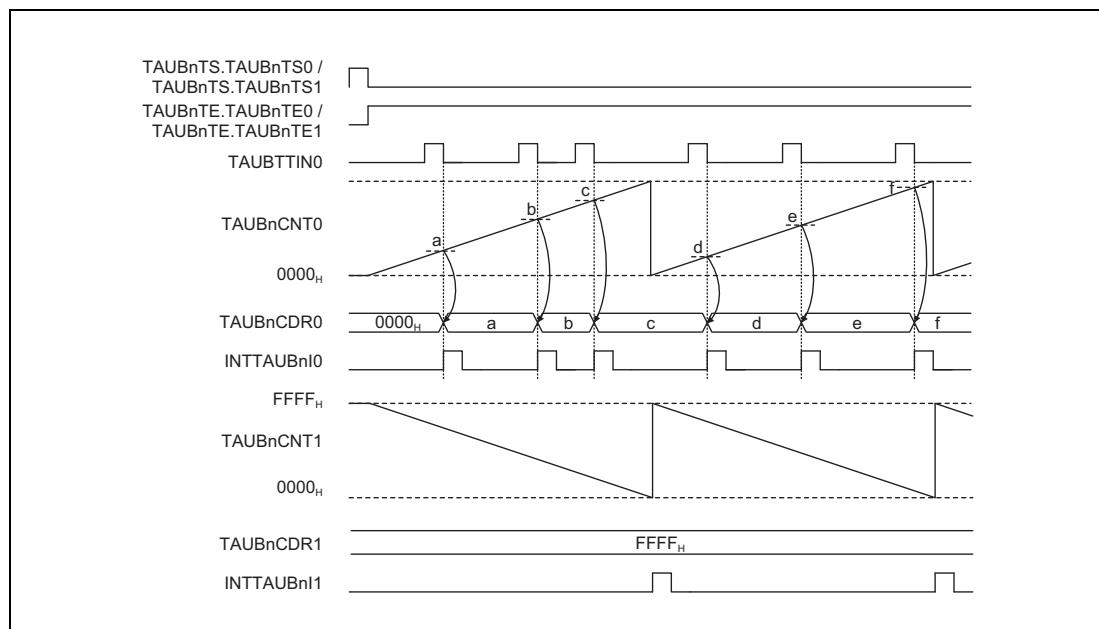
### 22.10.3 Example of Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUBnIm of the interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input position detection function exceeds FFFF<sub>H</sub>.



**Figure 22.19 Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function**

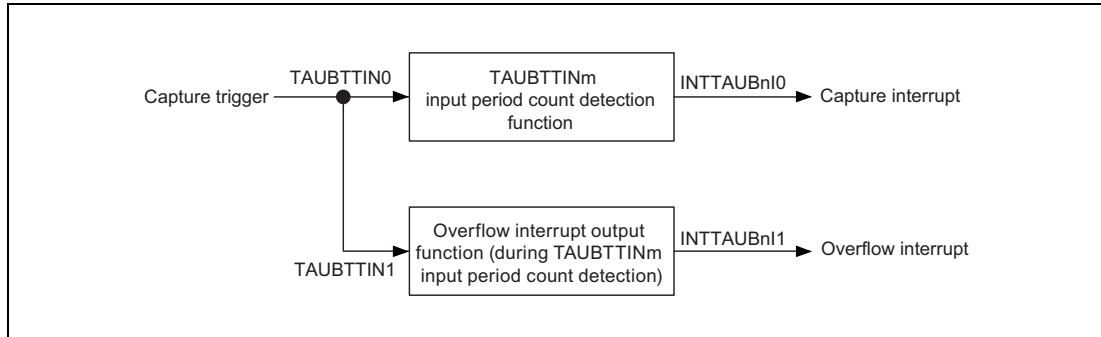
#### Timing diagram



**Figure 22.20 Interrupt Generation by Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function**

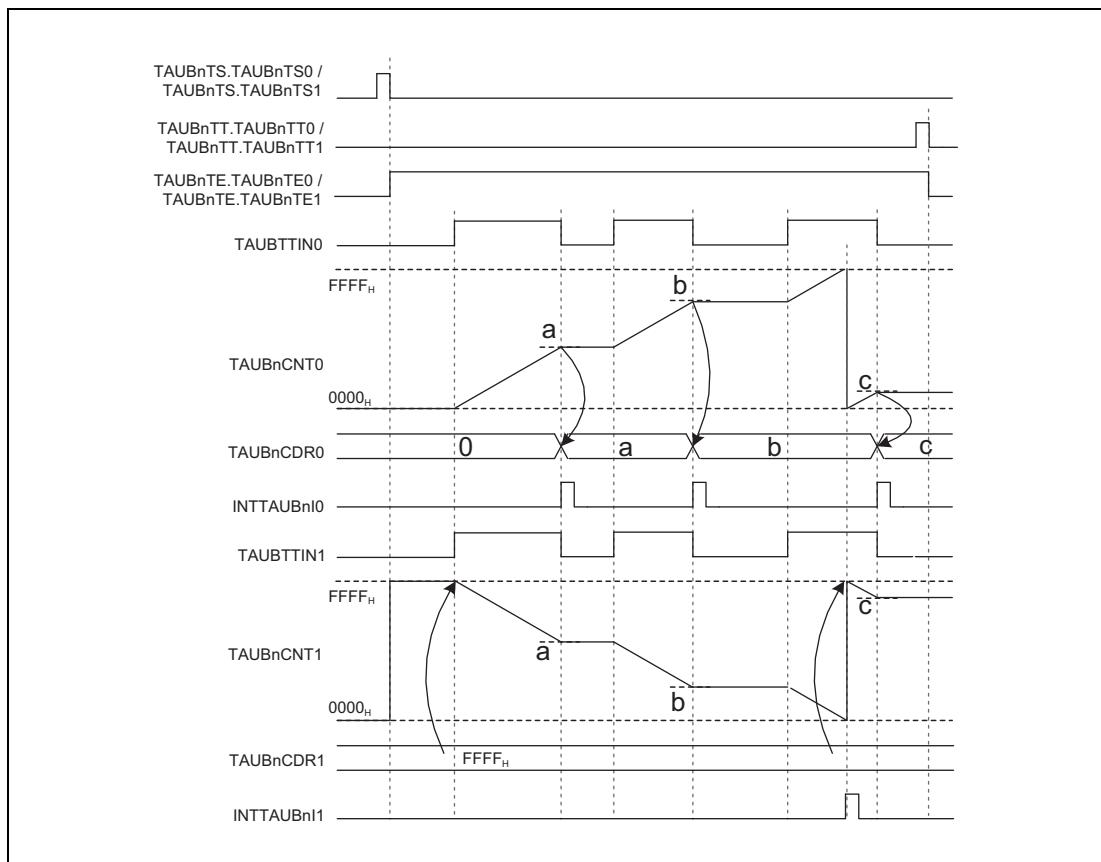
#### 22.10.4 Example of Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm input period count detection) can detect the overflow when TAUBnCNTm of the TAUBTTINm input period count detection function exceeds  $FFFF_H$ .



**Figure 22.21 Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (TAUBTTINm Input Period Count Detection)**

#### Timing diagram



**Figure 22.22 Interrupt Generation by Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)**

### 边沿检测

## 22.11 TAUBTTINm Edge Detection

边沿探测基于操作时钟，也就是说一个沿只能在下一次的时钟的上升沿被探测到。Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

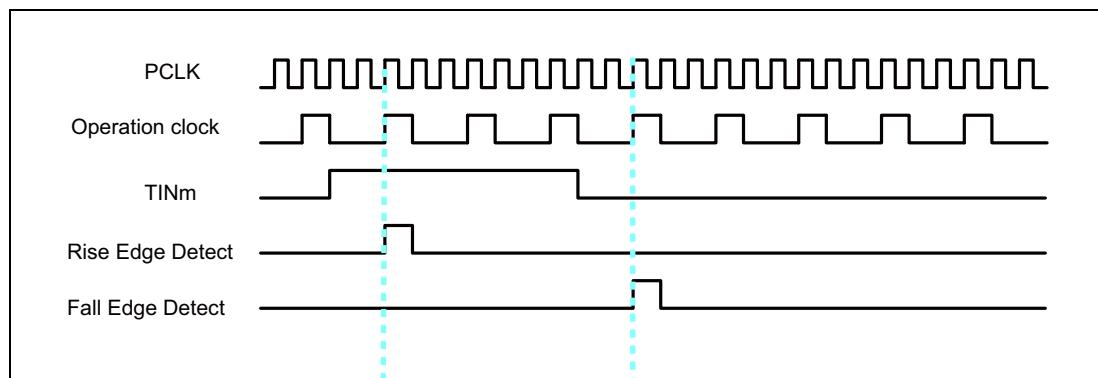


Figure 22.23 Basic Edge Detection Timing

**Figure 22.23** is an image of the operation timing. Actually, the delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn will be generated.

## 22.12 Independent Channel Operation Functions

以下部分由TAUB提供的独立channel 的操作模式

The following sections list the independent channel operation functions provided by the TAUB. For a general overview of independent channel operation functions, see **Section 22.2, Overview**.

### 22.12.1 Interval Timer Function

#### 22.12.1.1 Overview

##### Summary

此功能在用作隔一定的间隔产生INTTAUBnIm的定时器中断

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

一旦中断产生，TOUTm信号开始toggle，导致方波产生

##### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

如果counter 达到0x0000, INTTAUBnIm产生并且TAUBTTOUTm signal toggles,CNTm装载CDRm然后继续操作  
When the counter reaches 0000<sub>H</sub>, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles.

TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

CDRm可以在任何时候被重写，并且所改变的值在下次计数器开始下减的时候生效  
The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEM to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.  
在计数器运行过程中，可以被强制重启（设置TSM = 1）

##### Conditions

如果MD0 bit被设置为0启动或者重启后的第一个中断是不发生的，因此TAUBTTOUT不能toggle

If the TAUBnCMORM.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle. This results in an inverted TAUBTTOUTm signal output when TAUBnCMORM.TAUBnMD0 is set to 1. 结果导致TOUTm信号输出的反转（当MD0设置为1）

#### 22.12.1.2 Equations 公式:

$$\text{INTTAUBnIm cycle} = \text{count clock cycle} \times (\text{TAUBnCDRm} + 1)$$

$$\text{TAUBTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUBnCDRm} + 1) \times 2$$

### 22.12.1.3 Block Diagram and General Timing Diagram

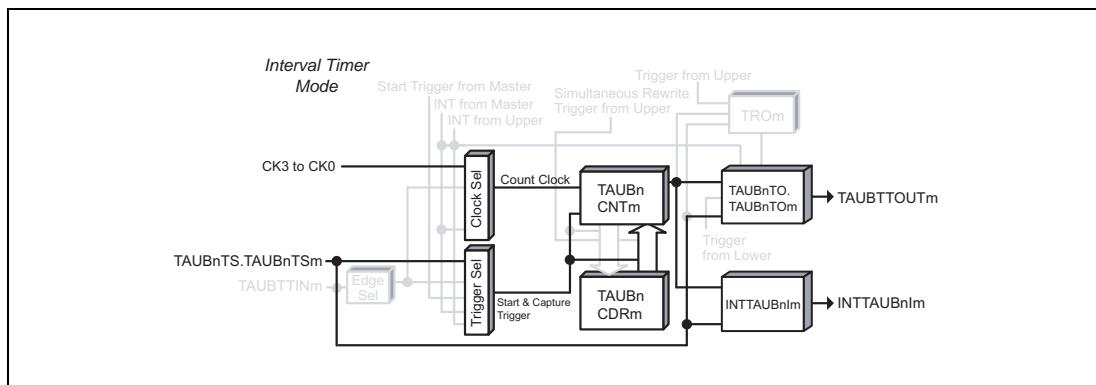


Figure 22.24 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnlm is generated at operation start ( $\text{TAUBnCMORm.TAUBnMD0} = 1$ )

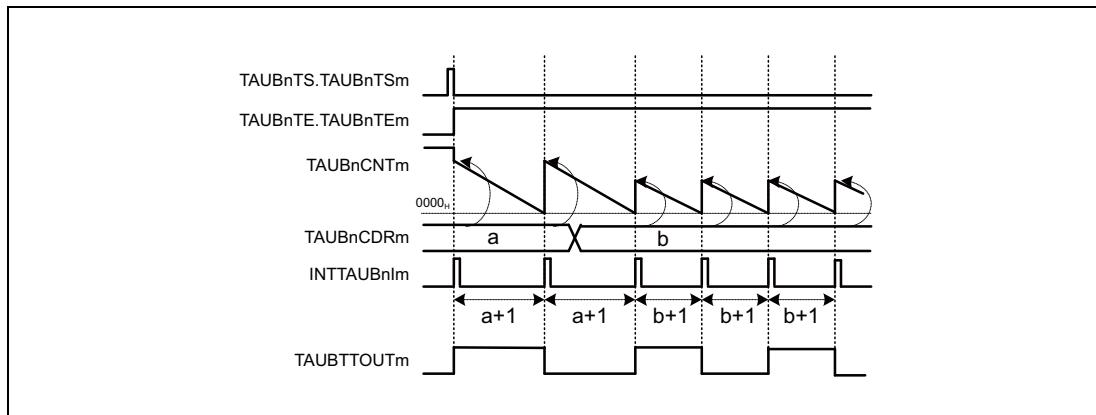


Figure 22.25 General Timing Diagram for Interval Timer Function

### 22.12.1.4 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.39 Contents of the TAUBnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnIm is not generated and TAUBTTOUTm does not toggle at operation start. 1: INTTAUBnIm is generated and TAUBTTOUTm toggles at operation start or restart.

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.40 Contents of the TAUBnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode

Table 22.41 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 0 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 0 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	Write 0 <sub>B</sub> .
TAUBnTDE.TAUBnTDEm	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

#### NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTOUTm can then be controlled independently of the interrupts.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 22.42 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

### 22.12.1.5 Operating Procedure for Interval Timer Function

Table 22.43 Operating Procedure for Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	<p>Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.39, Contents of the TAUBnCMORm Register for Interval Timer Function</b> and <b>Table 22.40, Contents of the TAUBnCMURm Register for Interval Timer Function</b></p> <p>Set the value of the TAUBnCDRm register</p> <p>Set the channel output mode by setting the control bits as described in <b>Table 22.41, Control Bit Settings for Independent Channel Output Mode 1</b></p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnlm is generated and TAUBTTOUTm toggles.</p>
During operation	<p>The TAUBnCDRm register value can be changed at any time. The TAUBnCNTm register can be read at all times.</p>	<p>TAUBnCNTm counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>• TAUBnCNTm reloads the TAUBnCDRm value and continues count operation</li> <li>• INTTAUBnlm is generated and TAUBTTOUTm toggles.</li> </ul>
Stop operation	<p>Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.</p>

Restart operation



### 22.12.1.6 Specific Timing Diagrams

#### (1) TAUBnCDRm = 0000<sub>H</sub>, count clock = PCLK/2

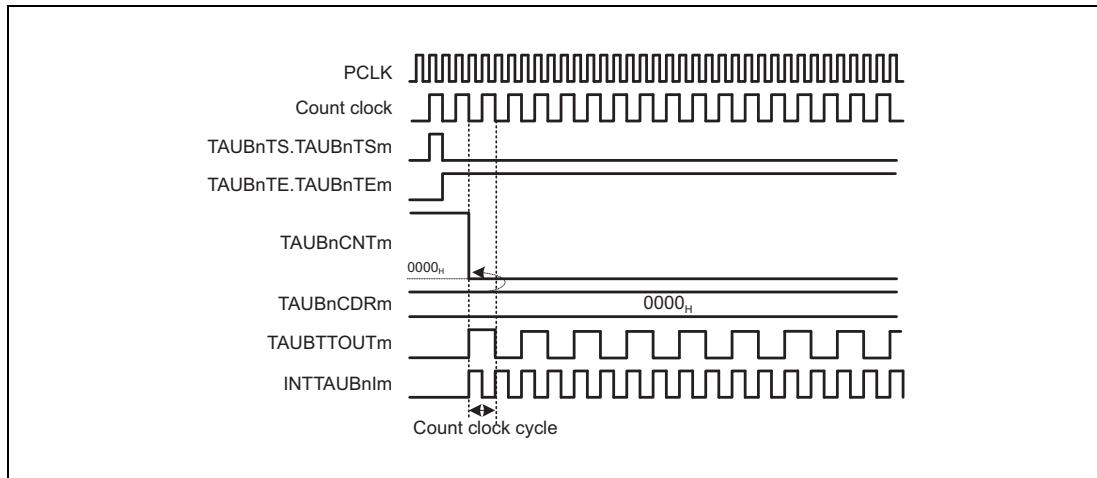


Figure 22.26 TAUBnCDRm = 0000<sub>H</sub>, Count Clock = PCLK/2

- TAUBnCDRm = 0000<sub>H</sub>, and the count clock = PCLK/2, the TAUBnCDRm value is written to TAUBnCNTm every count clock, meaning that TAUBnCNTm is always 0000<sub>H</sub>.
- INTTAUBnIm is generated every count clock, resulting in TAUBTOUTm toggling every count clock.

#### (2) TAUBnCDRm = 0000<sub>H</sub>, count clock = PCLK

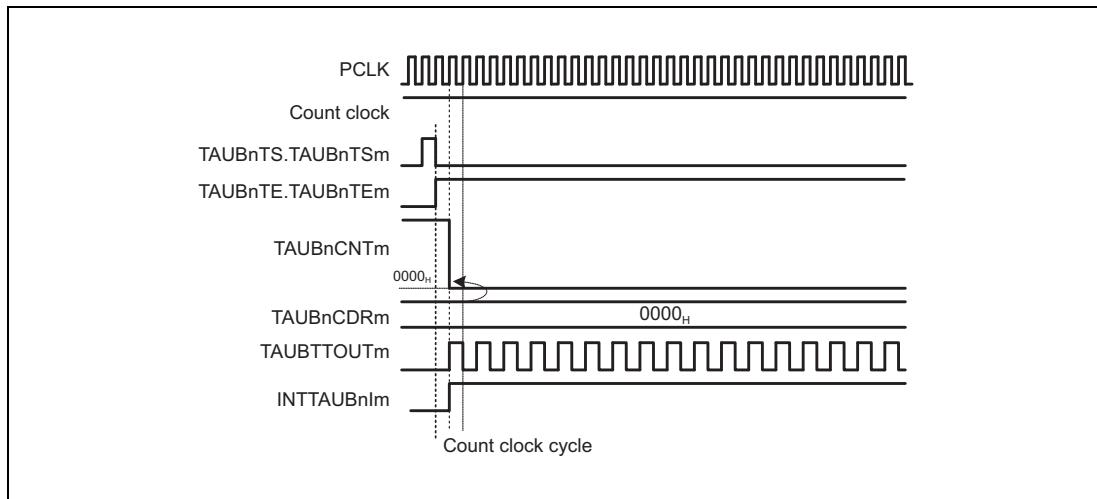


Figure 22.27 TAUBnCDRm = 0000<sub>H</sub>, Count Clock = PCLK

- TAUBnCDRm = 0000<sub>H</sub>, and the count clock = PCLK, the TAUBnCDRm value is written to TAUBnCNTm every PCLK clock, meaning that TAUBnCNTm is always 0000<sub>H</sub>.
- INTTAUBnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
- TAUBTOUTm is toggled every PCLK clock.

### (3) Operation stop and restart (TAUBnCMORm TAUBnMD0 = 1)

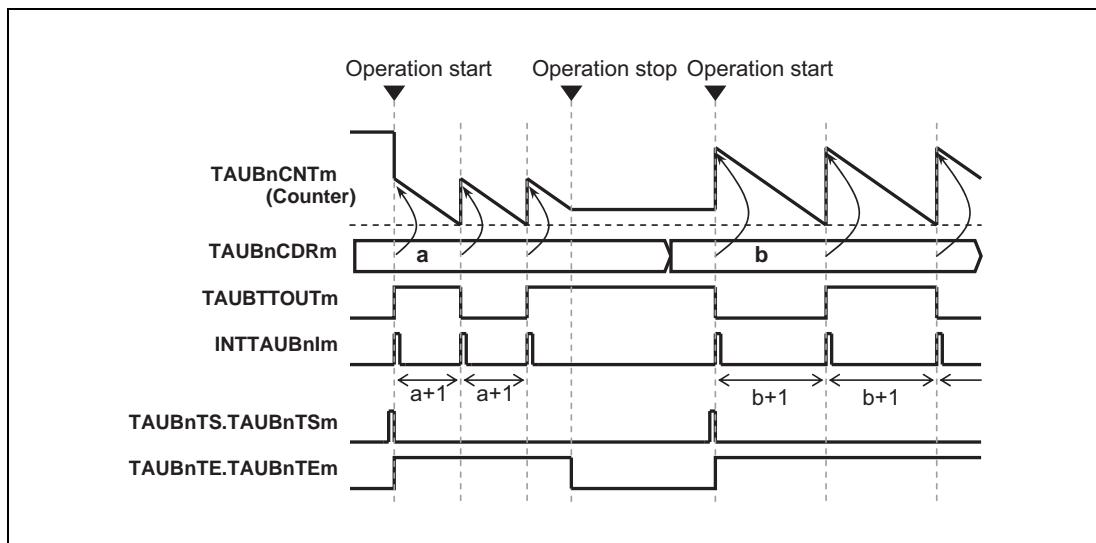


Figure 22.28 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEM to 0.
- TAUBnCNTm and TAUBTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1.

### (4) Operation stop and restart (TAUBnCMORm.TAUBnMD0 = 0)

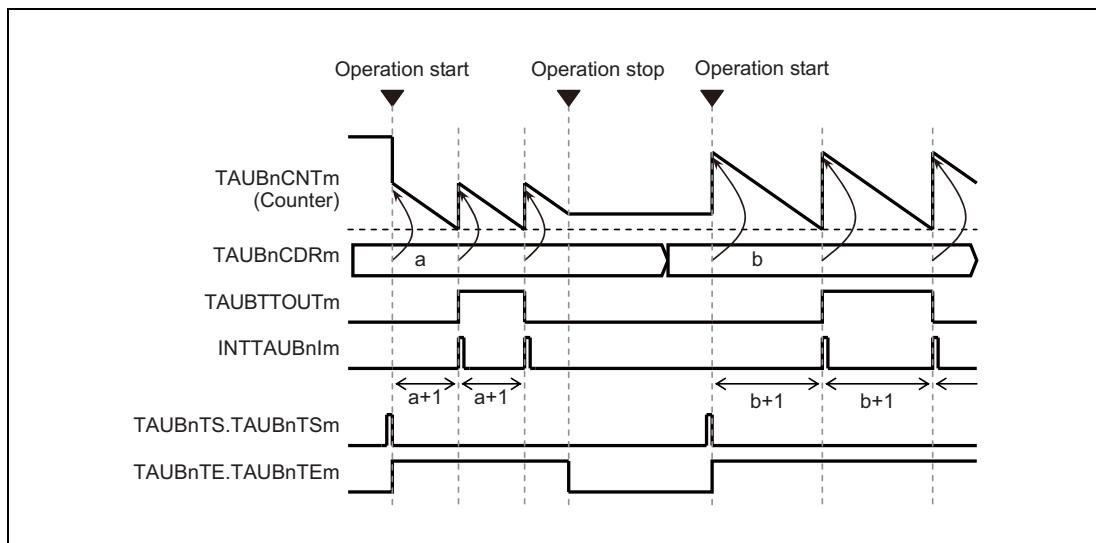


Figure 22.29 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 0

## (5) Forced restart (TAUBnCMORm.TAUBnMD0 = 1)

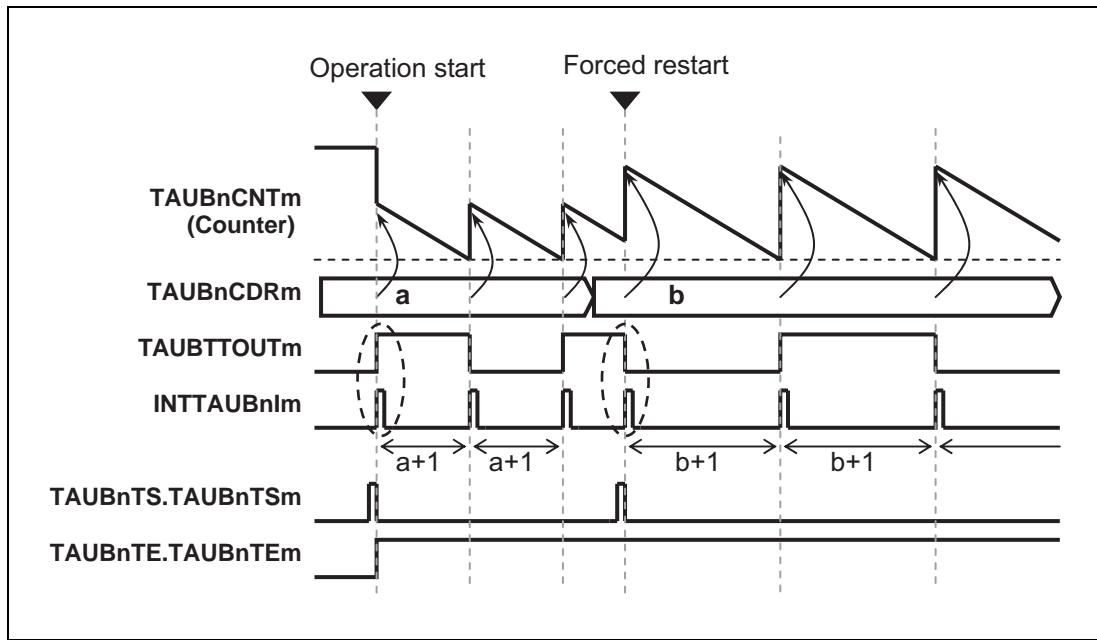


Figure 22.30 Forced Restart Operation, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt at start or restart is generated and the output TAUBTTOUTm toggles.

## (6) Forced restart (TAUBnCMORm.TAUBnMD0 = 0)

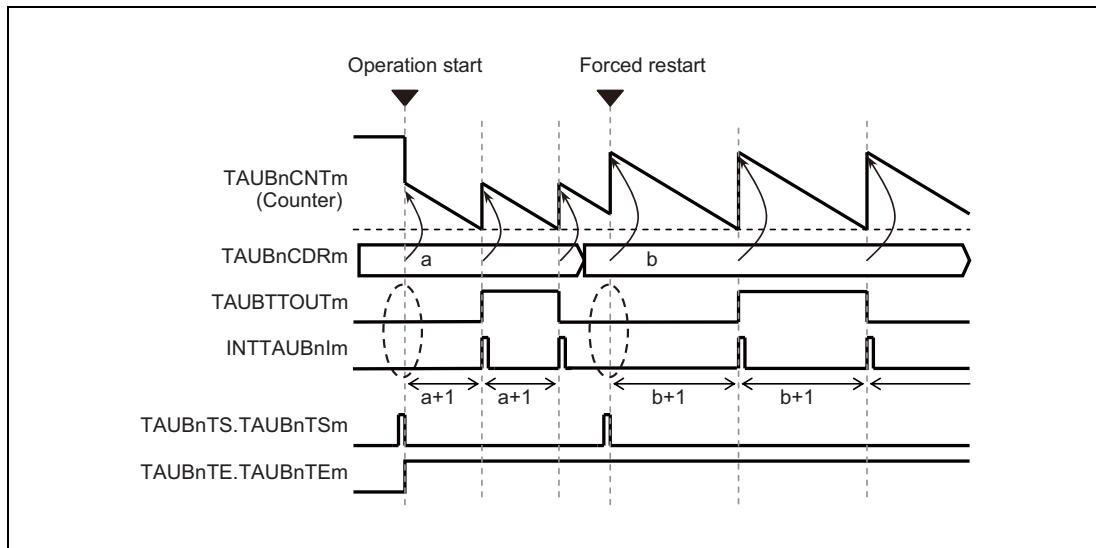


Figure 22.31 Forced Restart Operation (TAUBnCMORm.TAUBnMD0 = 0)

- The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle.

## 22.12.2 TAUBTTINm Input Interval Timer Function

### 22.12.2.1 Overview

#### Summary

此功能作为参考定时器，用于在一定的间隔或者检测到一个有效的TAUBTTINm输入边沿产生定时器中断(INTTAUBnIm)。This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals or when a valid TAUBTTINm input edge is detected.

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

INTTAUBnIm is generated when the counter reaches 0000<sub>H</sub> or by an effective TAUBTTINm input edge. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The type of edge used as the trigger is specified using the TAUBnCMURm.TAUBnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

### 22.12.2.2 Block Diagram and General Timing Diagram

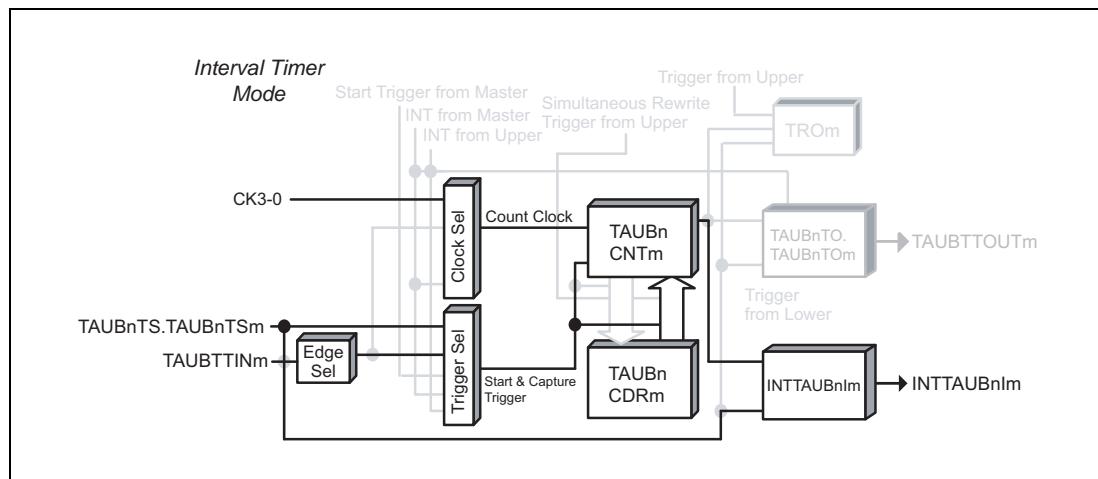


Figure 22.32 Block Diagram for TAUBTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1).
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

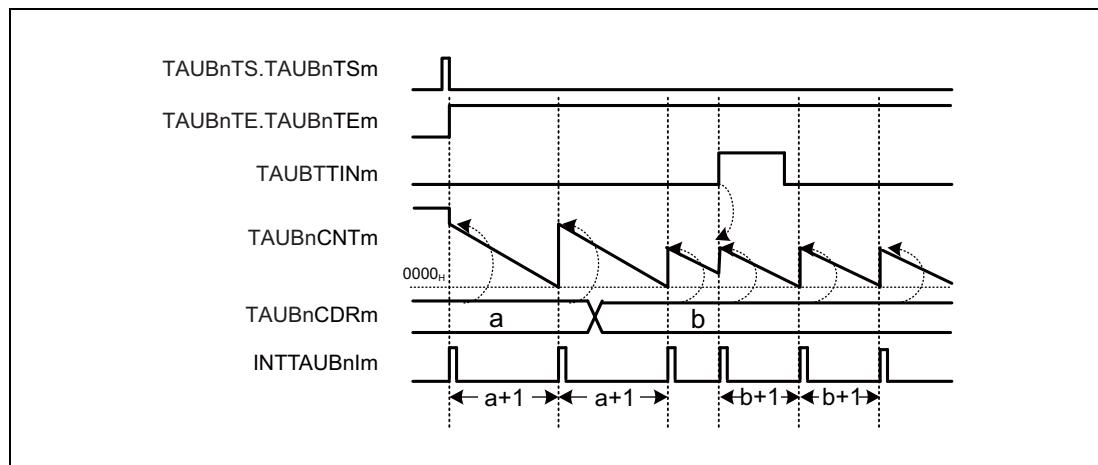


Figure 22.33 General Timing Diagram for TAUBTTINm Input Interval Timer Function

### 22.12.2.3 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.44 Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnlm not generated at operation start 1: Generates INTTAUBnlm at operation start

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.45 Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

**(3) Channel output mode**

This function does not use channel output mode.

**(4) Simultaneous rewrite**

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

**Table 22.46 Simultaneous Rewrite Settings for TAUBTTINm Input Interval Timer Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

### 22.12.2.4 Operating Procedure for TAUBTTINm Input Interval Timer Function

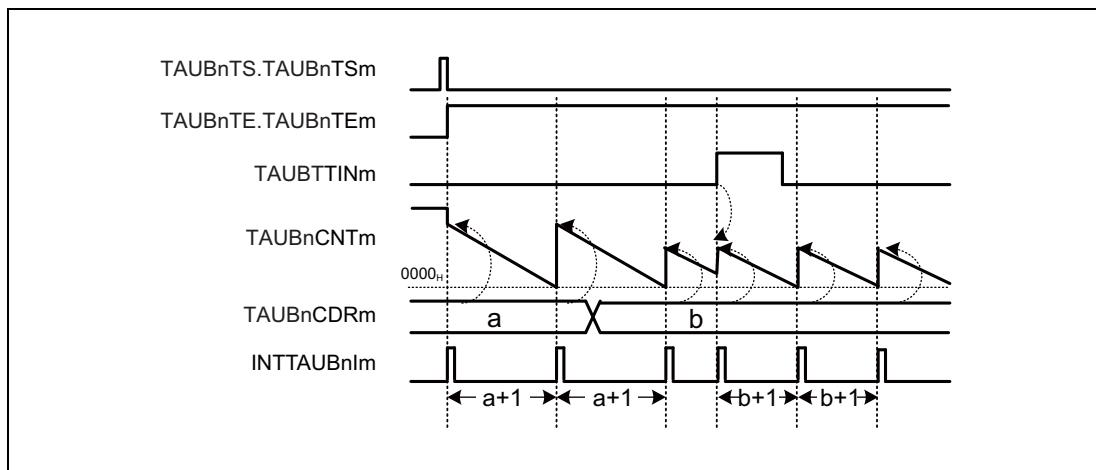
Table 22.47 Operating Procedure for TAUBTTINm Input Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	<p>Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.44, Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function</b> and <b>Table 22.45, Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function</b></p> <p>Set the value of the TAUBnCDRm register.</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEM is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.</p>
During operation	<p>The values of the TAUBnCMURm.TAUBnTIS[1:0] and the TAUBnCDRm register can be changed at any time. The TAUBnCNTm register can be read at all times. Detection of TAUBTTINm edge</p>	<p>TAUBnCNTm counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>• TAUBnCNTm reloads the TAUBnCDRm value and continues count operation</li> <li>• INTTAUBnIm is generated</li> </ul> <p>When a TAUBTTINm input valid edge is detected during count operation, TAUBnCNTm reloads the TAUBnCDRm value and continues count operation. Afterwards, this procedure is repeated.</p>
Stop operation	<p>Set TAUBnTT.TAUBnTTm to 1 TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEM is cleared to 0 and the counter stops. TAUBnCNTm stop and retain their current values.</p>



### 22.12.2.5 Specific Timing Diagrams

The timing diagrams in **Section 22.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by an effective TAUBTTINm input edge.



**Figure 22.34 Counter Triggered by Rising TAUBTTINm Input Edge  
(TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>), TAUBnCMORm.TAUBnMD0 = 1**

- If an effective TAUBTTINm input edge is detected, an interrupt INTTAUBnIm is generated. In this example, the effective edge is a rising edge (TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>).

## 22.12.3 Clock Divide Function

### 22.12.3.1 Overview

#### Summary

This function is used as a frequency divider. The frequency of the input signal TAUBTTINm is divided by a factor related to TAUBnCDRm, and an interrupt INTTAUBnIm is generated.

#### Prerequisites

- TAUBTTINm must have a fixed frequency
- The operation mode must be set to interval timer mode, see **Table 22.48, Contents of the TAUBnCMORm Register for Clock Divide Function**

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TSm) to 1.

This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value, using TAUBTTINm as the count clock.

When the counter value reaches 0000<sub>H</sub>, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEM = 0. TAUBnCNTm stops but retain their values. The function can be restarted by setting TAUBnTS.TAUBnTSm = 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.

#### Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

#### NOTE

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits.

### 22.12.3.2 Block Diagram and General Timing Diagram

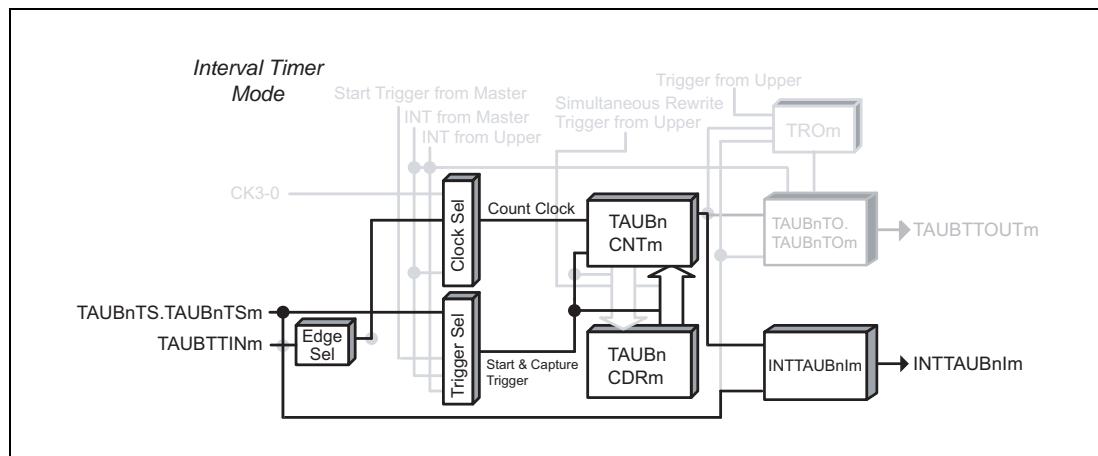


Figure 22.35 Block Diagram for Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

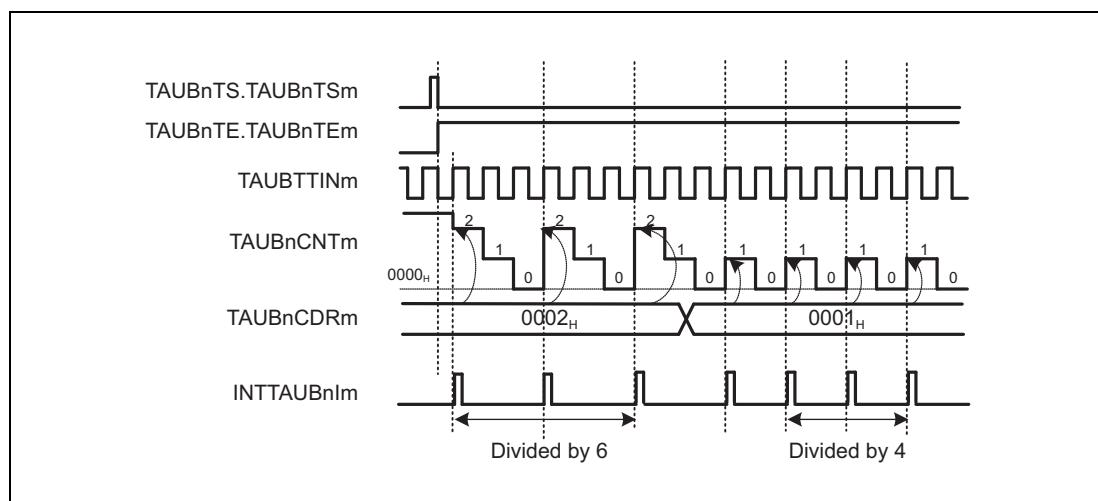


Figure 22.36 General Timing Diagram for Clock Divide Function

### 22.12.3.3 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.48 Contents of the TAUBnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 1 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnlm not generated at operation start 在操作开始的时候INTTAUBnlm不能产生 1: Generates INTTAUBnlm at operation start 在开始的时候产生INTTAUB

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.49 Contents of the TAUBnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

### (3) Channel output mode

This function does not use channel output mode.

### (4) Simultaneous rewrite

同时复写寄存器不能使用Clock Divide Function, 因此, 这些寄存器必须设置为0

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Clock Divide Function. Therefore, these registers must be set to 0.

**Table 22.50 Simultaneous Rewrite Settings for Clock Divide Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.3.4 Operating Procedure for Clock Divide Function

**Table 22.51 Operating Procedure for Clock Divide Function**

Operation	Status of TAUBn	
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.48, Contents of the TAUBnCMORm Register for Clock Divide Function</b> and <b>Table 22.49, Contents of the TAUBnCMURm Register for Clock Divide Function</b> Set the value of the TAUBnCDRm register. Set the control bits as described in <b>Table 22.49, Contents of the TAUBnCMURm Register for Clock Divide Function</b> .	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 is set to 1, INTTAUBnlm is generated.
During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	When a TAUBTTInm input edge is detected, TAUBnCNTm counts down. When the counter reaches 0000H: <ul style="list-style-type: none"><li>• TAUBnCNTm loads the TAUBnCDRm value and continues count operation</li><li>• INTTAUBnlm is generated</li></ul>
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

### 22.12.3.5 Specific Timing Diagrams

#### (1) TAUBnCDRm = 0000<sub>H</sub>

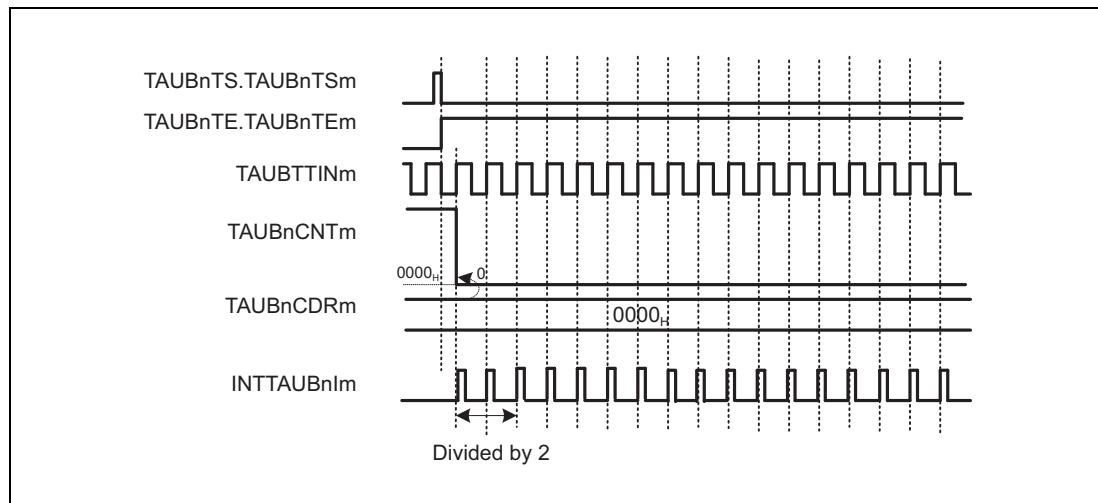


Figure 22.37 TAUBnCDRm = 0000<sub>H</sub>, TAUBnCMORm.TAUBnMD0 = 1,  
TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>

- If TAUBnCDRm is 0000<sub>H</sub>, TAUBnCNTm is also always 0000<sub>H</sub>.
- INTTAUBnIm is generated every count clock.

Figure 22.37 is an image of the operation timing. Actually, there is a delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn.

#### (2) Operation restart

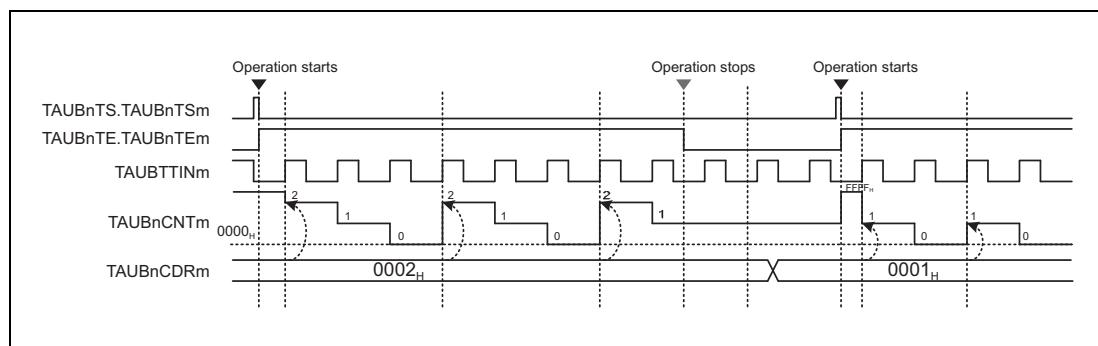


Figure 22.38 Operation Restart (TAUBnCMORm.TAUBnMD0 = 1,  
TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

## (3) Forced restart

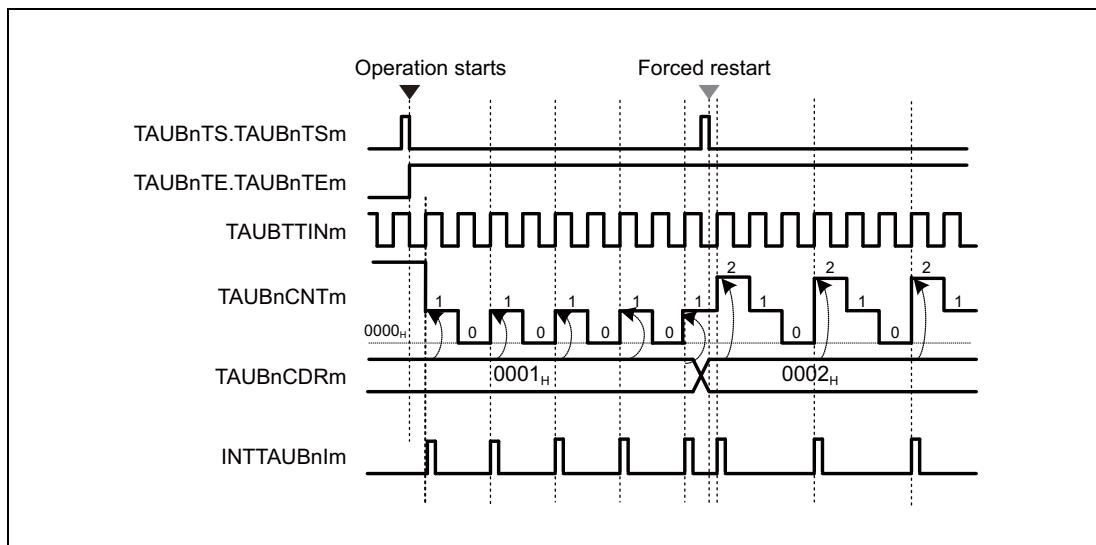


Figure 22.39 Forced Restart (TAUBnCMORm.TAUBnMD0 = 1,  
TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

To forcibly restart the counter.

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- The value of TAUBnCDRm is written to TAUBnCNTm and the count operation restarts.

## 22.12.4 External Event Count Function

### 22.12.4.1 Overview

#### Summary

**此功能用于事件定时器，当指定数目的有效沿 (TAUBTTINm input) 被探测到的时候**  
 This function is used as an event timer. It generates an interrupt (INTTAUBnIm) when a specific number of valid edges of TAUBTTINm input are detected.

#### Prerequisites

**操作模式被设置为event count mode**

- The operation mode must be set to event count mode, see **Table 22.52, Contents of the TAUBnCMORm Register for External Event Count Function**
- TAUBTTOUT 不能用于此模式**
- TAUBTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. When the counter starts, the current value of TAUBnCDRm is written to TAUBnCNTm. **一旦计数器开始工作，CDRm的当前值会被写入CNTm**

When an effective TAUBTTINm input edge is detected, the value of TAUBnCNTm reduces by 1. TAUBnCNTm retains this value until a valid TAUBTTINm input edge is detected or the counter is restarted.

When effective edges are detected (TAUBnCDRm + 1) times, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues to operate.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEM to 0. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

#### Conditions

**边沿触发的类型由TIS[1:0]bits决定**

The type of edge used as the trigger is specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10<sub>B</sub>, rising and falling edges trigger the counter.

### 22.12.4.2 Equations

Number of valid edges,

detected before INTTAUBnIm is generated = TAUBnCDRm + 1

**在INTTAUBnIm产生之前检测到有效沿的数目：TAUBnCDRm+1**

### 22.12.4.3 Block Diagram and General Timing Diagram

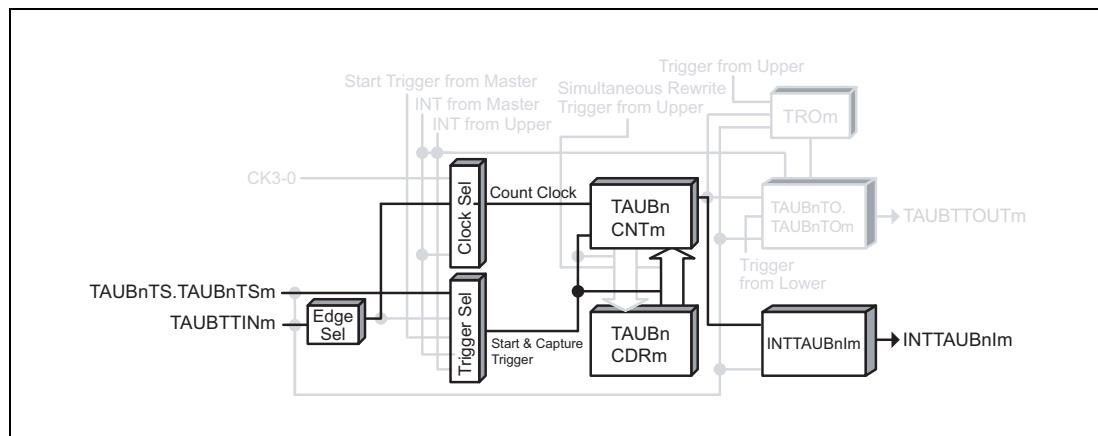


Figure 22.40 Block Diagram for External Event Count Function

The following settings apply to the general timing diagram.

- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

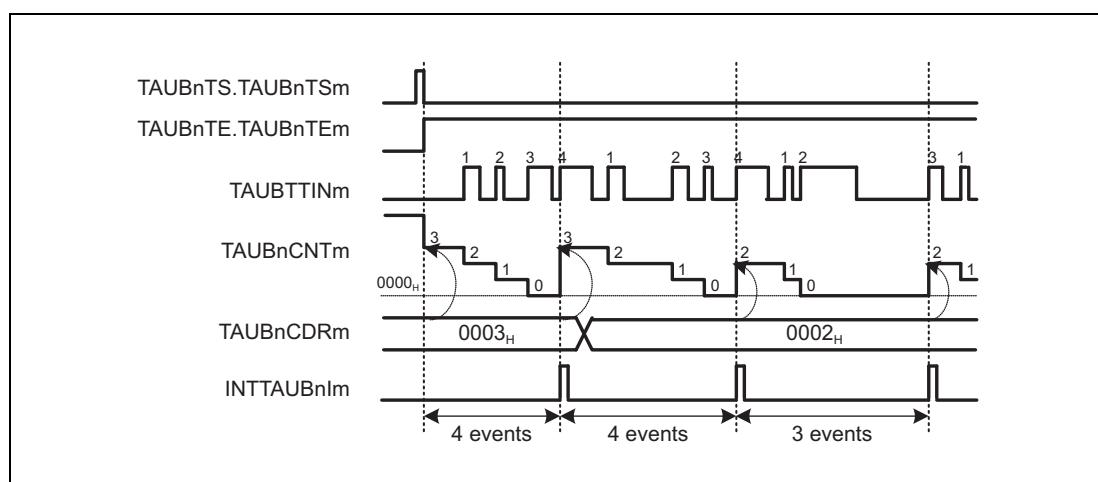


Figure 22.41 General Timing Diagram for External Event Count Function

#### 22.12.4.4 Register Settings

##### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.52 Contents of the TAUBnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 1 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0011 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

##### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.53 Contents of the TAUBnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected. 11: Setting prohibited

### (3) Channel output mode

The channel output mode is not used by this function.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the External Event Count Function.

Therefore, these registers must be set to 0.

**Table 22.54 Simultaneous Rewrite Settings for External Event Count Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.4.5 Operating Procedure for External Event Count Function

**Table 22.55 Operating Procedure for External Event Count Function**

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.52, Contents of the TAUBnCMORm Register for External Event Count Function</b> and <b>Table 22.53, Contents of the TAUBnCMURm Register for External Event Count Function</b> Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value and waits for detection of the TAUBTTINm input edge.
During operation	Detection of TAUBTTINm edges.  The value of TAUBnCDRm can be changed at any time.  The TAUBnCNTm register can be read at any time.	TAUBnCNTm performs count-down operation each time a TAUBTTINm input edge is detected. When effective edges are detected (TAUBnCDRm + 1) times: <ul style="list-style-type: none"><li>• TAUBnCNTm loads the TAUBnCDRm value and continues count operation</li><li>• INTTAUBnIm is generated.</li></ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

### 22.12.4.6 Specific Timing Diagrams

#### (1) TAUBnCDRm = 0000<sub>H</sub>

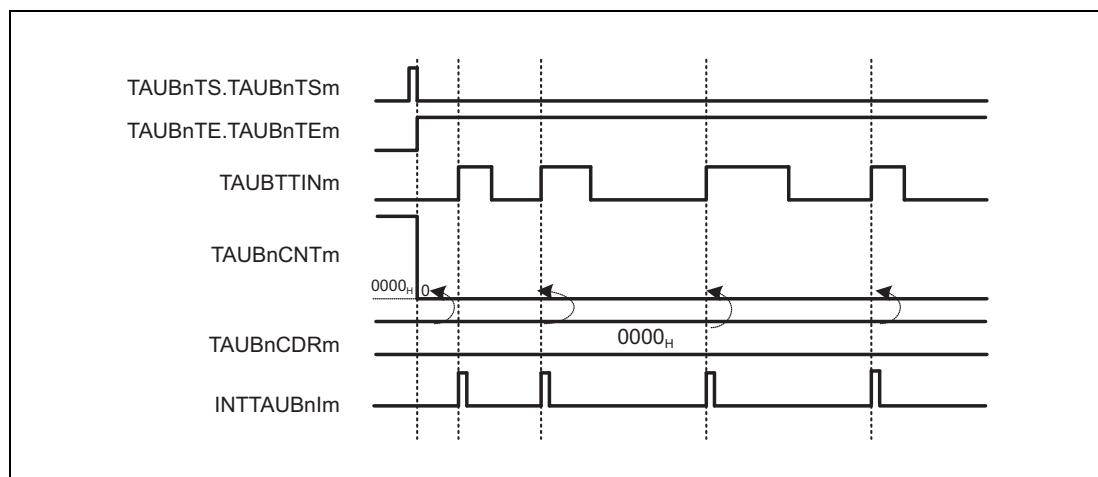


Figure 22.42 TAUBnCDRm = 0000<sub>H</sub>, TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>

- If 0000<sub>H</sub> = TAUBnCDRm, 0000<sub>H</sub> is loaded to TAUBnCNTm every time a valid TAUBTTINm input edge is detected.

This means, INTTAUBnIm is generated every time a valid TAUBTTINm input edge is detected.

#### (2) Operation stop and restart

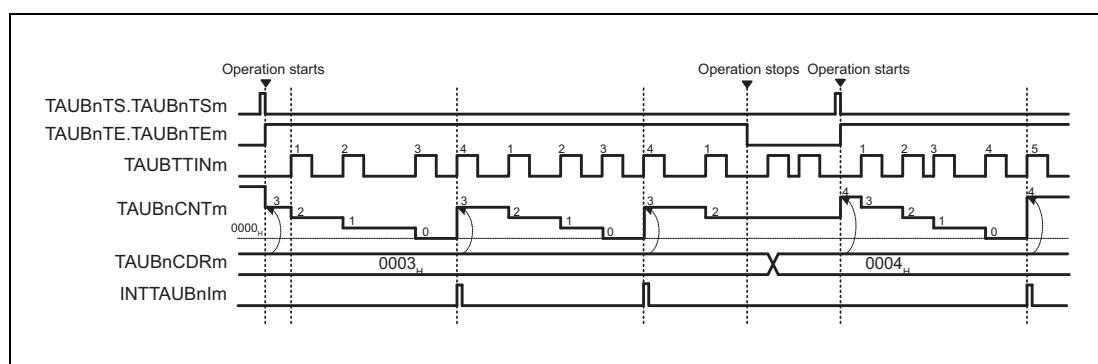


Figure 22.43 Operation Stop and Restart (TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained. TAUBTTINm continues and TAUBnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm loads the TAUBnCDRm value and restarts count operation.

### (3) Forced restart

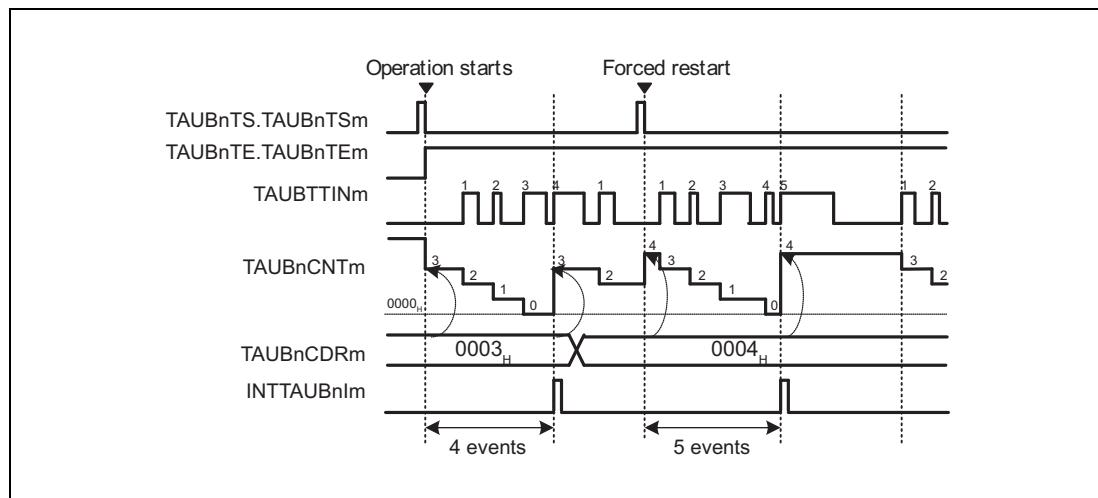


Figure 22.44 Forged Restart (TAUBnCMURm.TAUBnTIS[1:0] = 01<sub>B</sub>)

A forced restart applies the new TAUBnCDRm value to TAUBnCNTm immediately.

- The counter can be restarted (without stopping it first), by setting TAUBnTS.TAUBnTSm to 1 during operation.
- The value of TAUBnCDRm is loaded to TAUBnCNTm and the counter awaits the next valid TAUBTTINm input edge.

## 22.12.5 One-Pulse Output Function

### 22.12.5.1 Overview

#### Summary

此功能是当一个有效的TAUBTTINm input 边沿被检测的时候产生一个中断，然后在指定的间隔时间之后继续产生中断。This function generates an interrupt (INTTAUBnIm) when a valid TAUBTTINm input edge is detected and continues to generate interrupts at the specified interval. TAUBTTINm input signal pulses that occur within the defined interval are ignored. 在定义的时间间隔内TAUBTTINm输入的信号脉冲会被忽略

#### Prerequisites

操作模式必须设置为pulse one-count mode

- The operation mode must be set to pulse one-count mode. (See **Table 22.56, Contents of the TAUBnCMORm Register for One-Pulse Output Function**).
- Trigger detection must be disabled during counting (TAUBnCMORm.TAUBnMD0 = 0).

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input edge is detected. The value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from the TAUBnCDRm value, and an interrupt is generated.

计数器达到0x0001就会产生一个中断，计数器停止在0x0000并且等待下一轮有效的TAUBTTINm数输入沿  
When the counter reaches  $0001_H$  an interrupt is generated. The counter stops at  $0000_H$  and awaits the next effective TAUBTTINm input edge.

When the counter is counting down, further TAUBTTINm input signals are ignored, i.e. the counter does not reset.

CDRm的值可以在任何时间写入，在下一轮 的计数器开始后生效

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

#### Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] =  $00_B$ , falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] =  $01_B$ , rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] =  $10_B$ , rising and falling edges trigger the counter.

### 22.12.5.2 Equations

Interval between TAUBTTINm and INTTAUBnIm = count clock cycle  $\times$  TAUBnCDRm

### 22.12.5.3 Block Diagram and General Timing Diagram

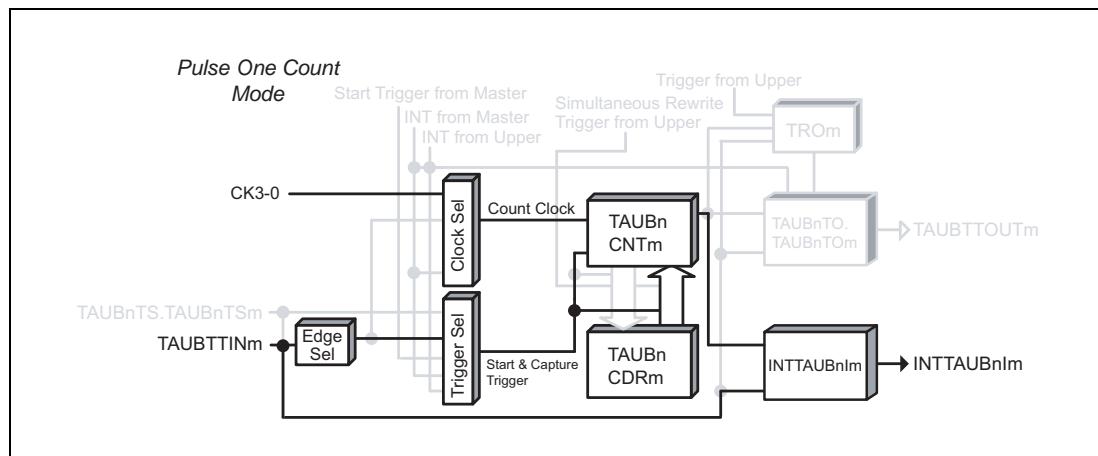


Figure 22.45 Block Diagram for One-Pulse Output Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

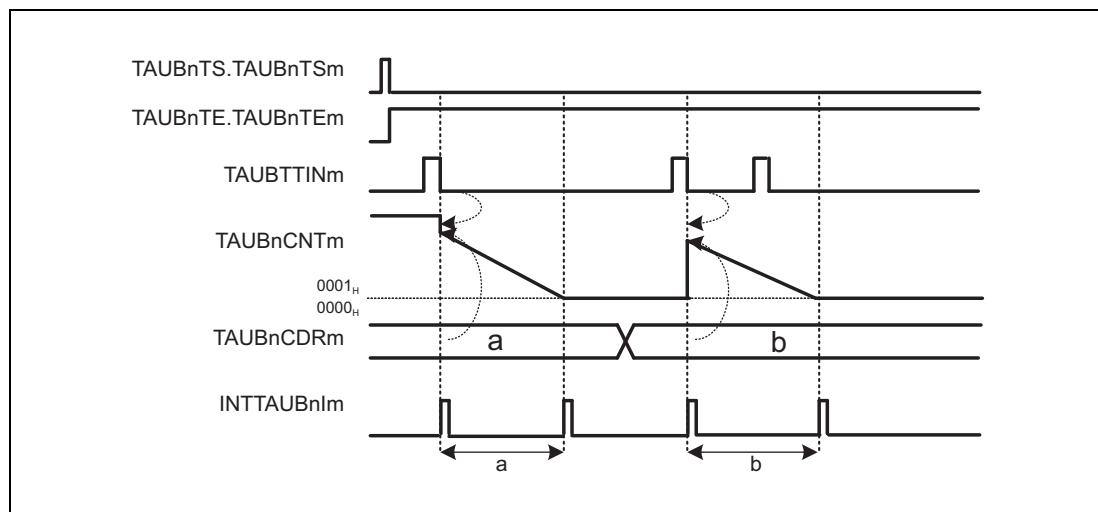


Figure 22.46 General Timing Diagram for One-Pulse Output Function

### 22.12.5.4 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 22.56 Contents of the TAUBnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 22.57 Contents of the TAUBnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

### (3) Channel output mode

This function does not use channel output mode.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the One-Pulse Output Function.  
Therefore, these registers must be set to 0.

**Table 22.58 Simultaneous Rewrite Settings for One-Pulse Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEM	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEM = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.5.5 Operating Procedure for One-Pulse Output Function

**Table 22.59 Operating Procedure for One-Pulse Output Function**

Operation	Status of TAUBn
Initial channel setting Set the TAUBnCMORm and TAUBnCMURm registers as described in <a href="#">Table 22.56, Contents of the TAUBnCMORm Register for One-Pulse Output Function</a> and <a href="#">Table 22.57, Contents of the TAUBnCMURm Register for One-Pulse Output Function</a> Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value.
During operation The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	INTTAUBnIm is generated when TAUBnCNTm starts. TAUBnCNTm counts down. When the counter reaches 0001H, INTTAUBnIm is generated. TAUBnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUBnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop operation Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

## 22.12.6 TAUBTTINm Input Pulse Interval Measurement Function

### 22.12.6.1 Overview

#### Summary

此功能捕获计数值，使用此值，然后溢出bit测量TAUBTTINm input signal的间隔  
This function captures the count value and uses this value and the overflow bit  
TAUBnCSRm.TAUBnOVF to measure the interval of the TAUBTTINm input signal.

#### Prerequisites

TAUBTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter TAUBnCNTm starts counting up from  $0000_H$ . When a valid TAUBTTINm edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter resets to  $0000_H$  and subsequently continues operation.  
 在TAUBTTINm沿被检测到之前，计数器达到 $0xFFFF$ ，就会溢出变为 $0x0000$ . counter reset为 $0x00$ ，然后继续  
 If the counter reaches  $FFFF_H$  before a valid TAUBTTINm edge is detected, it overflows to  $0000_H$ . The counter is reset to  $0000_H$  and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0]. 分别传递给CDRm和OVF bit要取决于COS[1:0]

Table 22.60 Effects of an Overflow

TAUBnCMORm. COS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input is then Detected	
	TAUBnCDRm	TAUBnCSRm. TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm. TAUBnOVF
00	Unchanged	0	TAUBnCNTm loaded to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm set to 0, TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] is 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

CDRm和OVF的组合可用于减小TAUBTTINm信号的间隔  
 The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the interval of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

The function can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm stops but retains its value. While the function is stopped, TAUBTTINm input valid edge detection and TAUBnCNTm capture are not performed.

The counter is reset to  $0000_H$  and subsequently continues operation.

#### Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the interrupt at start or restart is not generated.

**NOTE**

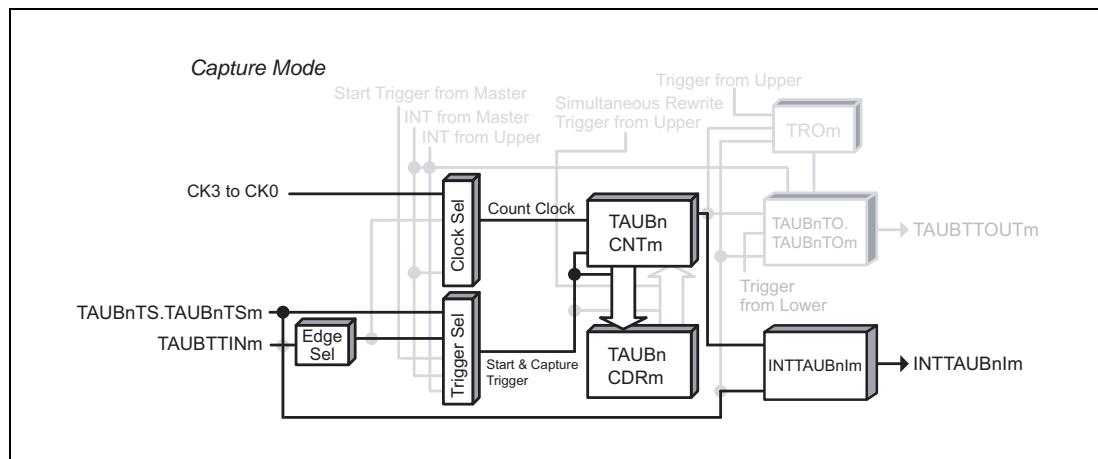
When  $\text{TAUBnCMORm.TAUBnCOS}[1:0] = 10_B$  or  $11_B$ , the value of  $\text{TAUBnCNTm}$  is not written to  $\text{TAUBnCDRm}$  when the first valid  $\text{TAUBTTINm}$  input edge occurs after an overflow. However, an interrupt is generated.

**22.12.6.2 Equations**

$\text{TAUBTTINm输入脉冲间隔} =$

$$\text{TAUBTTINm input pulse interval} = \text{count clock cycle} \times \\ [(\text{TAUBnCSRm.TAUBnOVF} \times (\text{FFFF}_H + 1)) + \text{TAUBnCDRm capture value} + 1]$$

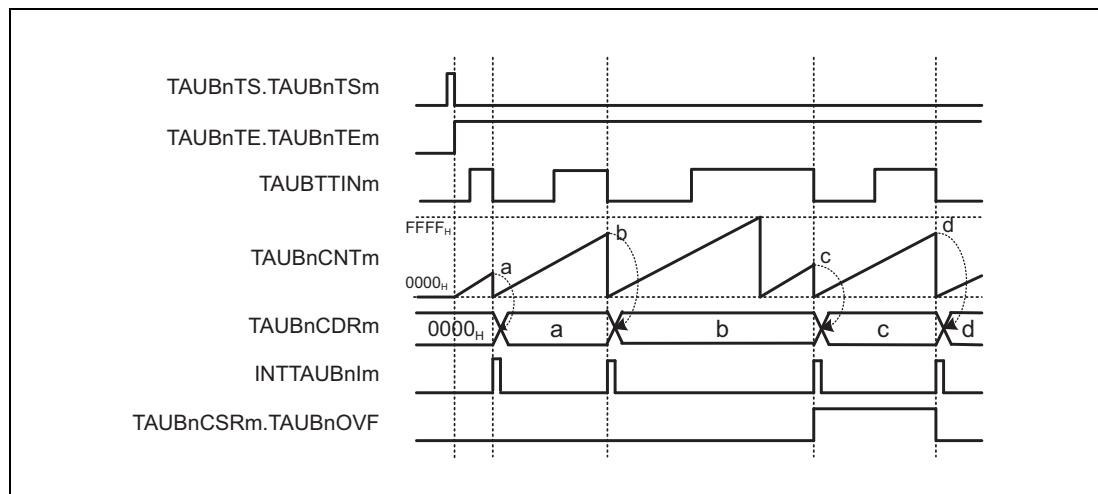
### 22.12.6.3 Block Diagram and General Timing Diagram



**Figure 22.47 Block Diagram for TAUBTTINm Input Pulse Interval Measurement Function**

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)
- When a valid TAUBTTINm input is detected after an overflow TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00<sub>B</sub>)



**Figure 22.48 General Timing Diagram for TAUBTTINm Input Pulse Interval Measurement Function**

### 22.12.6.4 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MD0						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.61 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	See Table 22.60, Effects of an Overflow
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0010 <sub>B</sub> .
0	TAUBnMD0	0: INTTAUBnlm not generated at operation start 1: Generates INTTAUBnlm at operation start

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.62 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

### (3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

**Table 22.63 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Measurement Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.6.5 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

**Table 22.64 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function**

Operation	Status of TAUBn	
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.61, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function</b> and <b>Table 22.62, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function</b>  The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm is cleared to 0000H. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMDO is set to 1.
During operation	Detection of TAUBTTINm edges.  The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time. The TAUBnCSCm.TAUBnCLOV bit can be set to 1. (The TAUBnCSRm.TAUBnOVF bit can be cleared to 0.)	TAUBnCNTm starts to count up from 0000H. When a TAUBTTINm valid edge is detected: <ul style="list-style-type: none"><li>• TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and returns to 0000H</li><li>• INTTAUBnIm is then generated.</li></ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.



### 22.12.6.6 Specific Timing Diagrams: Overflow Behavior

#### (1) TAUBnCMORm.TAUBnCOS[1:0] = 00<sub>B</sub>

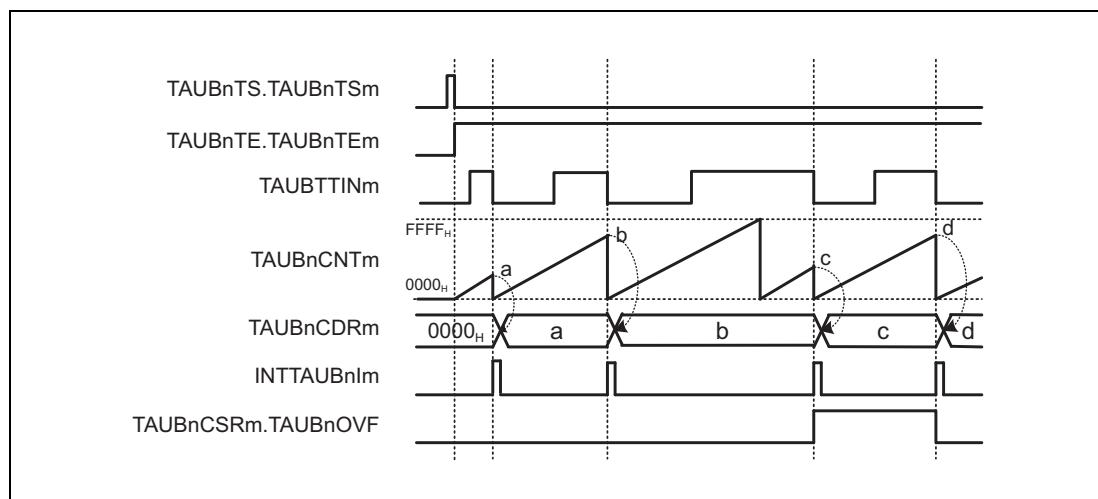


Figure 22.49 TAUBnCMORm.TAUBnCOS[1:0] = 00<sub>B</sub>, TAUBnCMORm.TAUBnMD0 = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRM.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is loaded to TAUBnCDRm and TAUBnCSRM.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRM.TAUBnOVF is cleared to 0.

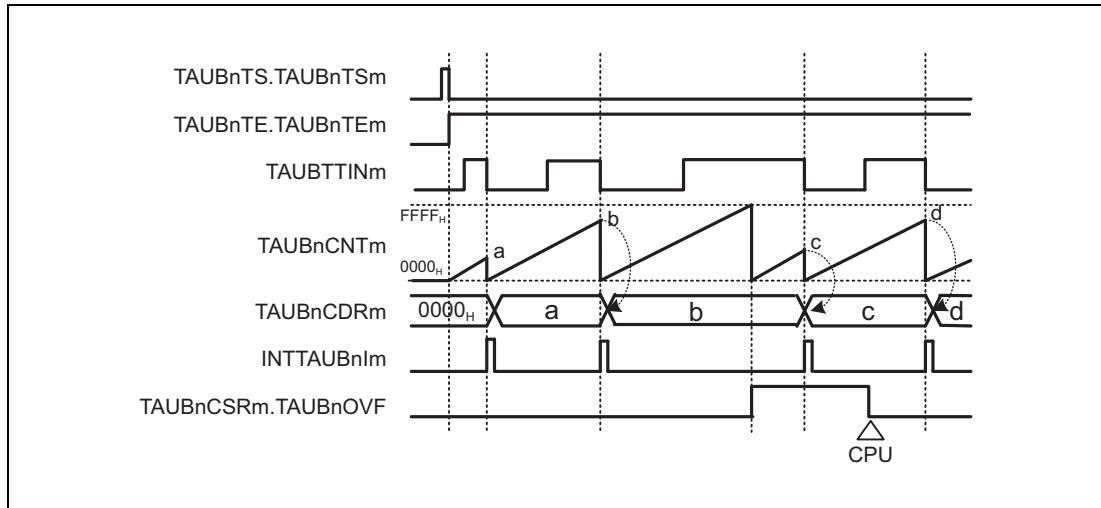
(2) TAUBnCMORm.TAUBnCOS[1:0] = 01<sub>B</sub>

Figure 22.50 TAUBnCMORm.TAUBnCOS[1:0] = 01<sub>B</sub>, TAUBnCMORm.TAUBnMD0 = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command. (TAUBnCSCm.TAUBnCLOV bit = 1)

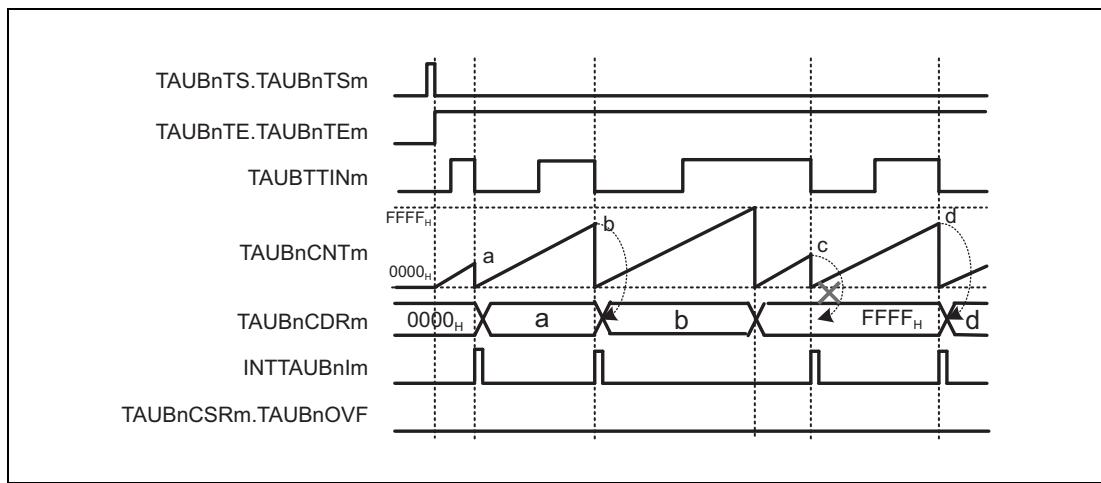
(3) TAUBnCMORm.TAUBnCOS[1:0] = 10<sub>B</sub>

Figure 22.51 TAUBnCMORm.TAUBnCOS[1:0] = 10<sub>B</sub>, TAUBnCMORm.TAUBnMD0 = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, TAUBnCDRm is set to FFFF<sub>H</sub> and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

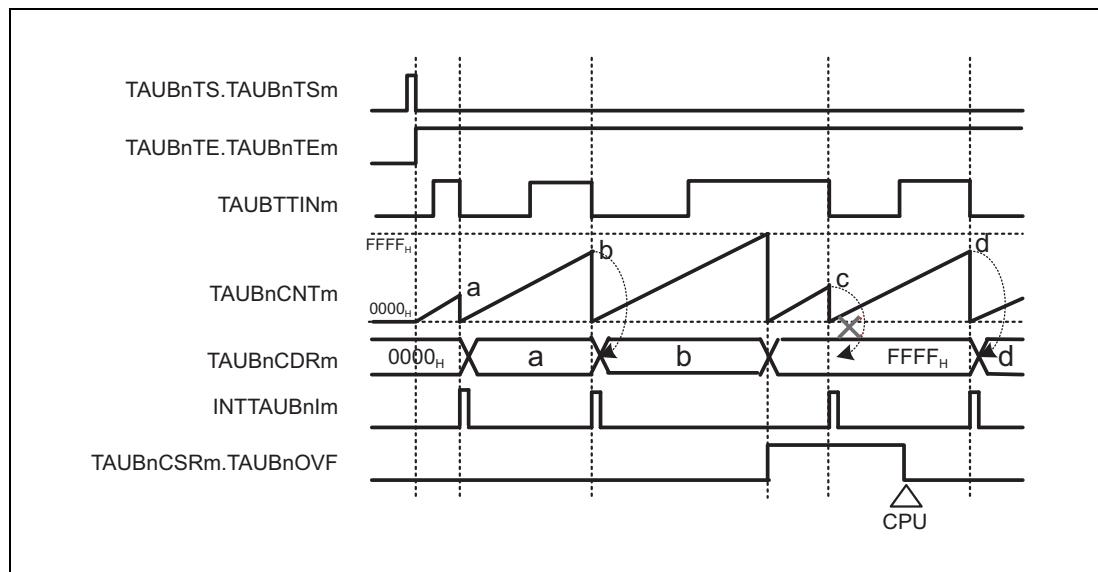
(4) TAUBnCMORm.TAUBnCOS[1:0] = 11<sub>B</sub>

Figure 22.52 TAUBnCMORm.TAUBnCOS[1:0] = 11<sub>B</sub>, TAUBnCMORm.TAUBnMD0 = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, TAUBnCDRm is set to FFFF<sub>H</sub>, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

## (5) When rising and falling edge detection are selected (TAUBnCMORm.TAUBnMD0 = 1)

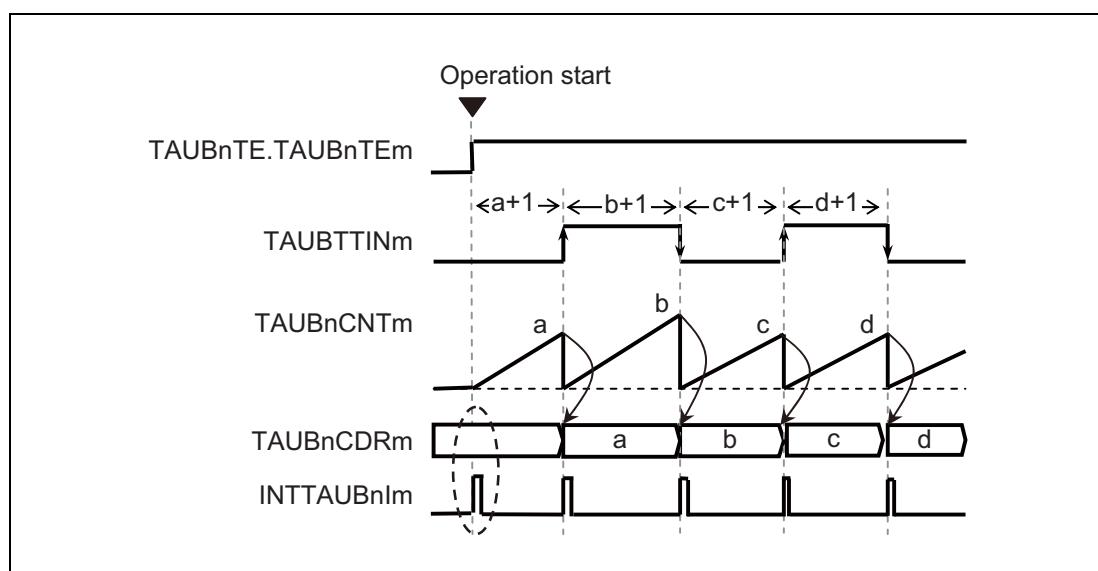


Figure 22.53 TAUBnCMORm.TAUBnMD0 = 1

Setting TAUBnCMURm.TAUBnTIS[1:0] to 10<sub>B</sub> (detection of both edges selected) measures the TAUBTTINm rising and falling edge intervals.

## (6) Operation stop and operation restart (TAUBnCMORm.TAUBnMD0 = 0)

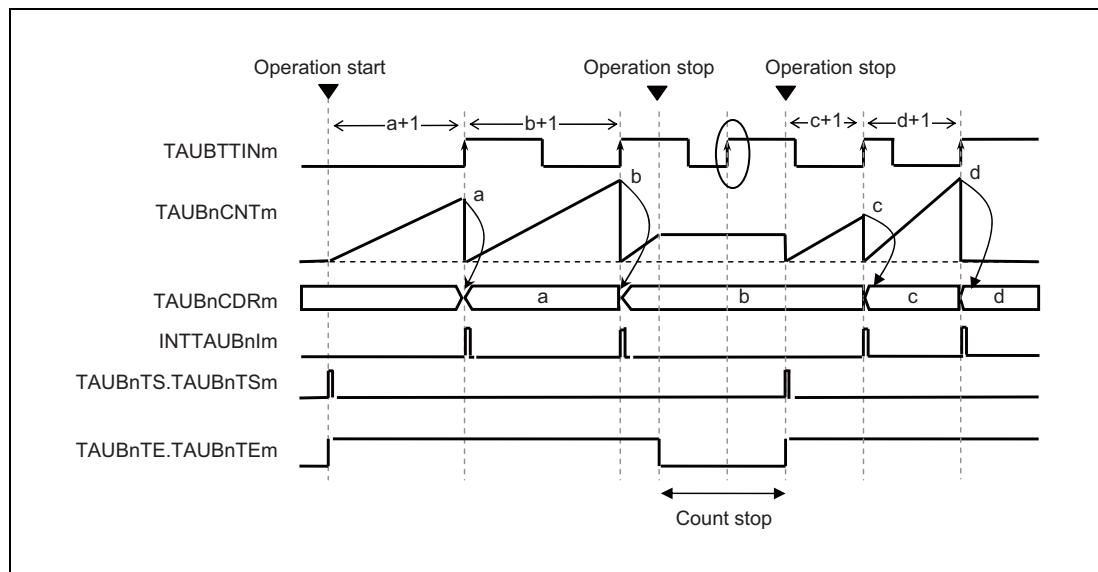


Figure 22.54 Operation Stop and Operation Restart (TAUBnCMORm.TAUBnMD0 = 0)

Setting TAUBnTT.TAUBnTTm to 1 clears TAUBnTE.TAUBnTEM to 0, which stops the count operation. At this time, TAUBnCNTm retains the status and stops.

When TAUBnTE.TAUBnTEM retains 0 (operation stopped), TAUBnTTINm input is ignored (edge detection is ignored and capture operation is not performed).

Setting TAUBnTS.TAUBnTSm to 1 clears the counter to 0000H and restarts count-up operation.

## 22.12.7 TAUBTTINm Input Signal Width Measurement Function

### 22.12.7.1 Overview

#### Summary

此功能测量TAUBTTINm信号的宽度，在信号的一个沿开始计算，不过另外一个沿。This function measures the width of a TAUBTTINm signal by starting counting on one edge of the TAUBTTINm signal and capturing the counter value on the opposite edge.

#### Prerequisites

TAUBTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. 当TAUBTTINm起始沿被探测到 This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm start edge is detected, the counter TAUBnCNTm starts counting up from  $0000_H$ . When a valid TAUBTTINm stop edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter retains its value (TAUBnCDRm + 1) and awaits the next valid TAUBTTINm input start edge. 在TAUBTTINm沿被检测到之前，计数器达到 $0xFFFF$ ，就会溢出变为 $0x0000$ .counter reset为 $0x00$ ，然后继续 If the counter reaches  $FFFF_H$  before a valid TAUBTTINm stop edge is detected, it overflows. The counter is reset to  $0000_H$  and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 22.65 Effects of an Overflow

TAUBnCMORm. COS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input Stop Edge is Detected	
	TAUBnCDRm	TAUBnCSRm. TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm. TAUBnOVF
00	Unchanged	0	TAUBnCNTm written to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm stops counting TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] = 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the width of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

This function cannot be forcibly restarted.

此功能不能被强制重启

#### NOTE

When TAUBnCMORm.TAUBnCOS[1] = 1, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

### 22.12.7.2 Equations TAUBTTINm 输入信号宽度

TAUBTTINm input signal width = count clock cycle ×

$$[(\text{TAUBnCSRm.OVF} \times (\text{FFFF}_H + 1)) + \text{TAUBnCDRm capture value} + 1]$$

### 22.12.7.3 Block Diagram and General Timing Diagram

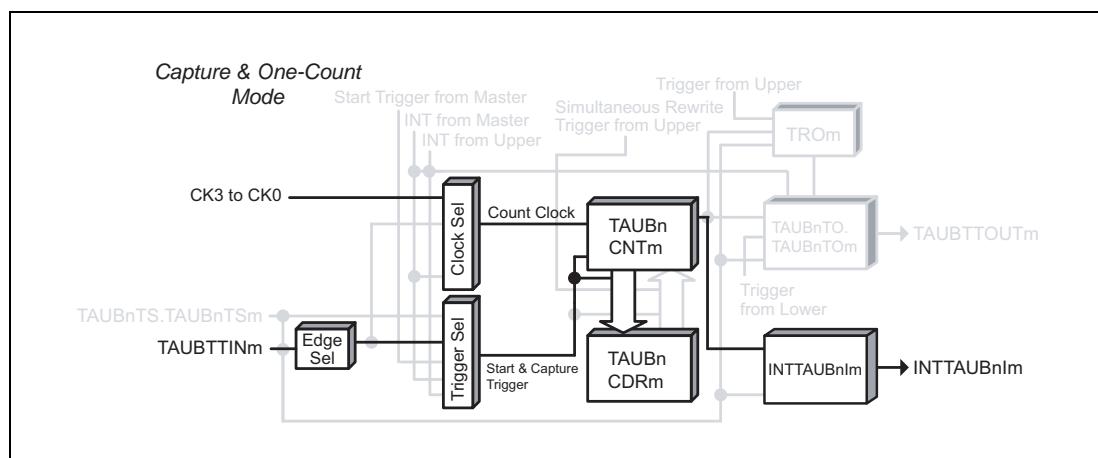


Figure 22.55 Block Diagram for TAUBTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement ( $\text{TAUBnCMURm.TAUBnTIS}[1:0] = 11_B$ )
- When a valid TAUBTTINm input is detected after an overflow, TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 ( $\text{TAUBnCMORM.TAUBnCOS}[1:0] = 00_B$ )

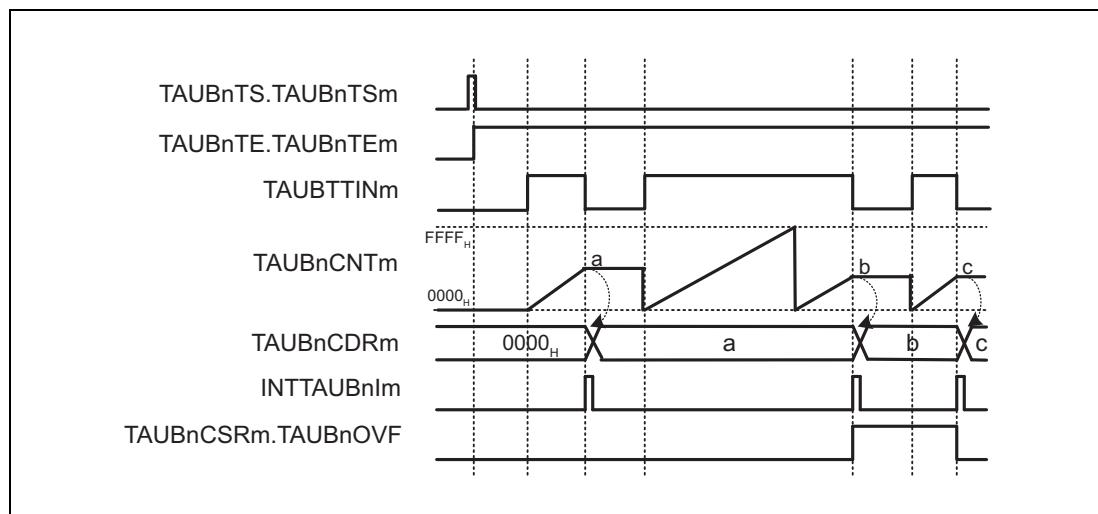


Figure 22.56 General Timing Diagram for TAUBTTINm Input Signal Width Measurement Function

### 22.12.7.4 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.66 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	See Table 22.65, Effects of an Overflow
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0110 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.67 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

#### (3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

**Table 22.68 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Measurement Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.7.5 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

**Table 22.69 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function**

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.66, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function</b> and <b>Table 22.67, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function</b>  The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEM is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a TAUBTTINm start edge is detected, TAUBnCNTm start edge to count up.
During operation	The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time. TAUBnCSCm.TAUBnCLOV bit can be set to 1.	TAUBnCNTm starts to count up from 0000H. When a TAUBTTINm valid edge is detected: <ul style="list-style-type: none"><li>• TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and retains its value</li><li>• INTTAUBnIm is then generated.</li><li>• The count stops at the value transferred to TAUBnCDRm + 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.</li></ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

### 22.12.7.6 Specific Timing Diagrams: Overflow Behavior

#### (1) TAUBnCMORm.TAUBnCOS[1:0] = 00<sub>B</sub>

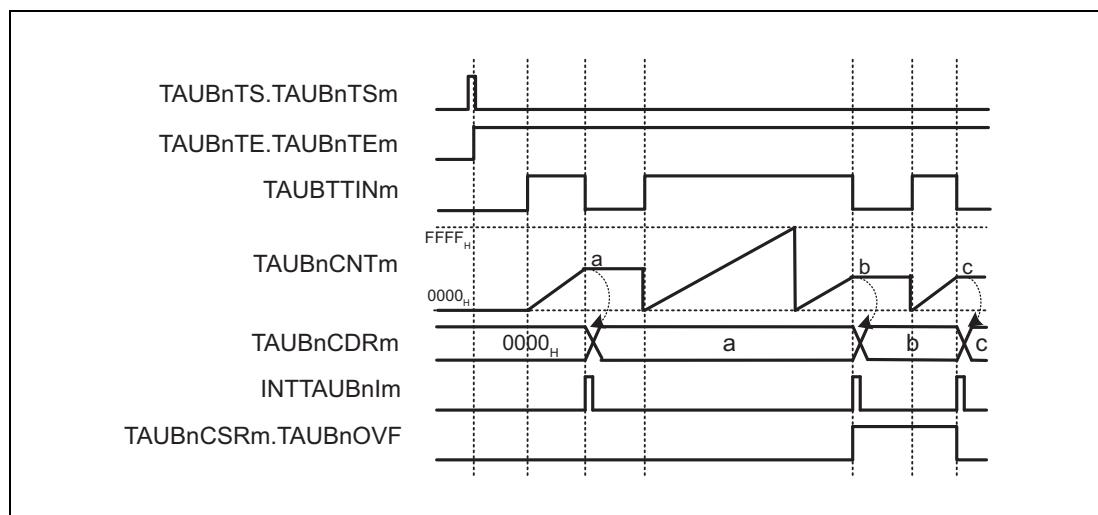


Figure 22.57 TAUBnCMORm.TAUBnCOS[1:0] = 00<sub>B</sub>, TAUBnCMORm.TAUBnMDO = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

#### (2) TAUBnCMORm.TAUBnCOS[1:0] = 01<sub>B</sub>

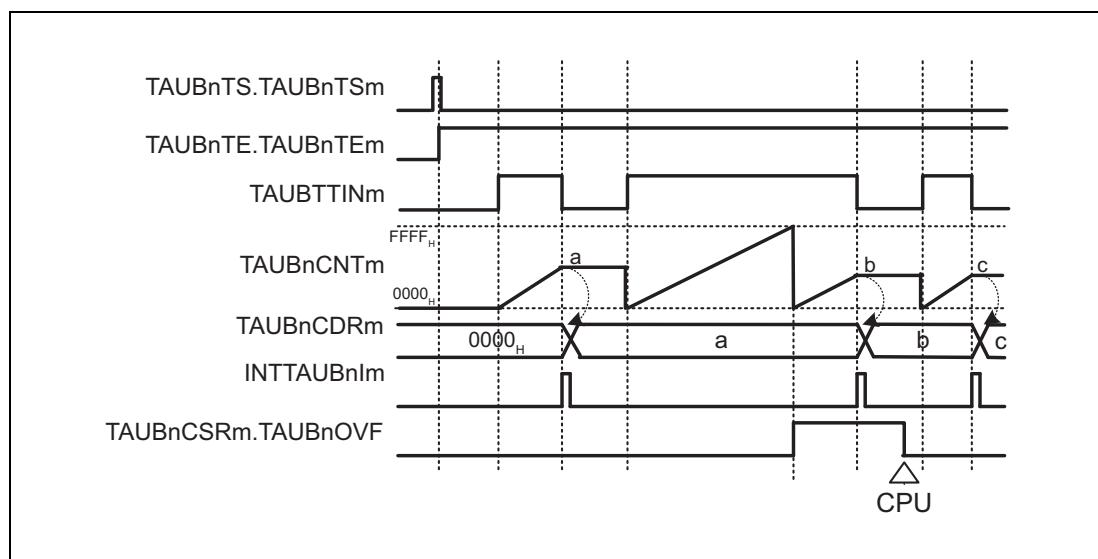


Figure 22.58 TAUBnCMORm.TAUBnCOS[1:0] = 01<sub>B</sub>, TAUBnCMORm.TAUBnMDO = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command (The TAUBnCSCm.TAUBnCLOV bit = 1).

**(3) TAUBnCMORm.TAUBnCOS[1:0] = 10<sub>B</sub>**

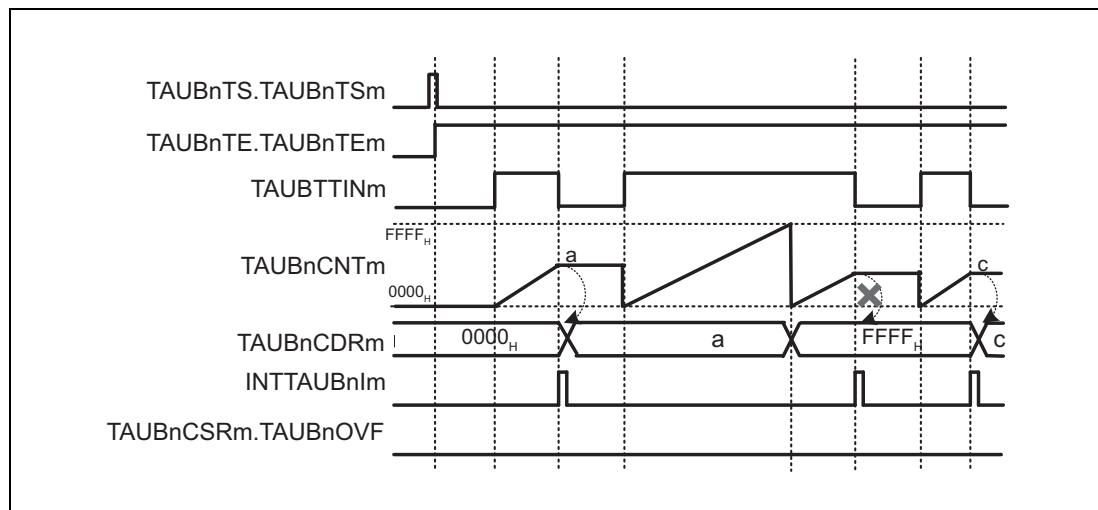


Figure 22.59 TAUBnCMORm.TAUBnCOS[1:0] = 10<sub>B</sub>, TAUBnCMORm.TAUBnMD0 = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, TAUBnCDRm is set to FFFF<sub>H</sub> and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

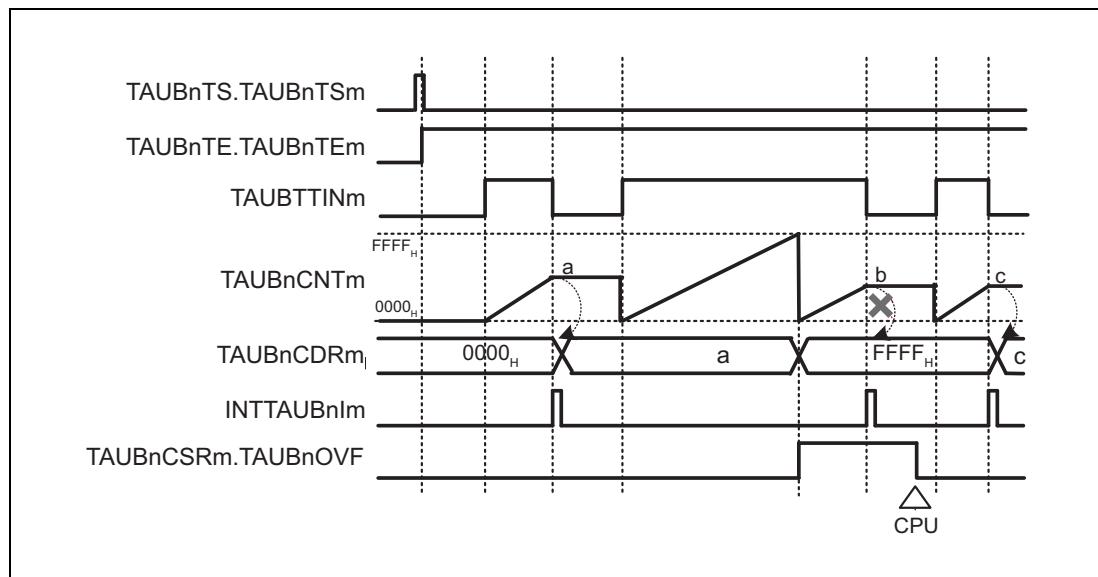
(4) TAUBnCMORm.TAUBnCOS[1:0] = 11<sub>B</sub>

Figure 22.60 TAUBnCMORm.TAUBnCOS[1:0] = 11<sub>B</sub>, TAUBnCMORm.TAUBnMDO = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, TAUBnCDRm is set to FFFF<sub>H</sub>, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

## (5) When an overflow occurs (high width measurement)

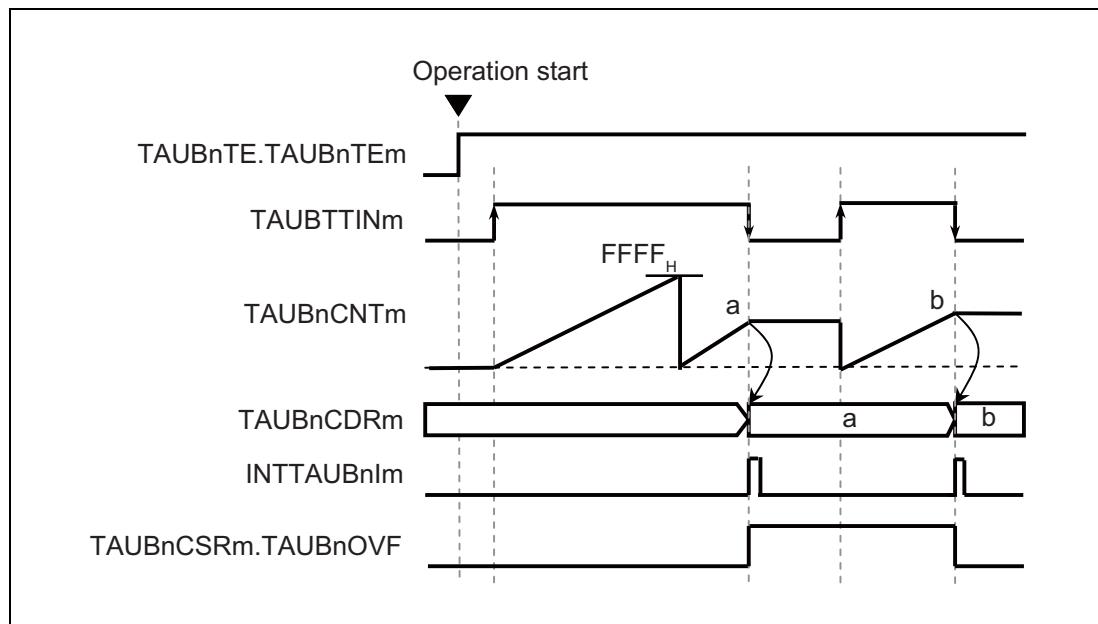


Figure 22.61 When an Overflow Occurs

When a capture trigger is input after the counter value has overflowed, the counter value is transferred to TAUBnCDRm and at the same time TAUBnCSRm.TAUBnOVF is set to 1.

TAUBnCSRm.TAUBnOVF is kept at 1 until the next capture trigger occurs.

If the next capture trigger is not accompanied by an overflow, TAUBnCSRm.TAUBnOVF is cleared to 0.

TAUBTTINm input signal width (example when TAUBnCSRm.TAUBnOVF is 1 and TAUBnCDRm is a)

$$\begin{aligned}
 &= \text{count clock cycle} \times ((10000_H \times \text{TAUBnCSRm.TAUBnOVF}) + (\text{TAUBnCDRm capture value} + 1)) \\
 &= \text{count clock cycle} \times ((10000_H \times 1) + (a+1)) \\
 &= \text{count clock cycle} \times (1000_H + a+1)
 \end{aligned}$$

## 22.12.8 TAUBTTINm Input Position Detection Function

### 22.12.8.1 Overview

#### Summary

在TAUBTTIN信号有效沿的时候捕获计数器的值来测量输入信号的时间间隔  
This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUBTTINm signal.

#### Prerequisites

TAUBTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The counter starts to count from 0000<sub>H</sub>.  
当计数器从0x0000开始，TAUBTTINm 输入停止  
When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The count operation continues.  
一旦计数器的值达到0xFFFF，计数器从0x0000开始计数  
When the counter reaches FFFF<sub>H</sub>, the counter restarts from 0000<sub>H</sub>.

#### NOTE

TAUBTTINm 输入信号以操作时钟被采样，结果，TOUTm的输出周期约有+/-1的误差

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTOUTm has an error of ± 1 operation clock cycle.

#### Conditions

如果MD0==0，启动后的第一个中断不能产生

If the TAUBnCMORm.MD0 bit is set to 0, the first interrupt after a start or restart is not generated.

### 22.12.8.2 Equations

Function duration at a TAUBTTINm input pulse =  
count clock cycle × (TAUBnCDRm capture value + 1)

### 22.12.8.3 Block Diagram and General Timing Diagram

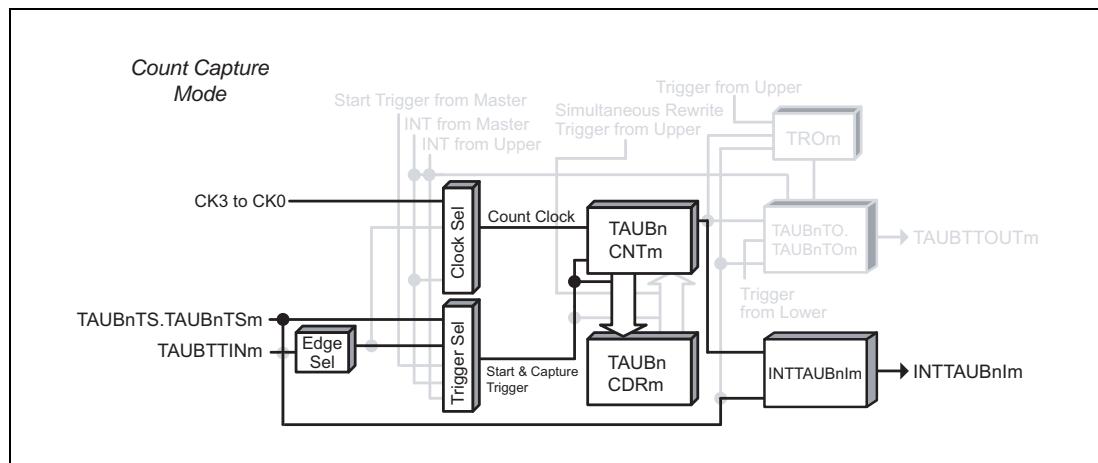


Figure 22.62 Block Diagram for TAUBTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

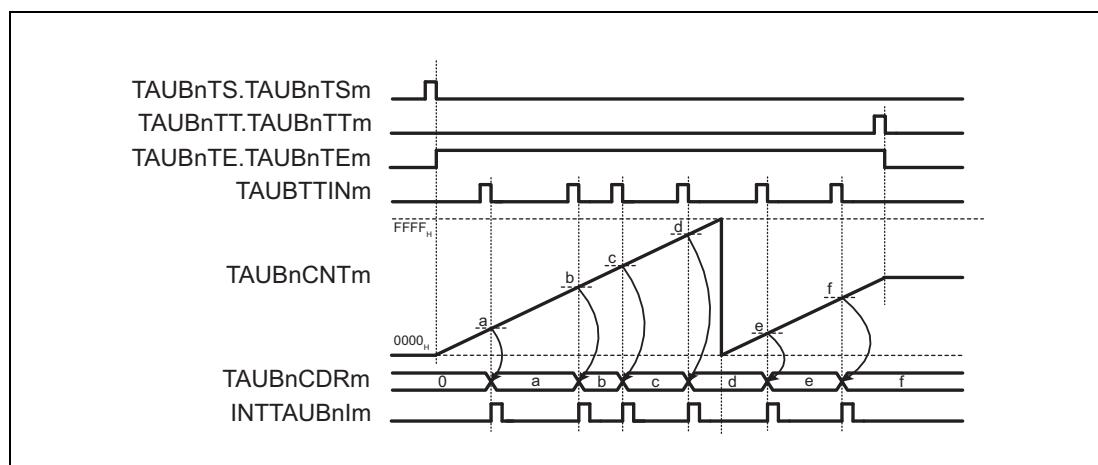


Figure 22.63 General Timing Diagram for TAUBTTINm Input Position Detection Function

### 22.12.8.4 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.70 Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 01 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1011 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.71 Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

### (3) Channel output mode

The channel output mode is not used by this function.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

**Table 22.72 Simultaneous Rewrite Settings for TAUBTTINm Input Position Detection Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.8.5 Operating Procedure for TAUBTTINm Input Position Detection Function

**Table 22.73 Operating Procedure for TAUBTTINm Input Position Detection Function**

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.70, Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function</b> and <b>Table 22.71, Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function</b>  The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMDO is set to 1.
During operation	The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time.	TAUBnCNTm starts to count up from 0000 <sub>H</sub> . When a TAUBTTINm valid edge is detected: <ul style="list-style-type: none"><li>• TAUBnCNTm transfers (captures) its value to TAUBnCDRm</li><li>• INTTAUBnIm is output.</li><li>• The counter value is not cleared to 0000<sub>H</sub> and TAUBnCNTm continues count operation.</li></ul> Afterwards, this procedure is repeated. If TAUBnCNTm reaches FFFF <sub>H</sub> , the counter restarts from 0000 <sub>H</sub> .
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

### 22.12.8.6 Specific Timing Diagrams

#### (1) Operation stop and restart

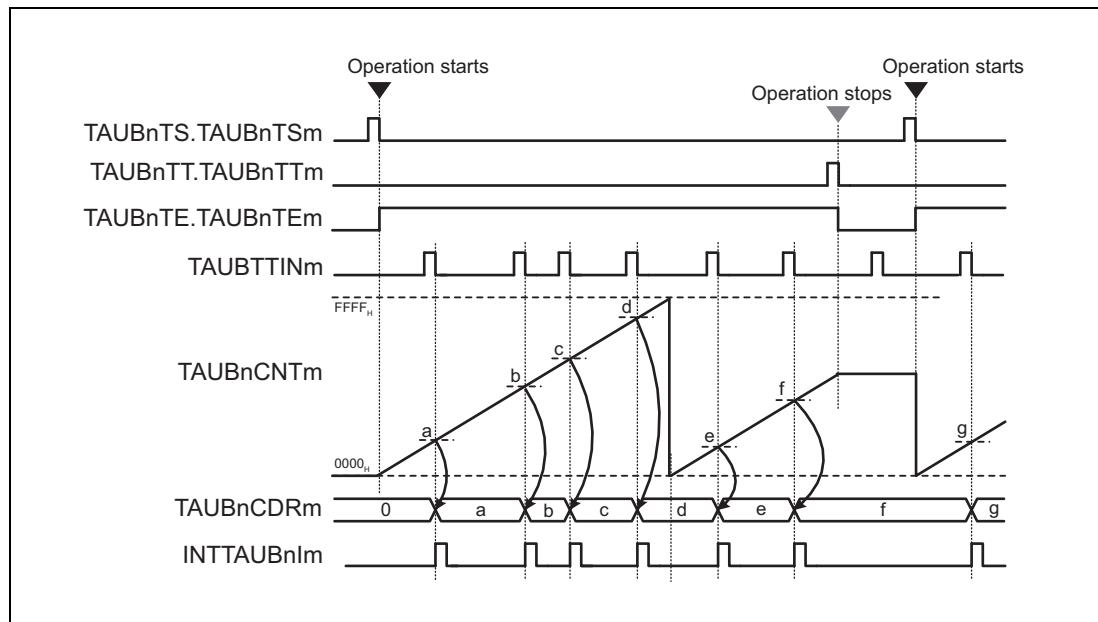


Figure 22.64 Operation Stop and Restart (TAUBnCMORm.TAUBnMD0 = 0,  
TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from  $0000_H$ .

## 22.12.9 TAUBTTINm Input Period Count Detection Function

### 22.12.9.1 Overview

#### Summary

**此功能测量输入信号的累计宽度**  
This function measures the cumulative width of a TAUBTTINm input signal.

#### Prerequisites

TAUBTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The counter awaits a valid TAUBTTINm input edge.

**一旦有效信号没检测到，计数器从0x0000开始计数**

When a valid TAUBTTINm input start edge is detected, the counter starts to count from 0000<sub>H</sub>.

**输入信号的停止沿被探测到，CNT的当前值被写入CDRm，中断也会产生。在下一次有效沿信号被你检**

When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter stops and retains its value (TAUBnCDRm + 1) until the next valid TAUBTTINm input start edge is detected.

**当下次有效输入信号沿被检测到的时候，计数器从停止时保留的值开始**

When a next valid TAUBTTINm input start edge is detected, the counter restarts from the value retained while stopping.

**如果达到0xFFFF，那么计数器会从0x0000开始**

If the counter reaches FFFF<sub>H</sub>, the counter restarts from 0000<sub>H</sub>.

#### NOTES

1. The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits.
2. As this function is to measure the TAUBTTINm input signal width, setting TAUBnTS.TAUBnTSm to 1 is disabled while TAUBnTE.TAUBnTEM = 1.

#### Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10<sub>B</sub>, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

### 22.12.9.2 Equations 累积TAUBTTINm输入宽度 =

Cumulative TAUBTTINm input width =

count clock cycle × (TAUBnCDRm capture value + 1)

### 22.12.9.3 Block Diagram and General Timing Diagram

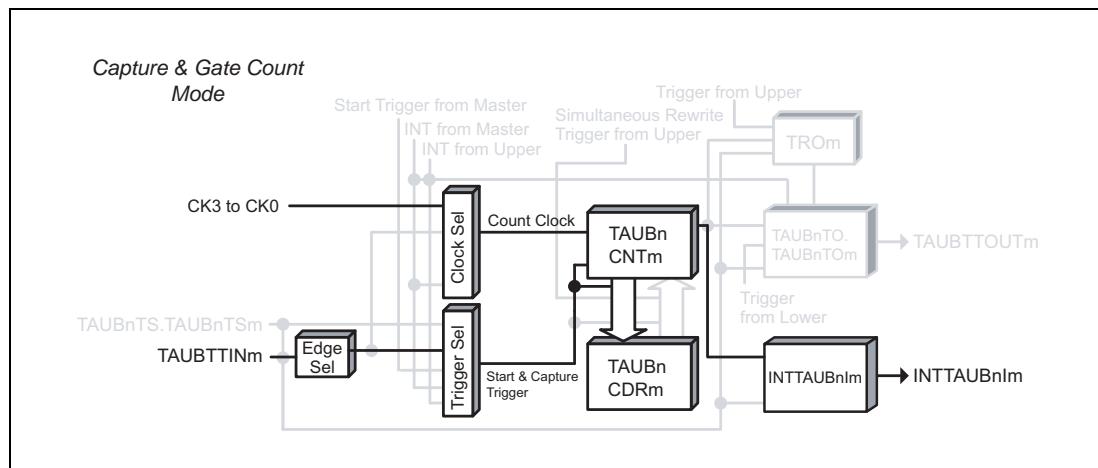


Figure 22.65 Block Diagram for TAUBTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
(TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>)

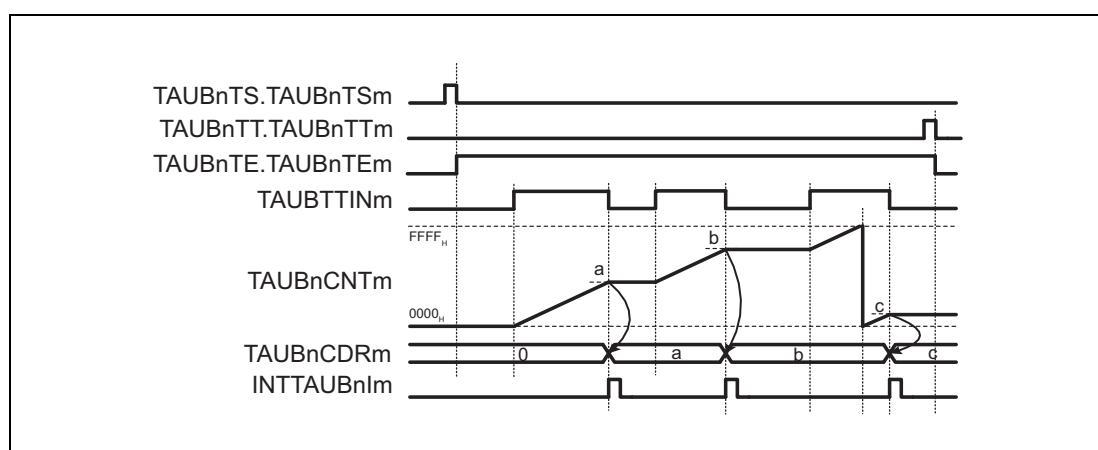


Figure 22.66 General Timing Diagram for TAUBTTINm Input Period Count Detection Function

### 22.12.9.4 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

R/W R/W R/W R R/W R/W R/W R/W R/W R/W R/W R R/W R/W R/W R/W R/W R/W

Table 22.74 Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 01 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1101 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0

R/W R R R R R R R/W R/W

Table 22.75 Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

#### (3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

**Table 22.76 Simultaneous Rewrite Settings for TAUBTTINm Input Period Count Detection Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.9.5 Operating Procedure for TAUBTTINm Input Period Count Detection Function

**Table 22.77 Operating Procedure for TAUBTTINm Input Period Count Detection Function**

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.74, Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function</b> and <b>Table 22.75, Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function</b>  The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEM is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	Detection of TAUBTTINm edges.  The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time.	When a TAUBTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUBnCNTm starts to count up from the stop value. When TAUBnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUBnCDRm and INTTAUBnIm is generated. Counting stops at the “value transferred to TAUBnCDRm + 1” value and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When TAUBnCNTm reaches FFFF <sub>H</sub> , the counter restarts from 0000 <sub>H</sub> . Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEM is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.



### 22.12.9.6 Specific Timing Diagrams

#### (1) Operation stop and restart

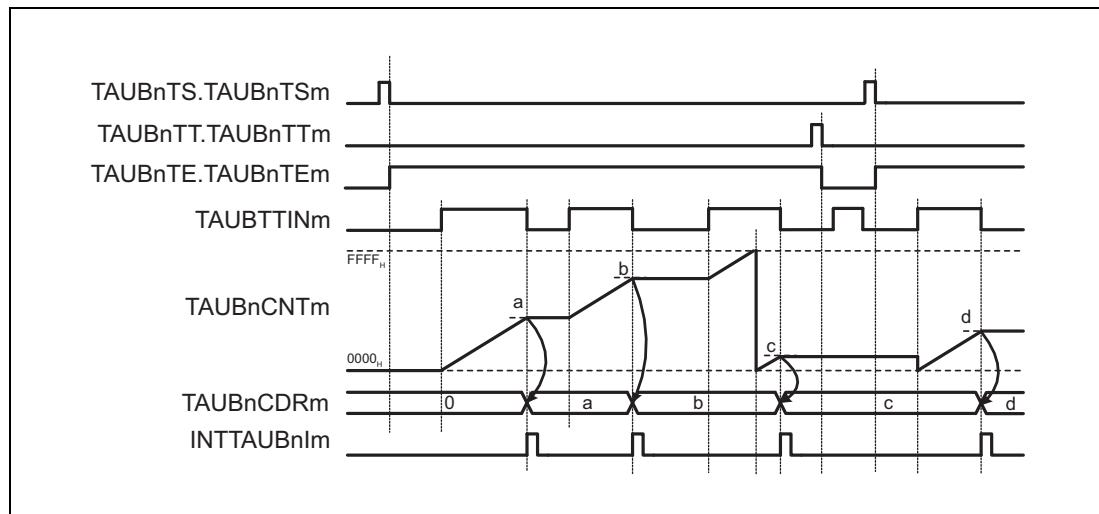


Figure 22.67 Operation Stop and Restart (TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000<sub>H</sub>.

## 22.12.10 TAUBTTINm Input Pulse Interval Judgment Function

### 22.12.10.1 Overview

#### Summary

当TAUBTTINm输入脉冲发生的时候，此功能输出CNTm和CDRm的比较结果。

This function outputs the result of a comparison between the count value (TAUBnCNTm) and the value in the channel data register (TAUBnCDRm) when a TAUBTTINm input pulse occurs. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true.  
一旦比较结果是true，那么INTTAUBnIm会产生。

#### Prerequisites

TAUBTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTS<sub>m</sub>) to 1. This in turn sets TAUBnTE.TAUBnTE<sub>m</sub> = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value. When a TAUBTTINm valid edge is detected or TAUBnTS.TAUBnTS<sub>m</sub> is set to 1, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. TAUBnCNTm reloads the value of TAUBnCDRm and subsequently continues operation, regardless of the result of the comparison.

如果比较结果为true，一个中断请求信号INTTAUBnIm就会产生，CNTm重载CDRm的值然后继续执行，这个跟比较的结果没有关系

如果在TAUBTTINm有效沿到来之前计数器达到0x0000<sub>H</sub>，那么CNTm溢出变为0xFFFF<sub>H</sub>，然后继续下减

If the counter reaches 0000<sub>H</sub> before a TAUBTTINm valid edge is detected, TAUBnCNTm overflows and is set to FFFF<sub>H</sub>. It then continues to count down.

CDRm可以在任何时候被写入，写入的新值在下一轮的启动下减的时候生效

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

#### Conditions

##### MD0 bit指定比较的类型

The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:

- If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
- If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.

### 22.12.10.2 Block Diagram and General Timing Diagram

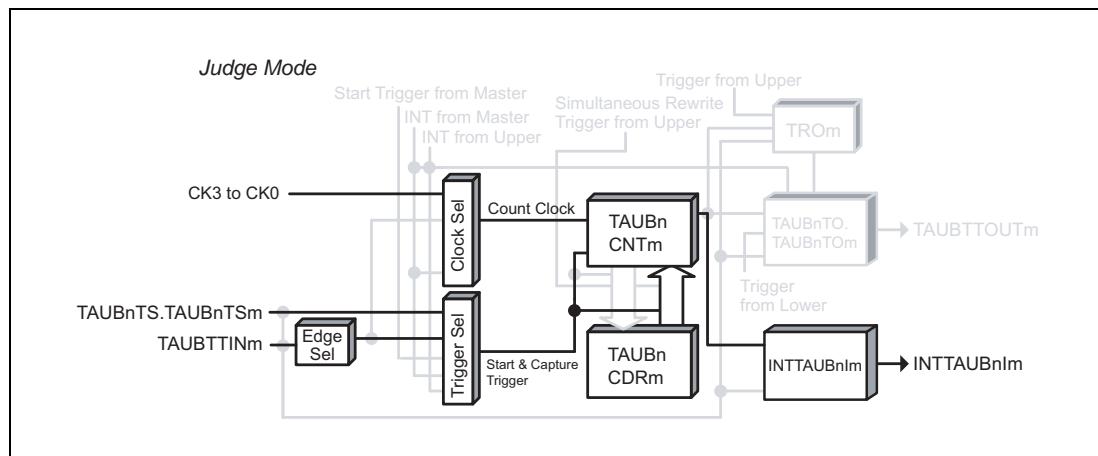


Figure 22.68 Block Diagram for TAUBTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

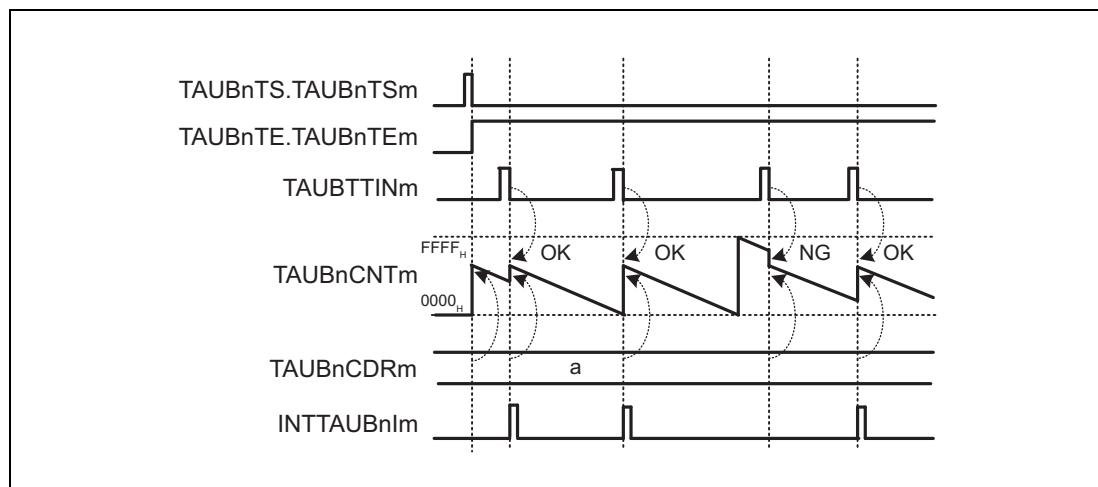


Figure 22.69 General Timing Diagram for TAUBTTINm Input Pulse Interval Judgment Function

### 22.12.10.3 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—		TAUBnMD[4:1]		TAUBn MDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

**Table 22.78** Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0001 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

**(2) TAUBnCMURm**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 22.79** Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

**(3) Channel output mode**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Judgment Function. Therefore, these registers must be set to 0.

**Table 22.80 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Judgment Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), sets these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.10.4 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

**Table 22.81 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function**

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.78, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function</b> and <b>Table 22.79, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function</b> Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value.
During operation	The following register can be changed at any time: <ul style="list-style-type: none"><li>• TAUBnCDRm register</li></ul>	When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated.  When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. If a TAUBTTINm input edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm.  Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.



**TAUBTTINm** 输入信号宽度判决功能

## 22.12.11 TAUBTTINm Input Signal Width Judgment Function

### 22.12.11.1 Overview

#### Summary

此功能是比较CNTm和CDRm的值来计算TAUBTTINm输入信号高电平或者低电平的宽度  
 This function compares the count value (TAUBnCNTm) for the high or low level width of a  
 TAUBTTINm input signal and the TAUBnCDRm value, and outputs the judgment result from the  
 interrupt request signal INTTAUBnIm.  
 在收到中断请求信号INTTAUBnIm的时候输出  
 判决信号

#### Prerequisites

TAUBTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. When a valid TAUBTTINm input start edge is detected, the current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value.

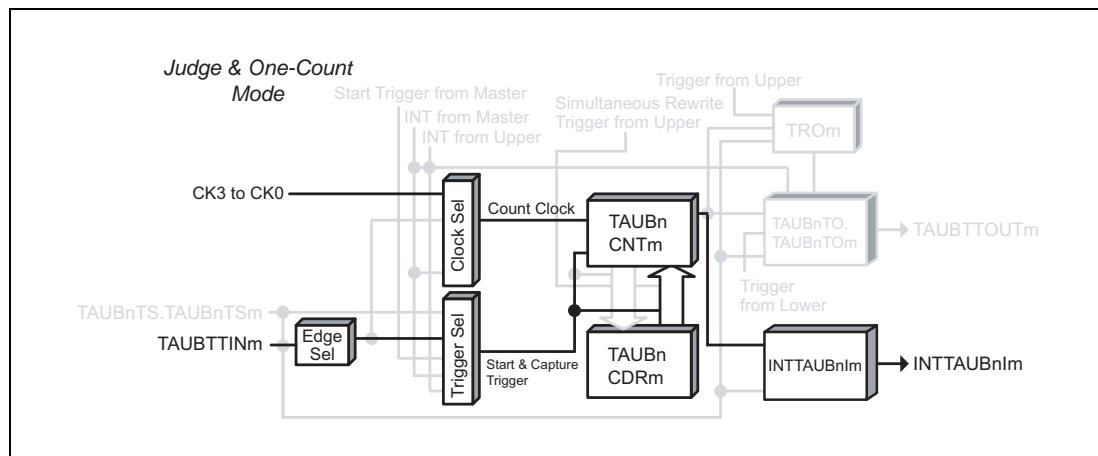
一旦有效的TAUBTTINm有效停止沿被检测到，此功能比较CNTm和CDR的当前值。  
 When a TAUBTTINm valid stop edge is detected, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. The counter TAUBnCNTm retains its value until the next TAUBTTINm valid start edge is detected, regardless of the result of the comparison.

如果在TAUBTTINm有效沿到来之前计数器达到0x0000,那么CNTm溢出变为0xFFFF,然后继续下减  
 If the counter reaches 0000<sub>H</sub> before a valid TAUBTTINm stop edge is detected, TAUBnCNTm overflows and is set to FFFF<sub>H</sub>. It then continues to count down.  
 CDRm可以在任何时候被写入，写入的新值在下一轮的启动下减的时候生效  
 The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

#### Conditions

- The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:
  - If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
  - If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.
- The TAUBnCMURm.TAUBnTIS[1:0] bits specify the type of width measurement:
  - TIS[1:0]指定宽度测量的类型**
  - 高电平宽度测量
    - For high width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>) the start edge is a rising TAUBTTINm edge and the stop edge is a falling TAUBTTINm edge.
  - 低电平宽度测量
    - For low width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 10<sub>B</sub>) the start edge is a falling TAUBTTINm edge and the stop edge is a rising TAUBTTINm edge.
- Forced restart is not possible for this function.  
**强制重启不能用于此功能**

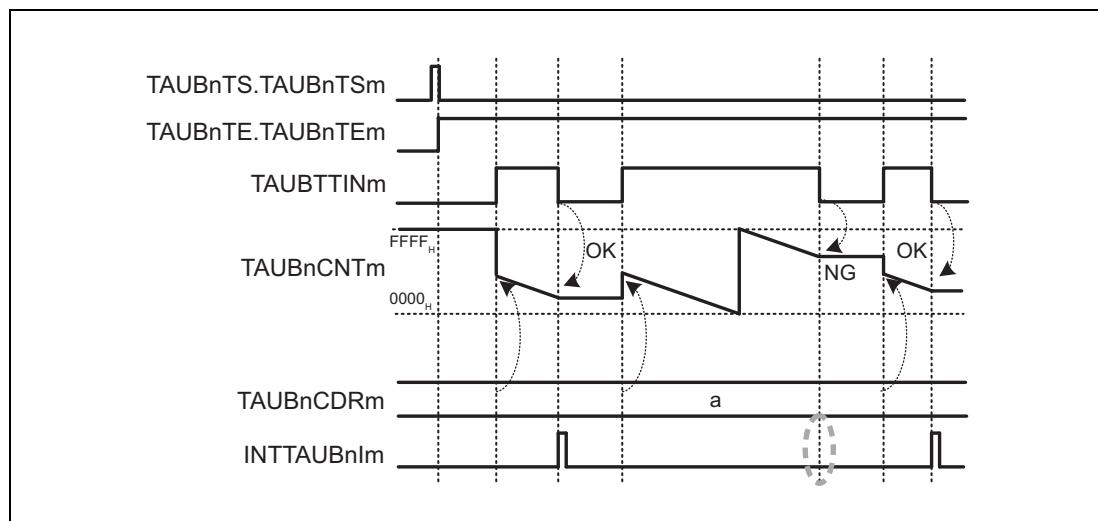
### 22.12.11.2 Block Diagram and General Timing Diagram



**Figure 22.70** Block Diagram for TAUBTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated when  $\text{TAUBnCNTm} \leq \text{TAUBnCDRm}$  ( $\text{TAUBnCMORM}. \text{TAUBnMD0} = 0$ )
- TAUBTTINm valid start edge = rising edge, TAUBTTINm valid stop edge = falling edge ( $\text{TAUBnCMURM}. \text{TAUBnTIS}[1:0] = 11_B$ )



**Figure 22.71** General Timing Diagram for TAUBTTINm Input Signal Width Judgment Function

### 22.12.11.3 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.82 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0111 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.83 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

### (3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Judgment Function. Therefore, these registers must be set to 0.

**Table 22.84 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Judgment Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), sets these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.11.4 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

**Table 22.85 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function**

	Operation	Status of TAUBn
Initial channel setting	<p>Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.82, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function</b> and <b>Table 22.83, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function</b></p> <p>Set the value of the TAUBnCDRm register</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	<p>The following register can be changed at any time:</p> <ul style="list-style-type: none"> <li>• TAUBnCDRm register</li> </ul>	<p>If a TAUBTTINm start edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm.</p> <p>When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated.</p> <p>When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm &gt; TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated.</p> <p>Afterwards, this procedure is repeated.</p>
Stop operation	<p>Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops.</p> <p>TAUBnCNTm stops and retains its current value.</p>

溢出中断输出功能 (在TAUBTTINm宽度测量 )

## 22.12.12 Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

### 22.12.12.1 Overview

#### Summary

此功能测量单独的TAUBTTINm输入信号的宽度。一旦输入宽度超过0xFFFF+1，中断就会产生  
This function measures the width of an individual TAUBTTINm input signal. An interrupt is generated  
if the TAUBTTINm input width is longer than  $FFFF_H + 1$ .

#### Prerequisites

- TAUBTTOUTm is not used for this function
- TAUBnCDRm的值可以被设定为0xFFFF  
The value of TAUBnCDRm must be set to FFFF<sub>H</sub>.

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation.

**当一个有效的TAUBTTINm输入起始信号被检测到的时候，计数器开始工作。** **0xFFFF 被写入TAUBnCNT**  
The counter starts when a valid TAUBTTINm input start edge is detected.  $FFFF_H$  is written to TAUBnCNTm and the counter starts to count down.

**当有效的停止沿被检测到，计数器停止并且保留当前的值**  
When a valid stop edge is detected, the counter stops and retains the current value.

**当下一次的TAUBTTINm输入信号被检测到，CNTm重新装载0xFFFF然后开始下减**  
When the next TAUBTTINm input start edge is detected, TAUBnCNTm loads  $FFFF_H$  and starts to count down.

**在停止沿检测到之前，计数器达到0x0000，中断就会产生**  
If the counter reaches  $0000_H$  before a stop edge is detected, an interrupt is generated.

#### Conditions

**起始可停止沿的指定由TIS[1:0]来决定**

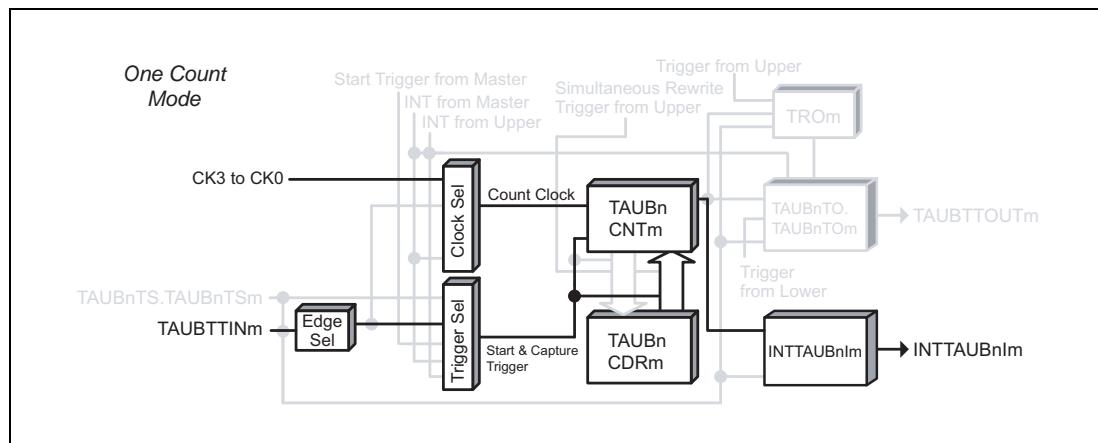
The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10<sub>B</sub>, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

#### NOTE

The counter cannot be restarted during operation.

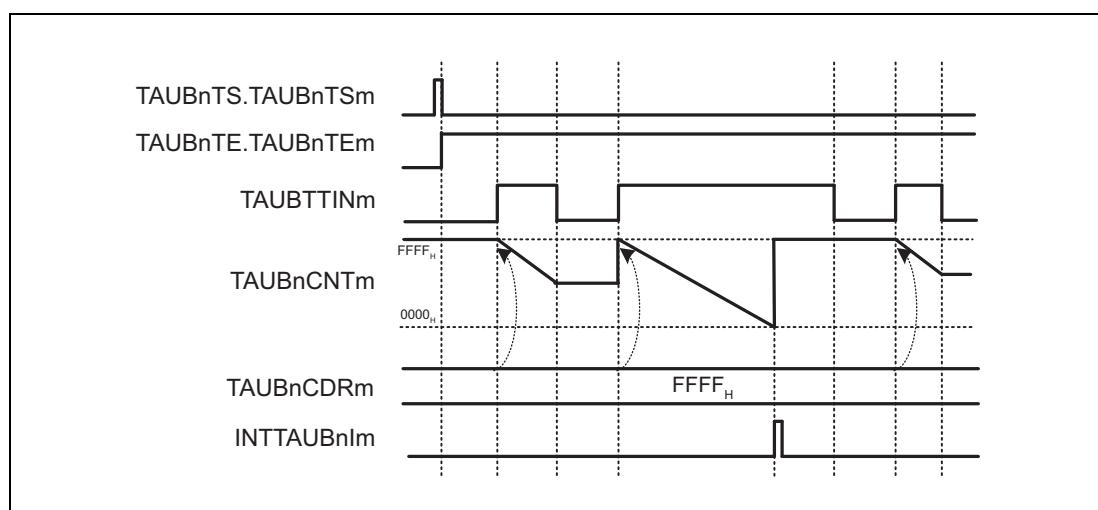
### 22.12.12.2 Block Diagram and General Timing Diagram



**Figure 22.72 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)**

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
(TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>)



**Figure 22.73 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)**

### 22.12.12.3 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

**Table 22.86** Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 0 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 22.87** Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

**(3) Channel output mode**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

**(4) Simultaneous rewrite**

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Width Measurement). Therefore, these registers must be set to 0.

**Table 22.88 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.12.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Table 22.89 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

	Operation	Status of TAUBn
Initial channel setting	<p>Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.86, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)</b> and <b>Table 22.87, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)</b></p> <p>Set the value of the TAUBnCDRm register to <math>FFFF_H</math>.</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p> <p>Detection of TAUBTTINm start edge</p>	<p>TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the start edge.</p> <p>When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value (<math>FFFF_H</math>).</p>
During operation	The TAUBnCNTm register can be read at any time.	<p>TAUBnCNTm counts down. When the counter reaches <math>0000_H</math>:</p> <ul style="list-style-type: none"> <li>INTTAUBnIm is generated</li> </ul> <p>When a TAUBTTINm input stop edge is detected during count operation:</p> <ul style="list-style-type: none"> <li>TAUBnCNTm stops and retains its current value.</li> </ul> <p>When a TAUBTTINm input start edge is detected while the counter is stopped:</p> <ul style="list-style-type: none"> <li>The TAUBnCDRm value (<math>FFFF_H</math>) is loaded to TAUBnCNTm again and the counter starts to count down.</li> </ul> <p>Afterwards, this procedure is repeated.</p>
Stop operation	<p>Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops.</p> <p>TAUBnCNTm stops and retains its current value.</p>

Restart operation

**溢出中断输出功能****在TAUBTTINm输入周期计算检测**

## 22.12.13 Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

### 22.12.13.1 Overview

#### Summary

**此功能测量TAUBTTINm输入信号的累积宽度**

This function measures the cumulative width of a TAUBTTINm input signal. An interrupt is generated and an overflow interrupt can be output if the cumulative TAUBTTINm input width is longer than  $FFFF_H$ . **如果累积宽度长于 $0xFFFF$ 那么中断就会产生，并且溢出中断就会输出**

#### Prerequisites

- TAUBTTOUTm is not used for this function  
**CDRm的值必须设置为 $0xFFFF$**
- The value of TAUBnCDRm must be set to  $FFFF_H$

#### Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation.

**当一个有效的TAUBTTINm输入起始信号被检测到的时候，计数器开始工作。 $0xFFFF$  被写入TAUBnCNT**  
The counter starts when a valid TAUBTTINm input start edge is detected.  $FFFF_H$  is written to TAUBnCNTm and the counter starts to count down.

**当一个有效的停止沿被检测到的时候，计数器停止计数并且保留当前的值，计数器等到下一个TAUBTTINm输入起始信号并且继续从当前值开始下减**  
When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUBTTINm input start edge and then continues to count down from the current value.

When the counter reaches  $0000_H$  an interrupt is generated.  $FFFF_H$  is written to TAUBnCNTm and the counter continues to count down until a TAUBTTINm input stop edge is detected.

#### Conditions

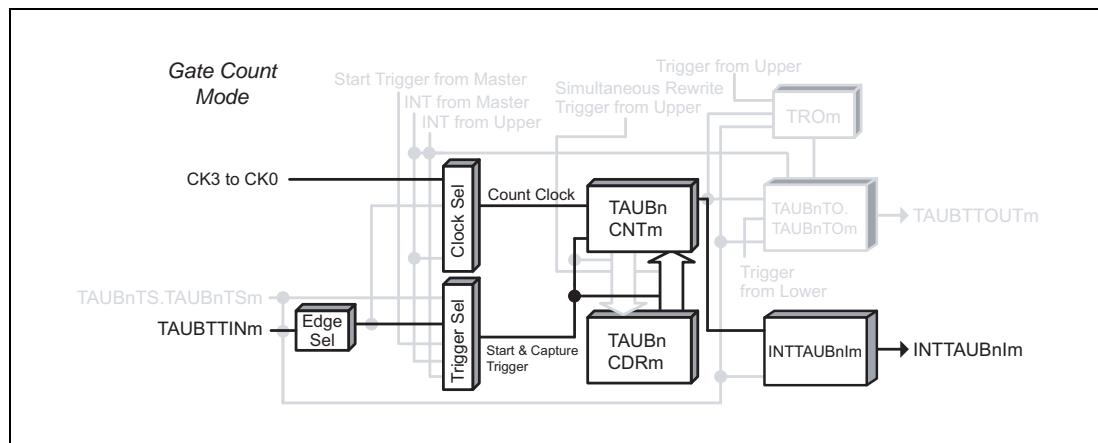
The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] =  $10_B$ , the TAUBTTINm input low width is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] =  $11_B$ , the TAUBTTINm input high width is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

#### NOTE

The counter cannot be restarted during operation.

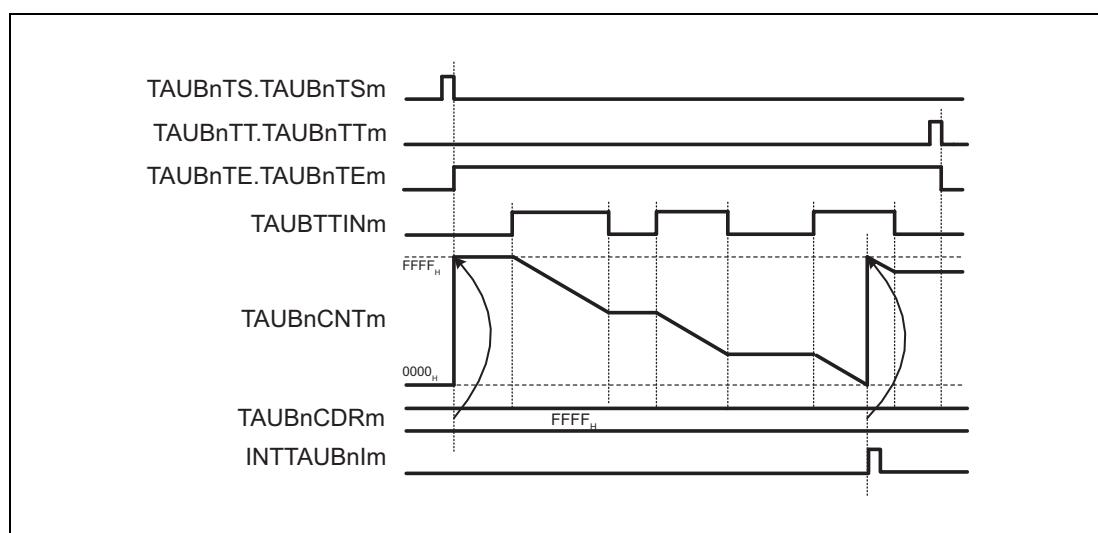
### 22.12.13.2 Block Diagram and General Timing Diagram



**Figure 22.74** Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
(TAUBnCMURm.TAUBnTIS[1:0] = 11<sub>B</sub>)



**Figure 22.75** General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

### 22.12.13.3 Register Settings

#### (1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MD0						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.90 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1100 <sub>B</sub> .
0	TAUBnMD0	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.91 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

**(3) Channel output mode**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

**(4) Simultaneous rewrite**

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

**Table 22.92 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

#### 22.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Table 22.93 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in <b>Table 22.90, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)</b> and <b>Table 22.91, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)</b>	Channel operation is stopped.
Start operation	<p>Set the value of the TAUBnCDRm register to <math>\text{FFFF}_H</math>.</p> <p>Set TAUBnTS.TAUBnTSm to 1 TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p> <p>Detection of TAUBTTINm start edge</p>	<p>TAUBnTE.TAUBnTEm is set to 1. TAUBnCNTm waits for detection of the start edge.</p> <p>When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value (<math>\text{FFFF}_H</math>).</p>
During operation	The TAUBnCNTm register can be read at all times.	<p>TAUBnCNTm counts down. When the counter reaches <math>0000_H</math>:</p> <ul style="list-style-type: none"> <li>INTTAUBnIm is generated</li> <li>TAUBnCNTm loads the TAUBnCDRm value (<math>\text{FFFF}_H</math>) and continues count operation</li> </ul> <p>When a TAUBTTINm input stop edge is detected during count operation:</p> <ul style="list-style-type: none"> <li>TAUBnCNTm stops and retains its current value.</li> </ul> <p>When a TAUBTTINm input start edge is detected while the counter is stopped:</p> <ul style="list-style-type: none"> <li>TAUBnCNTm starts to count down from the stop value.</li> </ul> <p>Afterwards, this procedure is repeated.</p>
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops.</p> <p>TAUBnCNTm stops and retains its current value.</p>



**独立信道的同时复写功能**

## 22.13 Independent Channel Simultaneous Rewrite Functions

The following describes functions that carry out simultaneous rewrite:

**同时复写触发产生功能类型1**

### 22.13.1 Simultaneous Rewrite Trigger Generation Function Type 1

#### 22.13.1.1 Overview

##### Summary

**在指定信道上产生的中断可以被低阶channel作为复写的触发信号**

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals. The upper channel is for generating the simultaneous rewrite trigger (TAUBnRDC.TAUBnRDCm = 1), and the lower channels are for conducting simultaneous rewrite when triggered from the upper channel (TAUBnRDC.TAUBnRDCm = 0).

##### Prerequisites

**2个或者多个channel低于高阶channel，每一个都使能同时复写**

- Two (or more) channels that are lower than the channel used as the upper channel, each with simultaneous rewrite enabled (TAUBnRDE.TAUBnRDEm = 1)
- 高阶channel的操作模式必须设置为Interval timer mode
- The operation mode of the upper channel must be set to interval timer mode, see **Table 22.94, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1.**
- For the operation modes that can be set to the lower channels, see **Table 22.37, Channel Functions and the Methods They Use for Simultaneous Rewrite.**
- 对于此功能，TOUTm不能用于所有的channels
- In this function, TAUBTOUTm is not used for all the channels.

##### Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) of the upper and lower channel(s) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The current value of the data register buffer of the upper channel (TAUBnCDRM buf) is written to the counter (TAUBnCNTM) and the counter starts to count down from this value.

The counter(s) of the lower channel(s) start to count as specified by their selected operating modes.

**当计数器达到 0x0000 中断就会产生，对应的TAUBnCNTM重载当前的CDRM的值然后继续执行**  
When a counter reaches 0000<sub>H</sub>, an interrupt is generated from the channel.

The corresponding TAUBnCNTM then reloads the current TAUBnCDRM buffer value and subsequently continues operation.

**如果中断产生的channel作为发生同时复写的触发channel并且是一个upper channel，那么同时复写就会在所有的低阶channel产生（前提是RSF被使能）**  
If the channel where the interrupt occurs is specified as the trigger channel for simultaneous rewrite (TAUBnRDC.TAUBnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUBnRSF.TAUBnRSFm = 1).

**数据寄存器的值被copy到对应的数据寄存器的buffer，一旦计数器开始下减，他会读取数据buf**  
The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

**数据寄存器的值可以随时改变，只是在同时复写产生的时候必须被传递至数据寄存器的buffer**  
The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

### Condition

- 监测INTTAUBnIm的channel，由RDC = 1来指定  
The channel which is monitored for INTTAUBnIm is specified by setting TAUBnRDC.TAUBnRDCm = 1 for the corresponding channel. The TAUBnRDC.TAUBnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.  
对于需要执行复写的channel RDC必须为0

#### 22.13.1.2 Equations

同时复写触发产生的周期 =

Simultaneous rewrite trigger generation cycle = count clock cycle  $\times$  (TAUBnCDRm + 1)

To control simultaneous rewrite, the following condition must be satisfied:

**[For PWM]**

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1)  $\times$  number of interrupts] – 1

**[For triangle PWM]**

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1)  $\times$  2  $\times$  number of interrupts] – 1

That is, the ratio of TAUBnCDRm + 1 and TAUBnCDRm\_master + 1 must be an integer. This integer corresponds to the number of interrupts.

Note that the cycle for the triangle PWM is twice the cycle for the PWM

### 22.13.1.3 Block Diagram and General Timing Diagram

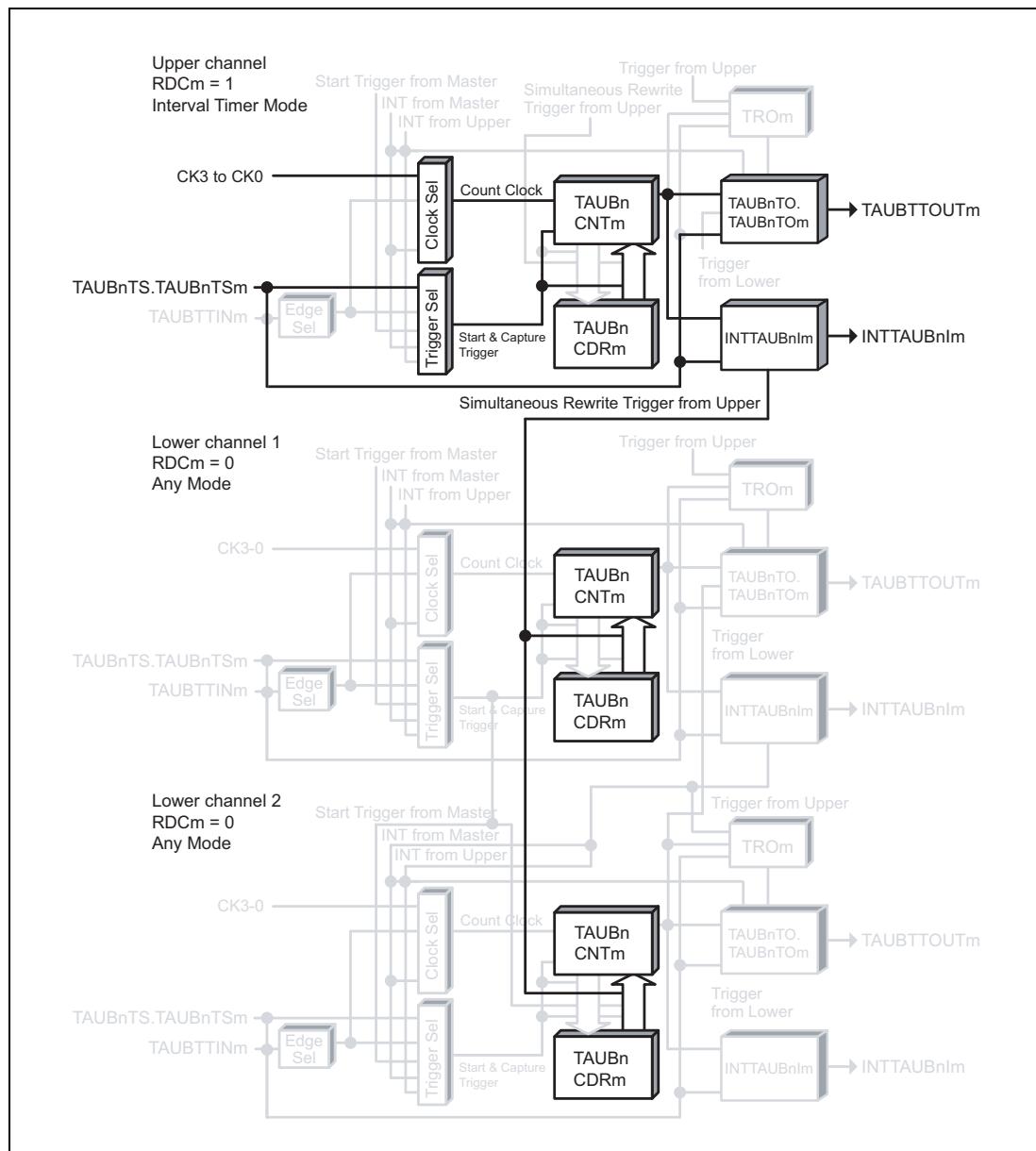


Figure 22.76 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

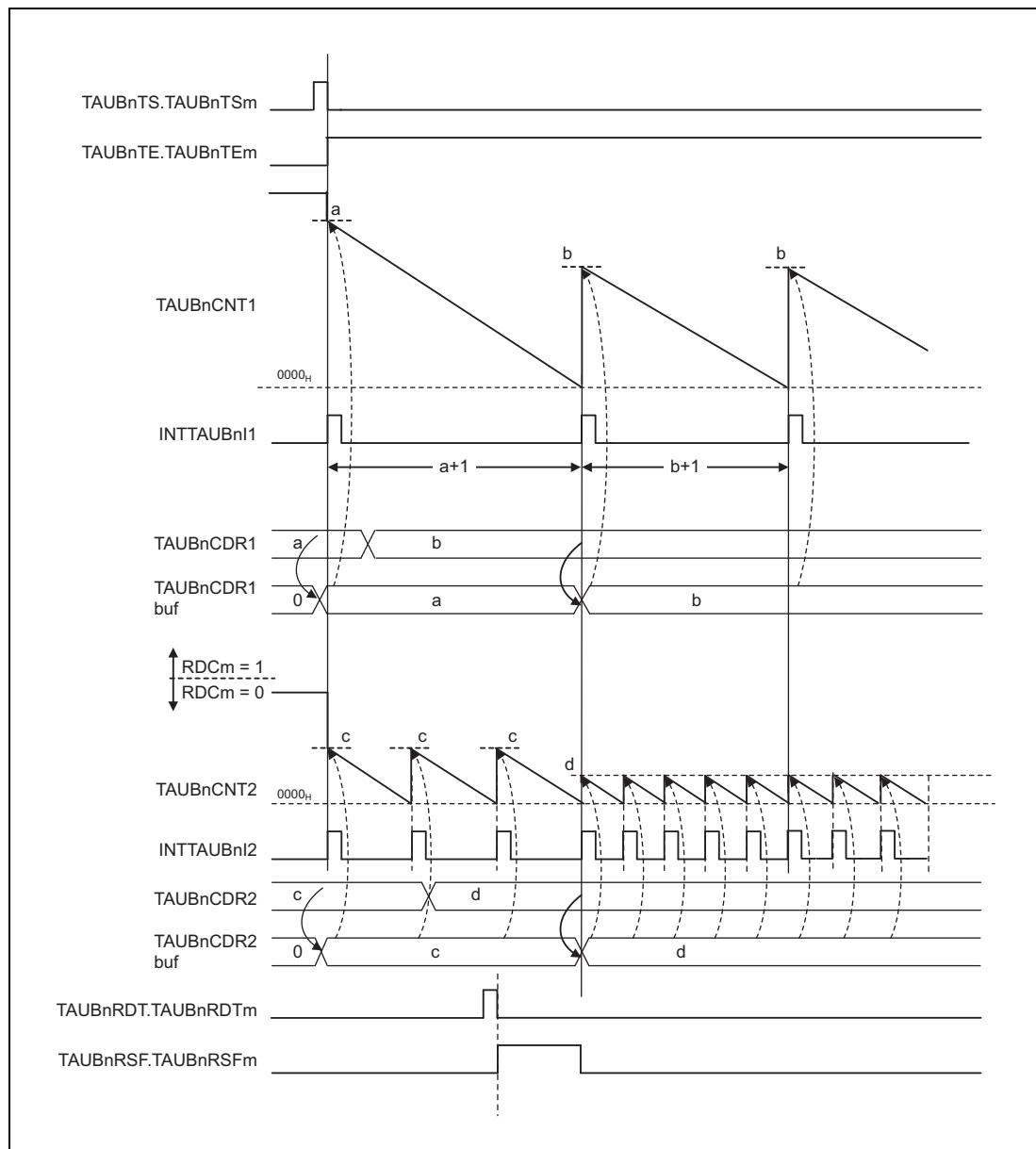


Figure 22.77 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

### 22.13.1.4 Register Settings for The Upper Channel

#### (1) TAUBnCMORm for the upper channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.94 Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for the upper channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.95 Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

**(3) Channel output mode for the upper channel**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function. However, it can be used in independent channel output mode controlled by software.

**(4) Simultaneous rewrite for the upper channel**

**Table 22.96 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	1: Channel is monitored for an INTTAUBnlm signal that is used as the simultaneous rewrite trigger

**22.13.1.5 Register Settings for the Lower Channel(s)****(1) TAUBnCMORm for the lower channel(s)**

For the TAUBnCMORm register of the lower channels, follow the TAUBnCMORm register settings for the operation mode that can be set. (See **Table 22.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**)

**(2) TAUBnCMURm for the lower channel(s)**

For the TAUBnCMURm register of the lower channels, follow the TAUBnCMURm register settings for the operation mode that can be set. (See **Table 22.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**)

**(3) Channel output mode for the lower channel(s)**

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 22.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

**(4) Simultaneous rewrite for the lower channel(s)**

**Table 22.97 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 1**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

#### **22.13.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1**

**Table 22.98** Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Operation	Status of TAUBn	
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers for the upper channel as described in <b>Table 22.94, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1</b> and <b>Table 22.95, Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1</b>  Set the TAUBnCMORm and TAUBnCMURm registers for the lower channel as described in <b>Section 22.13.1.5, Register Settings for the Lower Channel(s)</b>  Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.
During operation	TAUBnRDT.TAUBnRDTm, TAUBnCDR.TAUBnCDRm can be changed. TAUBnRSF.TAUBnRSFm can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000H: <ul style="list-style-type: none"> <li>• TAUBnCNTm reloads the TAUBnCDRm value and continues count operation</li> <li>• INTTAUBnIm is generated</li> </ul> Simultaneous rewrite is controlled when INTTAUBnIm is generated from the channel where TAUBnRDC.TAUBnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

## 22.14 Synchronous Channel Operation Functions

本节列出了所有有关Timer Array Unit B 提供的操作模式

This section lists all the synchronous channel operation functions provided by the Timer Array Unit B.

For a general overview of synchronous channel operation, see **Section 22.2, Overview**.

### 22.14.1 PWM Output Function PWM输出功能

#### 22.14.1.1 Overview

##### Summary

此功能使用master和多个slave channel 产生多个PWM输出

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the pulse width (duration) of the TAUBTTOUTm to be set. The pulse cycle is set in the master channel. The pulse width is set in the slave channel.

##### Prerequisites

- Two channels
- The operation mode of the master channel must be set to **interval timer mode**, see **Table 22.99, Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function**
- The operation mode of the slave channel(s) must be set to **one-count mode**, see **Table 22.102, Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function**
- **TAUBTTOUTm is not used for the master channel of this function**
- The channel output mode of the slave channel(s) must be set to **synchronous channel output mode 1**.

##### Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counters start to count down from these values. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) toggles, which realizes a PWM output.

- Master channel:

When the counter of the master channel reaches  $0000_H$ , pulse cycle time has elapsed and INTTAUBnIm is generated. The counter loads the TAUBnCDRm value and counts down.

- Slave channel:

INTTAUBnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUBnCDRm (slave) is written to TAUBnCNTm (slave) and the counter starts to count down from this value. The TAUBTTOUTm signal is set to the active level.

When the counter reaches  $0000_H$ , i.e. duty time has elapsed, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset to the inactive level. The counter returns to  $FFFF_H$  and awaits the next INTTAUBnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEM to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

### Conditions

Simultaneous rewrite can be used with this function. Please see **Section 22.6, Simultaneous Rewrite**.

#### 22.14.1.2 Equations

$$\text{Pulse cycle} = (\text{TAUBnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$\text{Duty cycle [%]} = (\text{TAUBnCDRm (slave)}) / (\text{TAUBnCDRm (master)} + 1) \times 100$$

- Duty cycle = 0%
  - TAUBnCDRm (slave) = 0000<sub>H</sub>
- Duty cycle = 100%
  - TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

#### 22.14.1.3 Block Diagram and General Timing Diagram

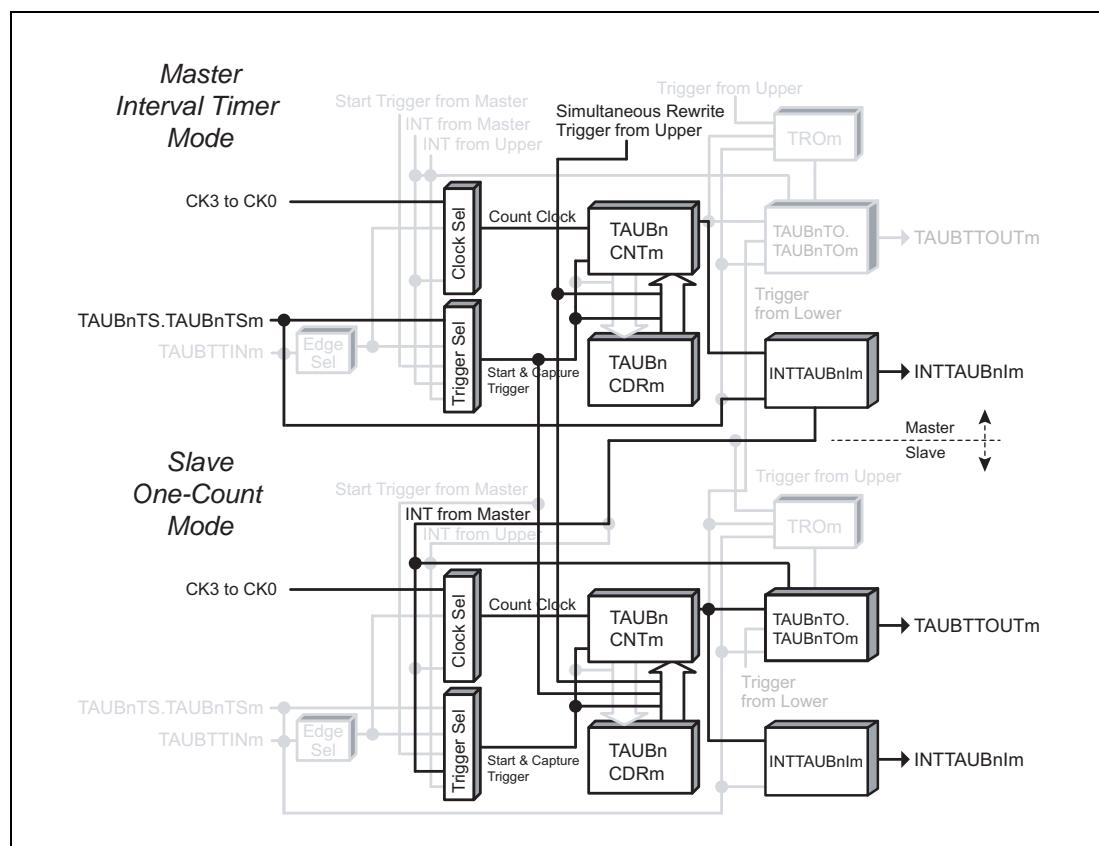


Figure 22.78 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

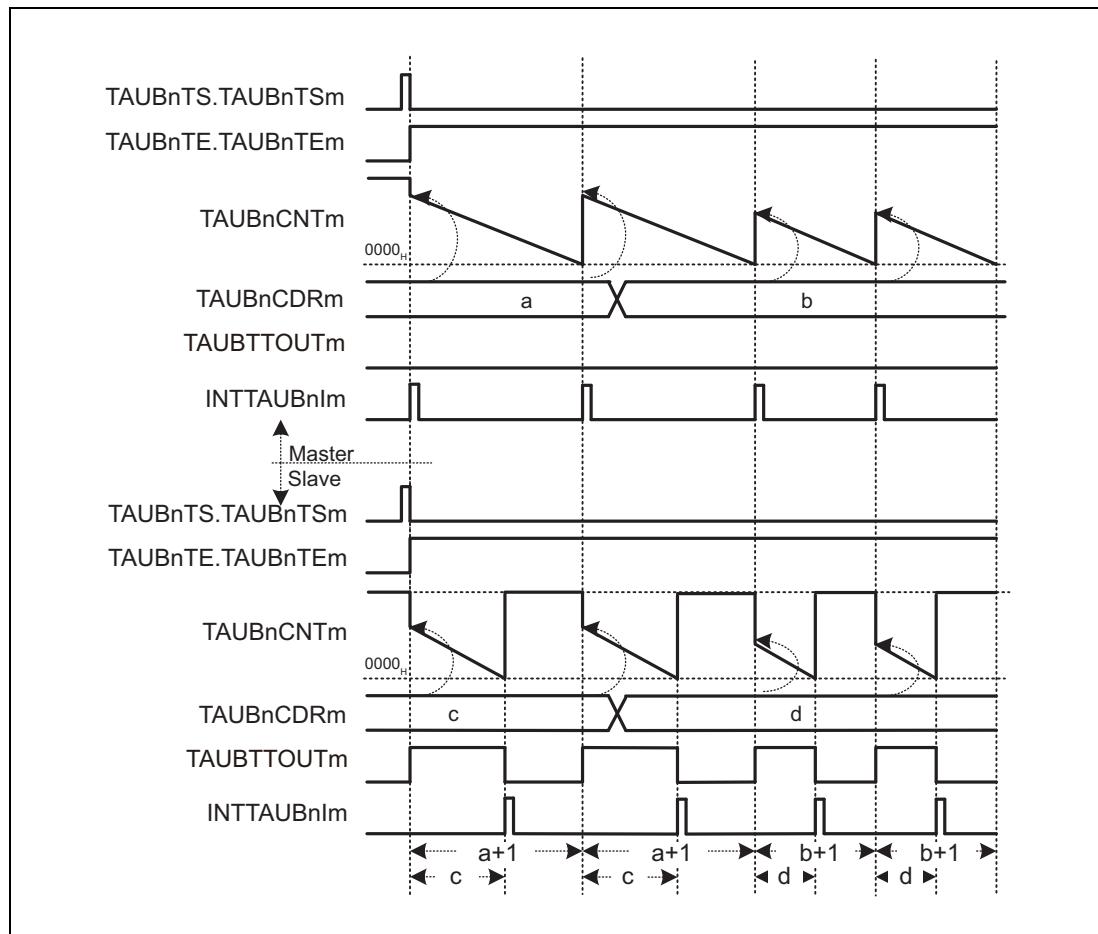


Figure 22.79 General Timing Diagram for PWM Output Function

#### NOTE

The interval between the starting to count and an interrupt being generated is the value of corresponding TAUBnCDRm + 1.

#### 22.14.1.4 Register Settings for the Master Channel

##### (1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.99 Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 1 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

##### (2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.100 Contents of the TAUBnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for the master channel

The channel output mode is not used by this function.

### (4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.101 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

#### NOTE

When used in TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel operating in **Section 22.13.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Configure the operation following the conditions below.

- The channel set to Simultaneous Rewrite Trigger Output Function Type 1: TAUBnRDCm = 1, TAUBnRDSm = 1

The setting value of TAUBnCDRm to this channel is as follows.

$$= ((\text{setting value of TAUBnCDRm of the master channel subject to simultaneous rewrite} + 1) \times \text{number of interrupts}) - 1$$

- Master channel: TAUBnRDCm = 0, TAUBnRDSm = 1
- Slave channel: TAUBnRDCm = 0, TAUBnRDSm = 1

Although the value of duty exceeds 100% when the setting value of TAUBnCDRm (slave) > the setting value of TAUBnCDRm (master) + 1, the output will be aggregated to 100%.

### 22.14.1.5 Register Settings for the Slave Channel(s)

#### (1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.102 Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 100 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.103 Contents of the TAUBnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for the slave channel(s)

**Table 22.104 Control Bit Settings for Independent Channel Output Mode 1**

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 1 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 0 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEM	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

### (4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.105 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEM	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.1.6 Operating Procedure for PWM Output Function

Table 22.106 Operating Procedure for PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	<p>Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.1.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.1.5, Register Settings for the Slave Channel(s)</b>.</p> <p>Set the values of the TAUBnCDRm registers of all channels</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEM (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnlm is generated on the master channel and TAUBTTOUTm (slave) is set.</p>
During operation	<p>TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time.</p> <p>TAUBnRDT.TAUBnRDTm can be changed during operation.</p>	<p>TAUBnCNTm of the master channel loads TAUBnCDRm and counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (master) is generated</li> <li>TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation</li> <li>TAUBnCNTm (slave) loads the TAUBnCDRm value and counts down</li> <li>TAUBTTOUTm (slave) is set to the active level</li> </ul> <p>When TAUBnCNTm (slave) reaches 0000H:</p> <ul style="list-style-type: none"> <li>The counter of TAUBnCNTm (slave) stops.</li> <li>INTTAUBnlm (slave) is generated</li> <li>TAUBTTOUTm (slave) is set to the inactive level</li> </ul>
Stop operation	<p>Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEM is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.</p>

Restart operation



### 22.14.1.7 Specific Timing Diagrams

#### (1) Duty cycle = 0%

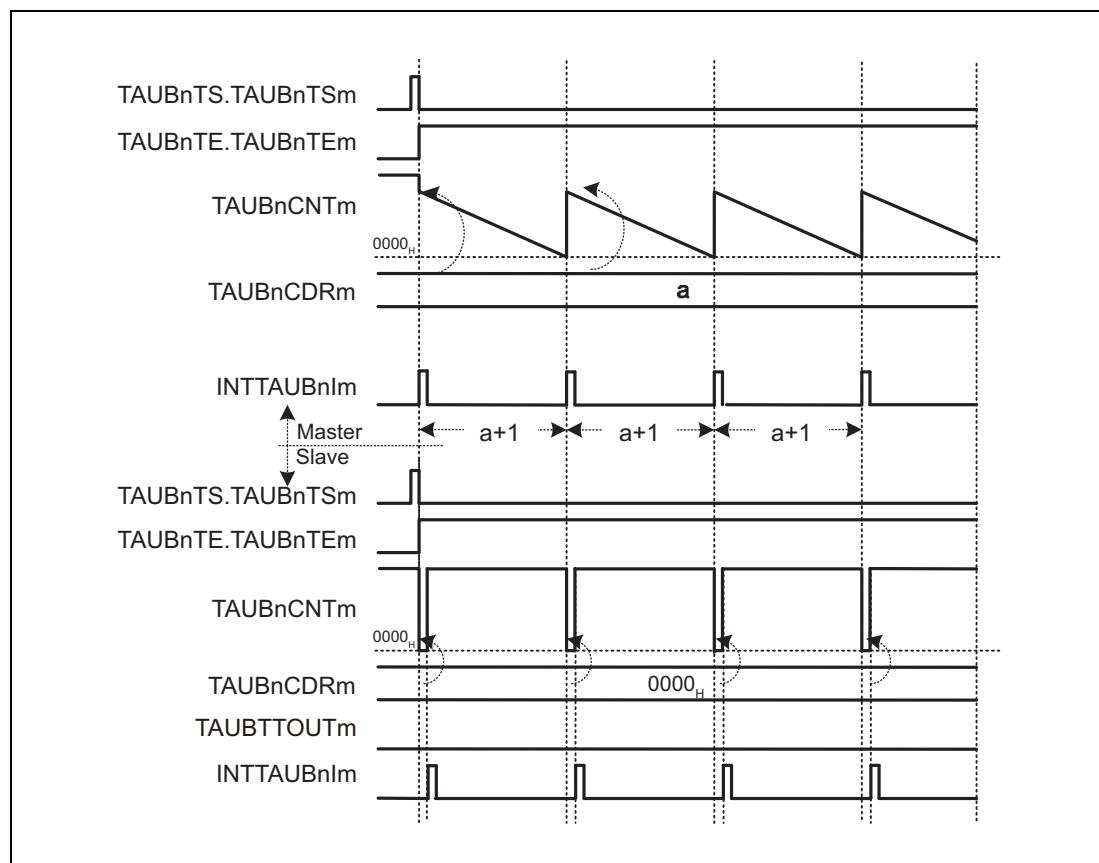
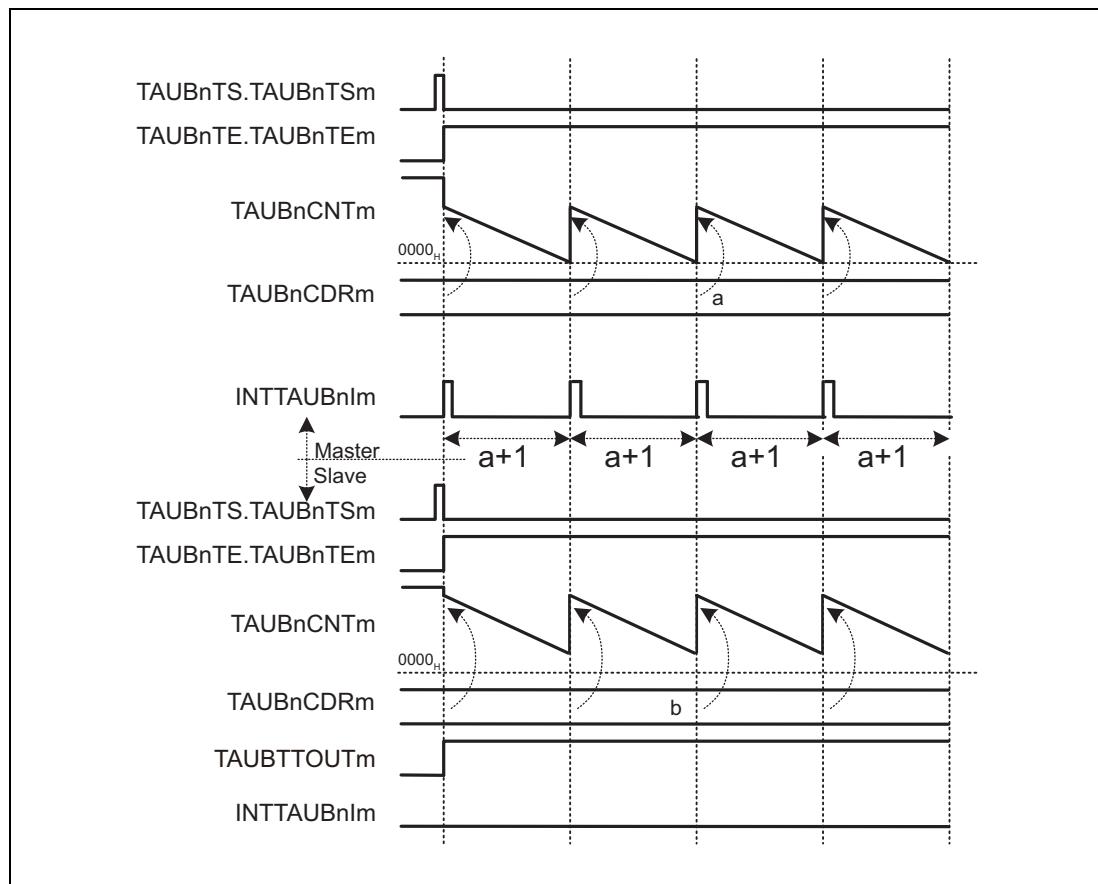


Figure 22.80 TAUBnCDRm (slave) =  $0000_H$ ,  
Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUBnIm),  $0000_H$  is written to TAUBnCNTm (slave). As a result, a slave channel interrupt (INTTAUBnIm) is generated at the same time and TAUBTTOUTm remains inactive.
- TAUBnCNTm (slave) generates an interrupt every time the value of TAUBnCDRm is loaded.

## (2) Duty cycle = 100%



**Figure 22.81 TAUBnCDRm (slave)  $\geq$  TAUBnCDRm (master) + 1,  
Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)**

If the value TAUBnCDRm (slave) is higher than the value TAUBnCDRm (master), the counter of the slave channel cannot reach 0000<sub>H</sub> and cannot generate interrupts. The TAUBTTOUTm remains at active state.

## (3) Operation stop and restart

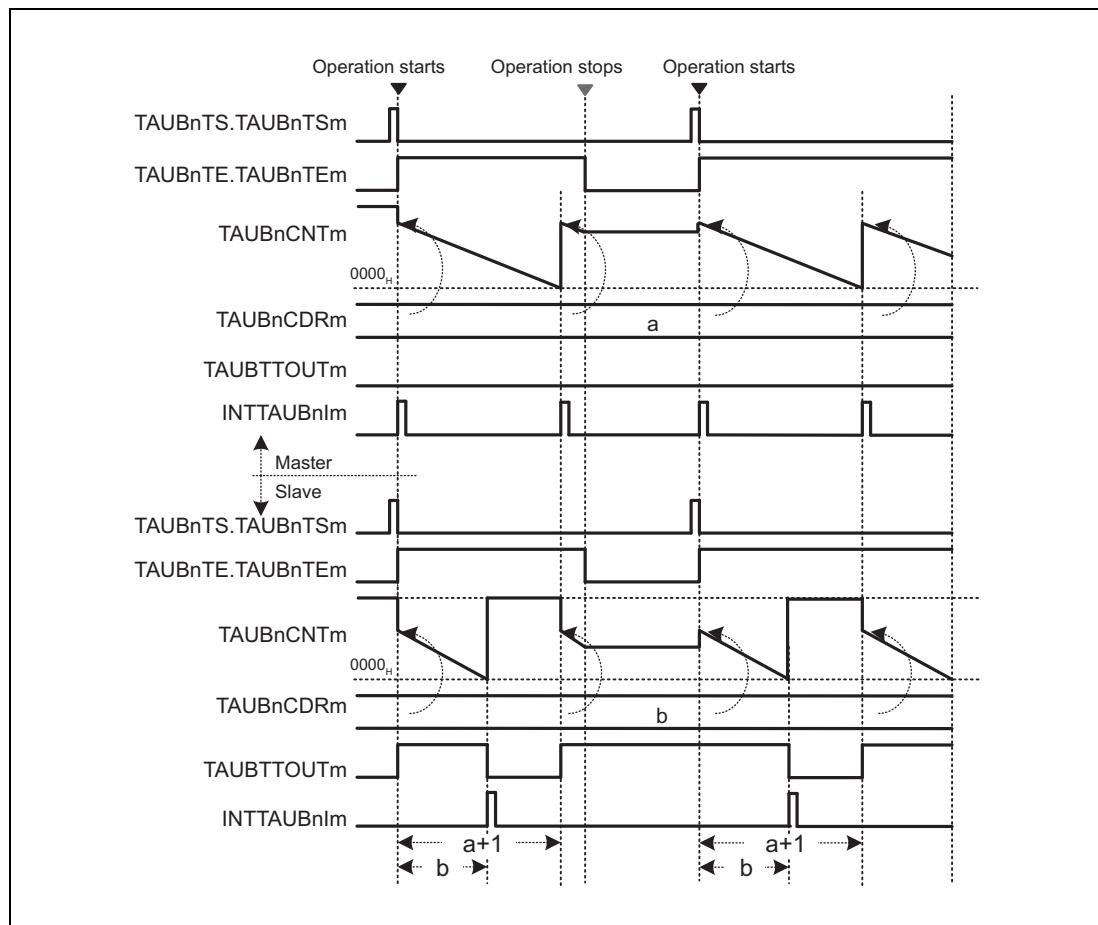


Figure 22.82 Operation Stop and Restart,  
Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm of the master and slave channel(s) to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm of master and slave channel(s) to 1. TAUBnCNTm of master and slave channel reload the current values of TAUBnCDRm and start to count down from these values.

#### (4) Operation stop and restart (Slave output, Initialization)

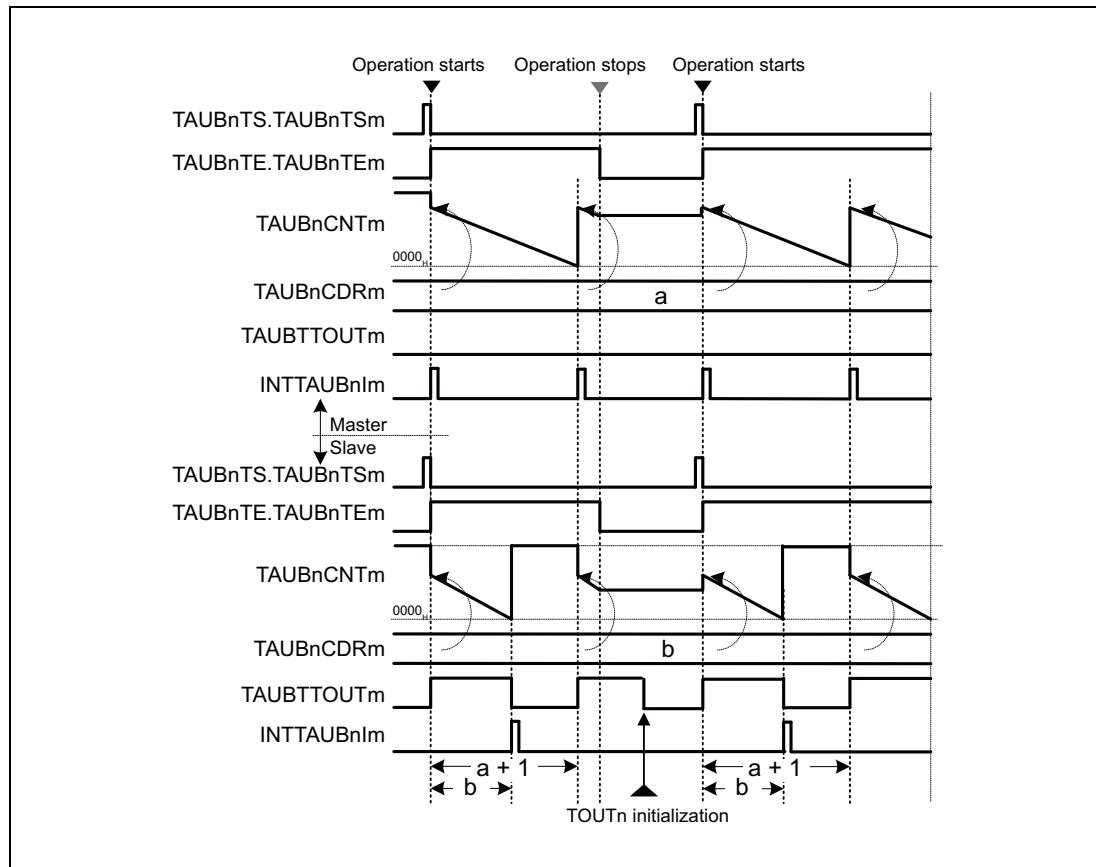


Figure 22.83 Operation Stop and Restart (Slave Output, Initialization)

When TAUBnTOE.TAUBnTOEm of the slave channel is set to 0 while TAUBnTE.TAUBnTEm = 0 and the inactive level of TAUBTTOUTm is written in the TAUBnTO.TAUBnTOm, the output level of TAUBTTOUTm (slave channel) becomes active when INTTAUBnIm is issued when the count operation is started.

## 22.14.2 One-Shot Pulse Output Function

### 22.14.2.1 Overview

#### Summary

This function outputs a signal pulse with a defined pulse width and a specific delay time compared to an external input signal pulse by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

#### Prerequisites

- Two channels
- The operation mode of the master channel must be set to one-count mode, see **Table 22.107, Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**
- The operation mode of the slave channel must be set to pulse one-count mode, see **Table 22.110, Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**
- TAUBTTOUTm is not used for the master channel of this function
- The channel output mode of the slave channel must be set to independent channel output mode 2.
- TAUBTTINm (master) has to be detected while TAUBnCNTm (master) and TAUBnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUBTTINm (slave).

#### Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) for master and slave channels to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation.

- Master channel:

When the next valid TAUBTTINm input edge is detected, the current value of TAUBnCDRm is written to TAUBnCNTm. The counter starts to count down from this value. If TAUBnCMORm.TAUBnMD0 = 0, a trigger (TAUBTTINm) which is detected within the delay time is ignored.

When the counter of the master channel reaches 0000<sub>H</sub>, INTTAUBnIm is generated. The counter returns to FFFF<sub>H</sub> and awaits the next valid TAUBTTINm input edge.

- Slave channel:

INTTAUBnIm generated on the master channel triggers the counter of the slave channel. The current value of TAUBnCDRm (slave) is written to TAUBnCNTm (slave) and the counter starts to count down from this value.

An interrupt is generated and the TAUBTTOUTm signal is set.

When the counter reaches 0001<sub>H</sub>, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter remains at 0000<sub>H</sub> and awaits the next INTTAUBnIm of the master channel.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEM to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

The counter of the master channel can be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSM to 1 during operation.

#### NOTES

1. If a forced restart of the counter is executed during operation, the width of the output signal does not correspond to the value of TAUBnCDRM (slave).
2. The TAUBTTINm input signal is sampled at the frequency of the operating clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of  $\pm 1$  operation clock cycle.

#### Conditions

- If TAUBnCMORm.TAUBnMD0 of the master channel is set to 0, during counting detected TAUBTTINm input edges are ignored.
- Simultaneous rewrite can be used with this function. Please see **Section 22.6, Simultaneous Rewrite**.

#### 22.14.2.2 Equations

$$\begin{aligned} \text{Delay from trigger input to pulse output} \\ = (\text{TAUBnCDRM (master)} + 1) \times \text{count clock cycle} \end{aligned}$$

$$\text{Pulse width} = (\text{TAUBnCDRM (slave)}) \times \text{count clock cycle}$$

### 22.14.2.3 Block Diagram and General Timing Diagram

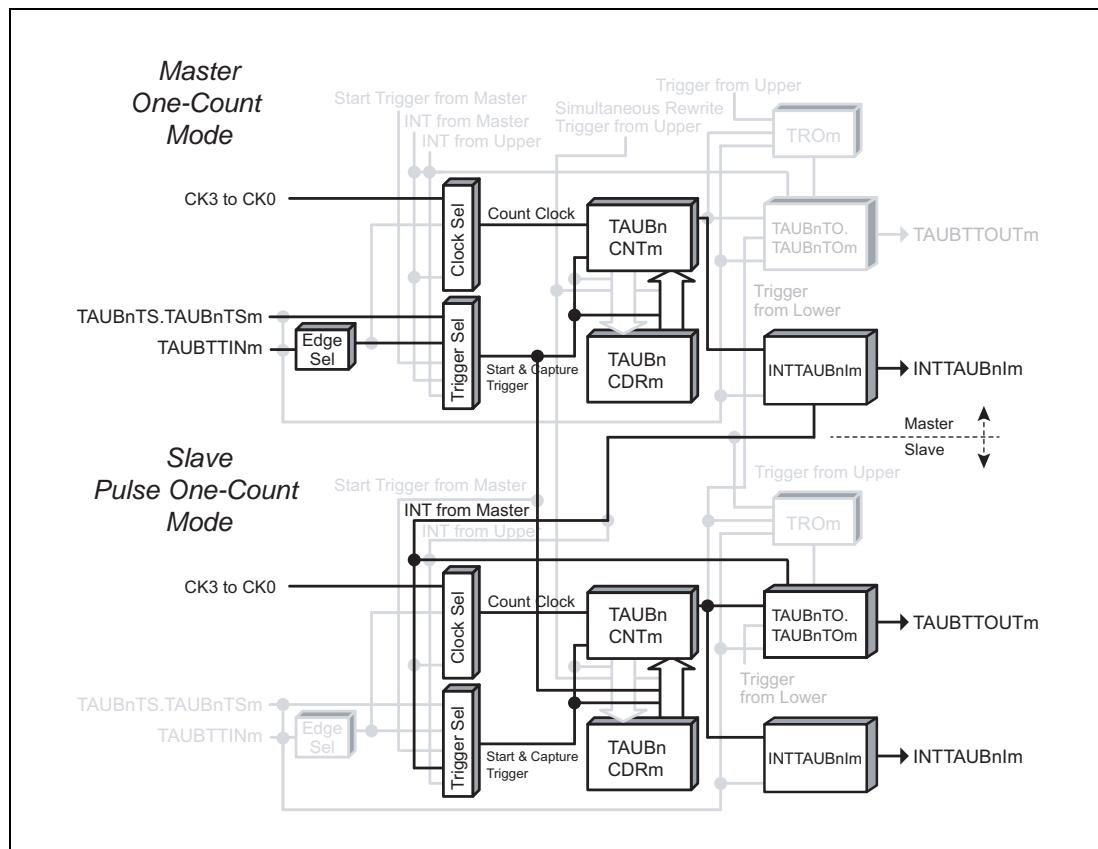


Figure 22.84 Block Diagram for One-Shot Pulse Output Function

The following settings apply to the general basic diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

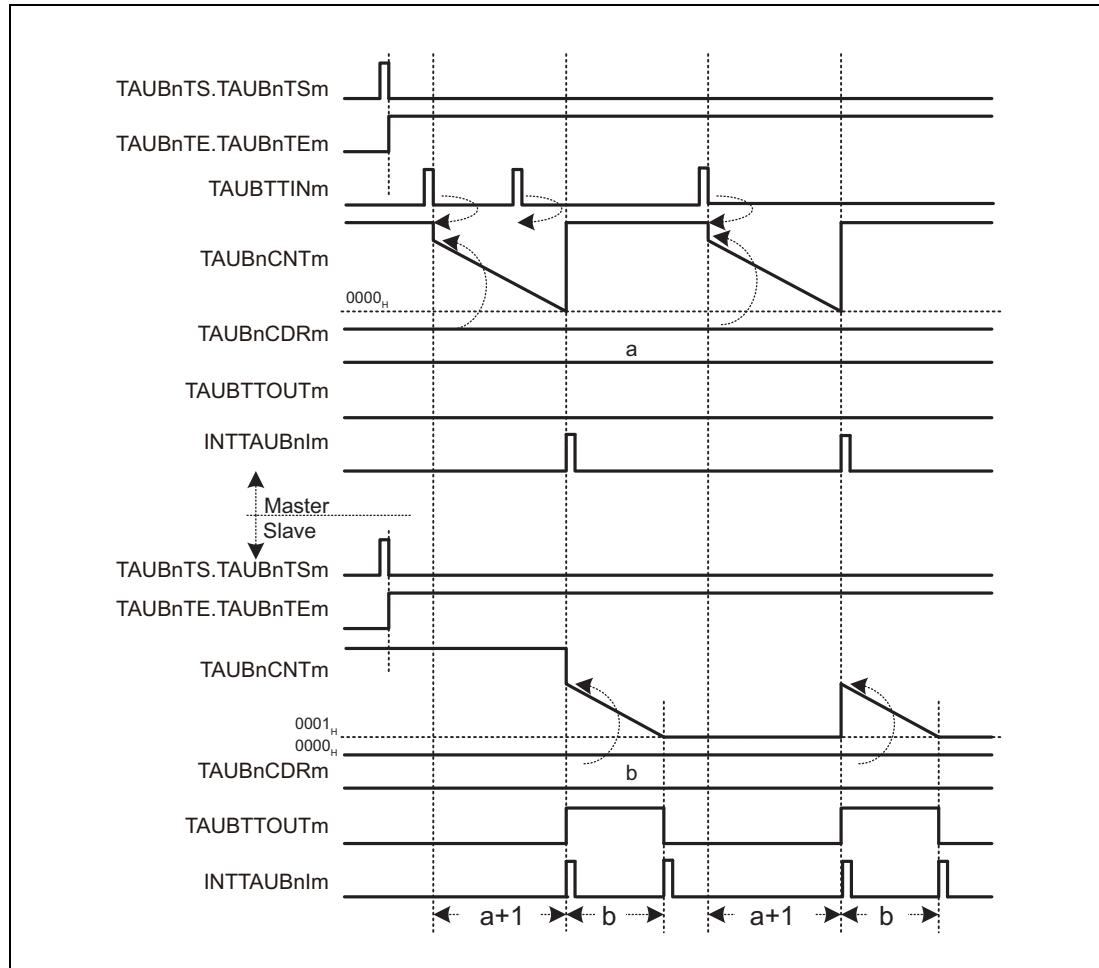


Figure 22.85 General Timing Diagram for One-Shot Pulse Output Function

### 22.14.2.4 Register Settings for the Master Channel

#### (1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.107 Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 1 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUBnMDO	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting

#### (2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.108 Contents of the TAUBnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

**(3) Channel output mode for the master channel**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

**(4) Simultaneous rewrite for the master channel**

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.109 Simultaneous Rewrite Settings for the Master Channel of the One-Shot Pulse Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.2.5 Register Settings for the Slave Channel

#### (1) TAUBnCMORm for the slave channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MD0						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.110 Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 100 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010 <sub>B</sub> .
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting The value of the MD0 bit of the master and slave channels must be identical.

#### (2) TAUBnCMURm for the slave channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.111 Contents of the TAUBnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for the slave channel

**Table 22.112 Control Bit Settings for Independent Channel Output Mode 2**

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	1: Enables independent channel output mode
TAUBnTOM.TAUBnTOMm	0: Independent channel output
TAUBnTOC.TAUBnTOCm	1: Operating mode 2
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	0: Disables dead time operation
TAUBnTDL.TAUBnTDLm	0: When dead time operation is disabled (TAUBnTDE.TAUBnTDEm = 0), set these bits to 0

### (4) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channels must be identical.

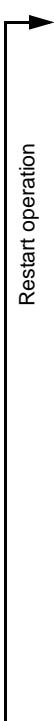
**Table 22.113 Simultaneous Rewrite Settings for the Slave Channel of the One-Shot Pulse Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 22.114 Operating Procedure for One-Shot Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	<p>Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.2.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.2.5, Register Settings for the Slave Channel</b>.</p> <p>Set the values of the TAUBnCDRm registers of all channels</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUBTTINm input.
During operation	<p>TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time.</p> <p>TAUBnRDT.TAUBnRDTm can be changed during operation.</p>	<p>When a valid TAUBTTINm input edge is detected, TAUBnCNTm of the master channel loads the TAUBnCDRm value and counts down. When the counter reaches <math>0000_H</math>:</p> <ul style="list-style-type: none"> <li>INTTAUBnIm (master) is generated</li> <li>TAUBnCNTm (master) is reset to <math>FFFF_H</math> and waits for the next valid TAUBTTINm input edge.</li> <li>TAUBnCNTm (slave) loads the TAUBnCDRm value and starts to count down</li> <li>INTTAUBnIm (slave) is generated</li> <li>TAUBTTOUTm (slave) is set to the active level.</li> </ul> <p>When TAUBnCNTm (slave) reaches <math>0001_H</math>:</p> <ul style="list-style-type: none"> <li>The counter of TAUBnCNTm (slave) stops.</li> <li>INTTAUBnIm (slave) is generated</li> <li>TAUBTTOUTm (slave) is set to an inactive level.</li> </ul>
Stop operation	<p>Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops.</p> <p>TAUBnCNTm and TAUBTTOUTm stop and retain their current values.</p>



### 22.14.2.7 Specific Timing Diagrams

#### (1) TAUBnCDRm (master) = 0000<sub>H</sub>

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

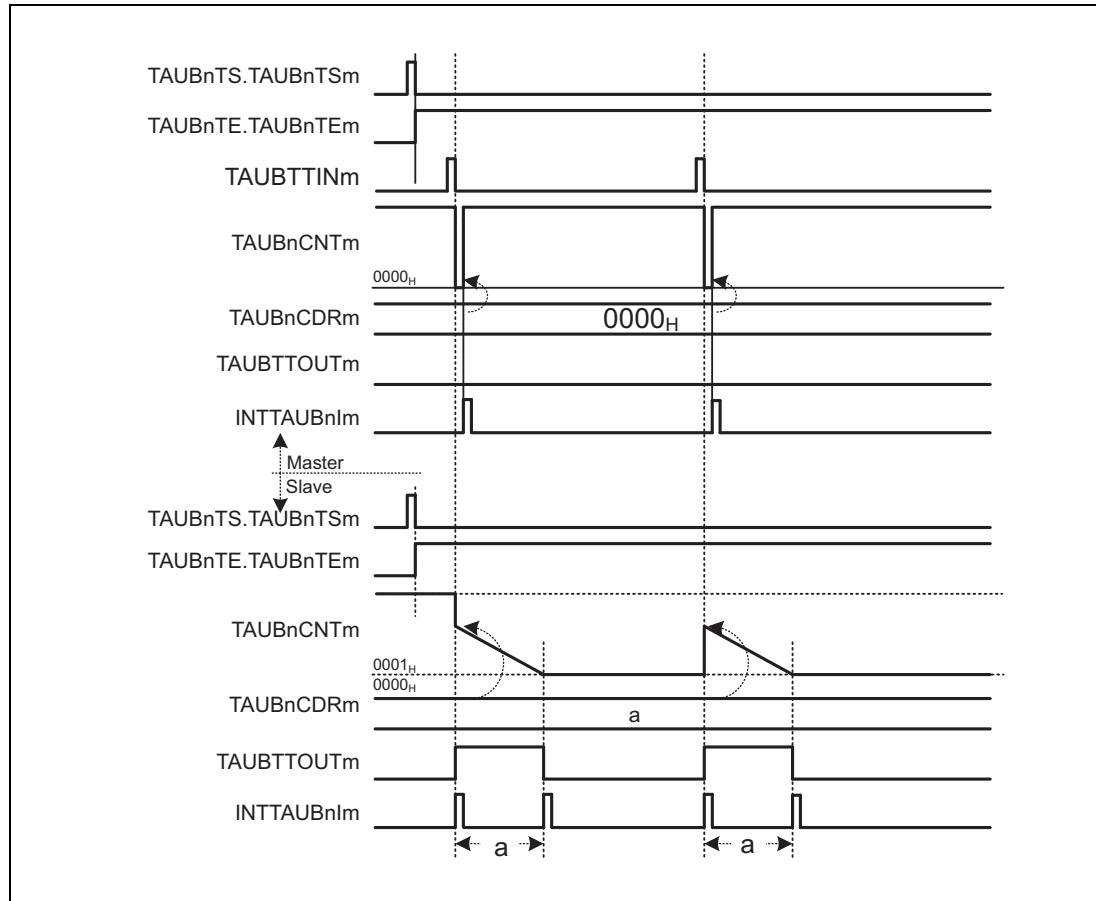


Figure 22.86 TAUBnCDRm (master) = 0000<sub>H</sub>

- When a valid TAUBTTINm input edge is detected, the value 0000<sub>H</sub> is written to TAUBnCNTm (master). The counter is set to 0000<sub>H</sub> for one count and returns to FFFF<sub>H</sub>. Thus, the slave channel starts to count down one count clock later to TAUBTTINm (master).

## (2) TAUBnCDRm (slave) = 0000<sub>H</sub>

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

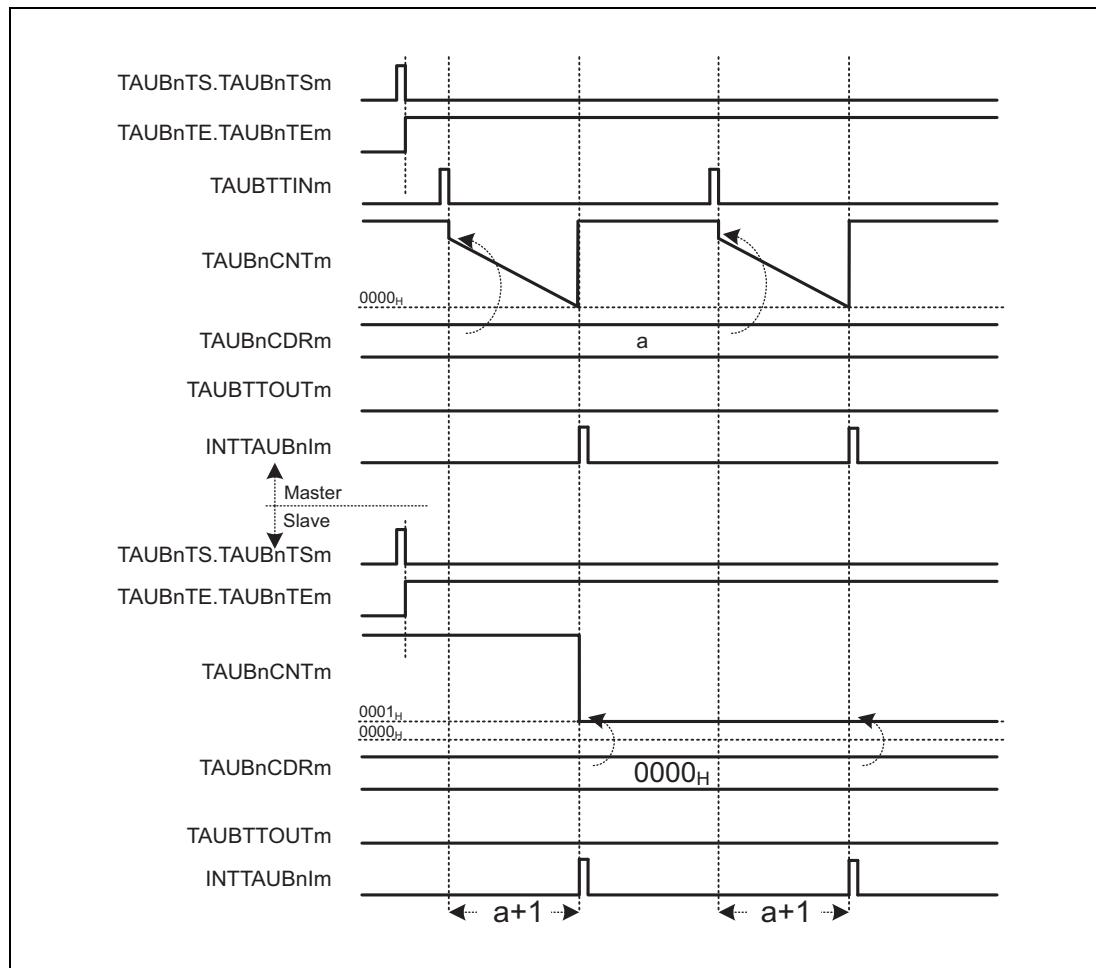


Figure 22.87 TAUBnCDRm (slave) = 0000<sub>H</sub>

- TAUBTTOUTm remains at inactive state, because the pulse width is zero.

### (3) TAUBnCMORm.TAUBnMD0 = 0

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

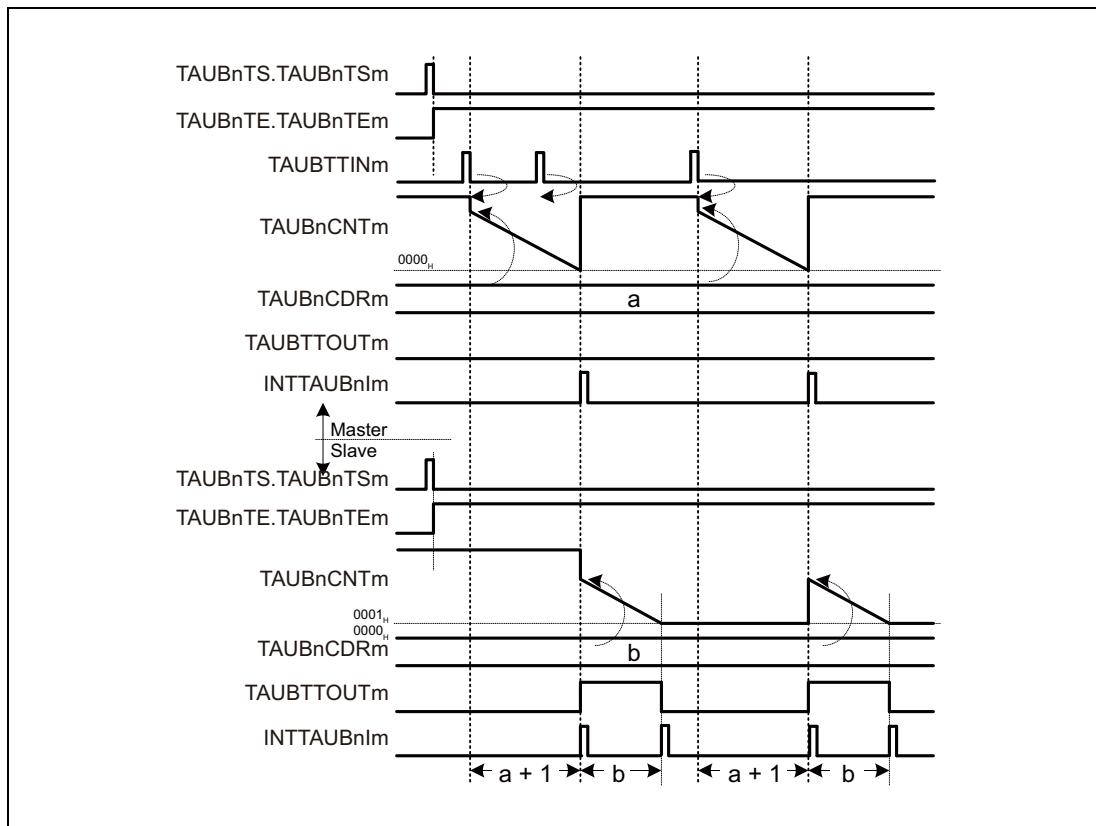


Figure 22.88 TAUBnCMORm.TAUBnMD0 = 0

- Even when an effective edge is input to TAUBTTINm while the counter of the master channel counts down, the counter continues counting down.

#### (4) TAUBnCMORm.TAUBnMD0 = 1

The following settings apply to this diagram.

- Start trigger detection enabled during counting (TAUBnCMORm.TAUBnMD0 = 1)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)

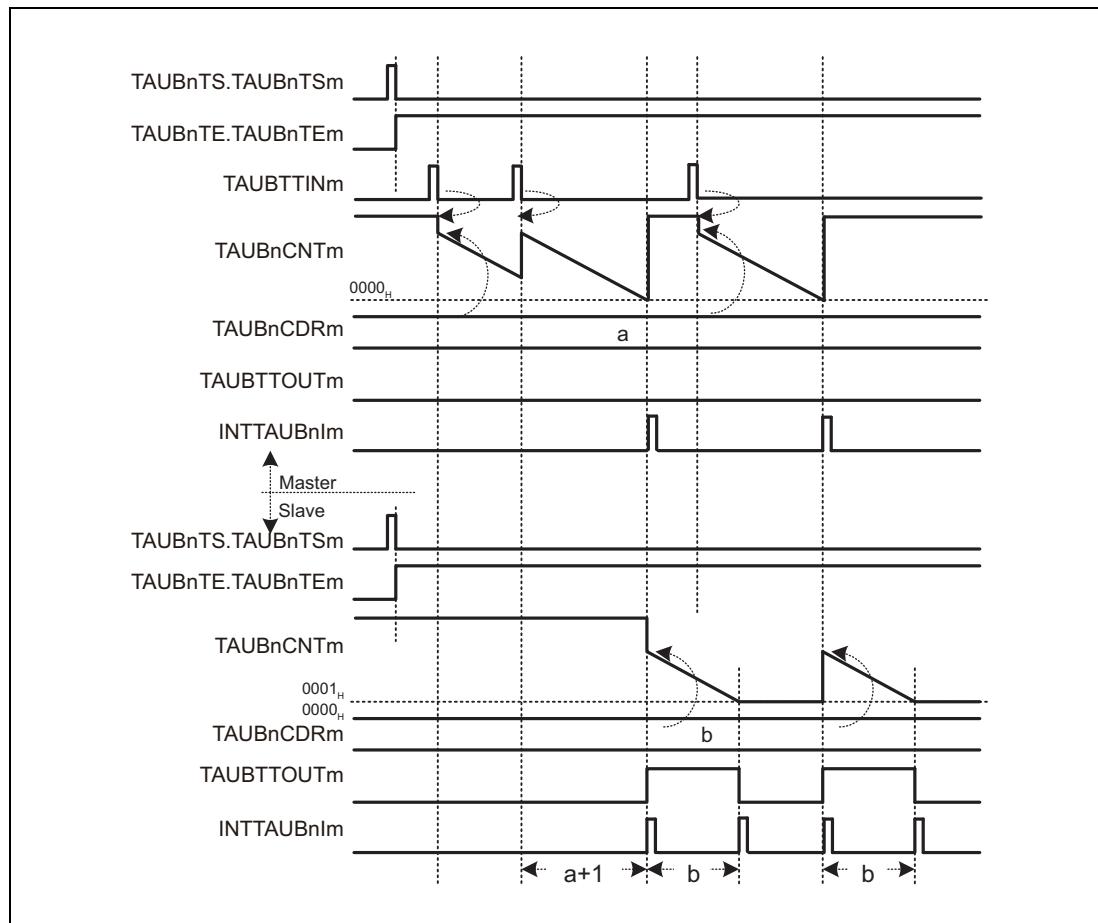
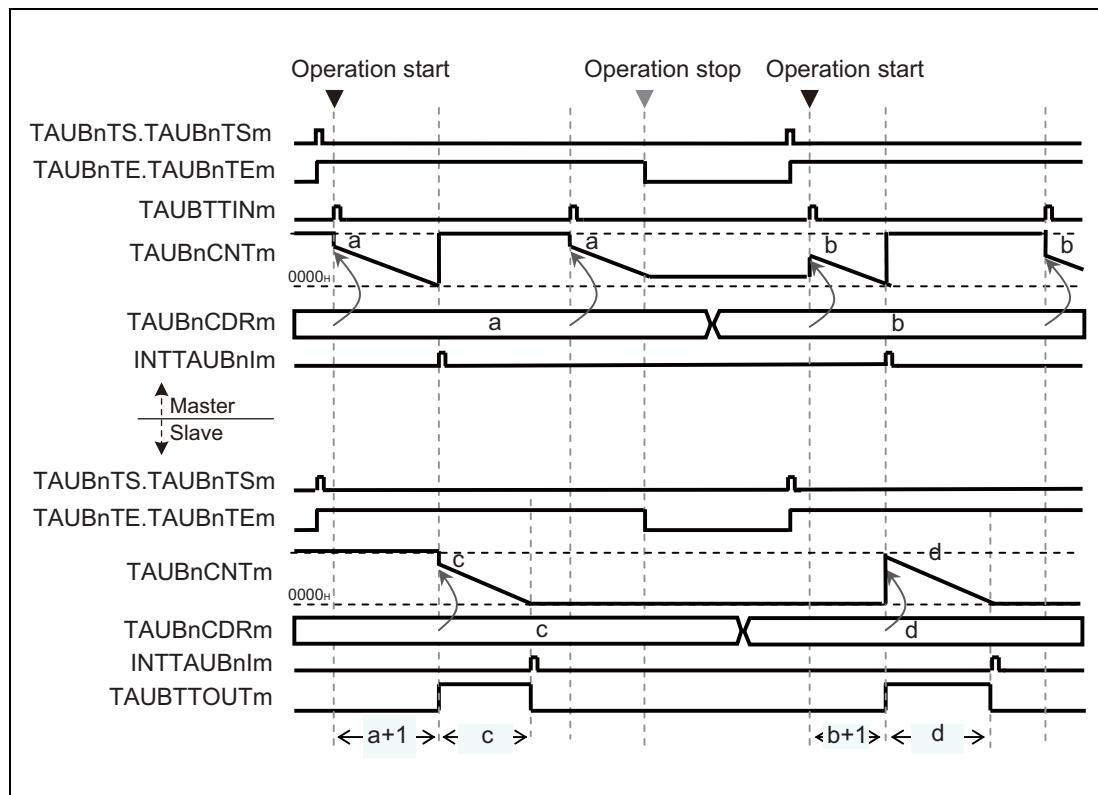


Figure 22.89 TAUBnCMORm.TAUBnMD0 = 1

- If a valid TAUBTTINm input edge is detected while the counter of the master channel counts down, TAUBnCNTm reloads the value of TAUBnCDRm. The counter restarts to count down.

This means the delay for INTTAUBnIm generation interval is extended by the value of TAUBnCNTm at the time when a valid TAUBTTINm input edge is detected.

### (5) Stopping and restarting the operation



**Figure 22.90 Stopping and Restarting the Operation**

Setting TTm of the master and slave channels to 1 clears TAUBnTE.TAUBnTEm to 0, thereby stopping the count operation. If this happens, TAUBnCNTm and TAUBTTOUTm stop operation with the values retained.

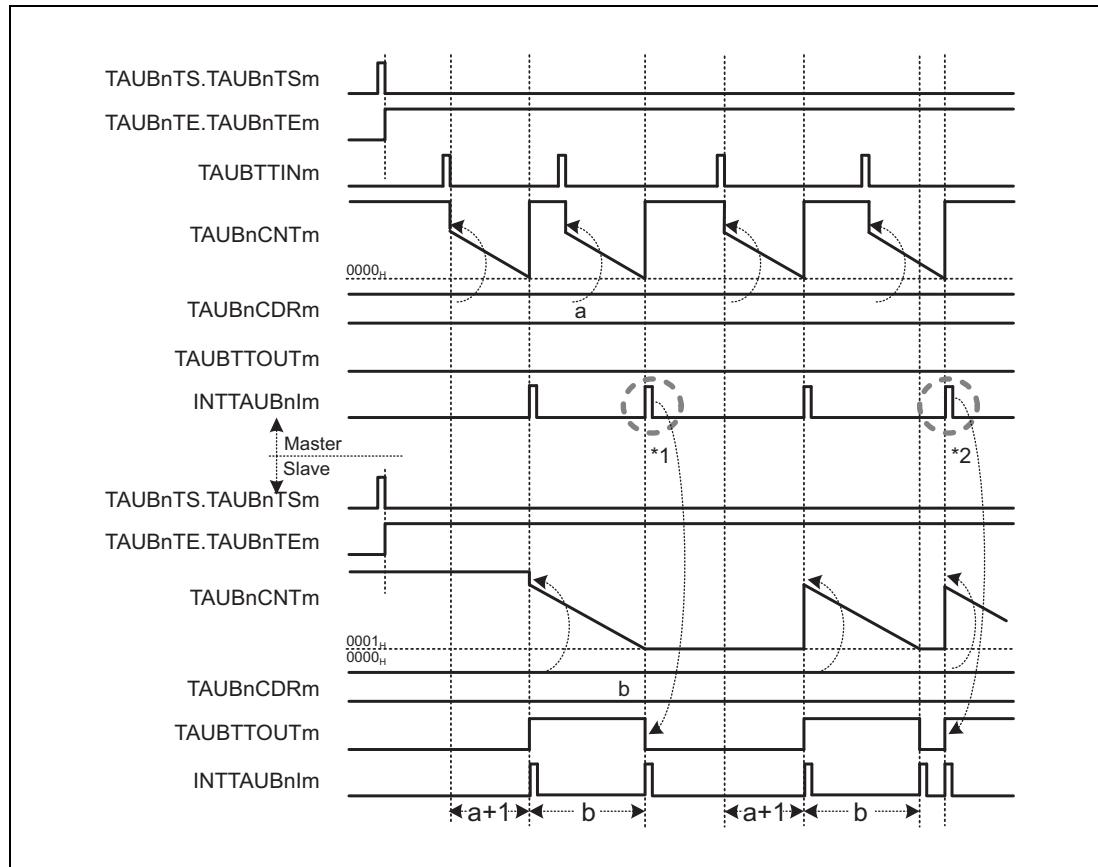
Setting TAUBnTS.TAUBnTSm of the master and slave channels to 1 concurrently sets TAUBnTE.TAUBnTEm to 1.

When the start trigger is detected while the TAUBnTE.TAUBnTEm is set to 1, the TAUBnCDRm value is transferred to TAUBnCNTm and the operation restarts.

### (6) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00<sub>B</sub>)



**Figure 22.91 Input Interval of TAUBTTINm ≤ Delay Time + Pulse Width + 1**

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001<sub>H</sub> or exactly when 0001<sub>H</sub> is reached (\*1), the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUBnCDRm (slave) is reloaded. An interrupt is generated and TAUBTTOUTm toggles. If TAUBnCNTm (master) has started to count down while the TAUBnCNTm (slave) is still counting (\*2), TAUBTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

### 22.14.3 Delay Pulse Output Function

#### 22.14.3.1 Overview

##### Summary

This function outputs two signals. The reference signal has a defined pulse width and pulse cycle specified using the master channel and slave channel 1.

Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by amount specified in slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1.  
The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified in slave channel 2.

##### Prerequisites

- Four channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 22.115, Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function**.
- The operation mode of slave channels 1 and 2 must be set to one-count mode, see **Table 22.118, Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function** and **Table 22.122, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function**.
- The operation mode of slave channel 3 must be set to pulse one-count mode, see **Table 22.125, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function**
- TAUBTOUTm is not used for the master channel and slave channel 2
- The channel output mode of slave channel 1 must be set to synchronous channel output mode 1.
- The channel output mode of slave channel 3 must be set to independent channel output mode 1.

##### Description

The counters of the channel group are enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM to 1, enabling count operation.

- Master channel:

The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value. INTTAUBnIm is generated on the master channel.

When the counter of the master channel reaches 0000<sub>H</sub>, pulse cycle time has elapsed and INTTAUBnIm is generated. The counter reloads the TAUBnCDRm value and counts down.

- Slave channels 1 and 2:

When the slave channels 1 and 2 detect an interrupt from the master channel, they start to count down from the current value of TAUBnCDRm. The TAUBTOUTm signal (slave 1) is set.

- Slave channel 1:

When the counter of slave channel 1 reaches  $0000_H$  (duty time has elapsed) INTTAUBnIm is generated and the TAUBTOUTm signal is reset. The counter returns to  $FFFF_H$  and awaits the next INTTAUBnIm of the master channel.

- Slave channel 2:

When the counter of slave channel 2 reaches  $0000_H$ , delay time has elapsed and INTTAUBnIm is generated. The counter returns to  $FFFF_H$  and awaits the next INTTAUBnIm of the master channel.

INTTAUBnIm (slave channel 2) triggers the counter of slave channel 3

- Slave channel 3:

When slave channel 3 detects an interrupt from slave channel 2, it starts to count down from the current value of TAUBnCDRm. INTTAUBnIm is generated and the TAUBTOUTm signal (slave 3) is set.

When the counter of slave channel 3 reaches  $0001_H$ , INTTAUBnIm is generated and the TAUBTOUTm signal is reset.

The output from slave channel 3 is the delayed PWM pulse.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEM to 0. TAUBnCNTm and TAUBTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

### Conditions

Simultaneous rewrite can be used with this function. Please see **Section 22.6, Simultaneous Rewrite**.

#### 22.14.3.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUBnCDRm (slave 1)) × count clock cycle

Delay = (TAUBnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUBnCDRm (slave 3)) × count clock cycle

Where the setting of the delay is within the following range:

$0000_H \leq \text{TAUBnCDRm (slave 2)} < \text{TAUBnCDRm (master)}$

**NOTES**

---

1. The output waveform of TAUBTOUTm (slave 3) is the output waveform of TAUBTOUTm (slave 1) delayed for the delay generated by slave 2. It cannot be delayed for more than the pulse cycle.
  2. When INTTAUBnIm of slave 2 occurs while slave 3 is counting, slave 3 restarts the operation. Therefore, the output waveform of TAUBTOUTm (slave 3) retains the active level. (In this case, TAUBTOUTm (Slave-CH-3) cannot output the waveform of the delayed basic pulse of TAUBTOUTm (Slave-CH-1).)
-

### 22.14.3.3 Block Diagram and General Timing Diagram

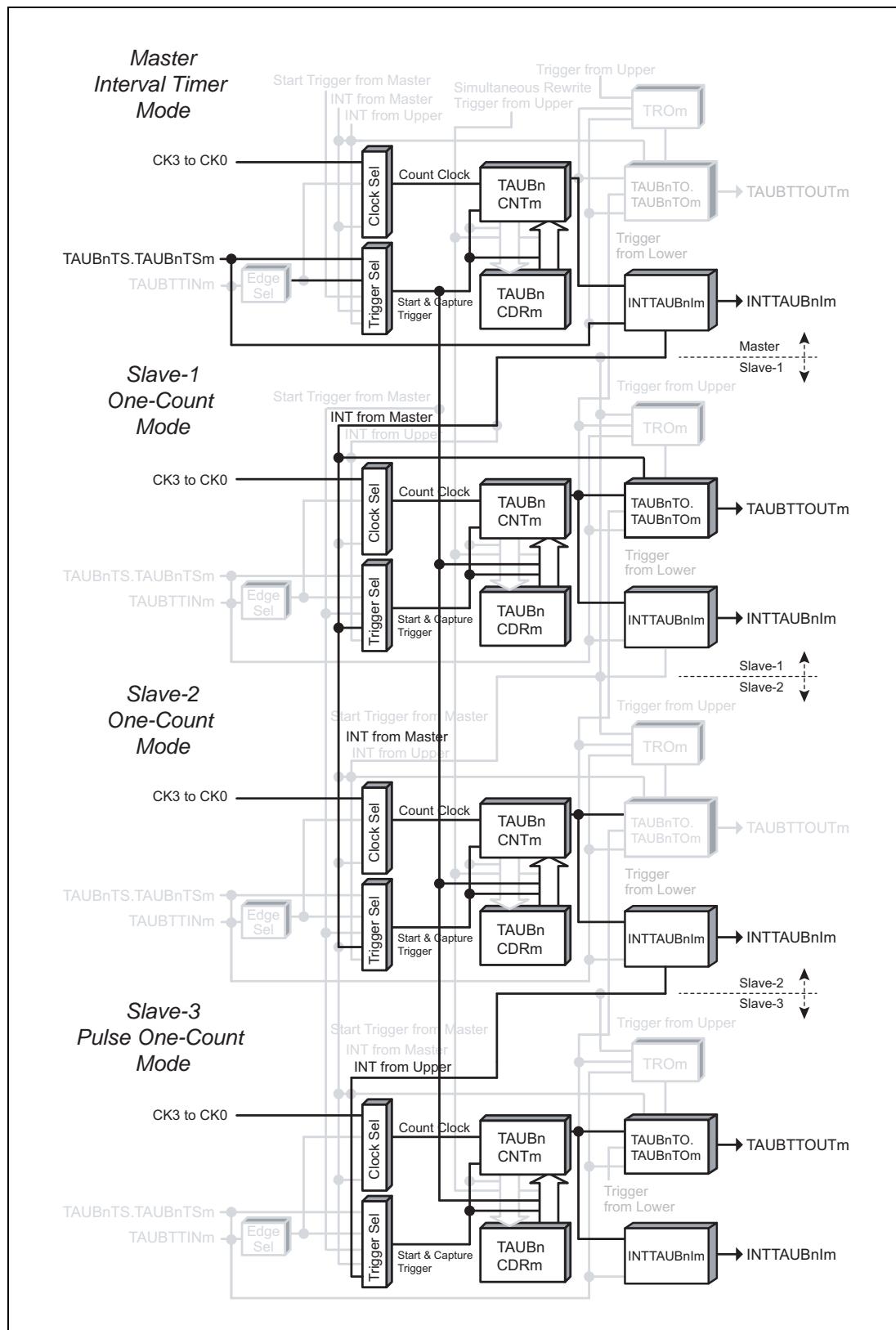


Figure 22.92 Block Diagram for Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

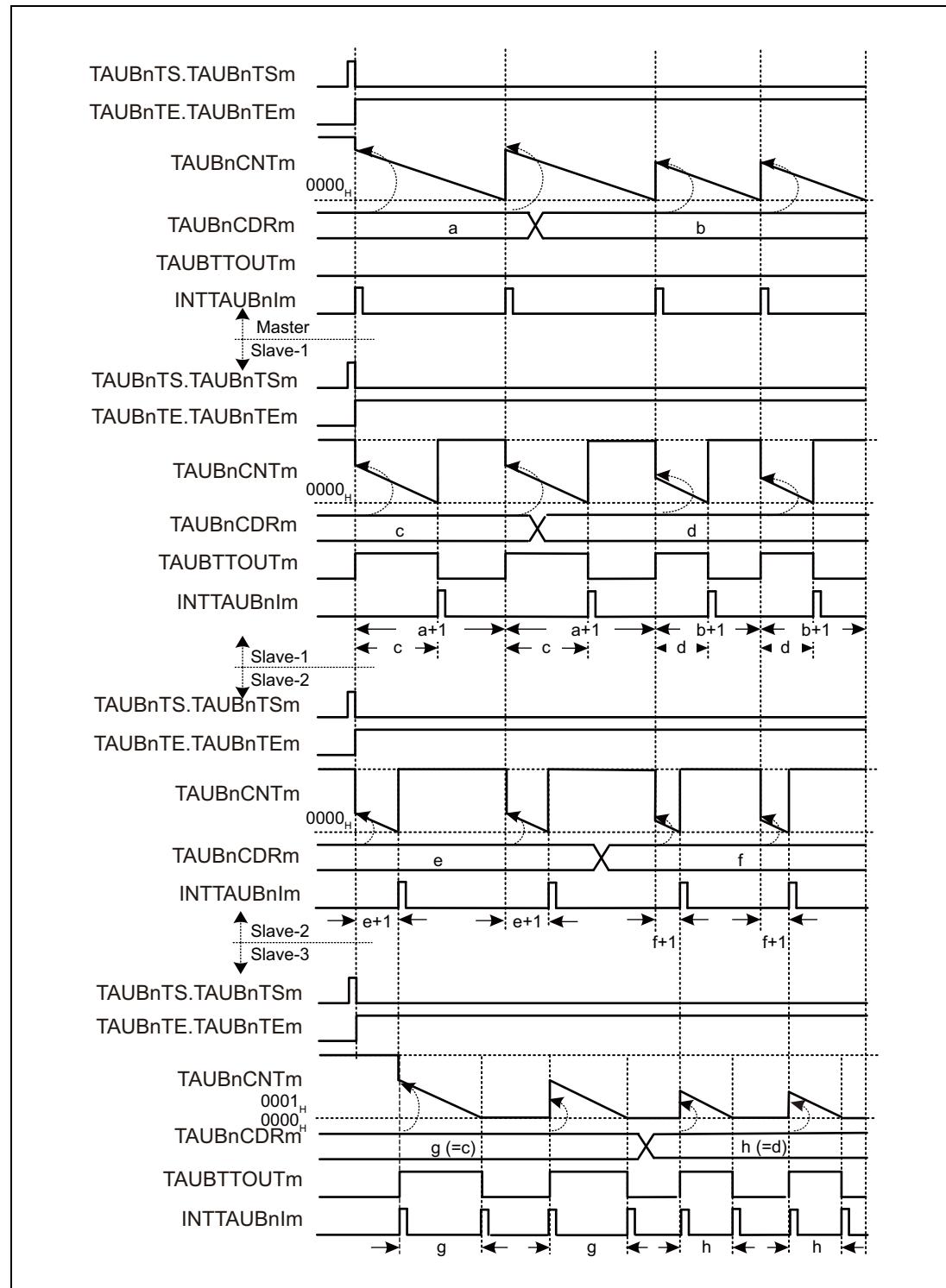


Figure 22.93 General Timing Diagram for Delay Pulse Output Function

### 22.14.3.4 Register Settings for the Master Channel

#### (1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.115 Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 1 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.116 Contents of the TAUBnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

**(3) Channel output mode for the master channel**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by the master channel of this function.

**(4) Simultaneous rewrite for the master channel**

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.117 Simultaneous Rewrite Settings for the Master Channel of the Delay Pulse Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.3.5 Register Settings for Slave Channel 1

#### (1) TAUBnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.118 Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 100 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 22.119 Contents of the TAUBnCMURm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for slave channel 1

**Table 22.120 Control Bit Settings for Slave Channel 1 of the Synchronous Channel Output Mode 1**

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 1 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 0 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

### (4) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.121 Simultaneous Rewrite Settings for Slave Channel 1 of the Delay Pulse Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.3.6 Register Settings For Slave Channel 2

#### (1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.122 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 100 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 22.123 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

**(3) Channel output mode for slave channel 2**

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

**(4) Simultaneous rewrite for slave channel 2**

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.124 Simultaneous Rewrite Settings for Slave Channel 2 of the Delay Pulse Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.3.7 Register Settings for Slave Channel 3

#### (1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.125 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 101 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.126 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for slave channel 3

Table 22.127 Control Bit Settings for Synchronous Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 0 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 1 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEM	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

### (4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 22.128 Simultaneous Rewrite Settings for Slave Channel 3 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEM	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.3.8 Operating Procedure for Delay Pulse Output Function

Table 22.129 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	Status of TAUBn
Initial channel setting	<p>Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.3.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel 1: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.3.5, Register Settings for Slave Channel 1</b>.</p> <p>Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.3.6, Register Settings For Slave Channel 2</b>.</p> <p>Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.3.7, Register Settings for Slave Channel 3</b>.</p> <p>Set the values of the TAUBnCDRm registers of all channels</p>	Channel operation is stopped.

Table 22.129 Operating Procedure for Delay Pulse Output Function (2/2)

	Operation	Status of TAUBn
Restart operation	Start operation	<p>Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p> <p>TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time.</p> <p>TAUBnRDT.TAUBnRDTm can be changed during operation.</p>
	During operation	<p>TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master channel and slave channels 1 and 2 start. INTTAUBnlm is generated on the master channel and TAUBTTOUTm (slave 1) is set.</p> <p>TAUBnCNTm of the master channel and slave channels 1 and 2 load TAUBnCDRm and count down.</p> <p>When the counter of the master channel reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (master) is generated</li> <li>TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation</li> <li>TAUBnCNTm (slave 1 and slave 2) reload the TAUBnCDRm value and start counting down</li> <li>TAUBTTOUTm (slave 1) is set</li> </ul> <p>When TAUBnCNTm (slave 1) reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (slave 1) is generated</li> <li>TAUBTTOUTm (slave 1) is reset</li> </ul> <p>When TAUBnCNTm (slave 2) reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (slave 2) is generated</li> <li>INTTAUBnlm (slave 3) is generated</li> <li>TAUBTTOUTm (slave 3) is set</li> <li>TAUBnCNTm (slave 3) reloads the TAUBnCDRm value and starts counting down</li> </ul> <p>When TAUBnCNTm (slave 3) reaches 0001H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (slave 3) is generated</li> <li>TAUBTTOUTm (slave 3) is reset</li> </ul>
	Stop operation	<p>Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>
		<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.</p>

### 22.14.3.9 Specific Timing Diagrams

#### (1) Duty cycle (slave 3) = 100%

The following values apply to the figure below.

- TAUBnCDRm (master) =  $000A_H$
- TAUBnCDRm (slave 1) =  $000B_H$
- TAUBnCDRm (slave 2) =  $0000_H$
- TAUBnCDRm (slave 3) =  $000B_H$

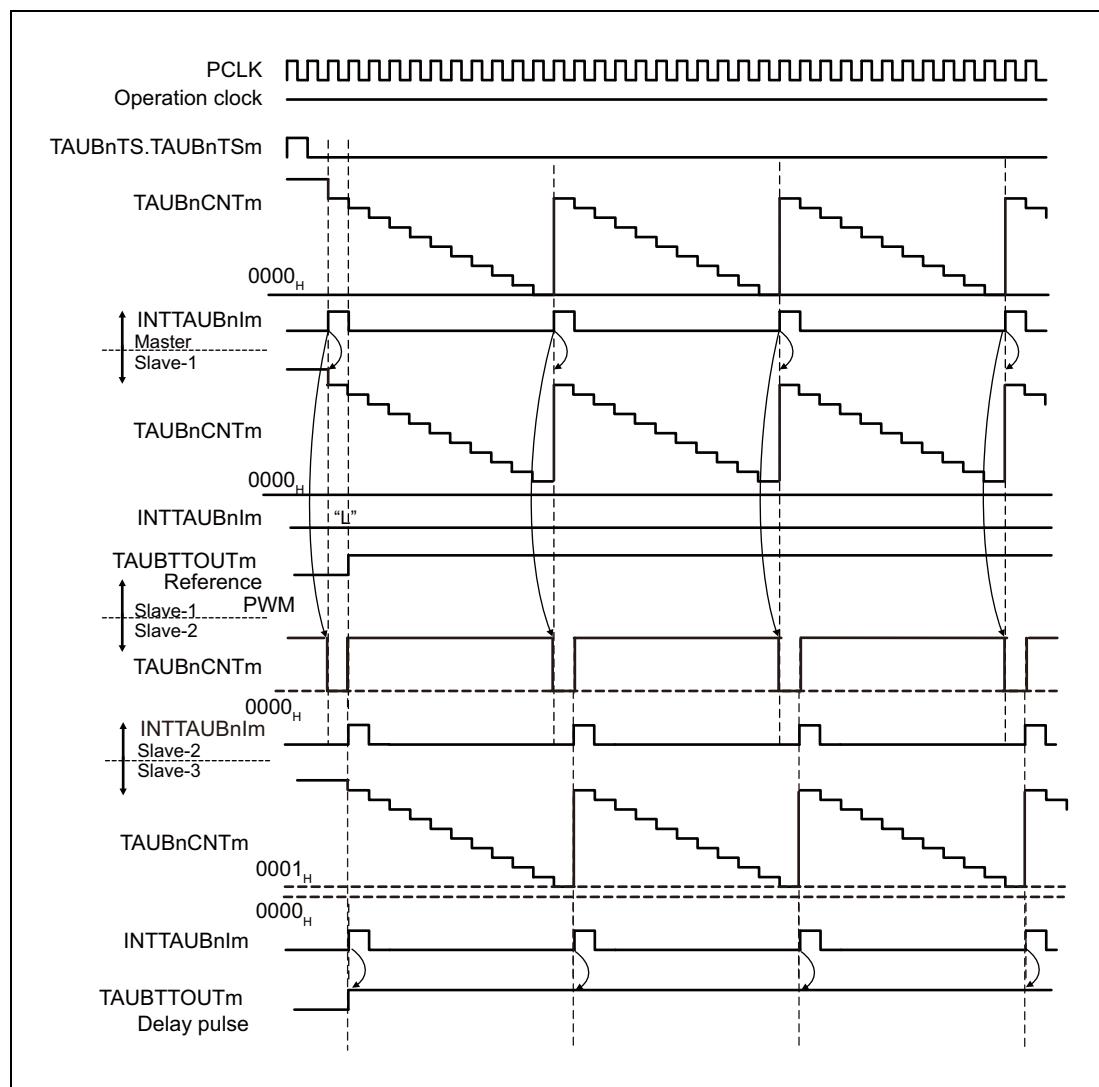


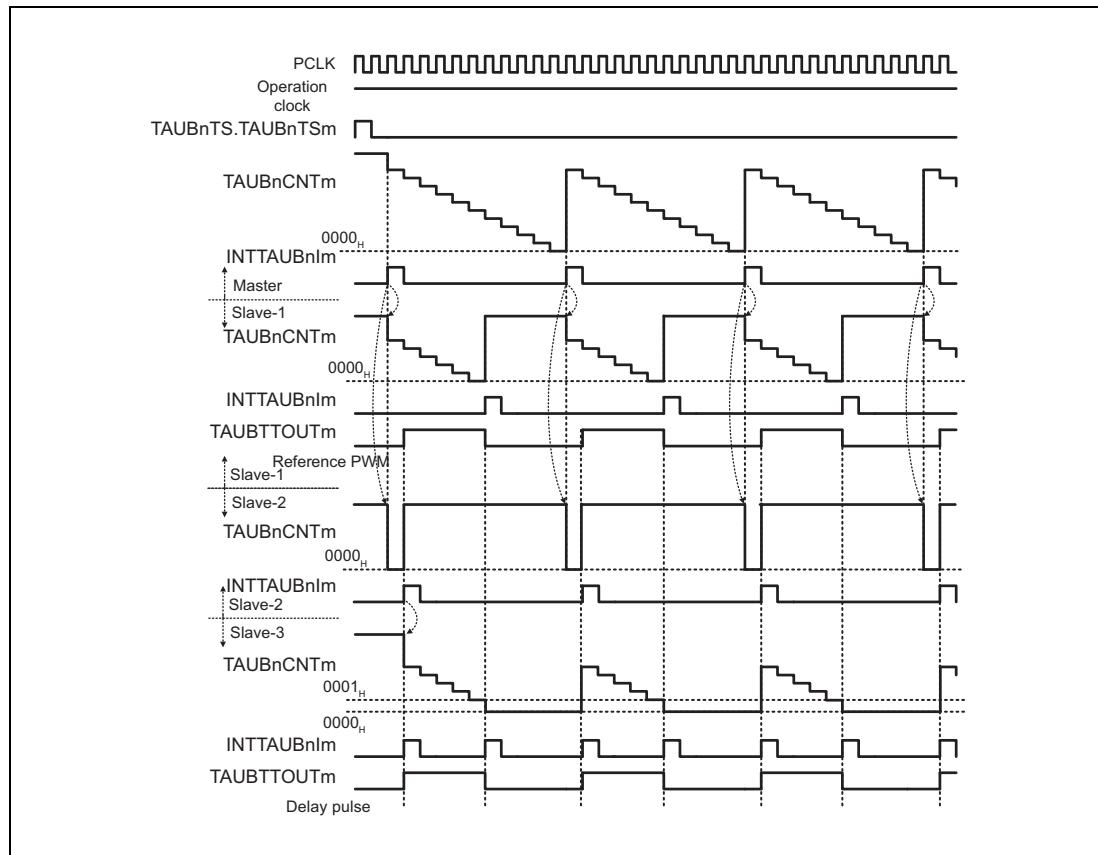
Figure 22.94 Duty Cycle (slave 3) = 100%

- If the value of TAUBnCDRm (slave 1 and 3) is higher than the value of TAUBnCDRm (master), the counter of slave channel 1 cannot reach  $0000H$  and cannot generate interrupt request signals. TAUBTTOUTm of channels 1 and 3 remain in the active state.

## (2) TAUBTOUTm (slave 1) = TAUBTOUTm (slave 3)

The following values apply to the figure below.

- TAUBnCDRm (master) =  $000A_H$
- TAUBnCDRm (slave 1) =  $0005_H$
- TAUBnCDRm (slave 2) =  $0000_H$
- TAUBnCDRm (slave 3) =  $0005_H$



**Figure 22.95 TAUBTOUTm (slave 1) = TAUBTOUTm (slave 3)**

- If  $\text{TAUBnCDRm} (\text{slave } 2) = 0000_H$ , the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

## 22.14.4 AD Conversion Trigger Output Function Type 1

### 22.14.4.1 Overview

#### Summary

此功能跟22.14.1 PWM Output Function 的功能一样，除了TAUBTTOUT没有输出  
This function is identical to Section 22.14.1, PWM Output Function except that TAUBTTOUTm is not output.

这是由软件设置slave的channel1 输出模式为Independent channel output mode  
This is achieved by setting the channel output mode of the slave to independent channel output mode controlled by software.

### 22.14.4.2 Block Diagram and General Timing Diagram

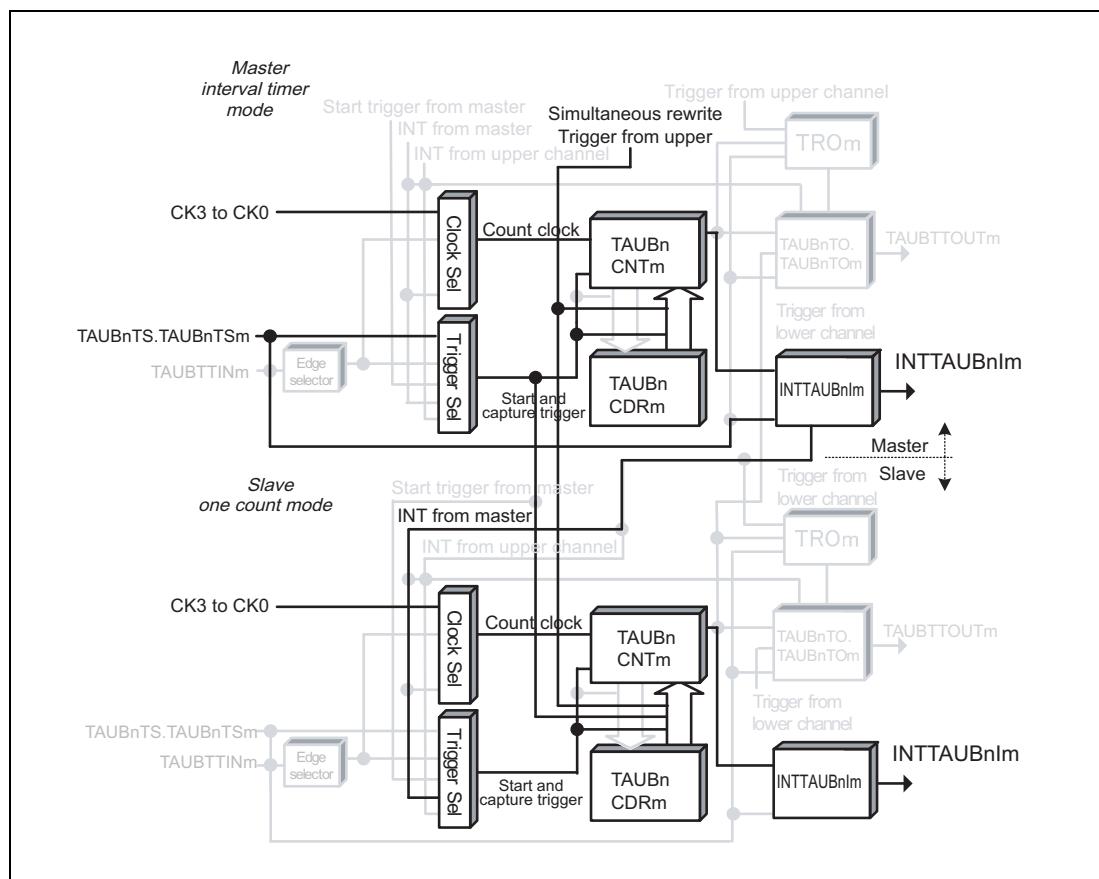


Figure 22.96 Block Diagram for AD Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

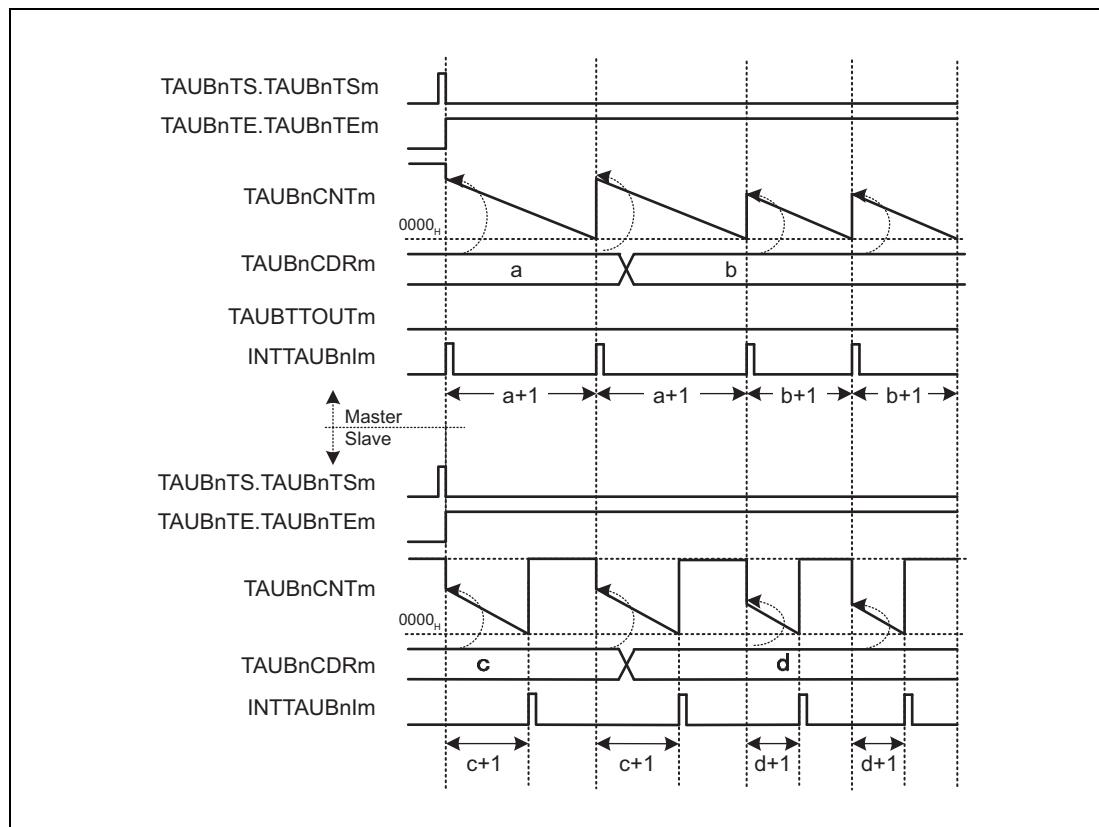


Figure 22.97 General Timing Diagram for AD Conversion Trigger Output Function Type 1

## 22.14.5 Triangle PWM Output Function

### 22.14.5.1 Overview

#### Summary

此功能是使用master和多个slave channel产生多个三角 PWM 输出  
This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

master channel 从2个pulse cycle中产生一个载波周期 第一个cycle 控制slave 计数器的下减状态  
The master channel generates a carrier cycle from two pulse cycles. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slaves counter.

#### Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 22.130, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function.**
- The operation mode of the slave channel(s) must be set to up down count mode, see **Table 22.134, Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle PWM Output Function.**
- The channel output mode of the slave channel(s) must be set to synchronous channel output mode
- The channel output mode of the master channel must be set to independent channel output mode
  1. slave channel 的输出模式必须设置为 synchronous channel output mode
  2. 以下设置为载波周期的下减状态 建立TOUTm在high level
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 0, TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnOEm is 0.

#### Functional description

The counters are enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1 for every channel. This in turn sets TAUBnTE.TAUBnTEM, enabling count operation. The current values of TAUBnCDRm (master and slave) are written to TAUBnCNTm (master and slave) and the counters start to count down from these values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel: 当master channel的计数器达到0x0000H(pulse cycle时间已过)  
When the counter of the master channel reaches 0000<sub>H</sub> (pulse cycle time has elapsed), INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then reloads the TAUBnCDRm value and counts down.

- **Slave channel:** **INTTAUBnIm** generated on the master channel triggers the counter of the slave channel:  
**如果slave的计数器当前在下减，它会改变计数方向**  
– If the slave counter currently counts down, it changes count direction.  
**如果slave的计数器当前在上增，CDRm被重载计数器开始下减**  
– If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.  
**无论slave的计数器在下减还是在上增，INTTAUBnIm中断就会产生，TAUBTTOUTm信号被set或reset**  
When the counter of the slave channel reaches  $0001_H$  while counting up or down, INTTAUBnIm is generated and the TAUBTTOUTm (slave) signal is set or reset.  
**计数器继续下减或者上升，等待下一次master channel的INTTAUBnIm中断**  
The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.  
**TOUTm在工作期间可以在positive和negative phase设置TAUBnTOL之间设置**  
TAUBTTOUTm can be switched between positive and negative phase setting  
TAUBnTOL.TAUBnTOLm during operation.

The counters can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEM to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

### Conditions

Simultaneous rewrite can be used with this function. Please see **Section 22.6, Simultaneous Rewrite**.

#### 22.14.5.2 Equations

$$\text{Pulse cycle} = (\text{TAUBnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$0000_H \leq \text{TAUBnCDRm (master)} < FFFF_H$$

$$\text{Carrier cycle (down/up)} = (\text{TAUBnCDRm (master)} + 1) \times 2 \times \text{count clock cycle}$$

$$\text{Duty cycle} =$$

$$[(\text{TAUBnCDRm (master)} + 1 - \text{TAUBnCDRm (slave)}) / (\text{TAUBnCDRm (master)} + 1)] \times 100$$

- Duty cycle = 100%  
 $\text{TAUBnCDRm (slave)} = 0000_H$
- Duty cycle = 0%  
 $\text{TAUBnCDRm (slave)} \geq \text{TAUBnCDRm (master)} + 1$

### 22.14.5.3 Block Diagram and General Timing Diagram

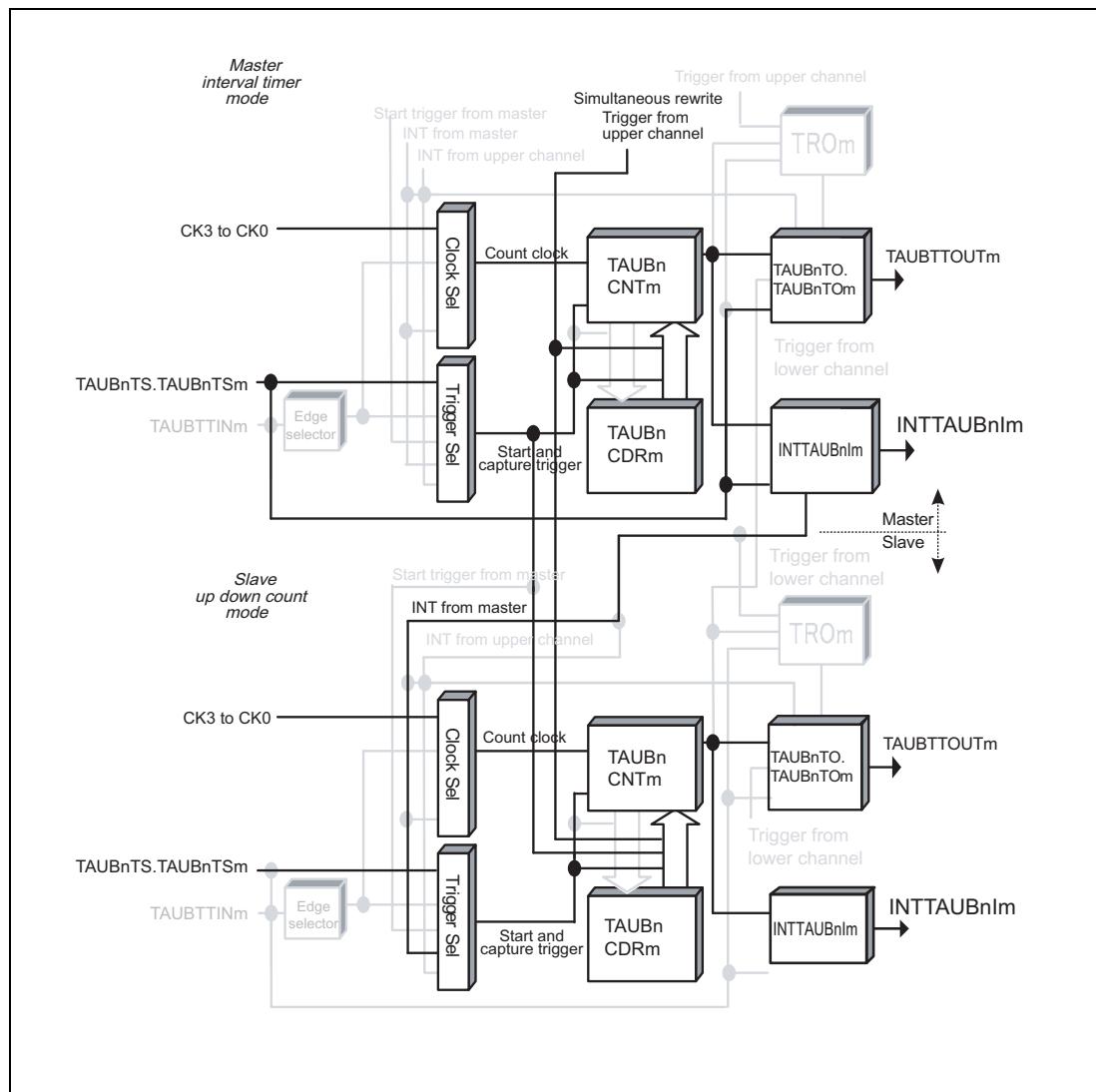


Figure 22.98 Block Diagram for Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

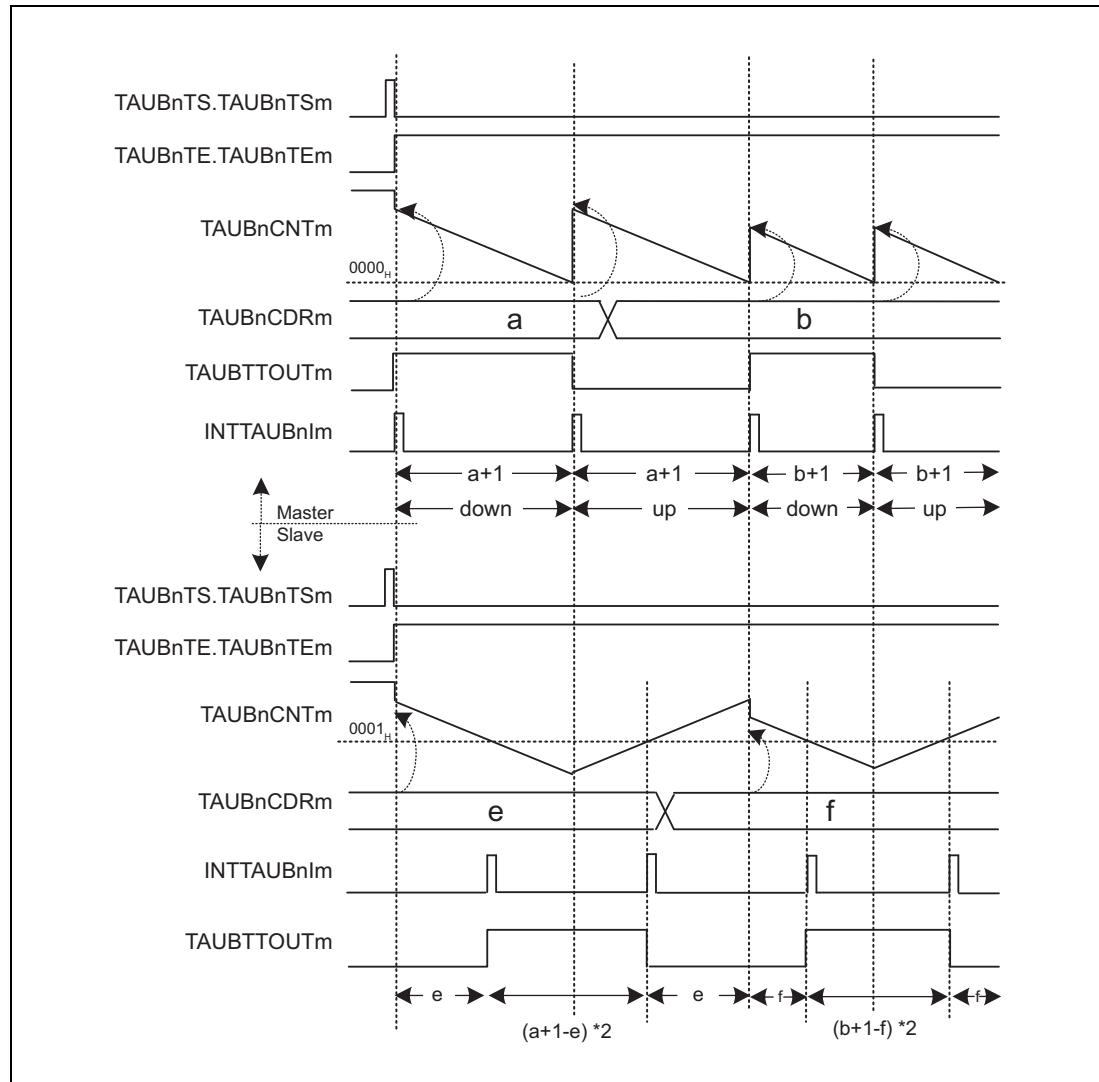


Figure 22.99 General Timing Diagram for Triangle PWM Output Function

#### 22.14.5.4 Register Settings for the Master Channel

##### (1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W						

Table 22.130 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 1 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMD0	0: INTTAUBnlm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnlm and toggles TAUBTTOUTm at operation start

##### (2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.131 Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for the master channel

Table 22.132 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 0 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 0 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	Write 0 <sub>B</sub> .
TAUBnTDE.TAUBnTDEm	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

### (4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 22.133 Simultaneous Rewrite Settings for the Master Channel of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

#### NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

### 22.14.5.5 Register Settings for the Slave Channel(s)

#### (1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.134 Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 111 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.135 Contents of the TAUBnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for the slave channel(s)

Table 22.136 Control Bit Settings for Synchronous Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 1 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 1 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEM	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

### (4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 22.137 Simultaneous Rewrite Settings for the Slave Channel of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEM	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMM	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.5.6 Operating Procedure for Triangle PWM Output Function

Table 22.138 Operating Procedure for Triangle PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	<p>Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.5.4, Register Settings for the Master Channel</b></p> <p>Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.5.5, Register Settings for the Slave Channel(s)</b></p> <p>Set the values of the TAUBnCDRm registers of all channels</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnlm (master) is generated on the master channel when TAUBnCMORm.TAUBnMD0 set to 1.</p>
During operation	<p>TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time.</p> <p>TAUBnRDT.TAUBnRDTm can be changed during operation.</p>	<p>TAUBnCNTm of the master and slave channels loads TAUBnCDRm and counts down. When the counter of the master channel reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (master) is generated</li> <li>TAUBTTOUTm (master) toggles</li> <li>TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation.</li> <li>TAUBnCNTm (slave) loads the TAUBnCDRm value or counts in the reverse direction.</li> </ul> <p>When TAUBnCNTm of the slave = 0001H:</p> <ul style="list-style-type: none"> <li>INTTAUBnlm (slave) is generated</li> <li>TAUBTTOUTm (slave) is set (in count-down status) or reset (in count-up status)</li> </ul>
Stop operation	<p>Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.</p>

Restart operation

### 22.14.5.7 Specific Timing Diagrams

#### (1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm =  $a = 5_{\text{H}}$
- Slave channel:
  - TAUBnCDRm =  $6_{\text{H}}$

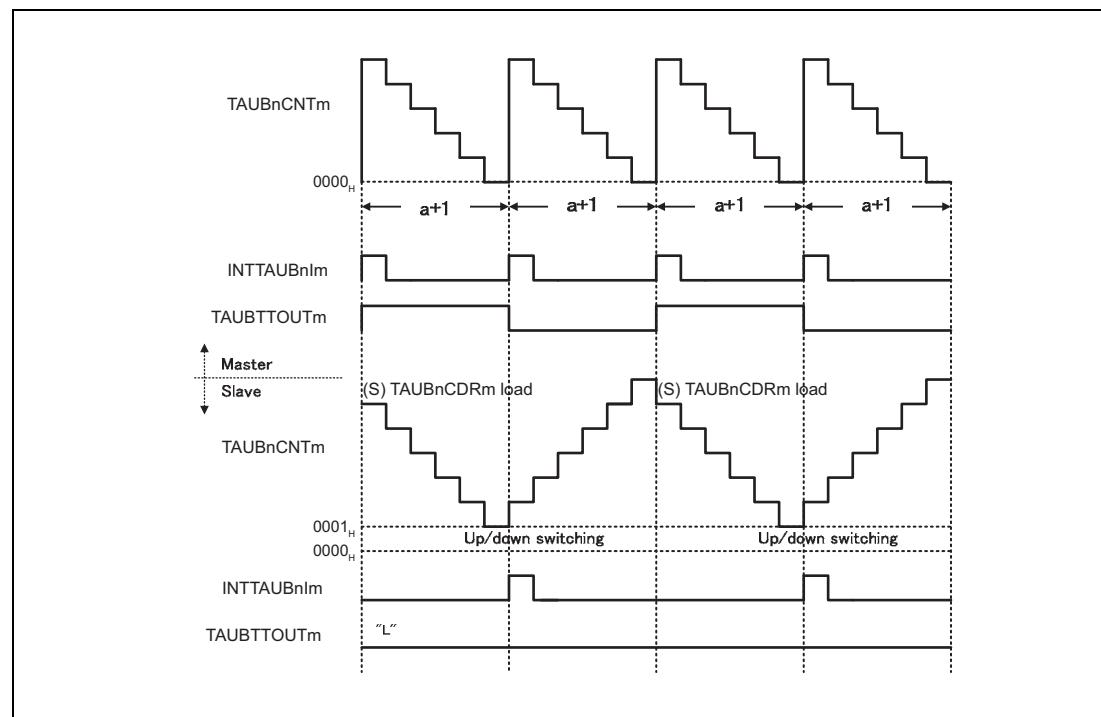


Figure 22.100 TAUBnCDRm (slave)  $\geq$  TAUBnCDRm (master) + 1

- If  $\text{TAUBnCDRm (slave)} \geq \text{TAUBnCDRm (master)} + 1$ , INTTAUBnIm of slave channel is not generated during counting down. The set signal is never detected, so TAUBTTOUTm remains at low state.

## (2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm =  $a = 5_{\text{H}}$
- Slave channel:
  - TAUBnCDRm =  $0_{\text{B}}$

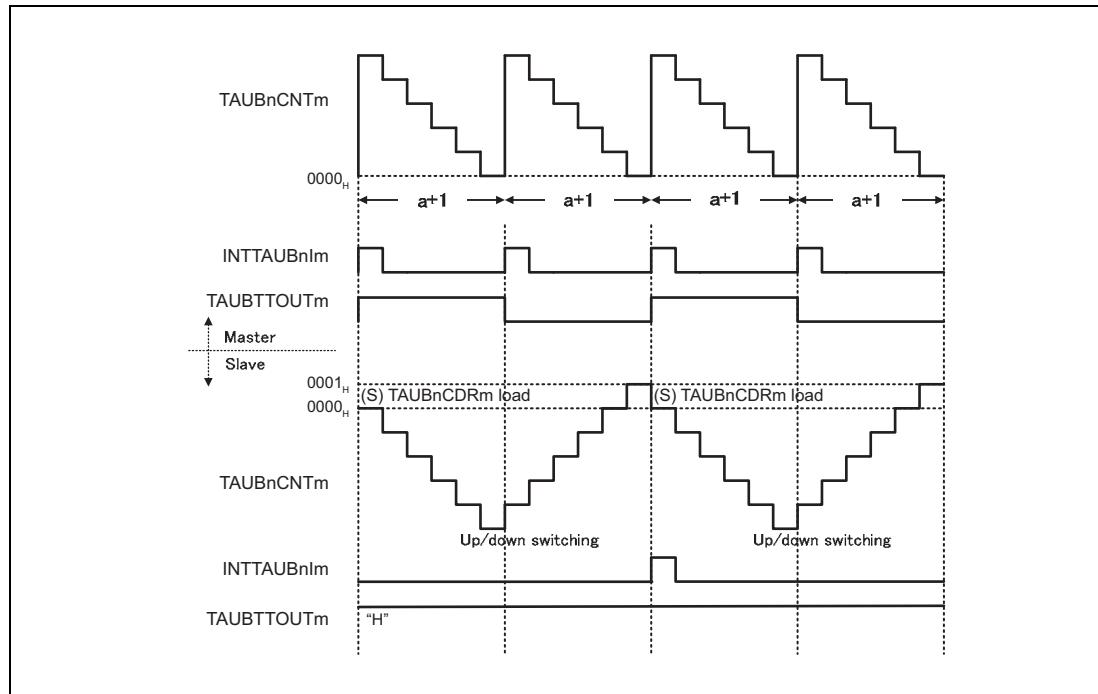


Figure 22.101 TAUBnCDRm (slave) = 0000<sub>H</sub>

- If TAUBnCDRm (slave) = 0000<sub>H</sub>, INTTAUBnIm of slave channel is not generated during counting up. The reset signal is never detected, so TAUBTTOUTm remains at high state.

## 22.14.6 Triangle PWM Output Function with Dead Time

### 22.14.6.1 Overview

#### Summary

This function generates multiple triangle PWM outputs with a defined dead time by using a master and two or more slave channels. The resulting PWM signals with the dead time are output via TAUBTTOUTm of the slave channels 2 and 3. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master generates a carrier cycle. The first pulse controls the down status and the second pulse controls the up status of the slaves counter.

An interrupt on slave 2 causes TAUBTTOUTm of the slave channels to be set or reset.

Depending on the settings of TAUBnTDL.TAUBnTDLm, delay time is added to positive or negative logic side of the signal (i.e. whether TAUBTTOUTm is set or reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

#### Prerequisites

- Three channels. Select an even channel CH (a) and an odd channel CH (a + 1) for slave channel 2 and 3 respectively.
- The operation mode of the master channel must be set to interval timer mode, see **Table 22.140, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even-numbered channel (a), and slave channel 3 is an odd-numbered channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operation mode of slave channel 2 must be set to up down mode, see **Table 22.144, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time**  
Furthermore, slave channel 2 must be an even channel
- The operation mode of slave channel 3 must be set to one-count mode, see **Table 22.148, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time**  
Furthermore, slave channel 3 must be an odd channel
- The channel output mode of the master channel must be set to independent channel output mode 1
- The channel output mode of the slave channels 2 and 3 must be set to synchronous channel output mode 2.
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.MD0 (master) bit is set to 0, TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.MD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnTOEm is 0.

**NOTE**

Slave channel 1 is not used for Triangle PWM Output Function with Dead Time. Slave channel 1 can be used as a separate timer (independent function).

**Functional description**

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEM = 1, enabling count operation. The current values of TAUBnCDRm is written to TAUBnCNTm and the counters start to count down from these values. If the master channel TAUBnCMORM.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:

When the counter of the master channel reaches  $0000_H$ , INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. The counter reloads the TAUBnCDRm value and counts down.

- Slave channel 2:

INTTAUBnIm generated on the master channel triggers the counter of the slave channel 2:

- If the slave counter currently counts down, it changes count direction.
- If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.

When the counter value of slave channel 2 reaches  $0001_H$ , INTTAUBnIm is generated.

- Slave channel 3:

INTTAUBnIm of slave channel 2 triggers the counter of slave channel 3. The current value of TAUBnCDRm (slave 3) is written to TAUBnCNTm (slave 3) and the counter starts to count down from this TAUBnCDRm value.

When the counter reaches  $0000_H$ , INTTAUBnIm is generated. The counter returns to  $FFFF_H$  and awaits the next INTTAUBnIm of slave channel 2.

The TAUBnTDL.TAUBnTDLm settings of the corresponding channel specify whether it is set/reset immediately, or after dead time has elapsed, as shown in **Table 22.139, Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2**.

The TAUBnTOL.TOLm settings specify whether set corresponds to a high signal (TAUBnTOL.TAUBnTOLm = 0) or a low signal (TAUBnTOL.TAUBnTOLm = 1).

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEM to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

TAUBnCDRm value of slave channel 2 can be set to  $0000_H$  to output 100% TAUBTTOUTm.

**NOTE**

If a forced restart is executed during operation, TAUBTTOUTm is not output as a triangle PWM signal.

### Conditions

Simultaneous rewrite can be used with this function. Please see **Section 22.6, Simultaneous Rewrite**.

TAUBnTOL.TAUBnTOLm and TAUBnTDL.TAUBnTDLm bits should be set before the counter starts, and slave channels 2 and 3 should have opposite TAUBnTOL.TAUBnTOLm settings or opposite TAUBnTDL.TAUBnTDLm settings.

**Table 22.139 Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2**

TAUBnTDL.TAUBnTDLm	Count Direction of Slave Channel 2 when Interrupt is Generated	TAUBTTOUTm Set/Reset Timing
0	Down	Set after dead time has elapsed
	Up	Reset immediately
1	Down	Set immediately
	Up	Reset after dead time has elapsed

### 22.14.6.2 Equations

$$\text{Pulse cycle} = (\text{TAUBnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$0000_{\text{H}} \leq \text{TAUBnCDRm (master)} < FFFF_{\text{H}}$$

$$\text{Carrier cycle (down/up)} = (\text{TAUBnCDRm (master)} + 1) \times 2 \times \text{count clock cycle}$$

$$\begin{aligned} \text{PWM signal width (positive phase)} &= [(\text{TAUBnCDRm (master)} + 1 - \\ &\quad \text{TAUBnCDRm (slave 2)}) \times 2 - (\text{TAUBnCDRm (slave 3)} + 1)] \times \text{count clock cycle} \end{aligned}$$

$$\begin{aligned} \text{PWM signal width (negative phase)} &= [(\text{TAUBnCDRm (master)} + 1 - \\ &\quad \text{TAUBnCDRm (slave 2)}) \times 2 + (\text{TAUBnCDRm (slave 3)} + 1)] \times \text{count clock cycle} \end{aligned}$$

### 22.14.6.3 Block Diagram and General Timing Diagram

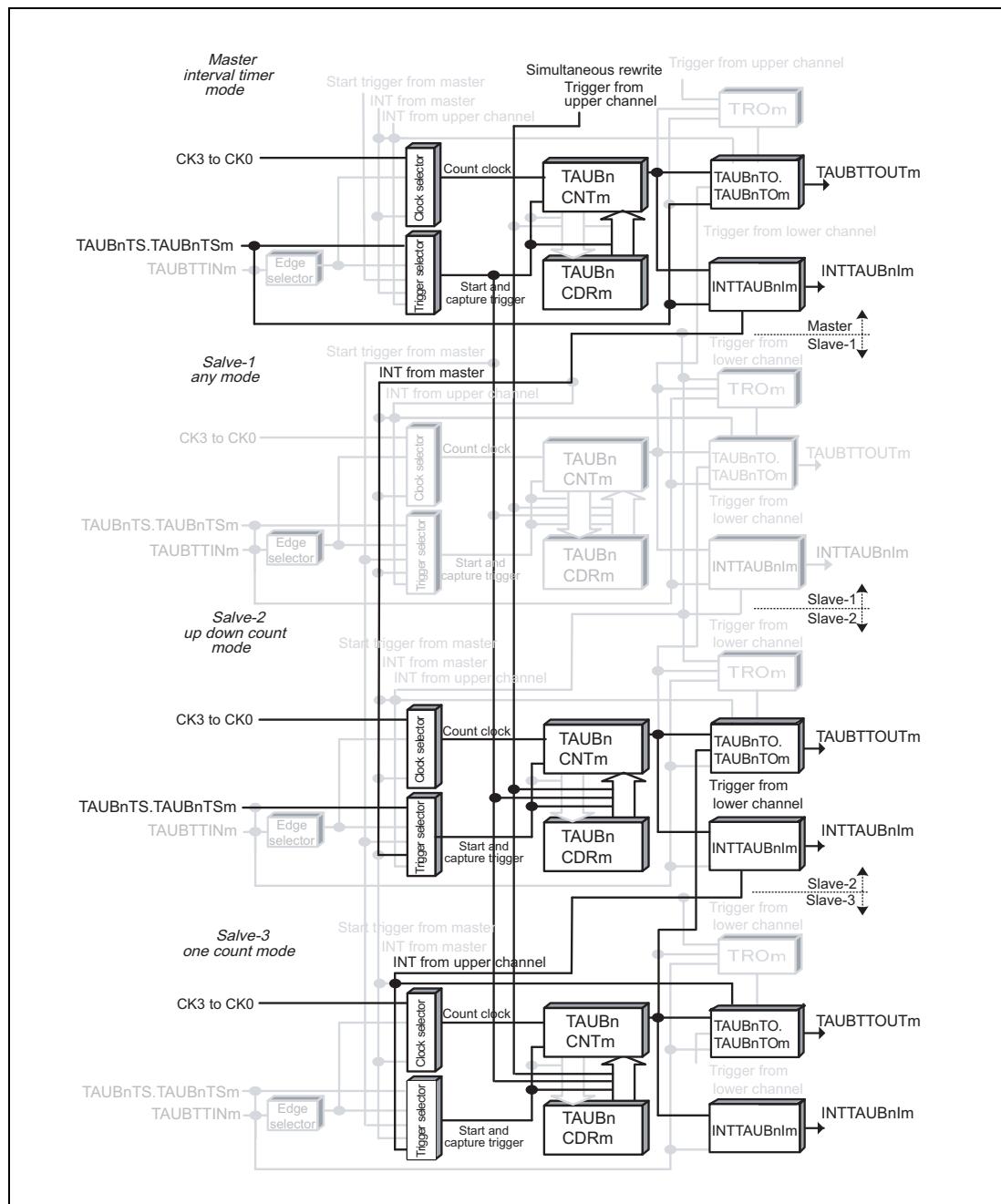


Figure 22.102 Block Diagram for Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Slave channel 2:
  - INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
  - TAUBnTDL.TAUBnTDLm = 0
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Enables start trigger detection during counting (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnTDL.TAUBnTDLm = 1
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)

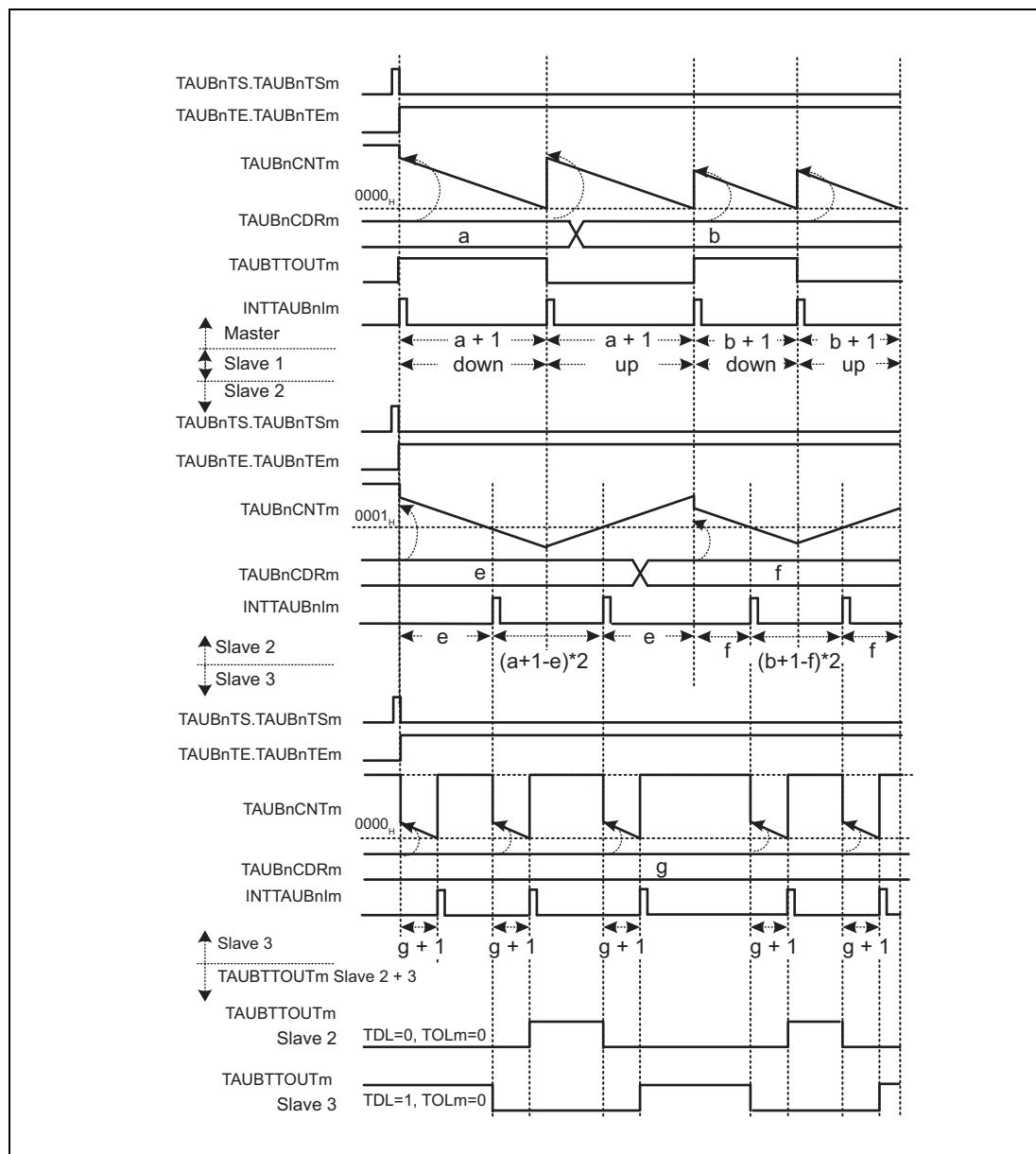


Figure 22.103 General Timing Diagram for Triangle PWM Output Function with Dead Time

#### 22.14.6.4 Register Settings for the Master Channel

##### (1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.140 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 1 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUBnMDO	0: INTTAUBnlm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnlm and toggles TAUBTTOUTm at operation start

##### (2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.141 Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for the master channel

Table 22.142 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 0 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 0 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	Write 0 <sub>B</sub> .
TAUBnTDE.TAUBnTDEm	Write 0 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	Write 0 <sub>B</sub> .

### (4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 22.143 Simultaneous Rewrite Settings for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

#### NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

### 22.14.6.5 Register Settings for Slave Channel 2

#### (1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.144 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 111 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001 <sub>B</sub> .
0	TAUBnMDO	Write 0 <sub>B</sub> .

#### (2) TAUBnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.145 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for slave channel 2

**Table 22.146 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output**

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 1 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 1 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase
<b>CAUTION</b>	
Set TAUBnTDL.TAUBnTDLm exclusively to the odd channel.	

### (4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.147 Simultaneous Rewrite Settings for Slave Channel 2 of the Triangle PWM Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.6.6 Register Settings for Slave Channel 3

#### (1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]	—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]	TAUBnCOS [1:0]	—	TAUBnMD[4:1]	—	TAUBn MDO						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 22.148 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0 <sub>B</sub> .
11	TAUBnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUBnSTS[2:0]	Write 110 <sub>B</sub> .
7, 6	TAUBnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUBnMDO	Write 1 <sub>B</sub> .

#### (2) TAUBnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 22.149 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode for slave channel 3

**Table 22.150 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output**

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1 <sub>B</sub> .
TAUBnTOM.TAUBnTOMm	Write 1 <sub>B</sub> .
TAUBnTOC.TAUBnTOCm	Write 1 <sub>B</sub> .
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1 <sub>B</sub> .
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase
<b>CAUTION</b>	
Set TAUBnTDL.TAUBnTDLm exclusively to the even channel.	

### (4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 22.151 Simultaneous Rewrite Settings for Slave Channel 3 of the Triangle PWM Output Function**

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

### 22.14.6.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 22.152 Operating Procedure for Triangle PWM Output with Dead Time

	Operation	Status of TAUBn
Initial channel setting	<p>Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.6.4, Register Settings for the Master Channel</b></p> <p>Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.6.5, Register Settings for Slave Channel 2</b></p> <p>Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in <b>Section 22.14.6.6, Register Settings for Slave Channel 3</b></p> <p>Set the values of the TAUBnCDRm registers of all channels</p>	Channel operation is stopped.
Start operation	<p>Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated when TAUBnCMORm.TAUBnMD0 is set to 1 on the master channel.</p>
During operation	<p>TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.</p>	<p>TAUBnCNTm of the master channel and slave channel 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000H:</p> <ul style="list-style-type: none"> <li>• INTTAUBnIm (master) is generated</li> <li>• TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation</li> <li>• TAUBnCNTm (slave 2) reloads the TAUBnCDRm value or counts in the reverse direction</li> </ul> <p>When TAUBnCNTm (slave 2) reaches 0001H:</p> <ul style="list-style-type: none"> <li>• INTTAUBnIm (slave 2) is generated</li> <li>• TAUBnCNTm of slave channel 3 loads the TAUBnCDRm value and counts down</li> </ul> <p>When TAUBnCNTm of slave channel 3 = 0000H:</p> <ul style="list-style-type: none"> <li>• INTTAUBnIm is generated</li> </ul>
Stop operation	<p>Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTOUTm stop and retain their current values.</p>



### 22.14.6.8 Specific Timing Diagrams

#### (1) Duty cycle = 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic ( $\text{TAUBnTOL} \cdot \text{TAUBnTOLm} = 0$ )
- Slave channel 3:
  - Negative logic ( $\text{TAUBnTOL} \cdot \text{TAUBnTOLm} = 1$ )

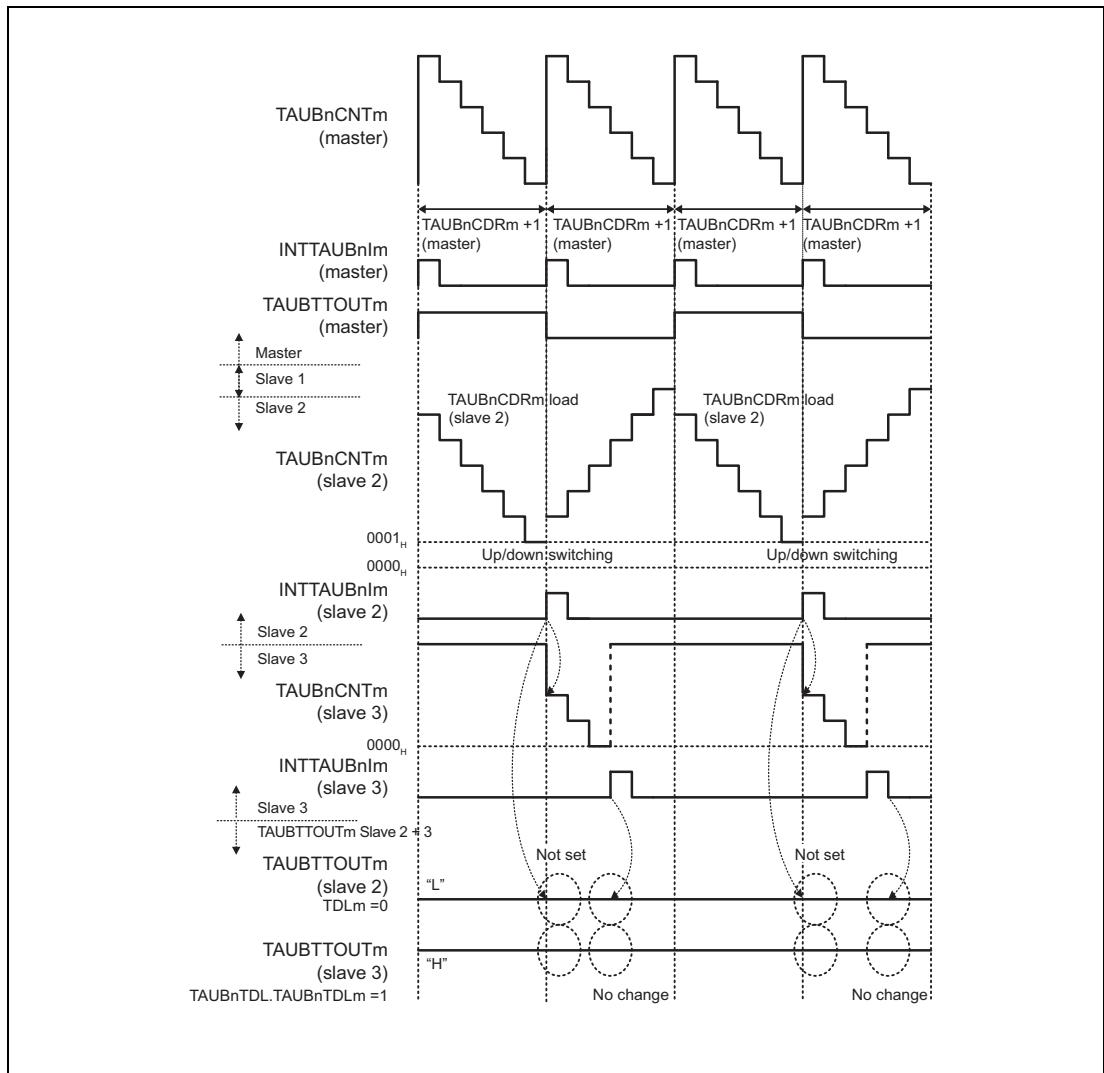


Figure 22.104  $\text{TAUBnCDRm} (\text{slave } 2) \geq \text{TAUBnCDRm} (\text{master}) + 1$

- If  $\text{TAUBnCDRm} (\text{slave } 2) \geq \text{TAUBnCDRm} (\text{master})$ , the counter of slave channel cannot reach  $0000_H$  during counting down. Therefore  $\text{TAUBTTOUTm}$  cannot toggle, i.e. it remains at its initial state. The interrupt from slave channel 2 occurs during count up, therefore it is a reset signal.

## (2) Duty cycle = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic ( $\text{TAUBnTOL}.\text{TAUBnTOLm} = 0$ )
- Slave channel 3:
  - Negative logic ( $\text{TAUBnTOL}.\text{TAUBnTOLm} = 1$ )

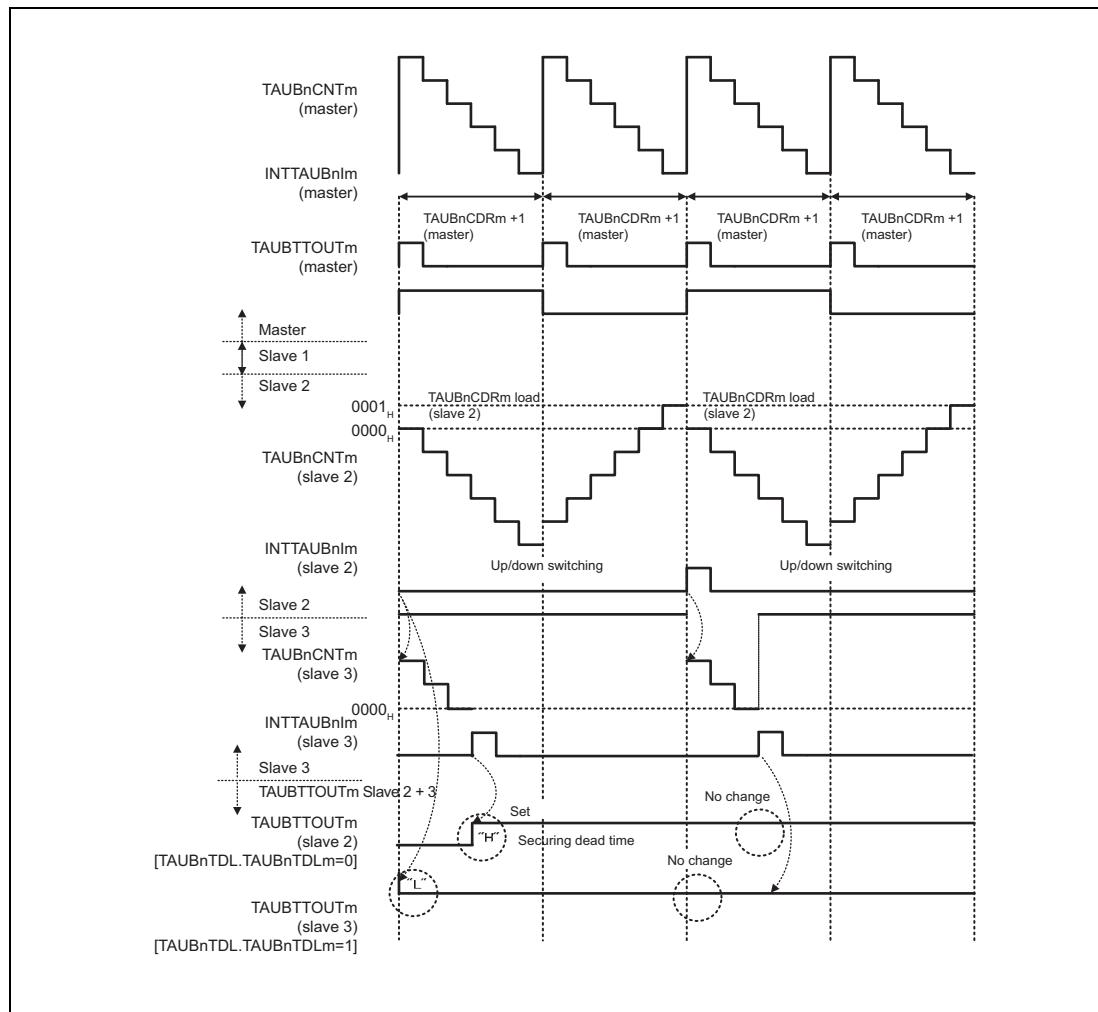


Figure 22.105  $\text{TAUBnCDRm}$  (slave 2) = 0000<sub>H</sub>

- If  $\text{TAUBnCDRm}$  (slave 2) = 0000<sub>H</sub> the counter of slave channel cannot reach 0001<sub>H</sub> while counting up and therefore cannot generate an  $\text{INTTAUBnIm}$  while counting up.
  - The set conditions for a channel in which  $\text{TAUBnTDL}.\text{TAUBnTDLm} = 0$  are met after dead time has elapsed.  $\text{TAUBTTOUTm}$  toggles but remains in the new state because the reset conditions are never satisfied for such a channel.
  - Slave channel 3 in the above diagram is set when the counter starts. However, the reset conditions for a channel in which  $\text{TAUBnTDL}.\text{TAUBnTDLm} = 1$  are never satisfied so  $\text{TAUBTTOUTm}$  remains in its initial state for such a slave channel.

### (3) TAUBTOUTm (slave 2) = 0% and TAUBTOUTm (slave 3) $\geq$ 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

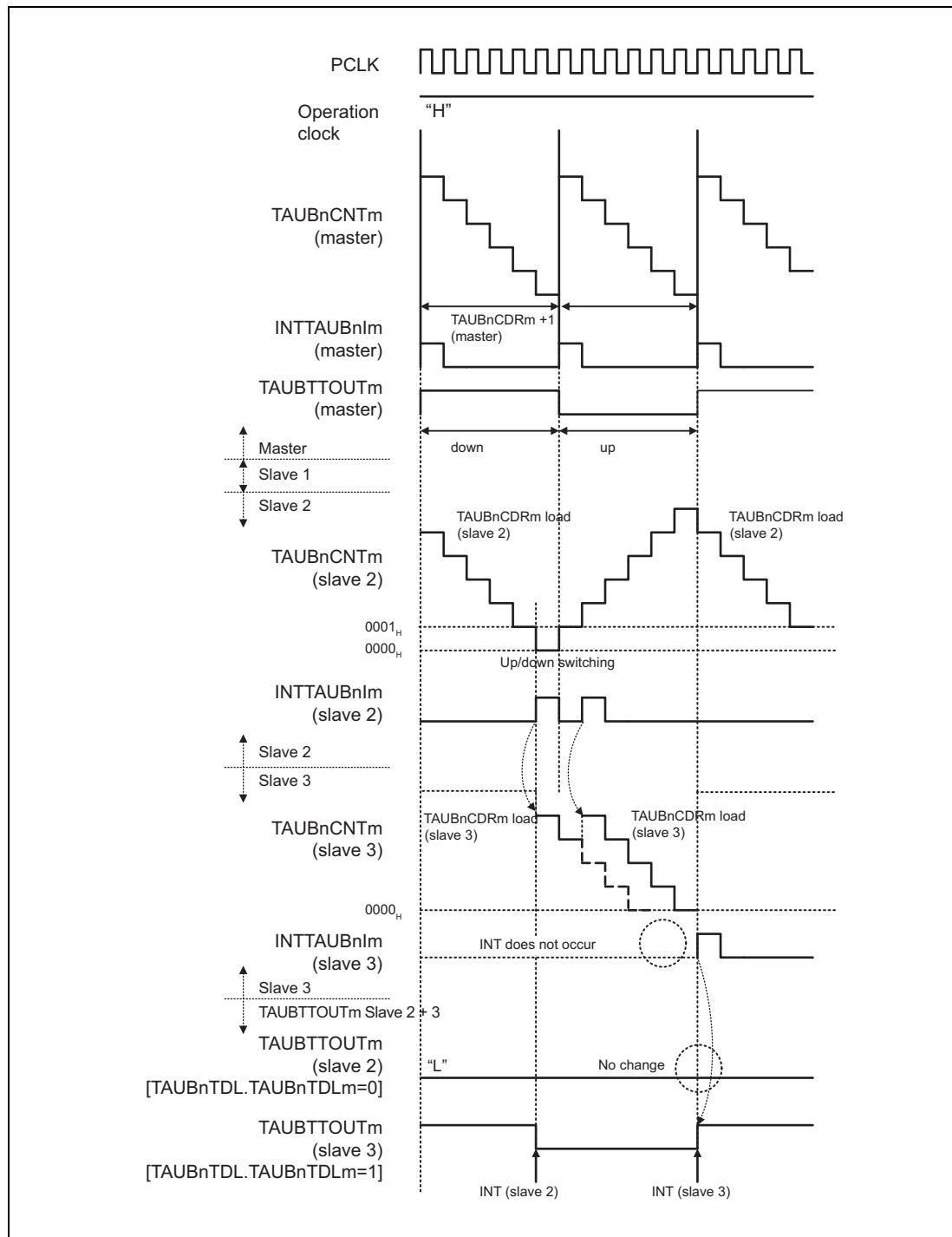


Figure 22.106 TAUBnCDRm (master) =  $0005_{\text{H}}$ , TAUBnCDRm (slave 2) =  $0005_{\text{H}}$   
TAUBnCDRm (slave 3) =  $0004_{\text{H}}$

- When the counter of slave channel 2 reaches  $0000_H$  after detecting that the counter reached  $0001_H$ , INTTAUBnIm (slave 2) is generated. The counter of slave channel 3 starts to count down.
- If another INTTAUBnIm (slave 2) is generated while the counter of slave channel 3 is still counting down, the value of TAUBnCDRm (slave 3) is reloaded and the counter restarts counting down from this value.
- In the diagram above, the first interrupt on channel 2 occurs while the counter is counting down, and the second while it is counting up.
- After the first interrupt, a slave for which TAUBnTDL.TAUBnTDLm = 0 waits for dead time to elapse before setting. However, if another interrupt occurs on slave 2, before the dead time has elapsed, the counter is counting up, so the signal acts as a reset signal, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 0 always remains inactive.
- TAUBTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 is set and reset as normal when the corresponding INTTAUBnIm is generated.

#### (4) TAUBTOUTm (slave 2) > 0% and TAUBTOUTm (slave 3) = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

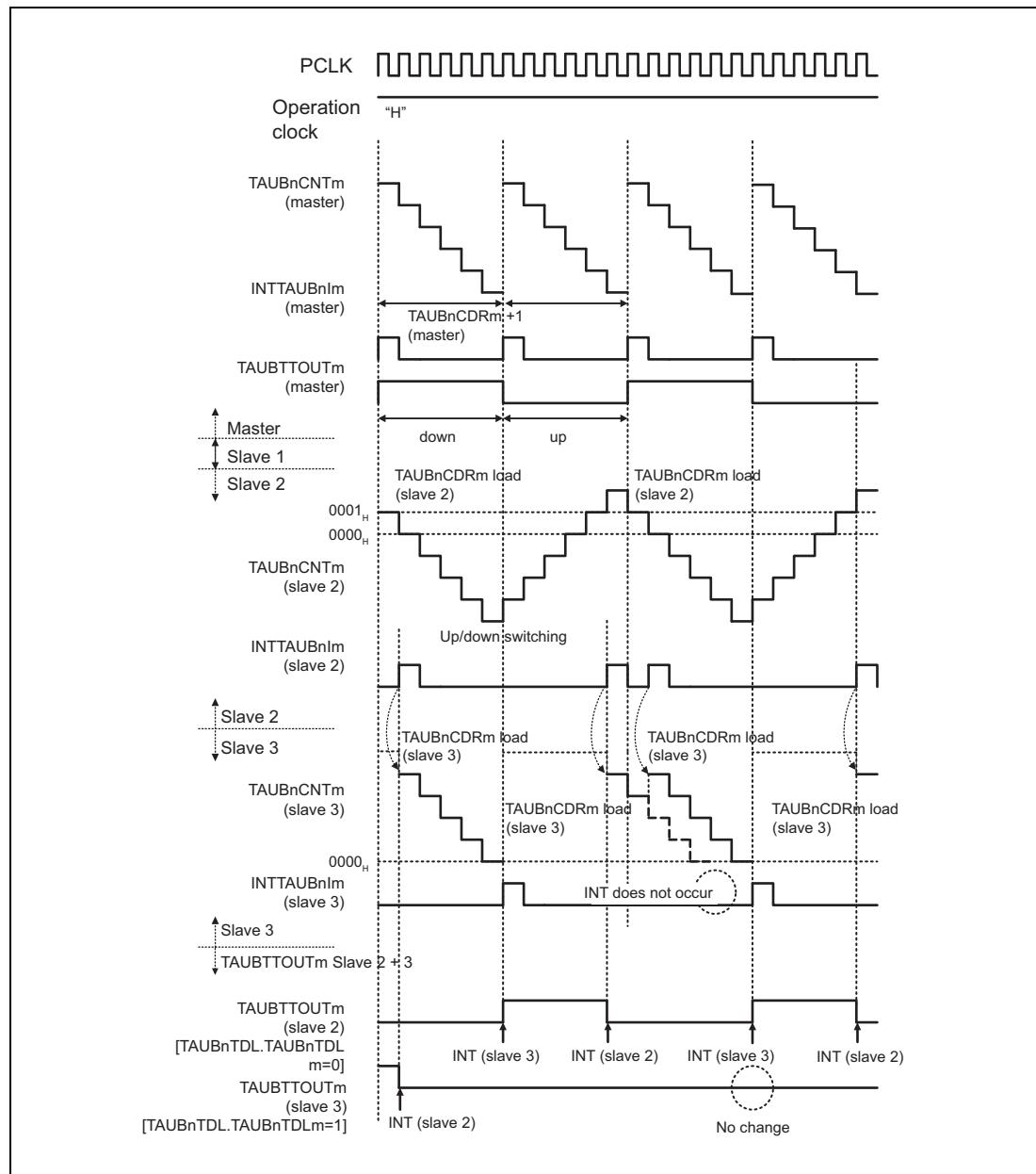


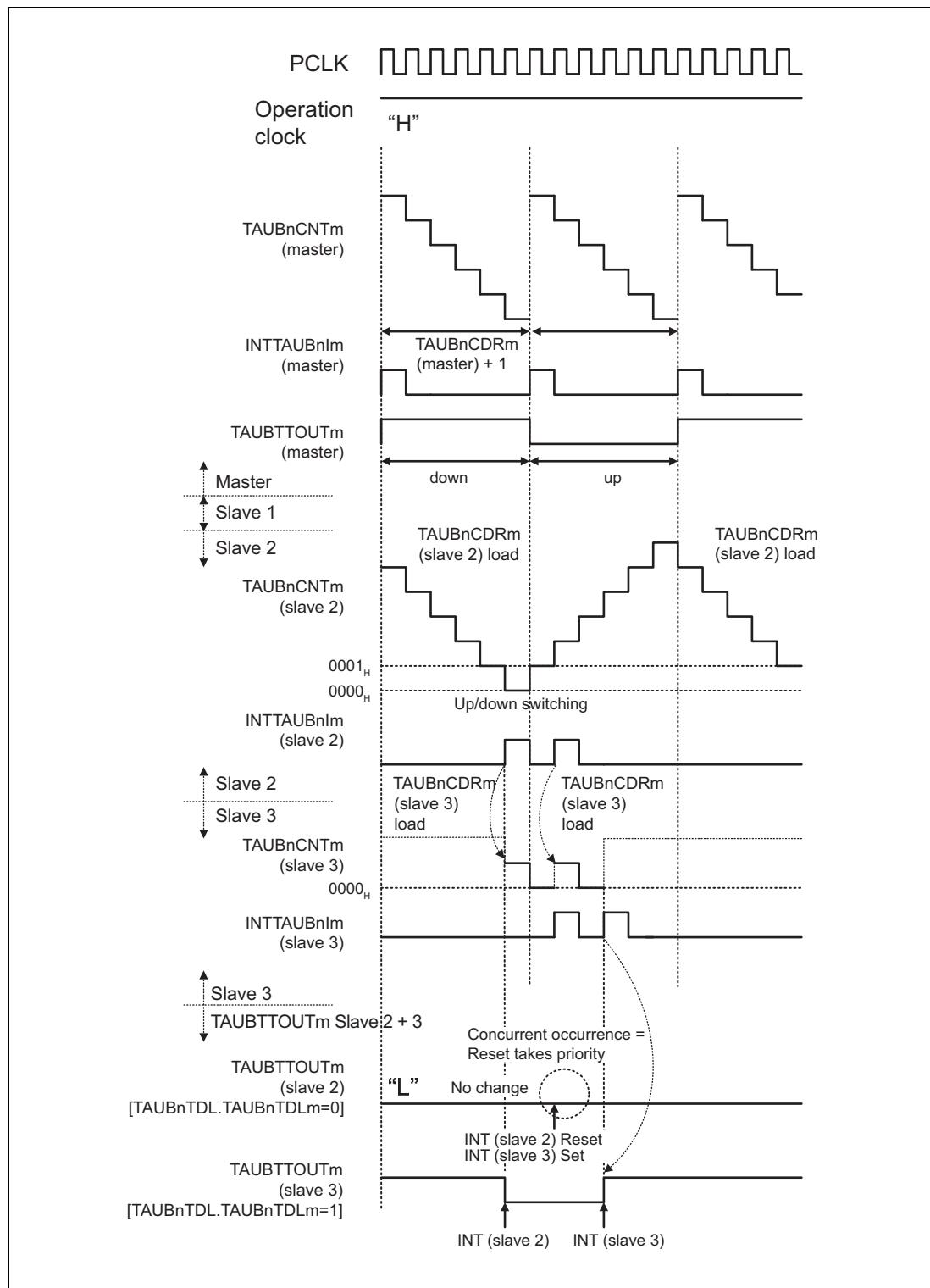
Figure 22.107 TAUBnCDRm (master) = 0005<sub>H</sub>, TAUBnCDRm (slave 2) = 0001<sub>H</sub>  
 TAUBnCDRm (slave 3) = 0004<sub>H</sub>  
 PWM Signal Width (negative phase) ≥ Carrier Cycle

- After the second interrupt on slave channel 2, a slave for which TAUBnTDL.TAUBnTDLM = 1 is reset after the dead time has elapsed. However if another interrupt occurs on slave 2 before the dead time has elapsed, slave channel 3 is restarted, and then if an interrupt on slave channel 3 is generated, the counter is counting up, so the signal acts as a setting signal, meaning that a channel for which TAUBnTDL.TAUBnTDLM = 1 always remains active.
- TAUBTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLM = 0 is set and reset as normal when the corresponding INTTAUBnIm is generated.

**(5) Inhibited INTTAUBnIm to set TAUBTOUTm positive phase period**

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)



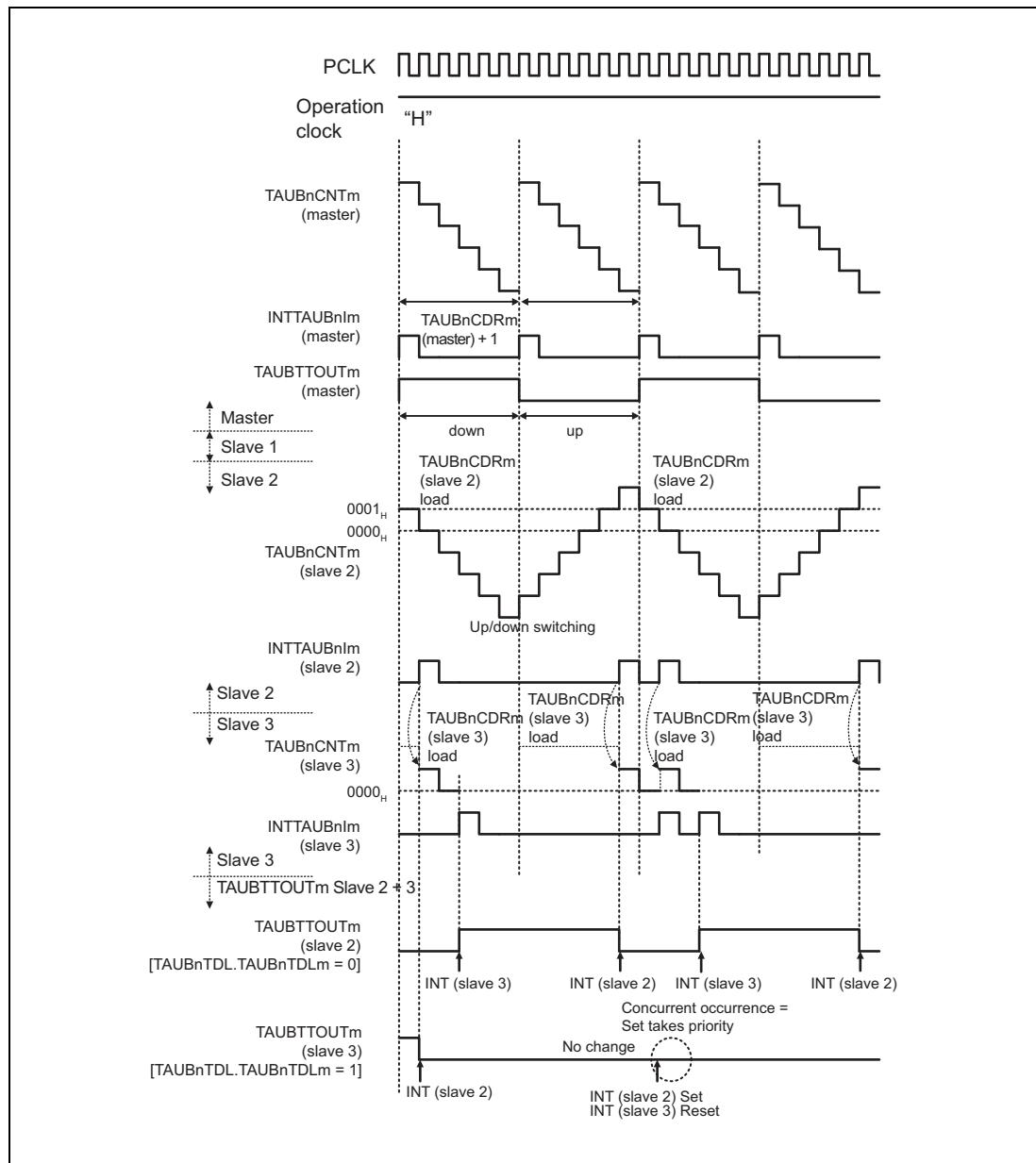
**Figure 22.108** TAUBnCDRm (master) =  $0005_H$ , TAUBnCDRm (slave 2) =  $0005_H$ ,  
 TAUBnCDRm (slave 3) =  $0001_H$   
 PWM Signal Width (positive phase) = 0

- The counter of slave channel 3 reaches  $0000_H$  and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 0 (slave channel 2 in this example).
- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, this reset signal is given priority if TAUBnTOL.TAUBnTOLm = 0 (if TAUBnTOL.TAUBnTOLm = 1, the set signal is given priority).
- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 remains in the value after reset.

### (6) Inhibited INTTAUBnIm to set TAUBTTOUTm negative phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)



**Figure 22.109** TAUBnCDRm (master) = 0005<sub>H</sub>, TAUBnCDRm (slave 2) = 0001<sub>H</sub>,  
TAUBnCDRm (slave 3) = 0001<sub>H</sub>  
PWM Signal Width (negative phase) = Carrier Cycle

- The counter of slave channel 3 reaches 0000<sub>H</sub> and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 1 (slave 3 in this example).

- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, the set signal is given priority if TAUBnTOL.TAUBnTOLm = 1 (if TAUBnTOL.TAUBnTOLm = 0, the reset signal is given priority).
- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 remains in the value after reset.

#### (7) Slave 2 TAUBnCDRm = 0000<sub>H</sub> (Duty cycle = 100%)

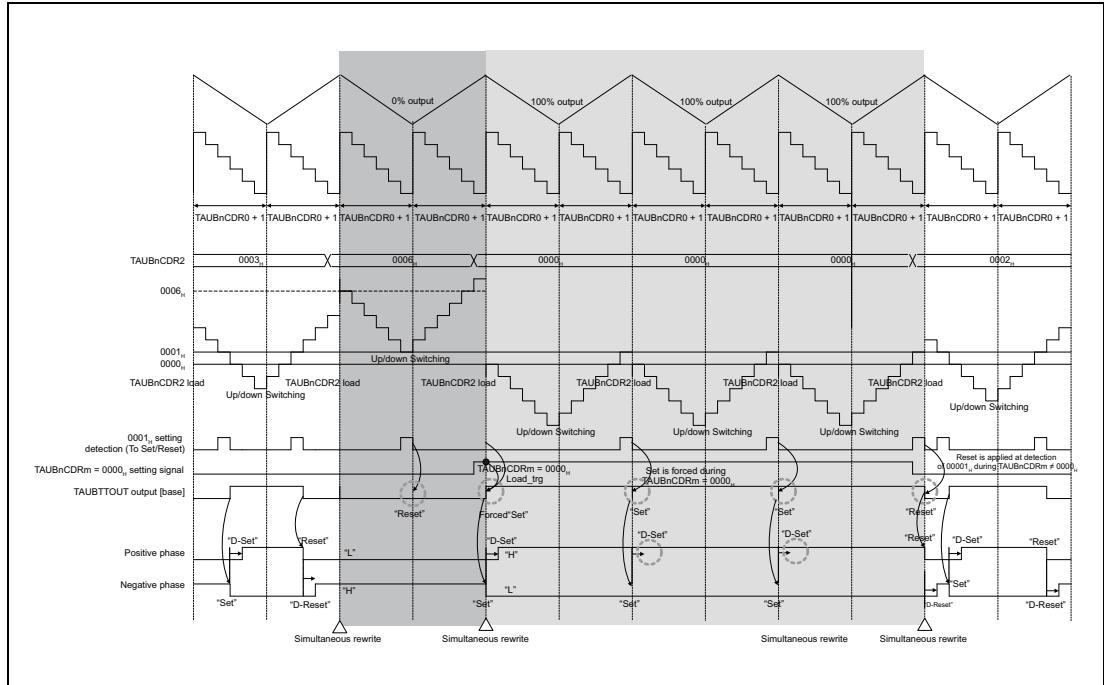


Figure 22.110 Slave 2 TAUBnCDRm = 0000<sub>H</sub> (Duty cycle = 100%)

When rewriting (slave channel 2) TAUBnCDRm  $\neq$  0000<sub>H</sub> to (slave channel 2) TAUBnCDRm = 0000<sub>H</sub> (100% output), set the negative phase side at the start of the carrier cycle, and set the positive phase side after dead time is secured.

When rewriting (slave channel 2) TAUBnCDRm = 0000<sub>H</sub> (100% output) to (slave channel 2) TAUBnCDRm  $\neq$  0000<sub>H</sub>, reset the positive phase side at the end of the carrier cycle, and reset the negative phase side after dead time is secured.

## 22.14.7 AD Conversion Trigger Output Function Type 2

### 22.14.7.1 Overview

#### Summary

This function is enabled by setting the channel output mode of the slave to independent channel output mode controlled by software.

### 22.14.7.2 Block Diagram and General Timing Diagram

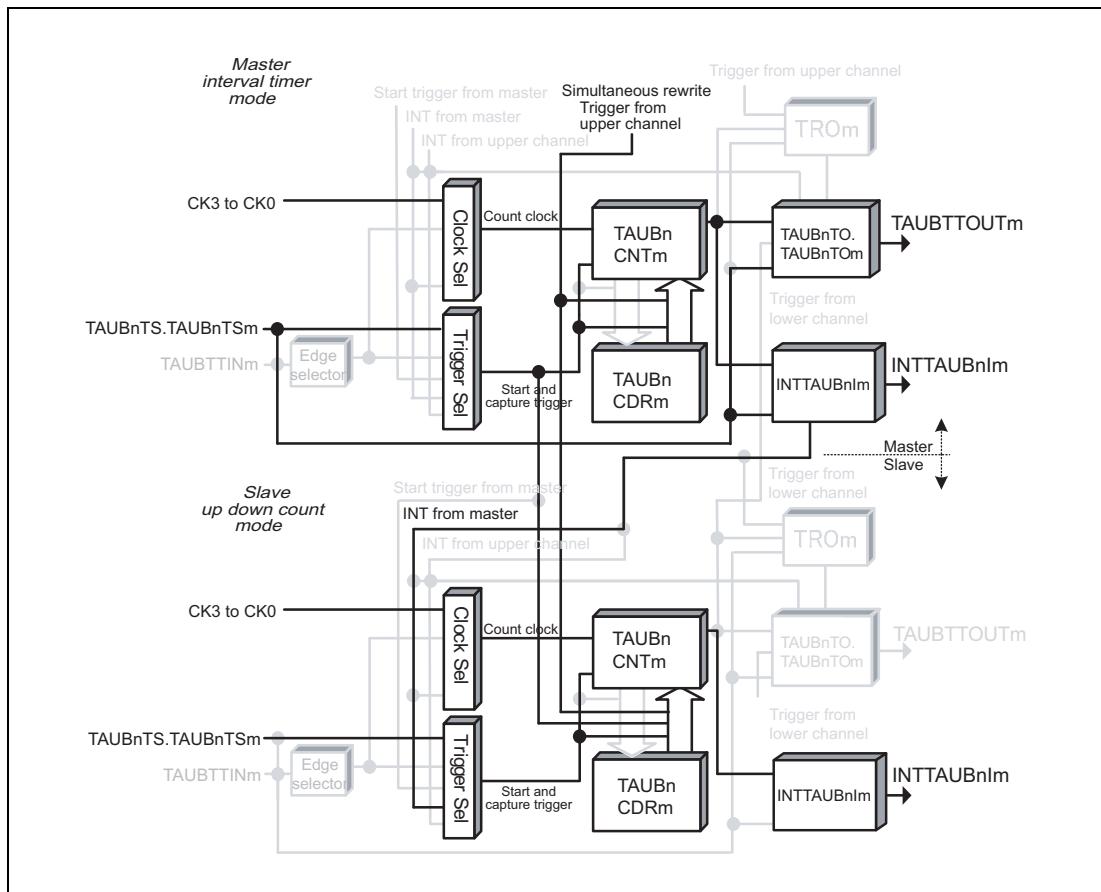
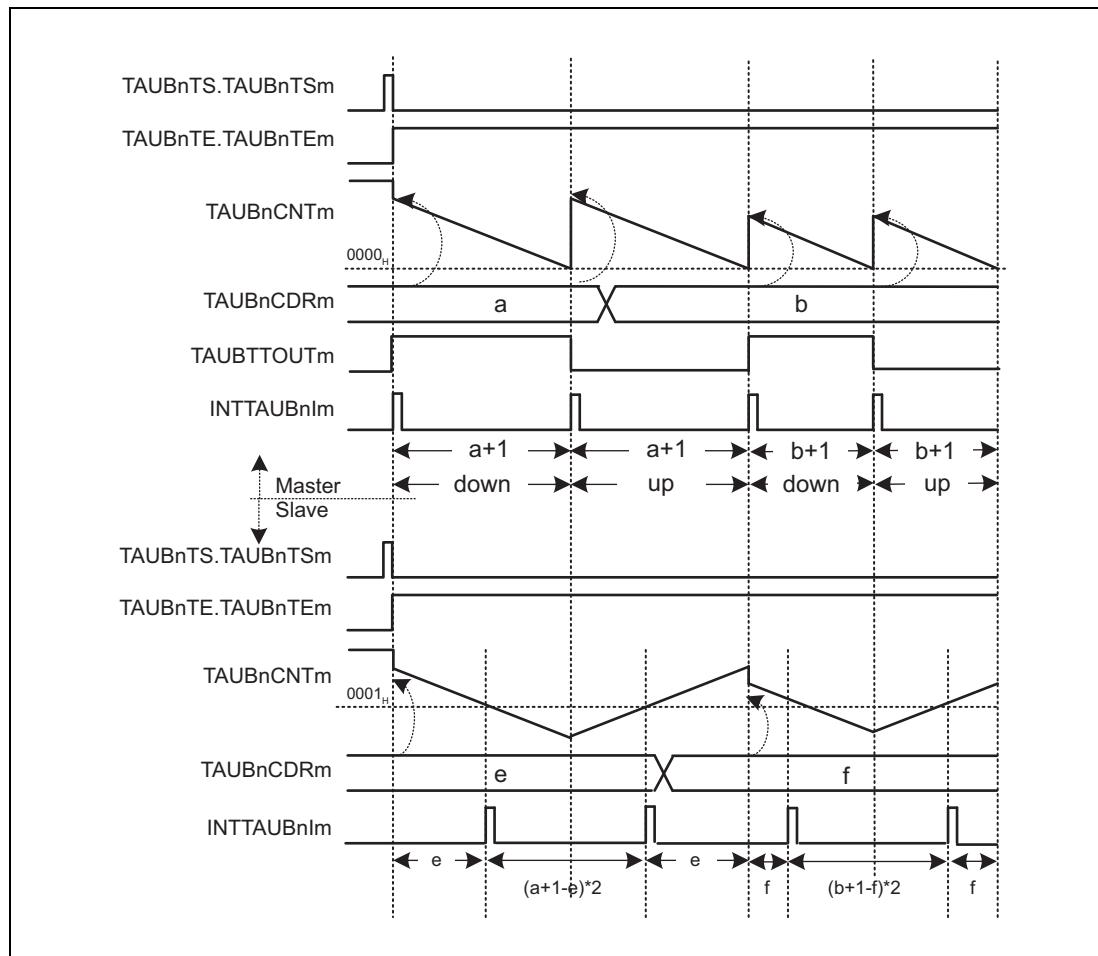


Figure 22.111 Block Diagram for AD Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)



**Figure 22.112 General Timing Diagram for AD Conversion Trigger Output Function Type 2**

## Section 23 Timer Array Unit D (TAUD)

This section contains a generic description of the timer array unit D (TAUD).

The first part of this section describes the RH850/F1L specific features such as the number of units and the register base addresses. The remainder of the section describes the functions and registers of the TAUD.

### 23.1 Features of RH850/F1L TAUD

#### 23.1.1 Number of Units and Channels

This microcontroller has the following number of TAUD units.

**Table 23.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units				1		
Name				TAUDn (n = 0)		

TAUDn has the following timers for the quantity channels.

**Table 23.2 TAUDn Unit Configurations and Channels**

Unit Name (Channel Name)	Channels per Unit	RH850/F1L 48 pins (16 ch)	RH850/F1L 64 pins (16 ch)	RH850/F1L 80 pins (16 ch)	RH850/F1L 100 pins (16 ch)	RH850/F1L 144 pins (16 ch)	RH850/F1L 176 pins (16 ch)
TAUD0	16	√	√	√	√	√	√

**Table 23.3 Indices**

Index	Description
n	Throughout this section, the individual TAUD units are identified by the index "n"; for example, TAUDnTOM is the TAUDn channel output mode register.
m	The TAUD has 16 channels. Throughout this section, the individual channels are identified by the index "m" (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

#### 23.1.2 Register Base Address

TAUDn base address is listed in the following table.

TAUDn register addresses are given as offsets from the base address.

**Table 23.4 Register Base Address**

Base Address Name	Base Address
<TAUD0_base>	FFE2 0000 <sub>H</sub>

### 23.1.3 Clock Supply

The TAUDn clock supply is shown in the following table.

**Table 23.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
TAUDn	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

### 23.1.4 Interrupt Requests

TAUDn interrupt requests are listed in the following table.

**Table 23.6 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>TAUDO</b>			
INTTAUD0I0	Channel 0 interrupt	0, 124	0 (Channels 0 to 7)
INTTAUD0I1	Channel 1 interrupt	39	15 (Channels 0 to 7)
INTTAUD0I2	Channel 2 interrupt	1, 150	0 (Channels 8 to 15)
INTTAUD0I3	Channel 3 interrupt	40	12 (Channels 8 to 15)
INTTAUD0I4	Channel 4 interrupt	2	1 (Channels 0 to 7)
INTTAUD0I5	Channel 5 interrupt	41	16 (Channels 0 to 7)
INTTAUD0I6	Channel 6 interrupt	3	1 (Channels 8 to 15)
INTTAUD0I7	Channel 7 interrupt	42	13 (Channels 8 to 15)
INTTAUD0I8	Channel 8 interrupt	4	2 (Channels 0 to 7)
INTTAUD0I9	Channel 9 interrupt	43	17 (Channels 0 to 7)
INTTAUD0I10	Channel 10 interrupt	5, 151	2 (Channels 8 to 15)
INTTAUD0I11	Channel 11 interrupt	44	14 (Channels 8 to 15)
INTTAUD0I12	Channel 12 interrupt	6, 152	3 (Channels 0 to 7)
INTTAUD0I13	Channel 13 interrupt	45	18 (Channels 0 to 7)
INTTAUD0I14	Channel 14 interrupt	7, 153	3 (Channels 8 to 15)
INTTAUD0I15	Channel 15 interrupt	46	15 (Channels 8 to 15)

### 23.1.5 Reset Sources

TAUDn reset sources are listed in the following table. TAUDn is initialized by these reset sources.

**Table 23.7 Reset Sources**

Unit Name	Reset Source
TAUDn	All reset sources (ISORES)

### 23.1.6 External Input/Output Signals

External input/output signals of TAUDn are listed below.

**Table 23.8 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal Name
<b>TAUD0</b>		
TAUDTTIN0	Channel 0 input* <sup>1</sup>	TAUD0I0
TAUDTTIN1	Channel 1 input* <sup>1</sup>	TAUD0I1
TAUDTTIN2	Channel 2 input* <sup>1</sup>	TAUD0I2
TAUDTTIN3	Channel 3 input* <sup>1</sup>	TAUD0I3
TAUDTTIN4	Channel 4 input* <sup>1</sup>	TAUD0I4
TAUDTTIN5	Channel 5 input* <sup>1</sup>	TAUD0I5
TAUDTTIN6	Channel 6 input* <sup>1</sup>	TAUD0I6
TAUDTTIN7	Channel 7 input* <sup>1</sup>	TAUD0I7
TAUDTTIN8	Channel 8 input* <sup>1</sup>	TAUD0I8
TAUDTTIN9	Channel 9 input* <sup>1</sup>	TAUD0I9
TAUDTTIN10	Channel 10 input* <sup>1</sup>	TAUD0I10
TAUDTTIN11	Channel 11 input* <sup>1</sup>	TAUD0I11
TAUDTTIN12	Channel 12 input* <sup>1</sup>	TAUD0I12
TAUDTTIN13	Channel 13 input* <sup>1</sup>	TAUD0I13
TAUDTTIN14	Channel 14 input* <sup>1</sup>	TAUD0I14
TAUDTTIN15	Channel 15 input* <sup>1</sup>	TAUD0I15
TAUDTTOUT0	Channel 0 output	TAUD0O0
TAUDTTOUT1	Channel 1 output	TAUD0O1
TAUDTTOUT2	Channel 2 output	TAUD0O2
TAUDTTOUT3	Channel 3 output	TAUD0O3
TAUDTTOUT4	Channel 4 output	TAUD0O4
TAUDTTOUT5	Channel 5 output	TAUD0O5
TAUDTTOUT6	Channel 6 output	TAUD0O6
TAUDTTOUT7	Channel 7 output	TAUD0O7
TAUDTTOUT8	Channel 8 output	TAUD0O8
TAUDTTOUT9	Channel 9 output	TAUD0O9
TAUDTTOUT10	Channel 10 output	TAUD0O10
TAUDTTOUT11	Channel 11 output	TAUD0O11
TAUDTTOUT12	Channel 12 output	TAUD0O12
TAUDTTOUT13	Channel 13 output	TAUD0O13
TAUDTTOUT14	Channel 14 output	TAUD0O14
TAUDTTOUT15	Channel 15 output	TAUD0O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions.  
For details, see **Section 2.12, Noise Filter & Edge/Level Detector**.

#### CAUTION

When port P0\_0 is used as TAUD0I2 or TAUD0O2, port P0\_0 (the RESETOUT signal) outputs a low level during a reset and after release from the reset state.

For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

### 23.1.7 Internal Input/Output Signals

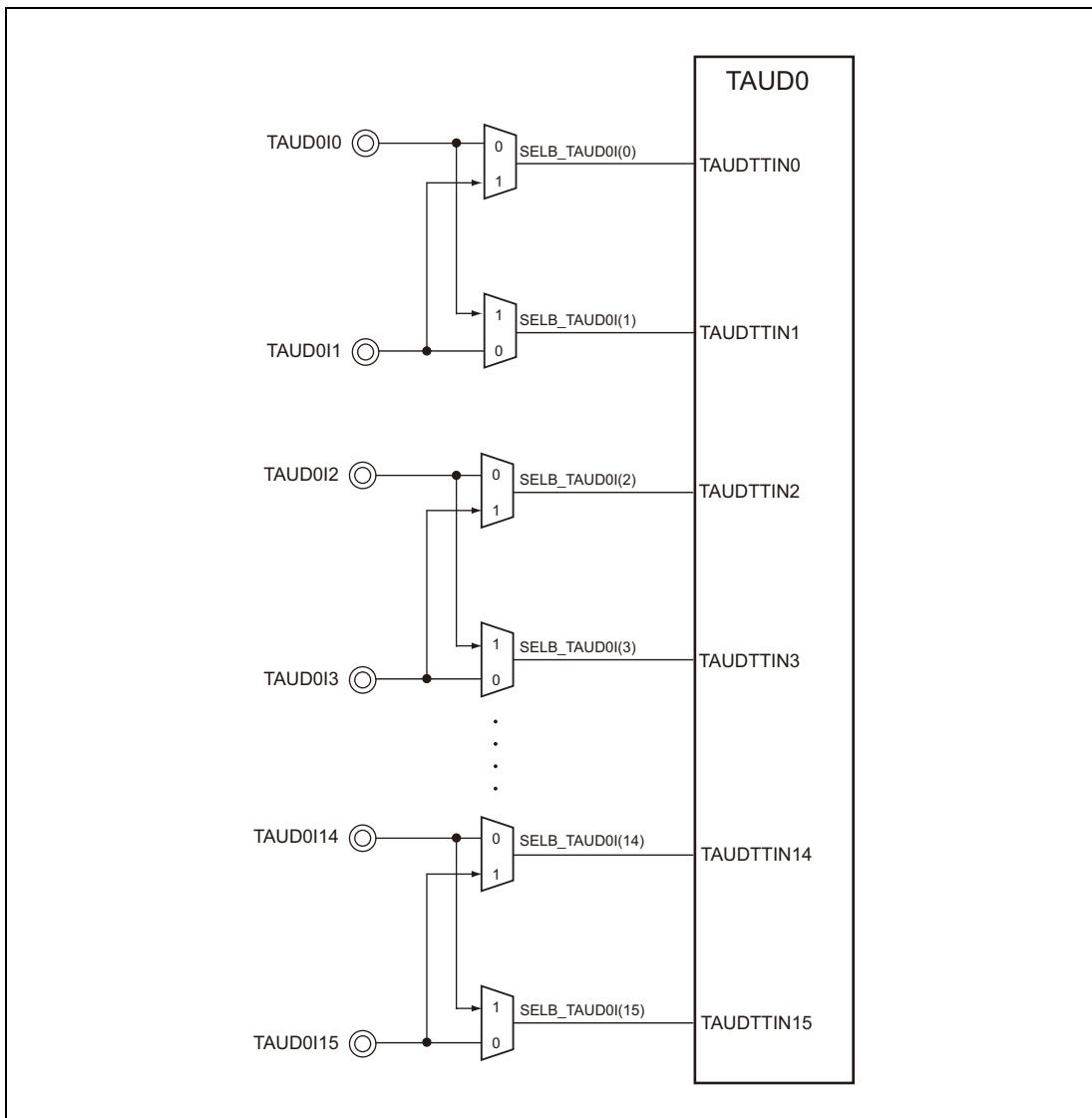
The internal input/output signals of TAUDn are listed below.

**Table 23.9 Internal Input/Output Signals**

Unit Signal Name	Description	Connected to
TAUDnTSSTm	Simultaneous channel start trigger input	PIC
TAUDnTUDCm (m = 0, 2, 8)	TAUD master up/down signal output	PIC

### 23.1.8 TAUD0 Input Selection

The output from port TAUD0Im (m = 0 to 15) can be input to TAUDTTINm (m = 0 to 15) as shown in the following figure.



**Figure 23.1 Selection of Signals Input to TAUD0**

The following table shows the method of selecting input signals to several TAUD0 inputs.

**Table 23.10 TAUD0 Input Selection**

Input Signal	Function	Settings
TAUDTTIN [m]	Port TAUD0I[m]	SELB_TAUD0I [m] = 0
	Port TAUD0I[m + 1]	SELB_TAUD0I [m] = 1
TAUDTTIN [m + 1]	Port TAUD0I[m + 1]	SELB_TAUD0I [m + 1] = 0
	Port TAUD0I[m]	SELB_TAUD0I [m + 1] = 1

### 23.1.8.1 SELB\_TAUD0I — TAUDTTINm Input Signal Selection Register

This register selects the TAUDTTINm input signals.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFBC 0200<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SELB_TAUD0I	15	SELB_TAUD0I	14	SELB_TAUD0I	13	SELB_TAUD0I	12	SELB_TAUD0I	11	SELB_TAUD0I	10	SELB_TAUD0I	9	SELB_TAUD0I	8	SELB_TAUD0I	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

**Table 23.11 SELB\_TAUD0I Register Contents**

Bit Position	Bit Name	Function																				
15 to 0	SELB_TAUD0Im	Selection of TAUDTTINm input signal																				
		<table border="1"> <thead> <tr> <th>TAUD Input</th> <th>Bit [m+1]</th> <th>Bit [m]</th> <th>Input signal</th> </tr> </thead> <tbody> <tr> <td>TAUDTTIN[m]</td> <td>x</td> <td>0</td> <td>Selection of port TAUD0I[m]</td></tr> <tr> <td></td> <td>x</td> <td>1</td> <td>Selection of port TAUD0I[m + 1]</td></tr> <tr> <td>TAUDTTIN[m+1]</td> <td>0</td> <td>x</td> <td>Selection of port TAUD0I[m + 1]</td></tr> <tr> <td></td> <td>1</td> <td>x</td> <td>Selection of port TAUD0I[m]</td></tr> </tbody> </table>	TAUD Input	Bit [m+1]	Bit [m]	Input signal	TAUDTTIN[m]	x	0	Selection of port TAUD0I[m]		x	1	Selection of port TAUD0I[m + 1]	TAUDTTIN[m+1]	0	x	Selection of port TAUD0I[m + 1]		1	x	Selection of port TAUD0I[m]
TAUD Input	Bit [m+1]	Bit [m]	Input signal																			
TAUDTTIN[m]	x	0	Selection of port TAUD0I[m]																			
	x	1	Selection of port TAUD0I[m + 1]																			
TAUDTTIN[m+1]	0	x	Selection of port TAUD0I[m + 1]																			
	1	x	Selection of port TAUD0I[m]																			
		(m = 0, 2, 4, 6, 8, 10, 12, 14)																				

#### CAUTION

Do not change the input signal of each channel during the timer counting.

## 23.2 Overview

### 23.2.1 Functional Overview

The TAUD has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The Timer Array Unit D is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUDnCNTm and a 16-bit data register TAUDnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

#### Independent and synchronous operation

Every channel can operate in different operation modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, this means that one channel can control when another starts to count, is reset, etc.

## 23.2.2 Terms

In this section, the following terms are used.

### **Independent / synchronous channel operation**

Independent or synchronous channel operation describes the dependency of channels on each other:

- If a channel operates independently of all other channels, this is called independent channel operation.
- If a channel operates depending on other channels, this is called synchronous channel operation.

### **Channel group**

In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

### **Operation mode**

An operation mode can be selected for every channel m. The operation mode defines the basic operation and features of a channel.

In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

Examples are “Capture Mode”, “Event Count Mode”, and “Interval Timer Mode”.

### **Channel output mode**

The channel output mode defines the operation of TAUDTOUTm

- of a single channel (independent output operation) or
- of all channels in a channel group (synchronous output operation).

Examples are “Independent Channel Output Mode 1” and “Synchronous Channel Output Mode 2 with Dead Time Output”.

### **Channel operation function**

The channel operation function defines the complete function and all features

- of a single channel (independent channel operation) or
- of all channels in a channel group (synchronous channel operation).

### **Upper / lower channel**

Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

Example:

For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

### 23.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

**Table 23.12 Functional List of TAUD Operations**

Operation Function	Example
<b>Independent Channel Operation Functions</b>	<b>Section 23.12</b>
Interval Timer Function	<b>Section 23.12.1</b>
TAUDTTINm Input Interval Timer Function	<b>Section 23.12.2</b>
Clock Divide Function	<b>Section 23.12.3</b>
External Event Count Function	<b>Section 23.12.4</b>
Delay Count Function	<b>Section 23.12.5</b>
One-Pulse Output Function	<b>Section 23.12.6</b>
TAUDTTINm Input Pulse Interval Measurement Function	<b>Section 23.12.7</b>
TAUDTTINm Input Signal Width Measurement Function	<b>Section 23.12.8</b>
TAUDTTINm Input Position Detection Function	<b>Section 23.12.9</b>
TAUDTTINm Input Period Count Detection Function	<b>Section 23.12.10</b>
TAUDTTINm Input Pulse Interval Judgment Function	<b>Section 23.12.11</b>
TAUDTTINm Input Signal Width Judgment Function	<b>Section 23.12.12</b>
Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)	<b>Section 23.12.13</b>
Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)	<b>Section 23.12.14</b>
One-Phase PWM Output Function	<b>Section 23.12.15</b>
<b>Independent Channel Real-Time Functions</b>	<b>Section 23.13</b>
Real-Time Output Function Type 1	<b>Section 23.13.1</b>
Real-Time Output Function Type 2	<b>Section 23.13.2</b>
<b>Independent Channel Simultaneous Rewrite Functions</b>	<b>Section 23.14</b>
Simultaneous Rewrite Trigger Generation Function Type 1	<b>Section 23.14.1</b>
Simultaneous Rewrite Trigger Generation Function Type 2	<b>Section 23.14.2</b>
<b>Synchronous Channel Operation Functions</b>	<b>Section 23.15</b>
PWM Output Function	<b>Section 23.15.1</b>
One-Shot Pulse Output Function	<b>Section 23.15.2</b>
Trigger Start PWM Output Function	<b>Section 23.15.3</b>
Delay Pulse Output Function	<b>Section 23.15.4</b>
Offset Trigger Output Function	<b>Section 23.15.5</b>
A/D Conversion Trigger Output Function Type 1	<b>Section 23.15.6</b>
Triangle PWM Output Function	<b>Section 23.15.7</b>
Triangle PWM Output Function with Dead Time	<b>Section 23.15.8</b>
A/D Conversion Trigger Output Function Type 2	<b>Section 23.15.9</b>
Interrupt Request Signals Culling Function	<b>Section 23.15.10</b>
<b>Synchronous Non-Complementary and Complementary Modulation Output Functions</b>	<b>Section 23.16</b>
Non-Complementary Modulation Output Function Type 1	<b>Section 23.16.1</b>
Non-Complementary Modulation Output Function Type 2	<b>Section 23.16.2</b>
Complementary Modulation Output Function	<b>Section 23.16.3</b>

### 23.2.4 TAUD I/O and Interrupt Request Signals

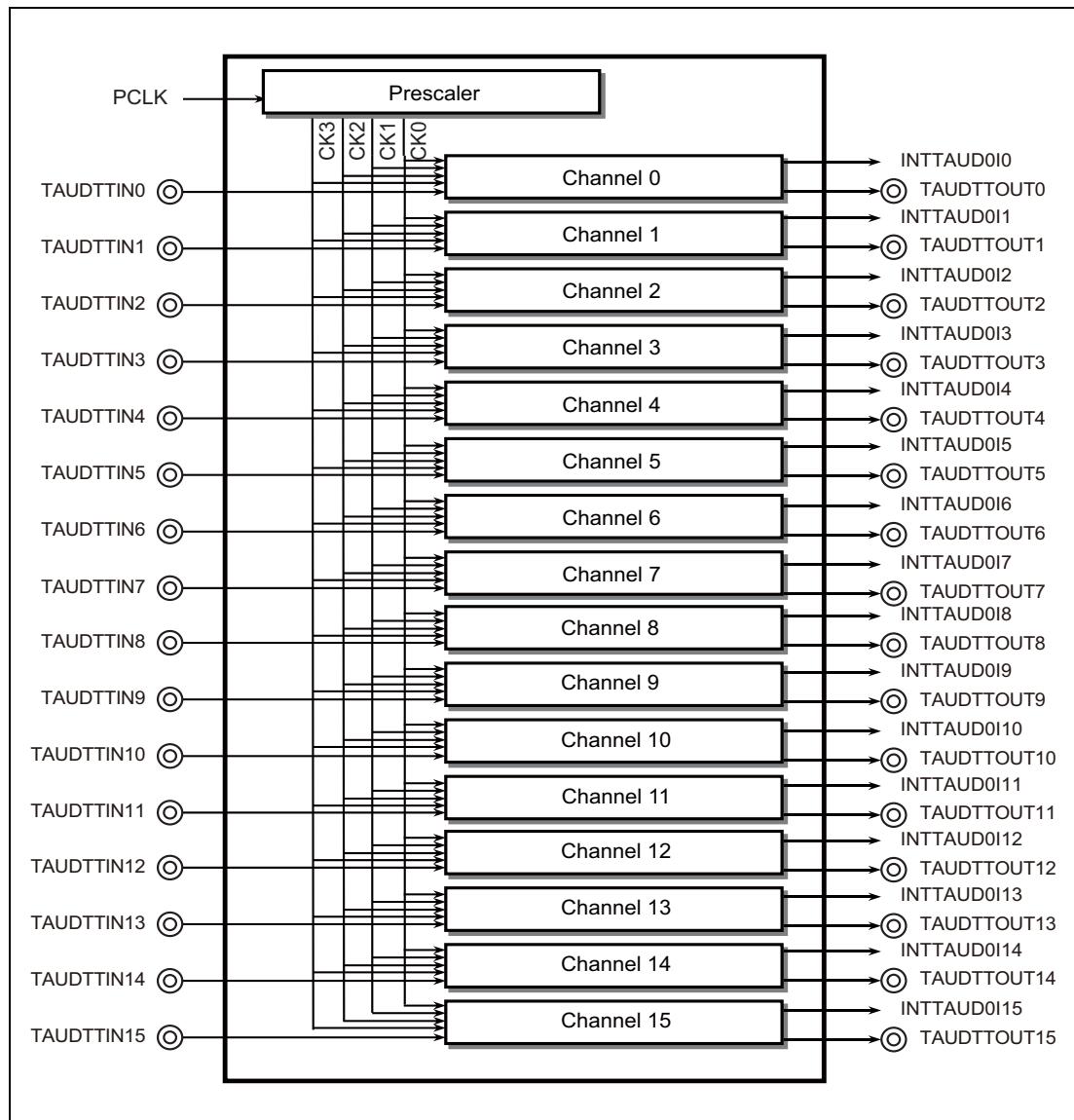
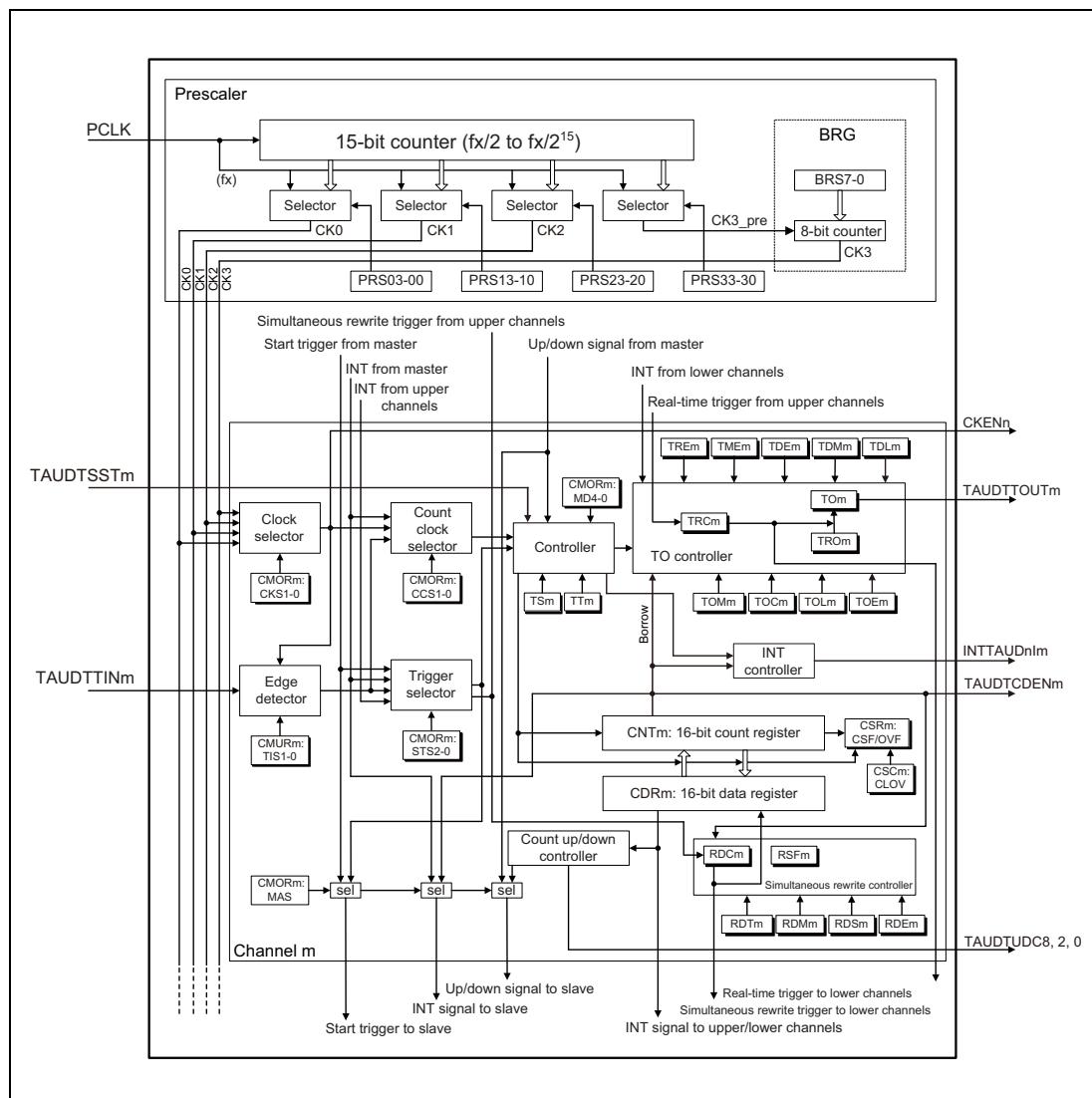


Figure 23.2 TAUD I/O and Interrupt Request Signals

### 23.2.5 Block Diagram

Figure 23.3 shows the main components of the TAUD.



**Figure 23.3 Block Diagram of the TAUD**

The module name “TAUD<sub>n</sub>” has been omitted from the register names for the sake of clarity in the above figure.

### 23.2.6 Description of Blocks

The following describes the functional blocks:

#### Prescaler block

The prescaler block provides up to four clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of  $2^0$  to  $2^{15}$ . The fourth count clock CK3 can be adjusted more precisely by using BRG to set an additional division factor that is not a power of 2.

#### Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUDnIm from master channel
- TAUDTTINm input signal valid edge

#### Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUDnCMORM.TAUDnMD[4:0])
- Counter start enable (TAUDnTS.TAUDnTSm) and counter stop (TAUDnTT.TAUDnTTm)  
When counter start is enabled, status flag TAUDnTE.TAUDnTEm is set.
- Count direction (up/down) (can be controlled by master channel)

#### Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUDnTE.TAUDnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUDnTSSTm
- TAUDTTINm input signal valid edge
- INTTAUDnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUDTTOUTm generation unit.

#### Simultaneous rewrite controller

Simultaneous rewrite control is a function that can be used in synchronous operating modes. The data registers (TAUDnCDRM) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

#### TAUDnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

## 23.3 Registers

### 23.3.1 List of Registers

TAUD registers are listed in the following table.

For details about <TAUDn\_base>, see **Section 23.1.2, Register Base Address**.

Table 23.13 List of Registers

Module	Register	Symbol	Address
<b>TAUDn prescaler registers</b>			
TAUDn	TAUDn prescaler clock select register	TAUDnTPS	<TAUDn_base> + 240 <sub>H</sub>
TAUDn	TAUDn prescaler baud rate setting register	TAUDnBRS	<TAUDn_base> + 244 <sub>H</sub>
<b>TAUDn control registers</b>			
TAUDn	TAUDn channel data register m	TAUDnCDRm	<TAUDn_base> + m × 4 <sub>H</sub>
TAUDn	TAUDn channel counter register m	TAUDnCNTm	<TAUDn_base> + 80 <sub>H</sub> + m × 4 <sub>H</sub>
TAUDn	TAUDn channel mode OS register m	TAUDnCMORm	<TAUDn_base> + 200 <sub>H</sub> + m × 4 <sub>H</sub>
TAUDn	TAUDn channel mode user register m	TAUDnCMURm	<TAUDn_base> + C0 <sub>H</sub> + m × 4 <sub>H</sub>
TAUDn	TAUDn channel status register m	TAUDnCSRm	<TAUDn_base> + 140 <sub>H</sub> + m × 4 <sub>H</sub>
TAUDn	TAUDn channel status clear trigger register m	TAUDnCSCm	<TAUDn_base> + 180 <sub>H</sub> + m × 4 <sub>H</sub>
TAUDn	TAUDn channel start trigger register	TAUDnTS	<TAUDn_base> + 1C4 <sub>H</sub>
TAUDn	TAUDn channel enable status register	TAUDnTE	<TAUDn_base> + 1C0 <sub>H</sub>
TAUDn	TAUDn channel stop trigger register	TAUDnTT	<TAUDn_base> + 1C8 <sub>H</sub>
<b>TAUDn output registers</b>			
TAUDn	TAUDn channel output enable register	TAUDnTOE	<TAUDn_base> + 5C <sub>H</sub>
TAUDn	TAUDn channel output register	TAUDnTO	<TAUDn_base> + 58 <sub>H</sub>
TAUDn	TAUDn channel output mode register	TAUDnTOM	<TAUDn_base> + 248 <sub>H</sub>
TAUDn	TAUDn channel output configuration register	TAUDnTOC	<TAUDn_base> + 24C <sub>H</sub>
TAUDn	TAUDn channel output active level register	TAUDnTOL	<TAUDn_base> + 040 <sub>H</sub>
TAUDn	TAUDn channel dead time output enable register	TAUDnTDE	<TAUDn_base> + 250 <sub>H</sub>
TAUDn	TAUDn channel dead time output mode register	TAUDnTDM	<TAUDn_base> + 254 <sub>H</sub>
TAUDn	TAUDn channel dead time output level register	TAUDnTDL	<TAUDn_base> + 54 <sub>H</sub>
TAUDn	TAUDn channel real-time output register	TAUDnTRO	<TAUDn_base> + 4C <sub>H</sub>
TAUDn	TAUDn channel real-time output enable register	TAUDnTRE	<TAUDn_base> + 258 <sub>H</sub>
TAUDn	TAUDn channel real-time output control register	TAUDnTRC	<TAUDn_base> + 25C <sub>H</sub>
TAUDn	TAUDn channel modulation output enable register	TAUDnTME	<TAUDn_base> + 50 <sub>H</sub>
<b>TAUDn reload data registers</b>			
TAUDn	TAUDn channel reload data enable register	TAUDnRDE	<TAUDn_base> + 260 <sub>H</sub>
TAUDn	TAUDn channel reload data mode register	TAUDnRDM	<TAUDn_base> + 264 <sub>H</sub>
TAUDn	TAUDn channel reload data control CH select register	TAUDnRDS	<TAUDn_base> + 268 <sub>H</sub>
TAUDn	TAUDn channel reload data control register	TAUDnRDC	<TAUDn_base> + 26C <sub>H</sub>
TAUDn	TAUDn channel reload data trigger register	TAUDnRDT	<TAUDn_base> + 44 <sub>H</sub>
TAUDn	TAUDn channel reload status register	TAUDnRSF	<TAUDn_base> + 48 <sub>H</sub>
<b>TAUDn Emulation Register</b>			
TAUDn	TAUDn emulation register	TAUDnEMU	<TAUDn_base> + 290 <sub>H</sub>

### 23.3.2 Details of TAUDn Prescaler Registers

#### 23.3.2.1 TAUDnTPS — TAUDn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3\_PRE for all channels of the PCLK prescaler. CK3 is generated by dividing CK3\_PRE by the factor specified in TAUDnBRS.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 240<sub>H</sub>

**Value after reset:** FFFF<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnPRS3 [3:0]				TAUDnPRS2 [3:0]				TAUDnPRS1 [3:0]				TAUDnPRS0 [3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R/W	R/W	R/W	R/W												

Table 23.14 TAUDnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUDnPRS3 [3:0]	Specifies CK3_PRE clock. CK3_PRE clock is an input clock to BRG unit which supplies the CK3 operation clock to all channels.
		<b>TAUDnPRS3[3:0] CK3_PRE Clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>

The above bits are rewritable only when all the counters using CK3 are stopped (TAUDnTE.TAUDnTEm = 0).

Table 23.14 TAUDnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUDnPRS2 [3:0]	Specifies the CK2 clock.
		<b>TAUDnPRS2[3:0]</b> <b>CK2 Clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUDnTE.TAUDnTEm = 0).		
7 to 4	TAUDnPRS1 [3:0]	Specifies the CK1 clock.
		<b>TAUDnPRS1[3:0]</b> <b>CK1 Clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUDnTE.TAUDnTEm = 0).		

Table 23.14 TAUDnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUDnPRS0 [3:0]	Specifies the CK0 clock.
		<b>TAUDnPRS0[3:0]</b>
		<b>CK0 Clock</b>
	0000 <sub>B</sub>	PCLK/2 <sup>0</sup>
	0001 <sub>B</sub>	PCLK/2 <sup>1</sup>
	0010 <sub>B</sub>	PCLK/2 <sup>2</sup>
	0011 <sub>B</sub>	PCLK/2 <sup>3</sup>
	0100 <sub>B</sub>	PCLK/2 <sup>4</sup>
	0101 <sub>B</sub>	PCLK/2 <sup>5</sup>
	0110 <sub>B</sub>	PCLK/2 <sup>6</sup>
	0111 <sub>B</sub>	PCLK/2 <sup>7</sup>
	1000 <sub>B</sub>	PCLK/2 <sup>8</sup>
	1001 <sub>B</sub>	PCLK/2 <sup>9</sup>
	1010 <sub>B</sub>	PCLK/2 <sup>10</sup>
	1011 <sub>B</sub>	PCLK/2 <sup>11</sup>
	1100 <sub>B</sub>	PCLK/2 <sup>12</sup>
	1101 <sub>B</sub>	PCLK/2 <sup>13</sup>
	1110 <sub>B</sub>	PCLK/2 <sup>14</sup>
	1111 <sub>B</sub>	PCLK/2 <sup>15</sup>

The above bits are rewritable only when all the counters using CK0 are stopped (TAUDnTE.TAUDnTEm = 0).

#### NOTE

The TAUDn clock input PCLK is specified in the first part of this section, **Section 23.1.3, Clock Supply**.

### 23.3.2.2 TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3\_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3\_PRE is specified in TAUDnTPS.TAUDnPRTS3[3:0].

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUDn\_base> + 244H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
TAUDnBRS[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 23.15 TAUDnBRS Register Contents**

Bit Position	Bit Name	Function
7 to 0	TAUDnBRS[7:0]	Specifies a CK3_PRE clock division factor for generating CK3.
		<b>TAUDnBRS[7:0] CK3 Clock</b>
		0000 0000 <sub>B</sub> CK3_PRE / 1
		0000 0001 <sub>B</sub> CK3_PRE / 2
		0000 0010 <sub>B</sub> CK3_PRE / 3
		0000 0011 <sub>B</sub> CK3_PRE / 4
		...
		1111 1110 <sub>B</sub> CK3_PRE / 255
		1111 1111 <sub>B</sub> CK3_PRE / 256

### 23.3.3 Details of TAUDn Control Registers

#### 23.3.3.1 TAUDnCDRm — TAUDn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUDnCMORm.TAUDnMD[4:1].

**Access:** This register can be read or written in 16-bit units.

- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

**Address:** <TAUDn\_base> + m × 4<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDnCDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.16 TAUDnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCDR [15:0]	Data register for capture/compare values

### 23.3.3.2 TAUDnCNTm — TAUDn Channel Counter Register

This is a channel m counter register.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAUDn\_base> + 80<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** FFFF<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUDnCNT[15:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 23.17 TAUDnCNTm Register Contents**

Bit Position	Bit Name	Function
15 to 0	TAUDnCNT [15:0]	16-bit counter value

A read value depends on a counter value, a changed operating mode, TAUDnTS.TAUDnTSm or TAUDnTT.TAUDnTTm bit value.

The initial read value of the counter depends on an operating mode and how the counter is stopped.

- Stop by a reset
- Stop by a counter stop trigger (TAUDnTT.TAUDnTTm = 1)

**Table 23.18** lists the initial counter read values after the counter is stopped (TAUDnTE.TAUDnTEM = 0) and re-enabled (TAUDnTS.TAUDnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUDnTS.TAUDnTSm = 1) with the counter waiting for a start trigger.

**Table 23.18 TAUDnCNTm Read Values after Re-Enabling Counter**

Mode Name	Count Method (Up/Down)	TAUDnCNTm Value		
		Start Value <sup>*1</sup>	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF <sub>H</sub>	Stop value	—
Judge mode	Count down	FFFF <sub>H</sub>	Stop value	—
Capture mode	Count up	0000 <sub>H</sub>	Stop value	—
Event count mode	Count down	FFFF <sub>H</sub>	Stop value	—
One-count mode	Count down	FFFF <sub>H</sub>	Stop value	Stop value
Capture and one-count mode	Count up	0000 <sub>H</sub>	Stop value	Capture value + 1 (TAUDnCDRm)
Judge and one-count mode	Count down	FFFF <sub>H</sub>	Stop value	TAUDnCNTm value – 1
Count-up/-down mode	Count down/up	FFFF <sub>H</sub>	Stop value	—
Pulse one-count mode	Count down	FFFF <sub>H</sub>	Stop value	0000 <sub>H</sub>
Count capture mode	Count up	0000 <sub>H</sub>	Stop value	—
Gate count mode	Count down	FFFF <sub>H</sub>	Stop value	Stop value
Capture and gate count mode	Count up	0000 <sub>H</sub>	Stop value	Stop value

Note 1. The value set for TAUDnCNTm when the operating mode is changed after a reset is deasserted.

### 23.3.3.3 TAUDnCMORm — TAUDn Channel Mode OS Register

This register controls channel m operation.

**Access:** This register can be read or written in 16-bit units. Writable only when the counter is stopped (TAUDnTE.TAUDnTEm = 0).

**Address:** <TAUDn\_base> + 200<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]	TAUDnCCS [1:0]	TAUDn MAS	TAUDnSTS[2:0]	TAUDnCOS [1:0]	—										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.19 TAUDnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function															
15, 14	TAUDnCKS[1:0]	Selects an operation clock. An operation clock is used for the TAUDTTINm input edge detection circuit. Setting of TAUDnCMORm.TAUDnCCS[1:0] bits also allows the operation clock to serve as the TAUDnCNTm counter clock.															
		<table border="1"> <thead> <tr> <th>TAUDn CKS1</th> <th>TAUDn CKS0</th> <th>Selection of Operation Clock</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>CK0</td></tr> <tr> <td>0</td><td>1</td><td>CK1</td></tr> <tr> <td>1</td><td>0</td><td>CK2</td></tr> <tr> <td>1</td><td>1</td><td>CK3</td></tr> </tbody> </table>	TAUDn CKS1	TAUDn CKS0	Selection of Operation Clock	0	0	CK0	0	1	CK1	1	0	CK2	1	1	CK3
TAUDn CKS1	TAUDn CKS0	Selection of Operation Clock															
0	0	CK0															
0	1	CK1															
1	0	CK2															
1	1	CK3															
13, 12	TAUDnCCS[1:0]	Selects a count clock for TAUDnCNTm counter.															
		<table border="1"> <thead> <tr> <th>TAUDn CCS1</th> <th>TAUDn CCS0</th> <th>Selection of Count Clock</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Operation clock specified by TAUDnCMORm.TAUDnCKS[1:0]</td></tr> <tr> <td>0</td><td>1</td><td>Valid edge of TAUDTTINm input signal</td></tr> <tr> <td>1</td><td>0</td><td>Setting prohibited</td></tr> <tr> <td>1</td><td>1</td><td>INTTAUDnlm signal of master channel</td></tr> </tbody> </table>	TAUDn CCS1	TAUDn CCS0	Selection of Count Clock	0	0	Operation clock specified by TAUDnCMORm.TAUDnCKS[1:0]	0	1	Valid edge of TAUDTTINm input signal	1	0	Setting prohibited	1	1	INTTAUDnlm signal of master channel
TAUDn CCS1	TAUDn CCS0	Selection of Count Clock															
0	0	Operation clock specified by TAUDnCMORm.TAUDnCKS[1:0]															
0	1	Valid edge of TAUDTTINm input signal															
1	0	Setting prohibited															
1	1	INTTAUDnlm signal of master channel															
11	TAUDnMAS	Specifies whether the channel is a master channel or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm_odd) are fixed to 0.															

Table 23.19 TAUDnCMORm Register Contents (2/3)

Bit Position	Bit Name	Function																																				
10 to 8	TAUDnSTS[2:0]	Selects an external start trigger.																																				
		<table border="1"> <thead> <tr> <th>TAUDn STS2</th> <th>TAUDn STS1</th> <th>TAUDn STS0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>Valid edge of TAUDTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>Valid edge of TAUDTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>Triggers simultaneous rewrite.</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>INTTAUDnIm is the start trigger of master channel</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>INTTAUDnIm of upper channel (<math>m - 1</math>) is the start trigger regardless of master setting</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>Dead time output signal of TAUDTTOUTm generating unit</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of master channel</td></tr> </tbody> </table>	TAUDn STS2	TAUDn STS1	TAUDn STS0	Functional Description	0	0	0	Software trigger	0	0	1	Valid edge of TAUDTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].	0	1	0	Valid edge of TAUDTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.	0	1	1	Triggers simultaneous rewrite.	1	0	0	INTTAUDnIm is the start trigger of master channel	1	0	1	INTTAUDnIm of upper channel ( $m - 1$ ) is the start trigger regardless of master setting	1	1	0	Dead time output signal of TAUDTTOUTm generating unit	1	1	1	Up/down output trigger signal of master channel
TAUDn STS2	TAUDn STS1	TAUDn STS0	Functional Description																																			
0	0	0	Software trigger																																			
0	0	1	Valid edge of TAUDTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].																																			
0	1	0	Valid edge of TAUDTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.																																			
0	1	1	Triggers simultaneous rewrite.																																			
1	0	0	INTTAUDnIm is the start trigger of master channel																																			
1	0	1	INTTAUDnIm of upper channel ( $m - 1$ ) is the start trigger regardless of master setting																																			
1	1	0	Dead time output signal of TAUDTTOUTm generating unit																																			
1	1	1	Up/down output trigger signal of master channel																																			
7, 6	TAUDnCOS[1:0]	<p>Specifies the timing for updating capture register TAUDnCDRm and overflow flag TAUDnCSRm.TAUDnOVF of channel m. These bits are valid only when channel m is in capture mode or capture one-count mode.</p> <table border="1"> <thead> <tr> <th>TAUDn COS1</th> <th>TAUDn COS0</th> <th>TAUDnCDRm</th> <th>TAUDnCSRm.TAUDnOVF</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Updated upon detection of a valid edge of TAUDTTINm input.</td><td>           Updated (cleared or set) upon detection of a TAUDTTINm input valid edge:           <ul style="list-style-type: none"> <li>If a counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is set.</li> <li>If no counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is cleared.</li> </ul> </td></tr> <tr> <td>0</td><td>1</td><td></td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr> <tr> <td>1</td><td>0</td><td>Updated upon detection of valid edge of TAUDTTINm input and at the occurrence of counter overflow:           <ul style="list-style-type: none"> <li>Detection of valid edge of TAUDTTINm input: Counter value is written into TAUDnCDRm.</li> <li>Occurrence of overflow: <math>FFFF_H</math> is loaded into TAUDnCDRm. The next detection of valid edge of TAUDTTINm input is ignored.</li> </ul> </td><td>Not set</td></tr> <tr> <td>1</td><td>1</td><td></td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr> </tbody> </table>	TAUDn COS1	TAUDn COS0	TAUDnCDRm	TAUDnCSRm.TAUDnOVF	0	0	Updated upon detection of a valid edge of TAUDTTINm input.	Updated (cleared or set) upon detection of a TAUDTTINm input valid edge: <ul style="list-style-type: none"> <li>If a counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is set.</li> <li>If no counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is cleared.</li> </ul>	0	1		Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.	1	0	Updated upon detection of valid edge of TAUDTTINm input and at the occurrence of counter overflow: <ul style="list-style-type: none"> <li>Detection of valid edge of TAUDTTINm input: Counter value is written into TAUDnCDRm.</li> <li>Occurrence of overflow: <math>FFFF_H</math> is loaded into TAUDnCDRm. The next detection of valid edge of TAUDTTINm input is ignored.</li> </ul>	Not set	1	1		Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.																
TAUDn COS1	TAUDn COS0	TAUDnCDRm	TAUDnCSRm.TAUDnOVF																																			
0	0	Updated upon detection of a valid edge of TAUDTTINm input.	Updated (cleared or set) upon detection of a TAUDTTINm input valid edge: <ul style="list-style-type: none"> <li>If a counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is set.</li> <li>If no counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is cleared.</li> </ul>																																			
0	1		Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.																																			
1	0	Updated upon detection of valid edge of TAUDTTINm input and at the occurrence of counter overflow: <ul style="list-style-type: none"> <li>Detection of valid edge of TAUDTTINm input: Counter value is written into TAUDnCDRm.</li> <li>Occurrence of overflow: <math>FFFF_H</math> is loaded into TAUDnCDRm. The next detection of valid edge of TAUDTTINm input is ignored.</li> </ul>	Not set																																			
1	1		Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.																																			
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																				

Table 23.19 TAUDnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function																																																																																										
4 to 0	TAUDnMD[4:0]	Specifies an operating mode.																																																																																										
		<table border="1"> <thead> <tr> <th>TAUDn MD4</th><th>TAUDn MD3</th><th>TAUDn MD2</th><th>TAUDn MD1</th><th>TAUDn MD0</th><th>Functional Description</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Judge mode</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Event count mode</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Judge and one-count mode</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Count-up/-down mode</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Pulse one-count mode</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr> </tbody> </table>	TAUDn MD4	TAUDn MD3	TAUDn MD2	TAUDn MD1	TAUDn MD0	Functional Description	0	0	0	0	1/0	Interval timer mode	0	0	0	1	1/0	Judge mode	0	0	1	0	1/0	Capture mode	0	0	1	1	0	Event count mode	0	1	0	0	1/0	One-count mode	0	1	0	1	1/0	Setting prohibited	0	1	1	0	0	Capture and one-count mode	0	1	1	1	1/0	Judge and one-count mode	1	0	0	0	0	Setting prohibited	1	0	0	1	0	Count-up/-down mode	1	0	1	0	1/0	Pulse one-count mode	1	0	1	1	1/0	Count capture mode	1	1	0	0	0	Gate count mode	1	1	0	1	0	Capture and gate count mode
TAUDn MD4	TAUDn MD3	TAUDn MD2	TAUDn MD1	TAUDn MD0	Functional Description																																																																																							
0	0	0	0	1/0	Interval timer mode																																																																																							
0	0	0	1	1/0	Judge mode																																																																																							
0	0	1	0	1/0	Capture mode																																																																																							
0	0	1	1	0	Event count mode																																																																																							
0	1	0	0	1/0	One-count mode																																																																																							
0	1	0	1	1/0	Setting prohibited																																																																																							
0	1	1	0	0	Capture and one-count mode																																																																																							
0	1	1	1	1/0	Judge and one-count mode																																																																																							
1	0	0	0	0	Setting prohibited																																																																																							
1	0	0	1	0	Count-up/-down mode																																																																																							
1	0	1	0	1/0	Pulse one-count mode																																																																																							
1	0	1	1	1/0	Count capture mode																																																																																							
1	1	0	0	0	Gate count mode																																																																																							
1	1	0	1	0	Capture and gate count mode																																																																																							
		<table border="1"> <thead> <tr> <th>Mode</th><th>Role of TAUDnMD0 Bit</th></tr> </thead> <tbody> <tr> <td>Interval timer mode Capture mode Count capture mode</td><td>Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated.</td></tr> <tr> <td>Event count mode Count-up/-down mode</td><td>This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).</td></tr> <tr> <td>One-count mode Pulse one-count mode</td><td>Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection.</td></tr> <tr> <td colspan="2"><b>CAUTION</b></td></tr> <tr> <td colspan="2"> <ul style="list-style-type: none"> <li>INTTAUDnIm signal is not output at the beginning of count operation in one-count mode.</li> <li>INTTAUDnIm signal is output at the beginning of count operation in pulse one-count mode.</li> </ul> </td></tr> <tr> <td>Gate count mode</td><td>This bit should be set to 0 (disables start trigger detection during counting).</td></tr> <tr> <td>Capture and one-count mode Capture and gate count mode</td><td>This bit should be set to 0. <b>CAUTION</b> INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.</td></tr> <tr> <td>Judge mode Judge and one-count mode</td><td>Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm &gt; TAUDnCDRm</td></tr> </tbody> </table>	Mode	Role of TAUDnMD0 Bit	Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated.	Event count mode Count-up/-down mode	This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).	One-count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection.	<b>CAUTION</b>		<ul style="list-style-type: none"> <li>INTTAUDnIm signal is not output at the beginning of count operation in one-count mode.</li> <li>INTTAUDnIm signal is output at the beginning of count operation in pulse one-count mode.</li> </ul>		Gate count mode	This bit should be set to 0 (disables start trigger detection during counting).	Capture and one-count mode Capture and gate count mode	This bit should be set to 0. <b>CAUTION</b> INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.	Judge mode Judge and one-count mode	Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm																																																																								
Mode	Role of TAUDnMD0 Bit																																																																																											
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated.																																																																																											
Event count mode Count-up/-down mode	This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).																																																																																											
One-count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection.																																																																																											
<b>CAUTION</b>																																																																																												
<ul style="list-style-type: none"> <li>INTTAUDnIm signal is not output at the beginning of count operation in one-count mode.</li> <li>INTTAUDnIm signal is output at the beginning of count operation in pulse one-count mode.</li> </ul>																																																																																												
Gate count mode	This bit should be set to 0 (disables start trigger detection during counting).																																																																																											
Capture and one-count mode Capture and gate count mode	This bit should be set to 0. <b>CAUTION</b> INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.																																																																																											
Judge mode Judge and one-count mode	Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm																																																																																											

### 23.3.3.4 TAUDnCMURm — TAUDn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUDTTINm input.

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUDn\_base> + C0H + m × 4H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 23.20 TAUDnCMURm Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	Specifies a valid edge of TAUDTTINm input signal.
TAUDn	TAUDn	Functional Description
TIS1	TIS0	
0	0	Falling edge
0	1	Rising edge
1	0	Detection of rising and falling edges (selects low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detection of rising and falling edges (selects high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge
<ul style="list-style-type: none"> <li>Edge detection of TAUDTTINm input signal is based on the operation clock selected by TAUDnCMORm.TAUDnCKS[1:0].</li> </ul>		

### 23.3.3.5 TAUDnCSRm — TAUDn channel status register

This register indicates the count direction and overflow status of channel m counter.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <TAUDn\_base> + 140<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnCSF	TAUDnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 23.21 TAUDnCSRm Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUDnCSF	<p>Indicates a count direction. 0: Count-up 1: Count-down</p> <p>The read value of this bit is valid only in the following mode: • Count-up/-down mode</p>
0	TAUDnOVF	<p>Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs.</p> <p>This bit is used only in the following modes: • Capture mode • Capture and one-count mode</p> <p>The function of this bit depends on the setting of control bit TAUDnCMORm.TAUDnCOS[1:0].</p>

### 23.3.3.6 TAUDnCSCm — TAUDn Channel Status Clear Register

This is a trigger register for clearing the overflow flag TAUDnCSRm.TAUDnOVF of channel m.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <TAUDn\_base> + 180<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 23.22 TAUDnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUDnCLOV	0: No function 1: Clears overflow flag TAUDnCSRm.TAUDnOVF.

### 23.3.3.7 TAUDnTS — TAUDn Channel Start Trigger Register

This register enables the counter operation of each channel.

**Access:** This register is a write-only register that can be written in 16-bit units.

**Address:** <TAUDn\_base> + 1C4<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TS15	TAUDn TS14	TAUDn TS13	TAUDn TS12	TAUDn TS11	TAUDn TS10	TAUDn TS09	TAUDn TS08	TAUDn TS07	TAUDn TS06	TAUDn TS05	TAUDn TS04	TAUDn TS03	TAUDn TS02	TAUDn TS01	TAUDn TS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 23.23 TAUDnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTSm	Enables the counter operation of channel m. 0: No function 1: Enables the counter operation and sets TAUDnTE.TAUDnTEM to 1. The counter operation is only enabled when TAUDnTE.TAUDnTEM is set to 1. Whether counting is started or not depends on a selected operating mode.

### 23.3.3.8 TAUDnTE — TAUDn Channel Enable Status Register

This register enables/disables a counter operation.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAUDn\_base> + 1C0H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TE15	TAUDn TE14	TAUDn TE13	TAUDn TE12	TAUDn TE11	TAUDn TE10	TAUDn TE09	TAUDn TE08	TAUDn TE07	TAUDn TE06	TAUDn TE05	TAUDn TE04	TAUDn TE03	TAUDn TE02	TAUDn TE01	TAUDn TE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.24 TAUDnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTEm	<p>Enables/disables the counter operation of channel m.            0: Disables counter operation.            1: Enables counter operation.</p> <p>This bit is set to 1 when trigger input of TAUDnTSSTm (synchronous channel start trigger signal) is detected or when TAUDnTS.TAUDnTSm is set to 1. This bit is set to 0 when TAUDnTT.TAUDnTTm is set to 1.</p>

### 23.3.3.9 TAUDnTT — TAUDn Channel Stop Trigger Register

This register stops the counter operation of each channel.

**Access:** This register is a write-only register that can be written in 16-bit units.

**Address:** <TAUDn\_base> + 1C8H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TT15	TAUDn TT14	TAUDn TT13	TAUDn TT12	TAUDn TT11	TAUDn TT10	TAUDn TT09	TAUDn TT08	TAUDn TT07	TAUDn TT06	TAUDn TT05	TAUDn TT04	TAUDn TT03	TAUDn TT02	TAUDn TT01	TAUDn TT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 23.25 TAUDnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTTm	<p>Stops the counter operation of channel m.            0: No function            1: Stops the counter operation and resets TAUDnTE.TAUDnTEm. TAUDnCNTm, TAUDnTO.TAUDnTOM, and TAUDTTOUTm retain the values provided before the counter is stopped.</p>

### 23.3.4 Details of TAUDn Simultaneous Rewrite Registers

#### 23.3.4.1 TAUDnRDE — TAUDn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUDnCDRm/TAUDnTOLm data register.

**Access:** This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

**Address:** <TAUDn\_base> + 260H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDE15	TAUDn RDE14	TAUDn RDE13	TAUDn RDE12	TAUDn RDE11	TAUDn RDE10	TAUDn RDE09	TAUDn RDE08	TAUDn RDE07	TAUDn RDE06	TAUDn RDE05	TAUDn RDE04	TAUDn RDE03	TAUDn RDE02	TAUDn RDE01	TAUDn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.26 TAUDnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDEm	Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite 1: Enables simultaneous rewrite

#### 23.3.4.2 TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

**Access:** This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

**Address:** <TAUDn\_base> + 268H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDS15	TAUDn RDS14	TAUDn RDS13	TAUDn RDS12	TAUDn RDS11	TAUDn RDS10	TAUDn RDS09	TAUDn RDS08	TAUDn RDS07	TAUDn RDS06	TAUDn RDS05	TAUDn RDS04	TAUDn RDS03	TAUDn RDS02	TAUDn RDS01	TAUDn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.27 TAUDnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDSm	Selects a channel that controls a simultaneous rewrite trigger. 0: Master channel 1: Another upper channel

### 23.3.4.3 TAUDnRDM — TAUDn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

**Access:** This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

**Address:** <TAUDn\_base> + 264H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDM15	TAUDn RDM14	TAUDn RDM13	TAUDn RDM12	TAUDn RDM11	TAUDn RDM10	TAUDn RDM09	TAUDn RDM08	TAUDn RDM07	TAUDn RDM06	TAUDn RDM05	TAUDn RDM04	TAUDn RDM03	TAUDn RDM02	TAUDn RDM01	TAUDn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.28 TAUDnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDMm	Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts to count 1: At the peak of cycle of triangular wave These bit settings are applied only when TAUDnRDE.TAUDnRDEM = 1 and TAUDnRDS.TAUDnRDSm = 0.

### 23.3.4.4 TAUDnRDC — TAUDn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUDnIm signal to trigger simultaneous rewrite.

**Access:** This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

**Address:** <TAUDn\_base> + 26CH

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDC15	TAUDn RDC14	TAUDn RDC13	TAUDn RDC12	TAUDn RDC11	TAUDn RDC10	TAUDn RDC09	TAUDn RDC08	TAUDn RDC07	TAUDn RDC06	TAUDn RDC05	TAUDn RDC04	TAUDn RDC03	TAUDn RDC02	TAUDn RDC01	TAUDn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.29 TAUDnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bit settings are applied only when TAUDnRDE.TAUDnRDEM = 1 and TAUDnRDS.TAUDnRDSm = 1.

### 23.3.4.5 TAUDnRDT — TAUDn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite enabling state.

**Access:** This register is a write-only register that can be written in 16-bit units.

**Address:** <TAUDn\_base> + 044H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDT15	TAUDn RDT14	TAUDn RDT13	TAUDn RDT12	TAUDn RDT11	TAUDn RDT10	TAUDn RDT09	TAUDn RDT08	TAUDn RDT07	TAUDn RDT06	TAUDn RDT05	TAUDn RDT04	TAUDn RDT03	TAUDn RDT02	TAUDn RDT01	TAUDn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 23.30 TAUDnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDTm	<p>Triggers a simultaneous rewrite enabling state.</p> <p>0: No function</p> <p>1: The simultaneous rewrite enabling flag (TAUDnRSFm) is set to 1. The system waits for a simultaneous rewrite trigger.</p> <p>These bits only apply when:</p> <ul style="list-style-type: none"> <li>• TAUDnRDE.TAUDnRDEm = 1</li> </ul>

### 23.3.4.6 TAUDnRSF — TAUDn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAUDn\_base> + 048H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RSF15	TAUDn RSF14	TAUDn RSF13	TAUDn RSF12	TAUDn RSF11	TAUDn RSF10	TAUDn RSF09	TAUDn RSF08	TAUDn RSF07	TAUDn RSF06	TAUDn RSF05	TAUDn RSF04	TAUDn RSF03	TAUDn RSF02	TAUDn RSF01	TAUDn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.31 TAUDnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRSFm	<p>Indicates simultaneous rewrite status.</p> <p>0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger.</p> <p>1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUDnRDTm = 1).</p>

### 23.3.5 Details of TAUDn Output Registers

#### 23.3.5.1 TAUDnTOE — TAUDn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 5C<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOE15	TAUDn TOE14	TAUDn TOE13	TAUDn TOE12	TAUDn TOE11	TAUDn TOE10	TAUDn TOE09	TAUDn TOE08	TAUDn TOE07	TAUDn TOE06	TAUDn TOE05	TAUDn TOE04	TAUDn TOE03	TAUDn TOE02	TAUDn TOE01	TAUDn TOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.32 TAUDnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOEm	Enables/disables the independent channel output function. 0: Disables the independent timer output function (controlled by software). 1: Enables the independent timer output function.

#### 23.3.5.2 TAUDnTO — TAUDn Channel Output Register

This register specifies and reads a TAUDTTOUTm level.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 58<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TO15	TAUDn TO14	TAUDn TO13	TAUDn TO12	TAUDn TO11	TAUDn TO10	TAUDn TO09	TAUDn TO08	TAUDn TO07	TAUDn TO06	TAUDn TO05	TAUDn TO04	TAUDn TO03	TAUDn TO02	TAUDn TO01	TAUDn TO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.33 TAUDnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOM	Specifies and reads a TAUDTTOUTm level. 0: Low level 1: High level Only TAUDnTOM bits for which Independent Channel Output function is disabled (TAUDnTOEm = 0) can be written.

### 23.3.5.3 TAUDnTOM — TAUDn Channel Output Mode Register

This register specifies the output mode of each channel.

**Access:** This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEM = 0).

**Address:** <TAUDn\_base> + 248H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOM15	TAUDn TOM14	TAUDn TOM13	TAUDn TOM12	TAUDn TOM11	TAUDn TOM10	TAUDn TOM09	TAUDn TOM08	TAUDn TOM07	TAUDn TOM06	TAUDn TOM05	TAUDn TOM04	TAUDn TOM03	TAUDn TOM02	TAUDn TOM01	TAUDn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 23.34 TAUDnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation

### 23.3.5.4 TAUDnTOC — TAUDn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUDnTOMm.

**Access:** This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEM = 0).

**Address:** <TAUDn\_base> + 24CH

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOC15	TAUDn TOC14	TAUDn TOC13	TAUDn TOC12	TAUDn TOC11	TAUDn TOC10	TAUDn TOC09	TAUDn TOC08	TAUDn TOC07	TAUDn TOC06	TAUDn TOC05	TAUDn TOC04	TAUDn TOC03	TAUDn TOC02	TAUDn TOC01	TAUDn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 23.35 TAUDnTOC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOCm	Specifies an output mode. 0: Operating mode 1 1: Operating mode 2 As listed below, the output mode depends on the setting of TAUDnTOM.TAUDnTOMm.
TAUDn TAU Dn TOMm TAUDn TOCm Functional Description		
0	0	Toggle mode: Toggle operation is conducted when INTTAUDnIm occurs.
0	1	Set/reset mode: Set when INTTAUDnIm occurs at the beginning of count operation, and reset when INTTAUDnIm is caused by detection of a match between TAUDnCNTm and TAUDnCDRm.
1	0	Synchronous channel operating mode 1: Set when INT occurs on master channels, and reset when INT occurs on slave channels.
1	1	Synchronous channel operating mode 2: Set when INTTAUDnIm occurs in count-down status, and reset when INTTAUDnIm occurs in count-up status.

### 23.3.5.5 TAUDnTOL — TAUDn Channel Output Level Register

This register specifies the output logic of channel output bit (TAUDnTO.TAUDnTOM).

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 040H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOL15	TAUDn TOL14	TAUDn TOL13	TAUDn TOL12	TAUDn TOL11	TAUDn TOL10	TAUDn TOL09	TAUDn TOL08	TAUDn TOL07	TAUDn TOL06	TAUDn TOL05	TAUDn TOL04	TAUDn TOL03	TAUDn TOL02	TAUDn TOL01	TAUDn TOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 23.36 TAUDnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOLm	<p>Specifies the output logic of channel m output bit (TAUDnTO.TAUDnTOM).</p> <p>0: Positive logic (active high) 1: Negative logic (active low)</p> <p>The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software.</p>

### 23.3.6 Details of TAUDn Dead Time Output Registers

#### 23.3.6.1 TAUDnTDE — TAUDn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

**Access:** This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

**Address:** <TAUDn\_base> + 250<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDE15	TAUDn TDE14	TAUDn TDE13	TAUDn TDE12	TAUDn TDE11	TAUDn TDE10	TAUDn TDE09	TAUDn TDE08	TAUDn TDE07	TAUDn TDE06	TAUDn TDE05	TAUDn TDE04	TAUDn TDE03	TAUDn TDE02	TAUDn TDE01	TAUDn TDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.37 TAUDnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDEM	<p>Enables/disables the dead time control operation of channel m.</p> <p>0: Disables dead time operation</p> <p>1: Enables dead time operation.</p> <p>The same setting should be made for both even and odd slave channels in pairs.</p> <p>These bit settings are applied when:</p> <ul style="list-style-type: none"> <li>• TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm = 1</li> </ul>

#### 23.3.6.2 TAUDnTDM — TAUDn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

**Access:** This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

**Address:** <TAUDn\_base> + 254<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDM15	TAUDn TDM14	TAUDn TDM13	TAUDn TDM12	TAUDn TDM11	TAUDn TDM10	TAUDn TDM09	TAUDn TDM08	TAUDn TDM07	TAUDn TDM06	TAUDn TDM05	TAUDn TDM04	TAUDn TDM03	TAUDn TDM02	TAUDn TDM01	TAUDn TDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.38 TAUDnTDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDMm	<p>Specifies the timing to add dead time during dead time output.</p> <p>0: When detecting the duty cycle of an upper even channel (duty dead time output).</p> <p>1: When detecting the TIN input edge of a lower odd channel (one-phase dead time output).</p> <p>The same setting should be made for both even and odd slave channels in pairs.</p> <p>These bit settings are applied when:</p> <ul style="list-style-type: none"> <li>• TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEM = 1</li> </ul>

### 23.3.6.3 TAUDnTDL — TAUDn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 54H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDL15	TAUDn TDL14	TAUDn TDL13	TAUDn TDL12	TAUDn TDL11	TAUDn TDL10	TAUDn TDL09	TAUDn TDL08	TAUDn TDL07	TAUDn TDL06	TAUDn TDL05	TAUDn TDL04	TAUDn TDL03	TAUDn TDL02	TAUDn TDL01	TAUDn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 23.39 TAUDnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDLm	<p>Selects a phase in which dead time is added.            0: Normal phase            1: Reverse phase</p> <p>These bit settings are applied when:</p> <ul style="list-style-type: none"> <li>• TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm,            TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEM = 1</li> </ul>

### 23.3.7 Details of TAUDn Real-time/Modulation Output Registers

#### 23.3.7.1 TAUDnTRE — TAUDn Channel Real-time Output Enable Register

This register enables/disables real-time output.

**Access:** This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

**Address:** <TAUDn\_base> + 258H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRE15	TAUDn TRE14	TAUDn TRE13	TAUDn TRE12	TAUDn TRE11	TAUDn TRE10	TAUDn TRE09	TAUDn TRE08	TAUDn TRE07	TAUDn TRE06	TAUDn TRE05	TAUDn TRE04	TAUDn TRE03	TAUDn TRE02	TAUDn TRE01	TAUDn TRE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.40 TAUDnTRE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTREm	<p>Enables or disables real-time output of channel m.</p> <p>0: Disables real-time output</p> <p>1: Enables real-time output.</p> <p>These bit settings are applied only when TAUDnTOE.TAUDnTOEm = 1. When TAUDnTRE.TAUDnTREm = 0, TAUDTTOUTm is not affected by real-time output.</p> <p>When TAUDnTRE.TAUDnTREm = 1, TAUDTTOUTm outputs the value of real-time output bit TAUDnTRO.TAUDnTROm in response to a timer operation.</p>

#### 23.3.7.2 TAUDnTRC — TAUDn Channel Real-time Output Control Register

This register controls the real-time output trigger of each channel.

**Access:** This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

**Address:** <TAUDn\_base> + 25CH

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRC15	TAUDn TRC14	TAUDn TRC13	TAUDn TRC12	TAUDn TRC11	TAUDn TRC10	TAUDn TRC09	TAUDn TRC08	TAUDn TRC07	TAUDn TRC06	TAUDn TRC05	TAUDn TRC04	TAUDn TRC03	TAUDn TRC02	TAUDn TRC01	TAUDn TRC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.41 TAUDnTRC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTRCm	<p>Specifies a channel on which the real-time output trigger for channel m is generated.</p> <p>0: Next upper channel with this bit set to 1</p> <p>1: Channel m</p> <p>These bit settings are applied only when TAUDnTRE.TAUDnTREm = 1.</p>

### 23.3.7.3 TAUDnTRO — TAUDn Channel Real-time Output Register

This register sets a value which is output to TAUDTTOUTm.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 04C<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRO15	TAUDn TRO14	TAUDn TRO13	TAUDn TRO12	TAUDn TRO11	TAUDn TRO10	TAUDn TRO09	TAUDn TRO08	TAUDn TRO07	TAUDn TRO06	TAUDn TRO05	TAUDn TRO04	TAUDn TRO03	TAUDn TRO02	TAUDn TRO01	TAUDn TRO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.42 TAUDnTRO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTROm	Sets a value which is output to TAUDTTOUTm. 0: Low 1: High TAUDnTROm value is not output to TAUDTTOUTm when TAUDnTRE.TAUDnTREm = 0, even if a real-time output trigger occurs.

### 23.3.7.4 TAUDnTME — TAUDn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUDn\_base> + 050<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TME15	TAUDn TME14	TAUDn TME13	TAUDn TME12	TAUDn TME11	TAUDn TME10	TAUDn TME09	TAUDn TME08	TAUDn TME07	TAUDn TME06	TAUDn TME05	TAUDn TME04	TAUDn TME03	TAUDn TME02	TAUDn TME01	TAUDn TME00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.43 TAUDnTME Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTMEm	Enables/disables modulation output for timer output and real-time output of channel m. 0: Disables modulation 1: Enables modulation These bit settings are applied only when TAUDnTOE.TAUDnTOEm and TAUDnTRE.TAUDnTREM = 1.

### 23.3.8 TAUDn Emulation Register

#### 23.3.8.1 TAUDnEMU — TAUDn Emulation Register

This register controls SVSTOP operations.

**Access:** This register can be read or written in 8-bit units.

Perform write operations when the counter is stopped (TAUDnTE.TAUDnTEm = 0) and (EPC.SVSTOP = 0).

**Address:** <TAUDn\_base> + 290<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
TAUDnSVSDIS	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

**Table 23.44 TAUDnEMU Register Contents**

Bit Position	Bit Name	Function
7	TAUDnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0).  When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 23.4 Operating Procedure

The following lists the general operation procedure for the TAUDn.

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUDTOUTm is also initialized and outputs a low level.

- (1) Set the TAUDnTPS and TAUDnBRS registers to specify the clock frequency of CK0 to CK3.
- (2) Configure the desired TAUDn function:
  - Set the operation mode
  - Set the channel output mode
  - Set any other control bits
- (3) Enable the counter by setting the TAUDnTS.TAUDnTSm bit to 1.  
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
- (4) If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUDnTT.TAUDnTTm bit to 1. The counter can be forcibly restarted by setting the TAUDnTS.TAUDnTSm bit to 1.
- (5) Stop the function by setting the TAUDnTT.TAUDnTTm bit to 1.

### NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 23.12, Independent Channel Operation Functions** and **Section 23.15, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUDnTE.TAUDnTEM = 0).

## 23.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels). Several rules apply to the settings of channels. These rules are detailed in **Section 23.5.1, Rules of Synchronous Channel Operation**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 23.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 23.6, Simultaneous Rewrite**

### 23.5.1 Rules of Synchronous Channel Operation

#### Number of master and slave channels

- Only even channels (CH0, CH2, CH4, ...) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.  
Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, ...) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.  
Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

#### Operation clock

- The same operation clock must be set for the master channel and the synchronized slave channel. This is achieved by setting the TAUDnCMORm.TAUDnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 23.4**.

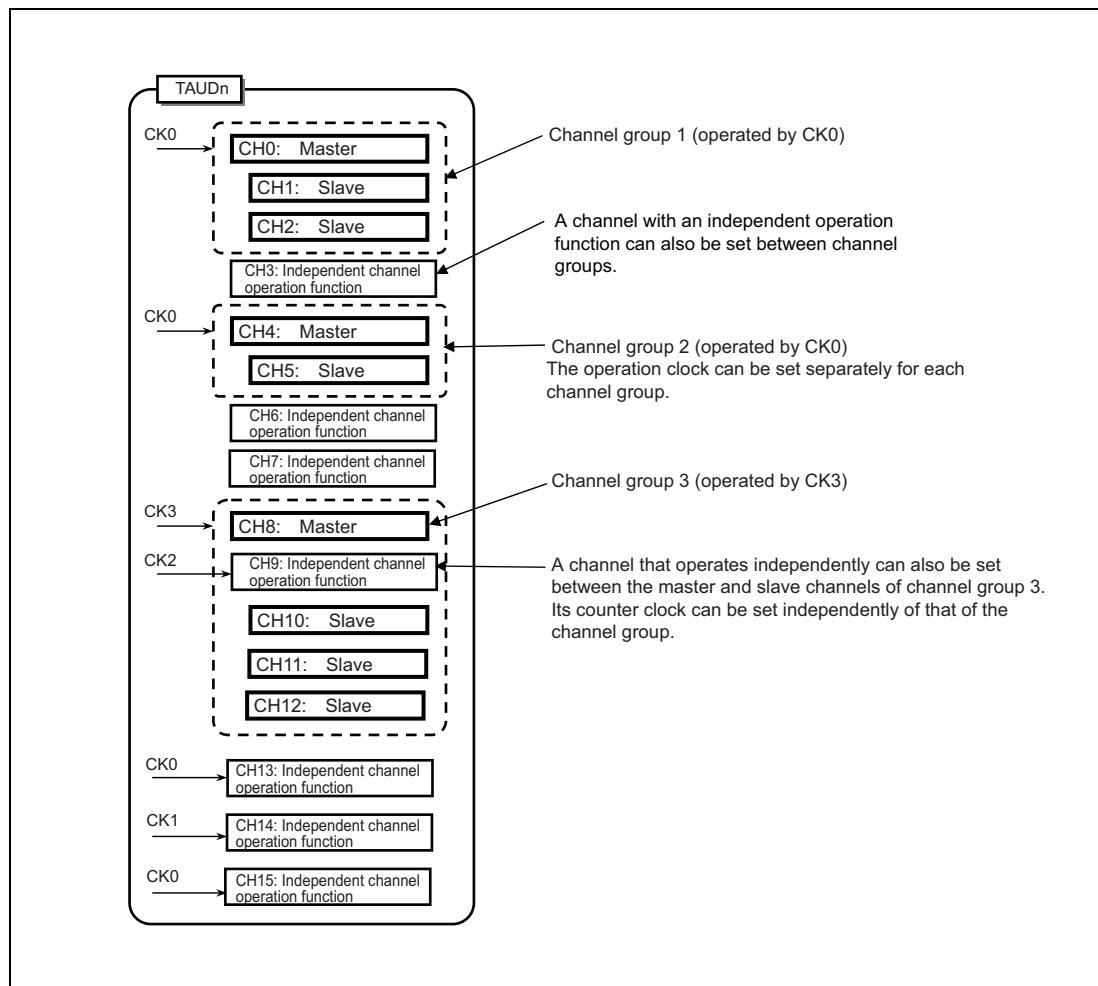


Figure 23.4 Grouping of Channels and Assignment of Count Clocks

#### Control trigger signal for master/slave channels

- Master channels can output control trigger signals to slave channels.
- Slave channels can use control trigger signals from master channels but cannot output control trigger signals for their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

### 23.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

#### 23.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUDnTS.TAUDnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUDnTT.TAUDnTTm bits of the channels should be set at the same time.

Setting to the TAUDnTS.TAUDnTSm bits to 1 also sets the corresponding TAUDnTE.TAUDnTEM bits to 1, enabling counting. The count start timing depends on operating mode.

#### 23.5.2.2 Simultaneous Start between the Units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about how to perform simultaneous start between the units, see **Section 27.8, Simultaneous Start Trigger Function**.

## 23.6 Simultaneous Rewrite

### 23.6.1 Overview of Operations

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUDnCDRm and TAUDnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUDnIm being issued on the upper channel specified by TAUDnRDC.TAUDnRDCm

There are four methods for simultaneous rewrite. These are listed in **Table 23.45**, along with how to specify them and when they cause simultaneous rewrite to be triggered.

**Table 23.45 Simultaneous Rewrite Methods and when They are Triggered**

Method	Simultaneous Rewrite Triggered when	TAUDnRDE. TAUDnRDEm	TAUDnRDS. TAUDnRDSm	TAUDnRDM. TAUDnRDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular cycle of the corresponding slave channel.	1	0	1
C1	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm	1	1	0/1
C2	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm that in turn is triggered by an external signal	1	1	0/1

**Table 23.46** lists which of these four methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 23.12, Independent Channel Operation Functions** and **Section 23.15, Synchronous Channel Operation Functions**.

Table 23.46 Channel Functions and the Methods They Use for Simultaneous Rewrite

Function	A	B	C1	C2	TAUDnTOL. TAUDnTOLm
Simultaneous Rewrite Trigger Output Function Type 1				✓	
PWM Output Function	✓		✓		✓
One-Shot Pulse Output Function	✓				
Trigger Start PWM Output Function	✓			✓	
Delay Pulse Output Function	✓				
Triangle PWM Output Function		✓	✓		✓
Triangle PWM Output Function with Dead Time	✓		✓		
Interrupt Request Signals Culling Function	✓	✓	✓		
AD Conversion Trigger Output Function Type 1	✓		✓		
AD Conversion Trigger Output Function Type 2		✓	✓		
Non-Complementary Modulation Output Function Type 1	✓		✓		
Non-Complementary Modulation Output Function Type 2	✓		✓		
Complementary Modulation Output Function	✓		✓		

**Note:** ✓: Available, (Blank): Unavailable

### 23.6.2 How to Control Simultaneous Rewrite

**Figure 23.5** shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

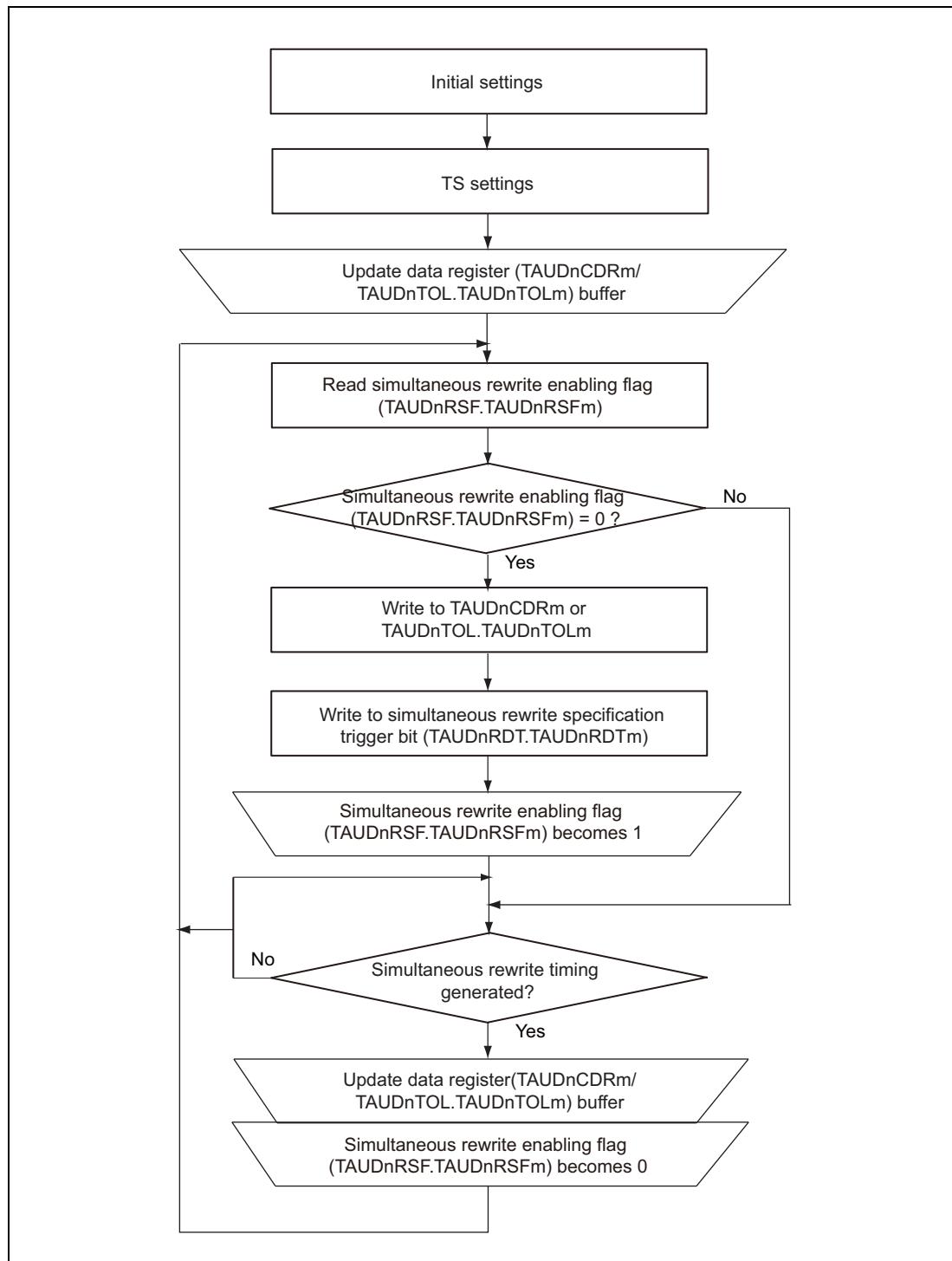


Figure 23.5 General Procedure for Simultaneous Rewrite

### 23.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUDnRDE.TAUDnRDEM = 1
- To select the type of simultaneous rewrite, set TAUDnRDM.TAUDnRDMm and TAUDnRDS.TAUDnRDSm according to the values listed in **Table 23.45, Simultaneous Rewrite Methods and when They are Triggered.**
- Specify a simultaneous rewrite trigger channel by using TAUDnRDC.TAUDnRDCm. (Prerequisite: TAUDnRDS.TAUDnRDSm has been set to the upper channel.)

### 23.6.2.2 Start Counter and Count Operation

- To start all the TAUDnCNTm counters of the channel group, set the corresponding TAUDnTS.TAUDnTSM bits to 1. The values of TAUDnTOL.TAUDnTOLm and the data registers (TAUDnCDRm) are loaded into the corresponding TAUDnTOL.TAUDnTOLm buffer (TAUDnTOL.TAUDnTOLm buf) and data buffer registers (TAUDnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1 sets the reload flag (TAUDnRSF.TAUDnRSFm) to 1, enabling simultaneous rewrite. TAUDnRSF.TAUDnRSFm remains set to 1 until simultaneous rewrite is completed.
- When the specified trigger for simultaneous rewrite is detected, the TAUDnRSF.TAUDnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1). If it is, simultaneous rewrite is carried out. Otherwise the simultaneous rewrite is not carried out and waits for the next trigger detection.

### 23.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and are applied the next time the counter starts or restarts.
- When simultaneous rewrite is complete, the TAUDnRSF.TAUDnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

### 23.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUDnRDE.TAUDnRDEM, TAUDnRDS.TAUDnRDSm, TAUDnRDM.TAUDnRDMm, and TAUDnRDC.TAUDnRDCm cannot be changed while the counter is in operation (TAUDnTE.TAUDnTEM = 1).
- TAUDnTOL.TAUDnTOLm can only be rewritten during operation with PWM output function or triangle PWM output function. For all other output functions, TAUDnTOL.TAUDnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUDTTOUTm outputs an invalid wave.
- When an upper channel is used as a channel issuing the simultaneous rewrite trigger (TAUDnRDS.TAUDnRDSm = 1), the TAUDnRDC.TAUDnRDCm bit controls all the lower channels. This means that if the TAUDnRDC.TAUDnRDCm bits of CH2 and CH7 are set to 1 and the TAUDnRDC.TAUDnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger (TAUDnRDE.TAUDnRDEM and TAUDnRDS.TAUDnRDSm = 1) but no upper channel is set (TAUDnRDC.TAUDnRDC[15:0] = 0), simultaneous rewrite cannot take place.

### 23.6.4 Types of Simultaneous Rewrite

In the following section, the four simultaneous rewrite methods are explained using timing diagrams.

#### 23.6.4.1 Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)

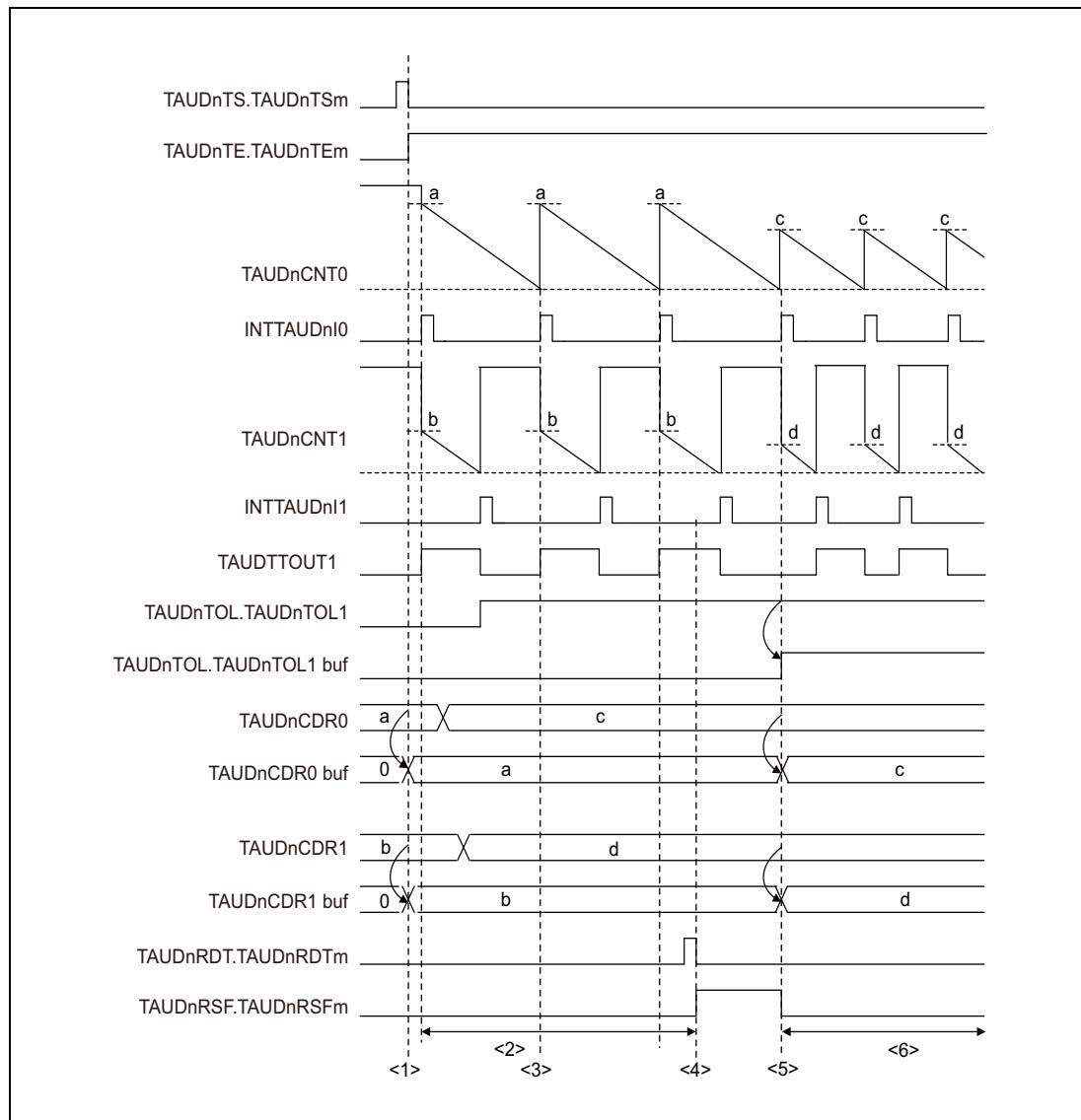


Figure 23.6 Simultaneous Rewrite when the Master Channel (Re)starts Counting

#### Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

#### Description:

- (1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer and TAUDnTOL.TAUDnTOLm value is copied to the TAUDnTOL.TAUDnTOLm buffer.
- (2) The TAUDnCDRm and TAUDnTOL.TAUDnTOLm registers can be written at any time.
- (3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0)

- (4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.
- (5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer and the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.
- (6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

### 23.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)

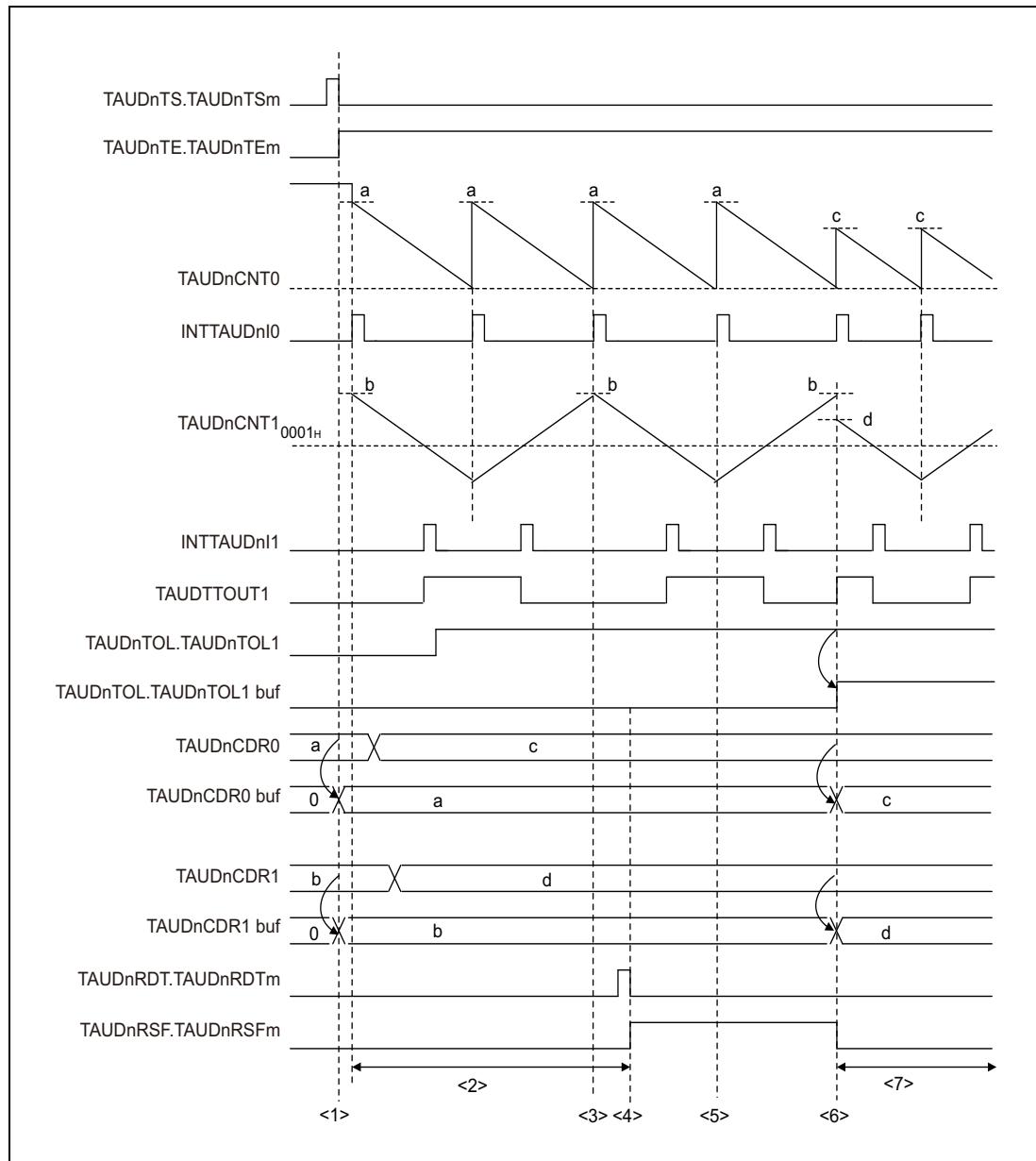


Figure 23.7 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel

#### Setting:

CH0 is the master channel which performs counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

#### Description:

- (1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.
- (2) The TAUDnCDRm and TAUDnTOL registers can be written at any time.
- (3) Simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

- (4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.
- (5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.
- (6) Simultaneous rewrite takes place at the top of the triangular cycle. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer, the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.
- (7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

#### 23.6.4.3 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)

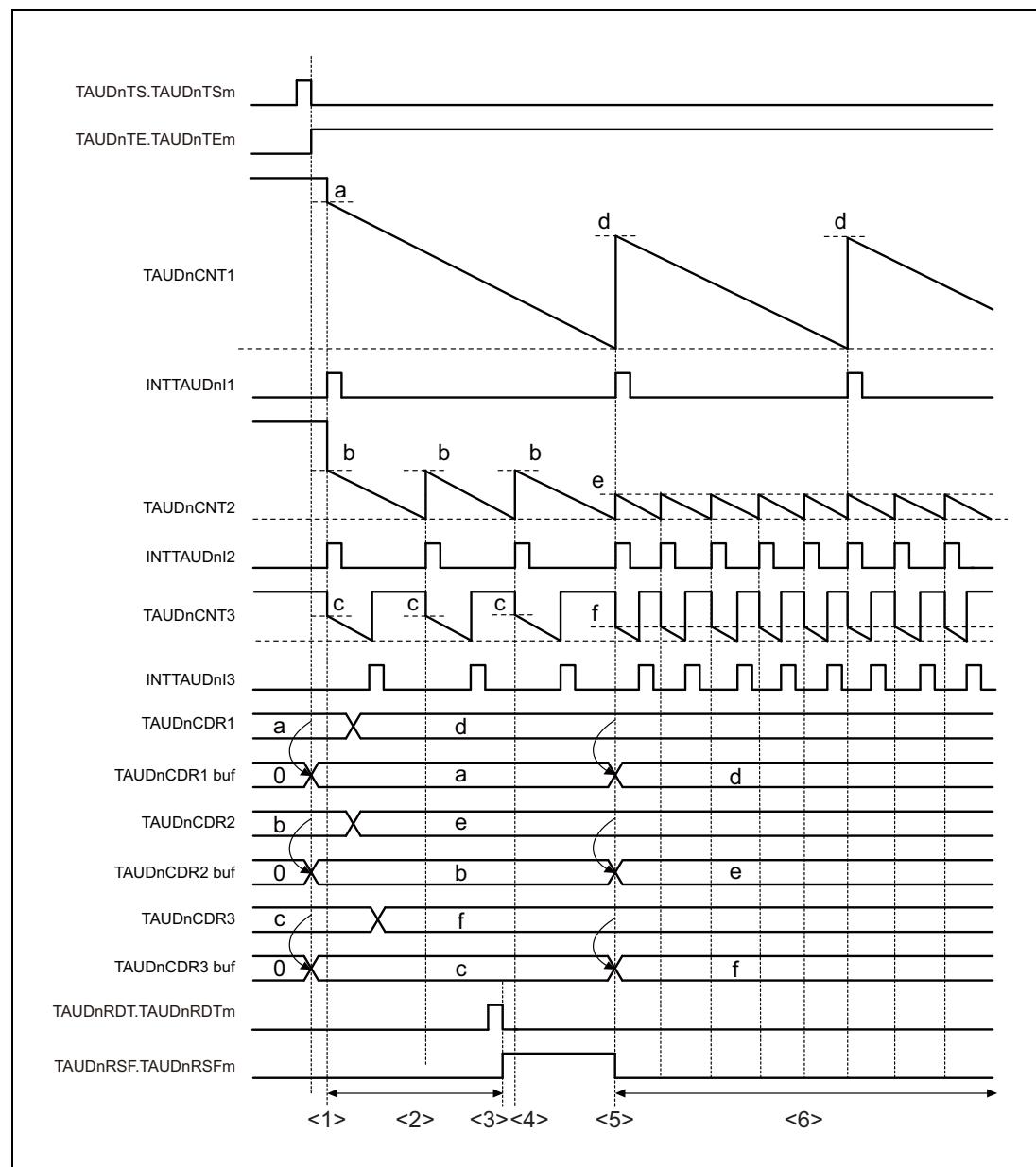


Figure 23.8 Simultaneous Rewrite When INTTAUDnIm Is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm

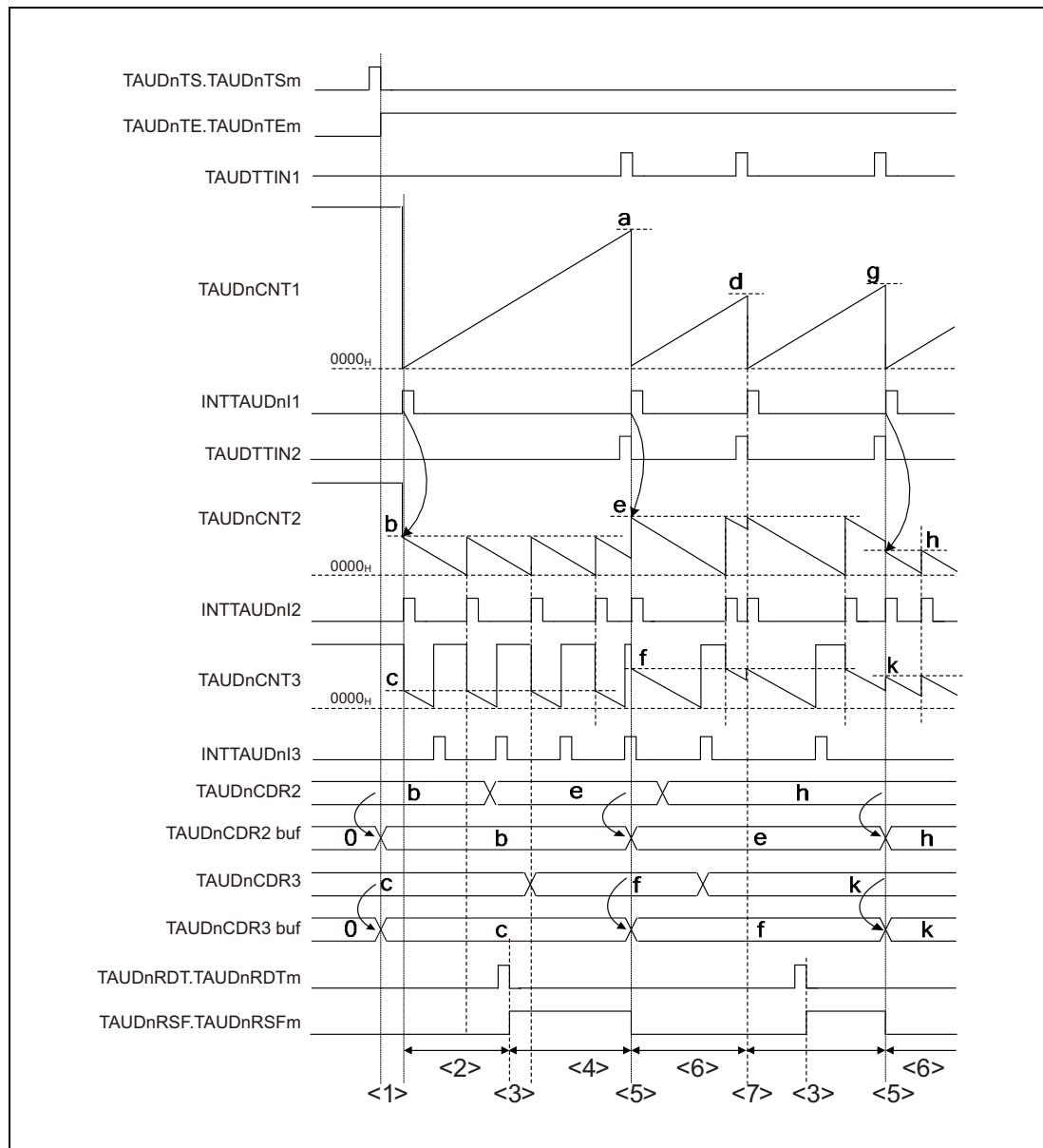
**Setting:**

CH1 is an upper channel which performs counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUDnRDC register specifies a channel which generates simultaneous rewrite triggers.

**Description:**

- (1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.
- (2) The TAUDnCDRm register is always ready to write.
- (3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.
- (4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.
- (5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000<sub>H</sub>. The TAUDnCDRm values are loaded into the corresponding TAUDnCDRm buffers.
- (6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be recharged.

#### 23.6.4.4 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2)



**Figure 23.9 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal**

**Setting:**

CH1 is an upper channel which performs counting up, CH2 is a master channel, and CH3 is the slave channel. The synchronous channel operation method C2 is applied. The TAUDnRDC register specifies which upper channel is monitored for an INTTAUDnIm trigger.

**Description:**

- (1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer. However, as TAUDnCDR1 operates in capture mode, TAUDnCDR1 value is not copied to the TAUDnCDR1 buffer.

- (2) The TAUDnCDRm register is always ready to write.
- (3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.
- (4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.
- (5) Simultaneous rewrite is triggered by INT1 which is caused by external signal TIN1. The TAUDnCDRm values are written to the corresponding TAUDnCDRm buffers.
- (6) The counters count down and await the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.
- (7) An external signal occurs at TIN2 but simultaneous rewrite does not take place because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

## 23.7 Channel Output Modes

The output of the TAUDTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUDnTOE.TAUDnTOEm = 0)  
When controlled by software, the value written in the output register bit (TAUDnTO.TAUDnTOm) is sent to the output pin (TAUDTTOUTm).
- By TAUD signals (TAUDnTOE.TAUDnTOEm = 1)  
When controlled by TAUD signals, the output level of TAUDTTOUTm is set or reset or toggled by internal signals. The value of TAUDnTO.TAUDnTOm is updated accordingly to reflect the value of TAUDTTOUTm.
  - Independently (TAUDnTOM.TAUDnTOMm = 0)  
In case of independent operation, the output of the TAUDTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUDnTOM.TAUDnTOMm = 0).
  - Synchronously (TAUDnTOM.TAUDnTOMm = 1)  
In case of synchronous operation, the output of the TAUDTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUDnTOM.TAUDnTOMm = 1).

The TAUDnTO.TAUDnTOm bit can always be read to determine the current value of TAUDTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

### Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 23.47, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 23.7.2, Channel Output Modes Controlled Independently by TAUDn Signals**
- **Section 23.7.3, Channel Output Modes Controlled Synchronously by TAUDn Signals**

### Batch operation of TAUDnTOm bit

Whether a set value is reflected to the TAUDnTOm bit or not is controlled by the TAUDnTOE.TAUDnTOEm bit.

The TAUDnTOm setting is written only to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 0 when a write to the TAUDnTO register is attempted. No TAUDnTOm setting is reflected to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 1.

#### NOTE

---

TAUDnTO.TAUDnTOm bit is placed so that its bit number corresponds to a channel number.

---

### Output logic

Positive logic or negative logic of the output is specified by control bit TAUDnTOL.TAUDnTOLm.

The value of TAUDnTOL.TAUDnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUDnTOL.TAUDnTOLm result in an undefined TAUDTTOUTm signal output.

See **Section 23.6, Simultaneous Rewrite.**

The various channel output modes and the channel output control bits are listed in **Table 23.47.**

**Table 23.47 Channel Output Modes**

Channel Output Mode	TAUDn TOE. TAUDn TOEm	TAUDn TOM. TAUDn TOMm	TAUDn TOC. TAUDn TOCm	TAUDn TDE. TAUDn TDEM	TAUDn TRE. TAUDn TREM	TAUDn TME. TAUDn TMEm	TAUDn TDM. TAUDn TDMm
<b>By software</b>							
Independent channel output mode controlled by software	0	X					
<b>By TAUD signals, independently</b>							
Independent channel output mode 1	1	0	0	0	0	0	0
with real-time output					1		
Independent channel output mode 2			1		0		
<b>By TAUD signals, synchronously</b>							
Synchronous channel output mode 1	1	1	0	0	0	0	0
with non-complementary modulation output					1	X	
Synchronous channel output mode 2			1	0	0	0	0
with dead time output				1			
with one-phase PWM output							1
with complementary modulation output					1	1	0
with non-complementary modulation output			1	0			

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

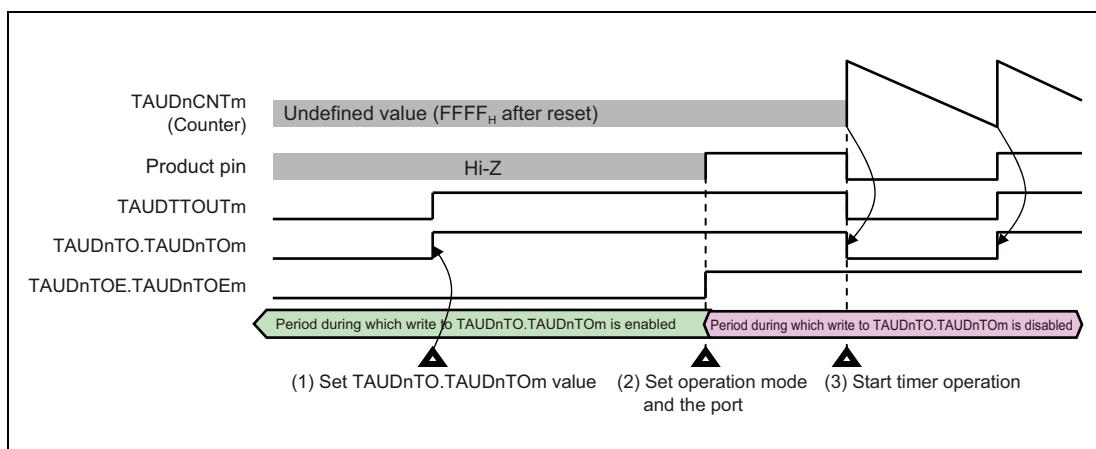
**NOTES**

1. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1):
  - TAUDnTOM.TAUDnTOMm
  - TAUDnTOC.TAUDnTOCm
  - TAUDnTDE.TAUDnTDEM
  - TAUDnTRE.TAUDnTREM
  - TAUDnTDM.TAUDnTDMm
2. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1) except in channel output modes with modulation output:
  - TAUDnTME.TAUDnTMEM
  - TAUDnTDL.TAUDnTDLm

### 23.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUDTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUDnTOE.TAUDnTOEm = 0).

- (1) Set TAUDnTO.TAUDnTOM to specify the initial level of the TAUDTTOUTm output.
- (2) Set channel output mode according to **Table 23.47, Channel Output Modes**, and the output logic using the TAUDnTOL.TAUDnTOLm bit.
- (3) Start the counter (TAUDnTS.TAUDnTSm = 1).



**Figure 23.10 General Procedure for Specifying a TAUDTTOUTm Channel Output Mode**

## 23.7.2 Channel Output Modes Controlled Independently by TAUDn Signals

This section lists the channel output modes that are controlled independently by TAUDn signals. The control bits used to specify a mode are listed in **Table 23.47, Channel Output Modes**.

### 23.7.2.1 Independent Channel Output Mode 1

#### Set/reset conditions

In this output mode, TAUDTTOUTm toggles when INTTAUDnIm is detected. The value of TAUDnTOL.TAUDnTOLm is ignored.

#### Prerequisites

There are no prerequisites other than those shown in **Table 23.47, Channel Output Modes**.

### 23.7.2.2 Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUDnTRO.TAUDnTROm bit of the trigger channel is output to TAUDTTOUTm. The trigger channel is specified by setting the corresponding TAUDnTRC.TAUDnTRCm bit to 1. It controls all lower channels for which TAUDnTRC.TAUDnTRCm = 0.

#### Set/reset conditions

The value of TAUDnTRO.TAUDnTROm bit is sent to TAUDTTOUTm only when an INTTAUDnIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUDTTINm input edge/counter start

The type of trigger is set using the TAUDnCMORm.TAUDnMD[4:1] bits.

#### Prerequisites

Both the master and slave channels can be set as a trigger generation channel. A channel for which TAUDnTRC.TAUDnTRCm is set to 1 serves as a trigger generation channel regardless of the value of TAUDnTRE.TAUDnTREm.

If there is no channel for which TAUDnTRC.TAUDnTRCm is set to 1 or if TAUDnTRC.TAUDnTRC0 = 0, real-time output cannot take place.

This can be seen in **Figure 23.11**.

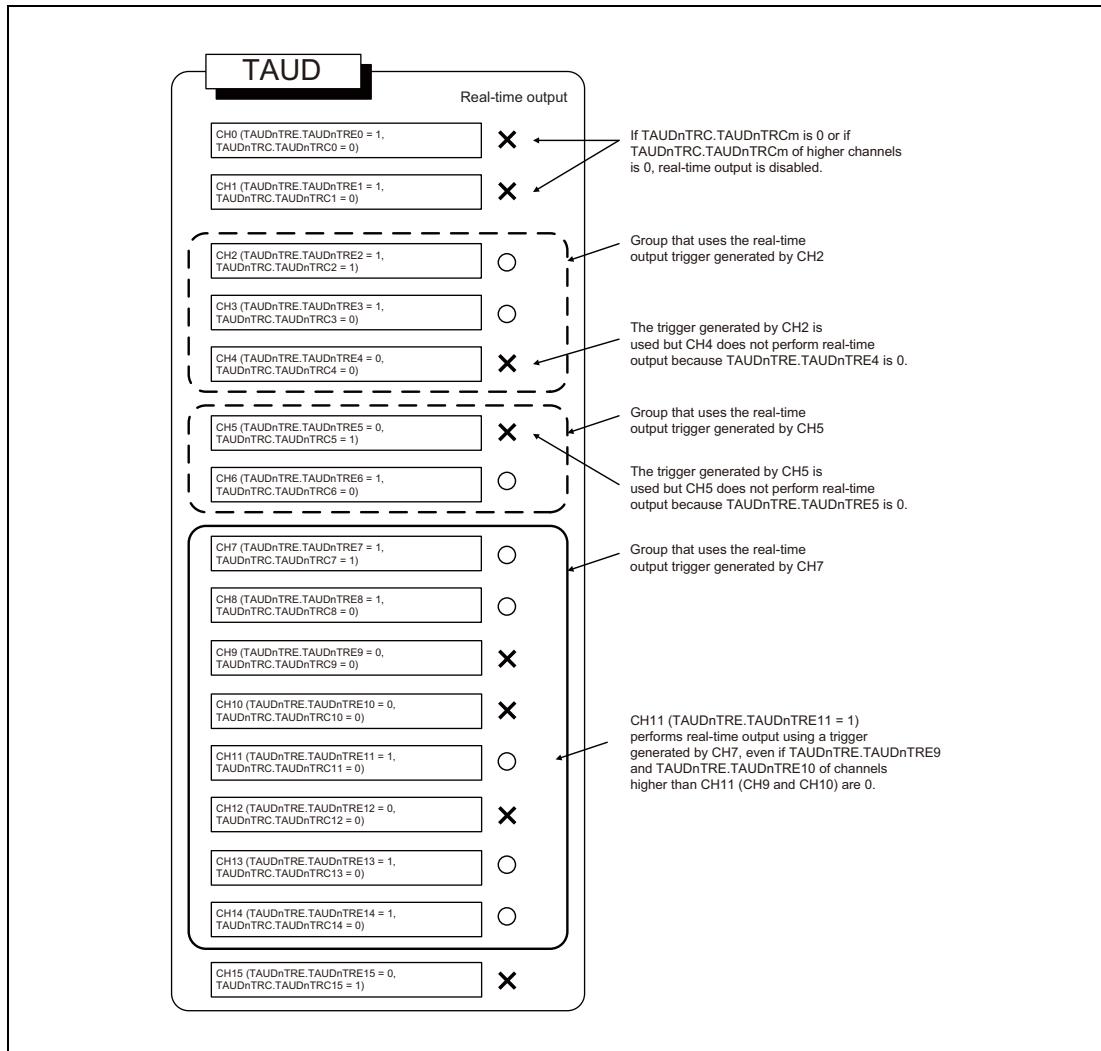


Figure 23.11 Real-Time Output

### 23.7.2.3 Independent Channel Output Mode 2

#### Set/reset conditions

In this output mode, TAUDTTOUTm is set when INTTAUDnIm occurs at the time of count start, and reset when INTTAUDnIm occurs due to a match between TAUDnCNTm and TAUDnCDRm.

#### Prerequisites

There are no prerequisites other than those shown in **Table 23.47, Channel Output Modes**.

### 23.7.3 Channel Output Modes Controlled Synchronously by TAUDn Signals

This section lists the channel output modes that are controlled synchronously by TAUDn signals. The control bits used to specify a mode are listed in **Table 23.47, Channel Output Modes**.

#### 23.7.3.1 Synchronous Channel Output Mode 1

##### **Set/reset conditions**

In this output mode, INTTAUDnIm of master channel serves as a set signal and INTTAUDnIm of the slave channel as a reset signal. If INTTAUDnIm of master channel and INTTAUDnIm of the slave channel are generated at the same time, INTTAUDnIm of the slave channel (reset signal) has priority over INTTAUDnIm (set signal) of master channel, i.e., the master channel is ignored.

##### **Prerequisites**

There are no prerequisites other than those shown in **Table 23.47, Channel Output Modes**.

#### 23.7.3.2 Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

##### **Set/reset conditions**

In this output mode, TAUDTTOUTm outputs the result of an AND operation between the PWM output and the real-time output bit (TAUDnTRO.TAUDnTROM) of a channel.

The phase period to which the dead time is added is specified using the TAUDnTDL.TAUDnTDLm bit; for positive phase set TAUDnTDL.TAUDnTDLm = 0 and for negative phase set TAUDnTDL.TAUDnTDLm = 1.

##### **Prerequisites**

A set of at least three channels is required to generate the PWM output. The master channel and slave channel 1 generate a period, and slave channel 2 generates the duty cycle. In typical applications, five more slave channels are also used that operate in the same manner as slave channel 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUDnTRO.TAUDnTROM, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed, its new value is applied upon detection of INTTAUDnIm on the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on the master channel.

#### 23.7.3.3 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is a triangle PWM output at TAUDTTOUTm. For details, see **Section 23.15.7, Triangle PWM Output Function**.

##### **Set/reset conditions**

TAUDnCNTm of the slave channel counts down and up alternatively. When it passes 0001<sub>H</sub> it generates an interrupt, causing TAUDTTOUTm to toggle.

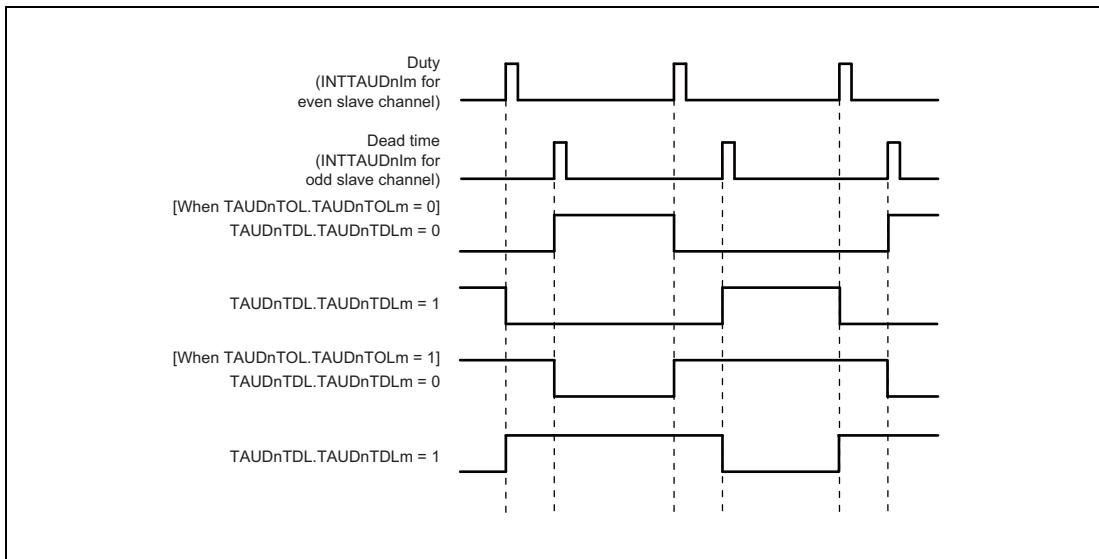
### Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUDTTOUTm should be set to 0 before the function starts.

#### 23.7.3.4 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in **Figure 23.12**.

##### Set/reset conditions



**Figure 23.12 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output**

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

### Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel  
The master channel should be set to interval timer mode.
- One even slave channel  
The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)  
The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

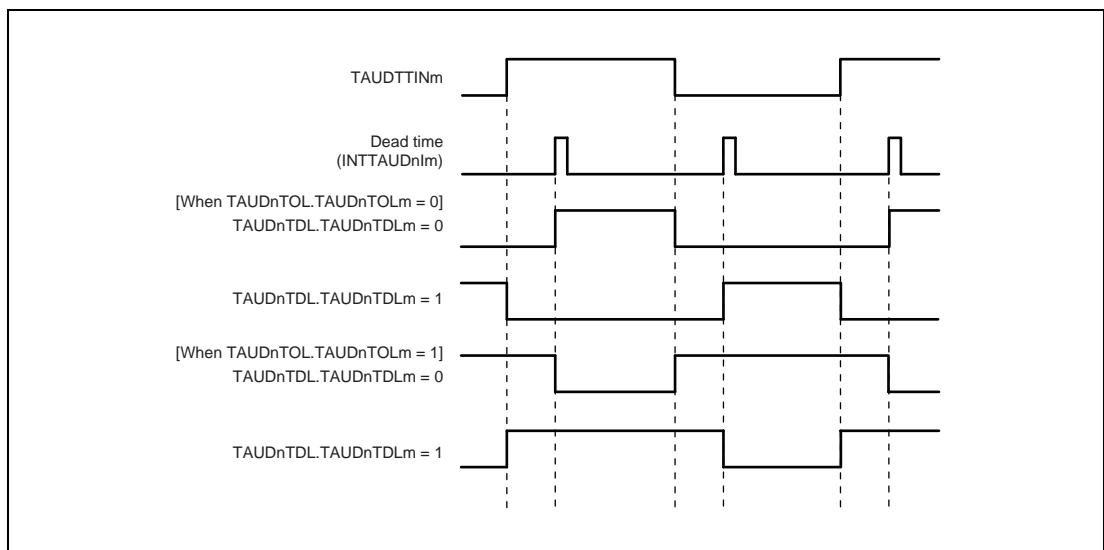
- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm

- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

### 23.7.3.5 Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in **Figure 23.13**.

#### Set/reset conditions



**Figure 23.13 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output**

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

#### Prerequisites

One-phase PWM output control requires a set of two channels:

- One even slave channel
  - One odd slave channel (even channel + 1)
- The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm
- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

### 23.7.3.6 Synchronous Channel Output Mode 2 with Complementary Modulation Output

#### Set/reset conditions

In this output mode, TAUDTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of real-time output bit (TAUDnTRO.TAUDnTROm), the modulation output bit (TAUDnTME.TAUDnTMEm), and the output level bit (TAUDnTOL.TAUDnTOLm) of a pair of slave channels.

For details, see **Section 23.16.3, Complementary Modulation Output Function**.

#### Prerequisites

A set of at least four channels is required for this mode. The master channel and slave channel 1 generate a period, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time. Slave channels 2 and 3 are a pair. In typical applications, four more channels are also used, which operates in the same manner as slave channels 2 and 3 respectively.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed during operation, its new value is applied upon detection of INTTAUDnIm at the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on an even slave channel.

### 23.7.3.7 Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference from synchronous channel output mode 1 with non-complementary modulation output is the PWM wave shape.

Mode 1 has a square wave while mode 2 has a triangular wave.

## 23.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUDnTS.TAUDnTSm is set to 1 in each operating mode.

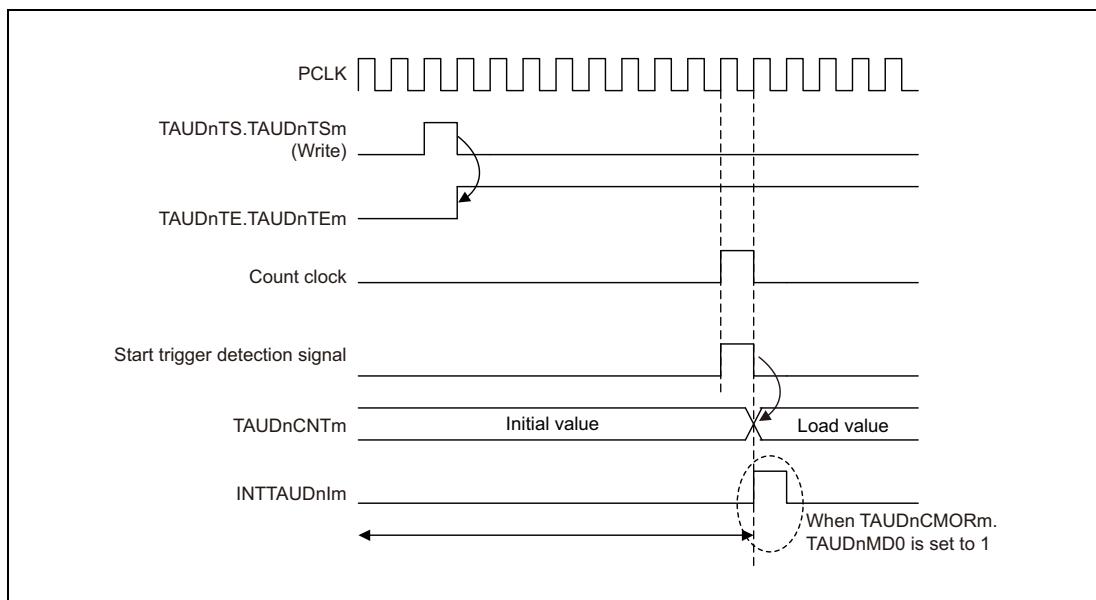
In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

### CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

### 23.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUDnTS.TAUDnTSm is set to 1. The value of data register is also loaded when the counter starts.



**Figure 23.14 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode**

### NOTE

Make sure to set TAUDnCMORm.TAUDnMD0 to 0 when using the count-up/-down mode.

### 23.8.2 Event Count Mode

The value of data register is loaded as soon as TAUDnTS.TAUDnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

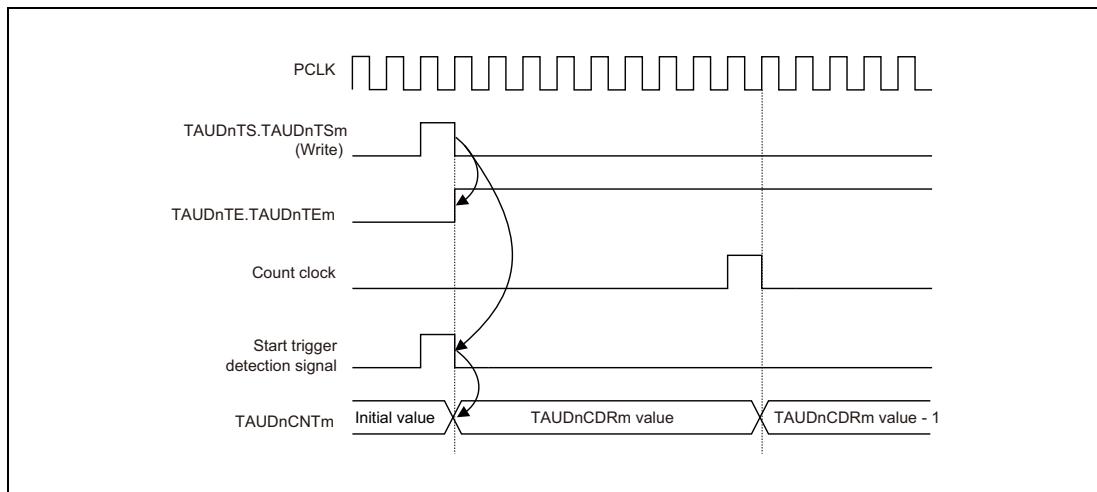


Figure 23.15 Start Timing in Event Count Mode

### 23.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUDTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

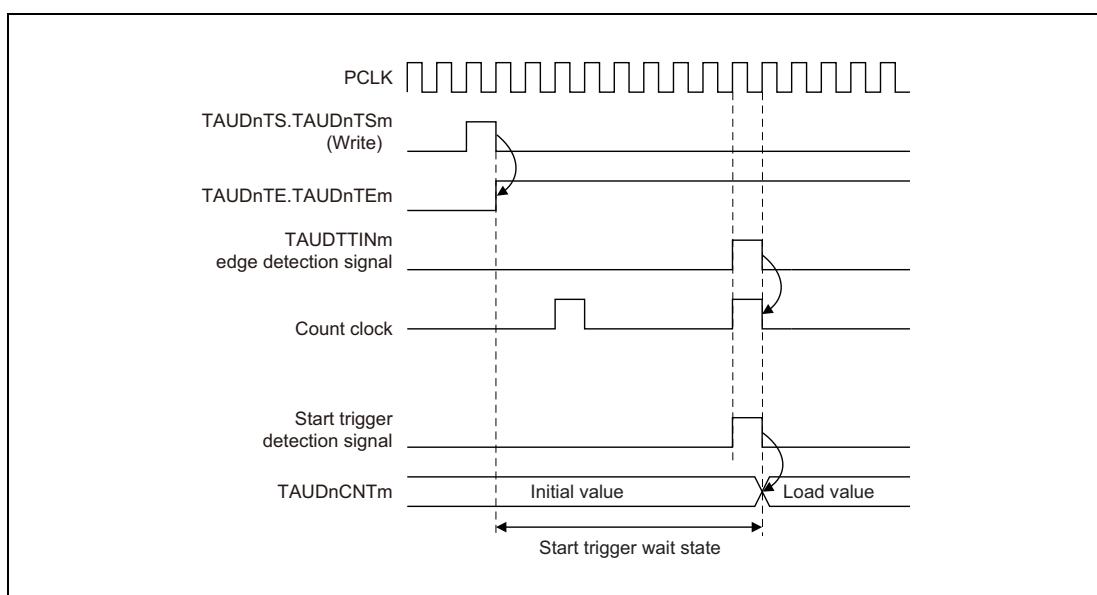


Figure 23.16 Count Start Timing in Other Operating Modes

## 23.9 TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUDnIm is generated using the TAUDnCMORm.TAUDnMD0 bit. The generation of INTTAUDnIm when the TAUDnCMORm.TAUDnMD0 bit starts counting and the effect to TAUDTTOUTm depend on the selected function. For details, refer to the description of TAUDnCMORm.TAUDnMD0 of each function.

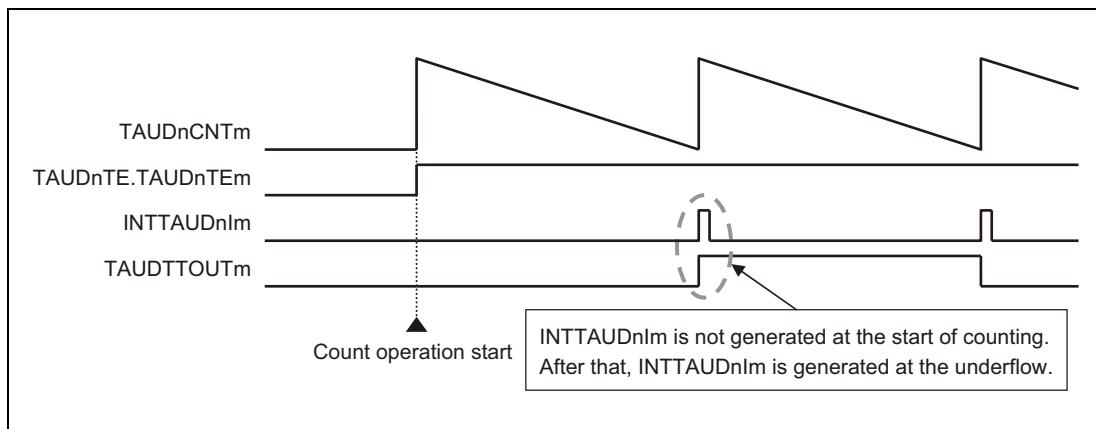


Figure 23.17 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 0)

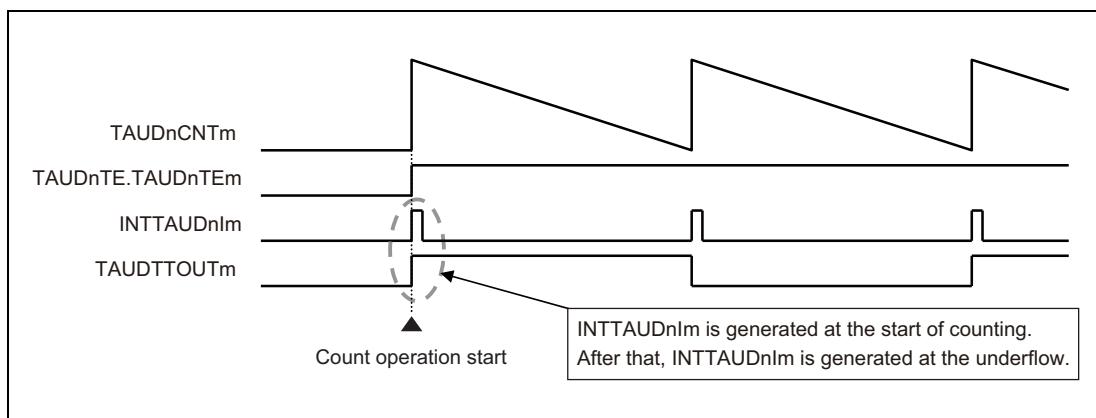


Figure 23.18 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 1)

## 23.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches  $\text{FFFF}_H$  and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches  $0000_H$  at the same time as the first channel overflows ( $\text{TAUDnCNTm} = \text{FFFF}_H$ ).
- Set  $\text{TAUDnCDRm}$  of the second channel to  $\text{FFFF}_H$ .
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same  $\text{TAUDTTINm}$  input.
- The trigger detection settings ( $\text{TAUDnCMORm.TAUDnSTS[2:0]}$  and  $\text{TAUDnCMURm.TAUDnTIS[1:0]}$ ) must be identical for both channels.

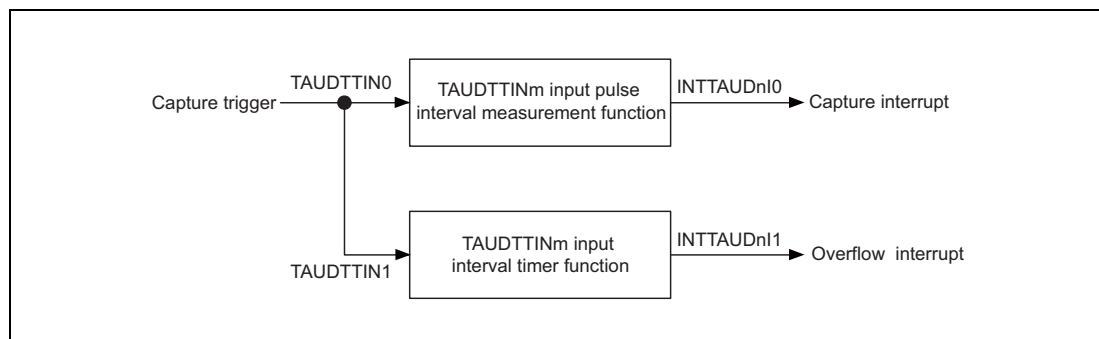
### Result:

The down-counter of the second channel reaches  $0000_H$  at exactly the same time as the up-counter of the first channel overflows ( $\text{TAUDnCNTm} = \text{FFFF}_H$ ). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

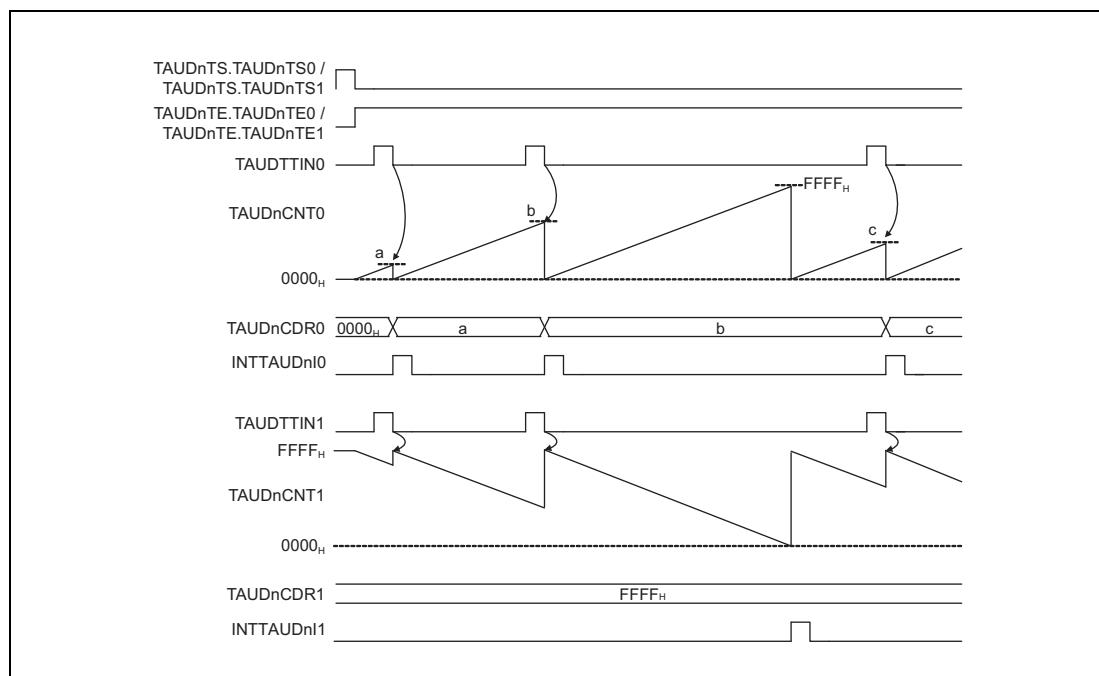
### 23.10.1 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the TAUDTTINm input interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input pulse interval measurement function exceeds  $FFFF_H$ .



**Figure 23.19 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function**

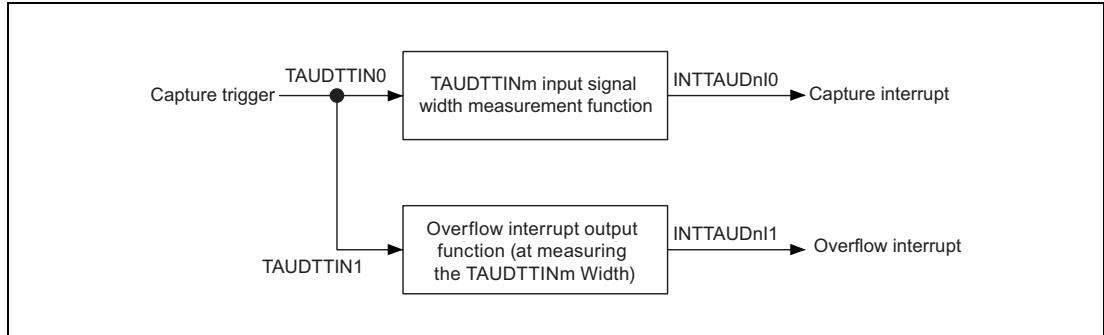
#### Timing diagram



**Figure 23.20 Interrupt Generation via Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function**

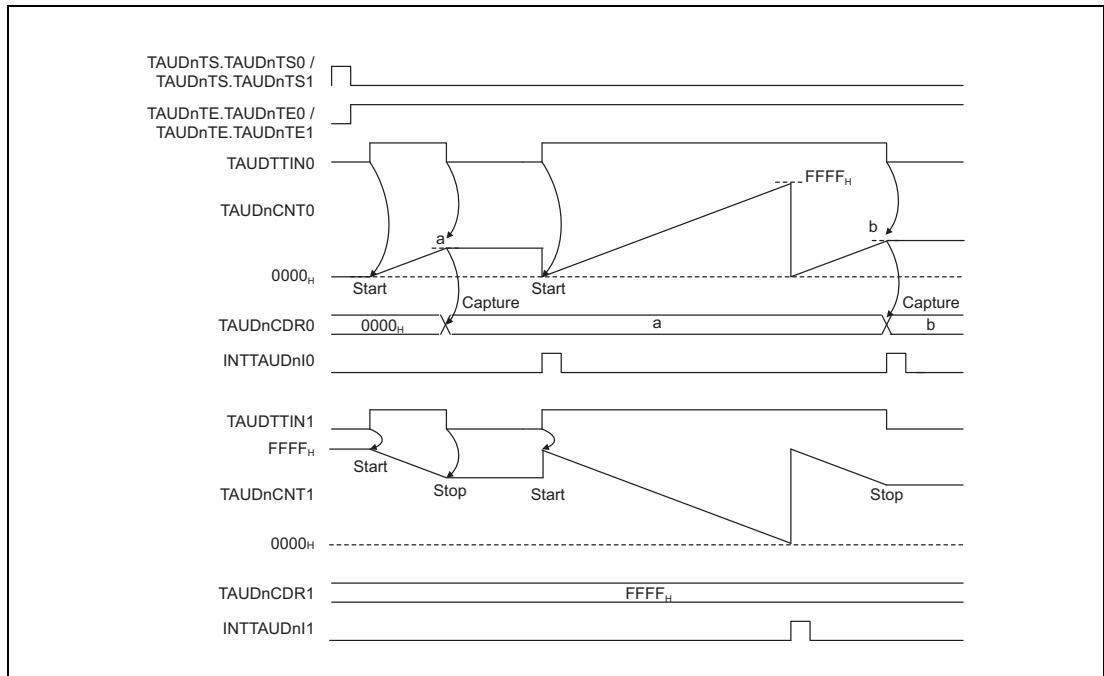
### 23.10.2 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at measuring the TAUDTTINm width) can detect the overflow when TAUDnCNTm of the TAUDTTINm input signal width measurement function exceeds  $FFFF_{H}$ .



**Figure 23.21 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)**

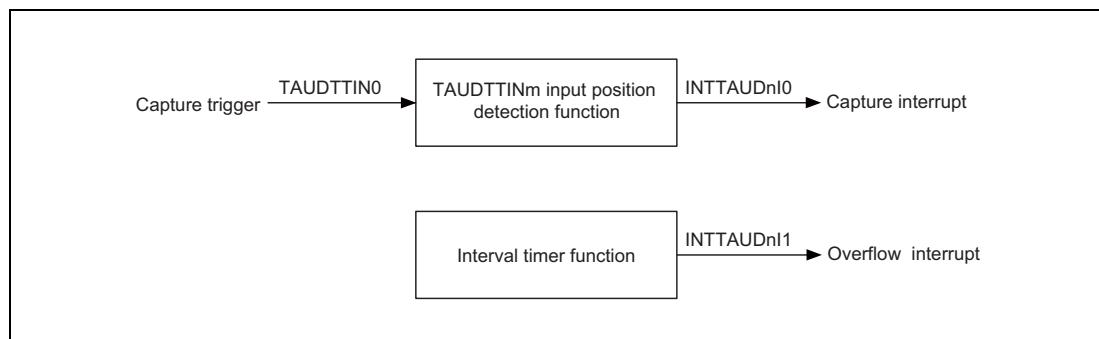
#### Timing diagram



**Figure 23.22 Interrupt Generation via Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)**

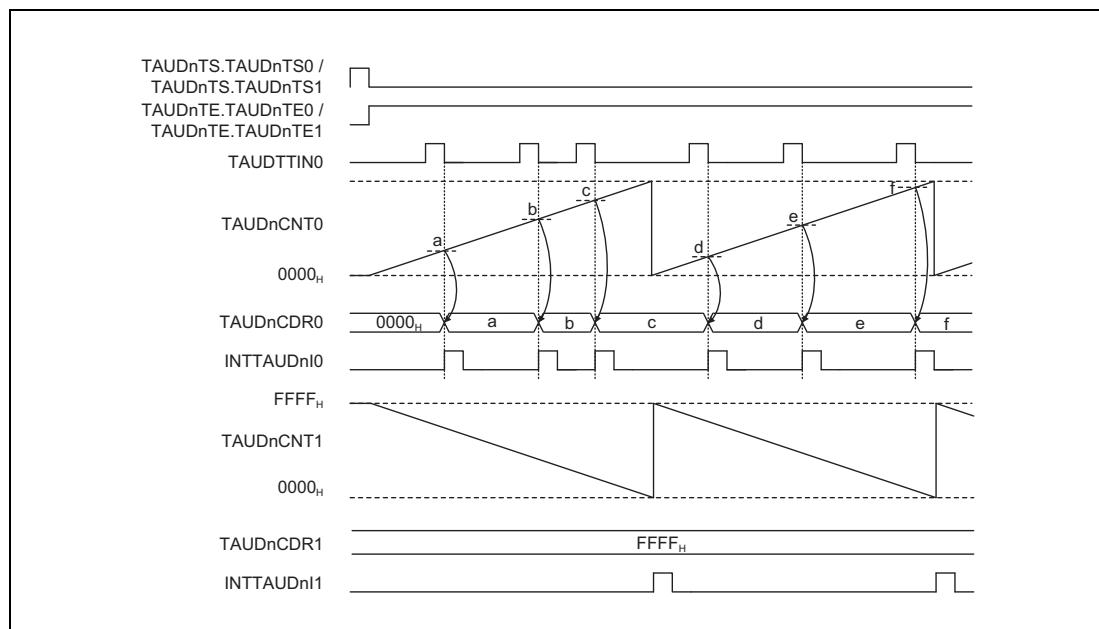
### 23.10.3 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUDnIm of the interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input position detection function exceeds  $FFFF_H$ .



**Figure 23.23 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function**

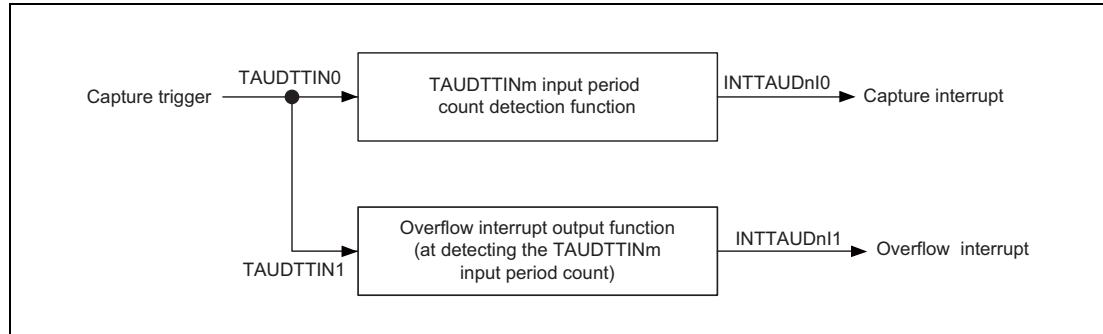
#### Timing diagram



**Figure 23.24 Interrupt Generation via Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function**

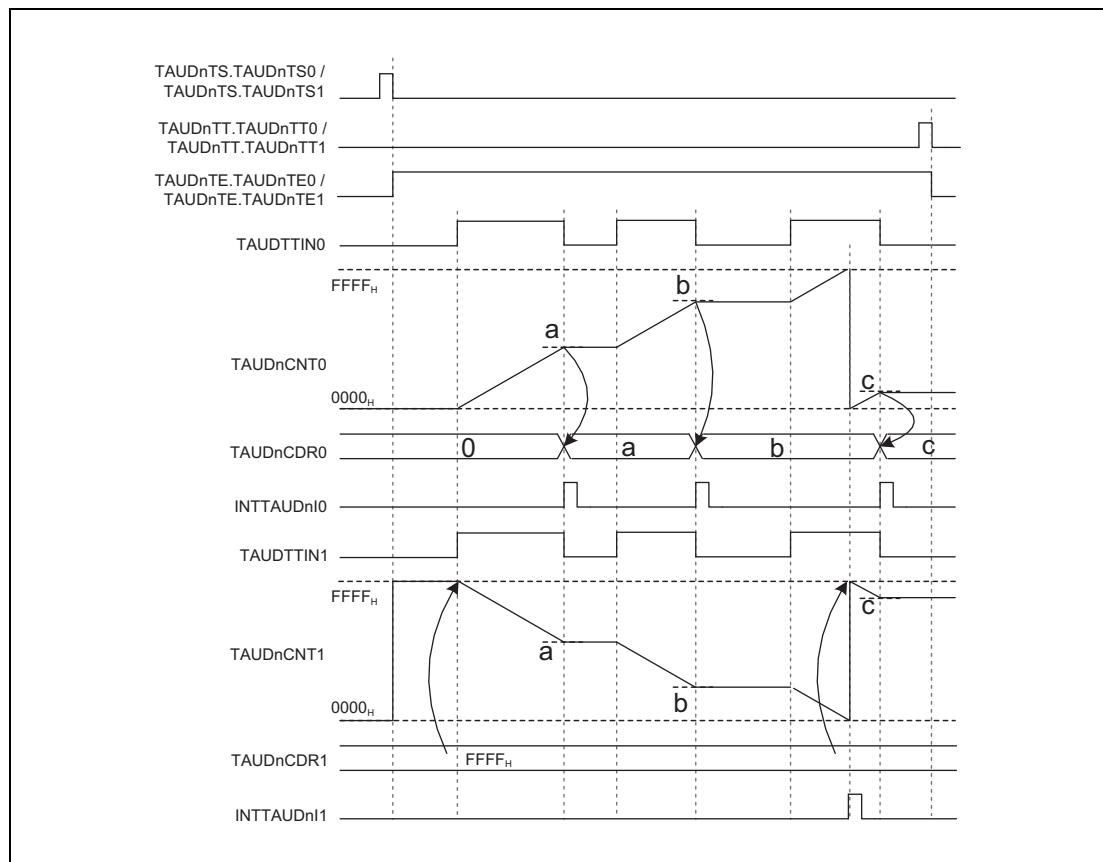
### 23.10.4 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at detecting the TAUDTTINm input period count) can detect the overflow when TAUDnCNTm of the TAUDTTINm input period count detection function exceeds  $FFFF_H$ .



**Figure 23.25 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)**

#### Timing diagram

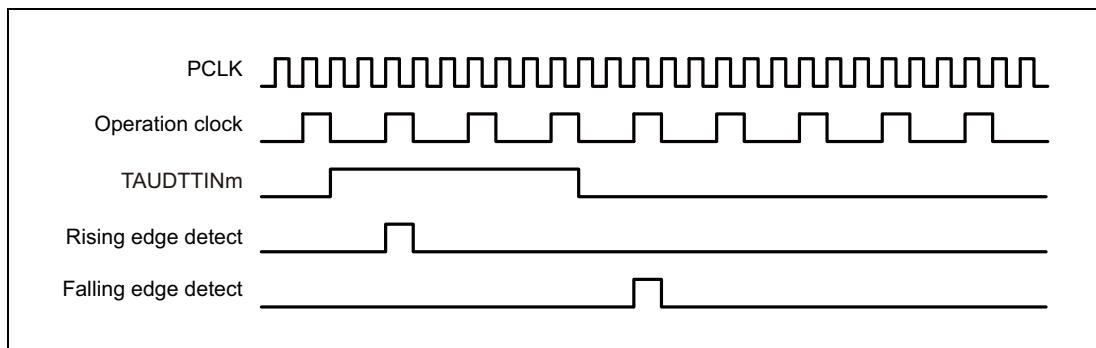


**Figure 23.26 Interrupt Generation via Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)**

## 23.11 TAUDTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

**Figure 23.27** shows when edge detection takes place.



**Figure 23.27 Basic Edge Detection Timing**

**Figure 23.27** shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDnIm pin and TAUDn causes a delay time.

## 23.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the Timer Array Unit D. For a general overview of independent channel operation, see [Section 23.2, Overview](#).

This section describes functions that generate interrupts at regular intervals or with a specified delay.

### 23.12.1 Interval Timer Function

#### 23.12.1.1 Overview

##### Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

##### Prerequisites

- The operation mode must be set to Interval Timer Mode, see [Table 23.48, Contents of the TAUDnCMORm Register for Interval Timer Function](#).
- The channel output mode must be set to Independent Channel Output Mode 1, see [Section 23.7, Channel Output Modes](#).

##### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is written to TAUDnCNTm and the counter starts to count down from this value.

When the counter reaches 0000<sub>H</sub>, INTTAUDnIm is generated and the TAUDTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and subsequently continues operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1, which in turn sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm stop but retain their values. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.

##### Conditions

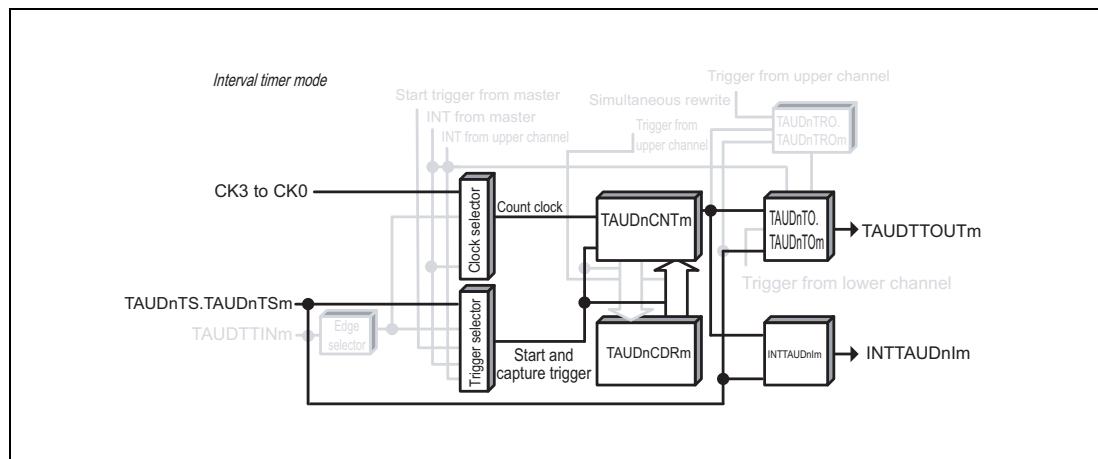
If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDTTOUTm does not toggle. This results in a negative TAUDTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details see [Section 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts](#).

#### 23.12.1.2 Equations

$$\text{INTTAUDnIm cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1)$$

$$\text{TAUDTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1) \times 2$$

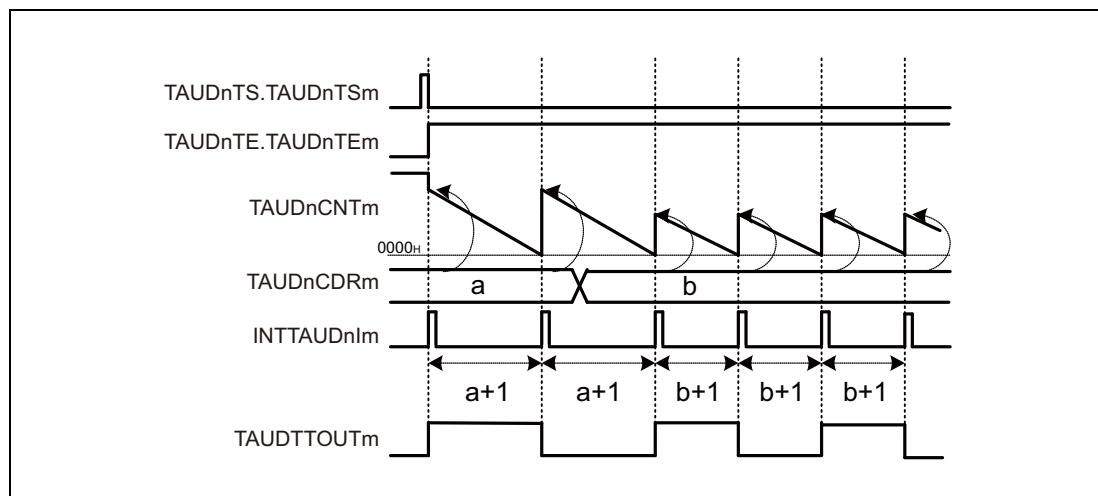
### 23.12.1.3 Block Diagram and General Timing Diagram



**Figure 23.28 Block Diagram of Interval Timer Function**

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).



**Figure 23.29 General Timing Diagram of Interval Timer Function**

### 23.12.1.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]	TAUDnCCS [1:0]	TAUDnMAS	TAUDnSTS[2:0]	TAUDnCOS [1:0]	—	TAUDnMD[4:1]	TAUDnMDO								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.48 Contents of the TAUDnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Triggers the counter by software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated to toggle TAUDTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDTOUTm at the beginning of an operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 23.49 Contents of the TAUDnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode

**Table 23.50 Control Bit Settings in Independent Channel Output Mode 1**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set these bits to 0
TAUDnTRC.TAUDnTRCM	
TAUDnTME.TAUDnTMEM	0: Disables modulation

#### NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Section 23.7, Channel Output Modes**.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the interval timer function. Therefore, these registers should be set to 0.

**Table 23.51 Simultaneous Rewrite Settings for Interval Timer Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.12.1.5 Operating Procedure for Interval Timer Function

Table 23.52 Operating Procedure for Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.48, Contents of the TAUDnCMORm Register for Interval Timer Function</b>, and <b>Table 23.49, Contents of the TAUDnCMURm Register for Interval Timer Function</b>.</p> <p>Set the value of TAUDnCDRm register.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.50, Control Bit Settings in Independent Channel Output Mode 1</b>.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMDO = 1, INTTAUDnIm is generated and TAUDTTOUTm toggles.</p>
During Operation	<p>The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.</p>	<p>TAUDnCNTm counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues.</li> <li>INTTAUDnIm is generated and TAUDTTOUTm toggles.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

Restart operation



### 23.12.1.6 Specific Timing Diagrams

#### (1) TAUDnCDRm = 0000<sub>H</sub>, count clock = PCLK/2

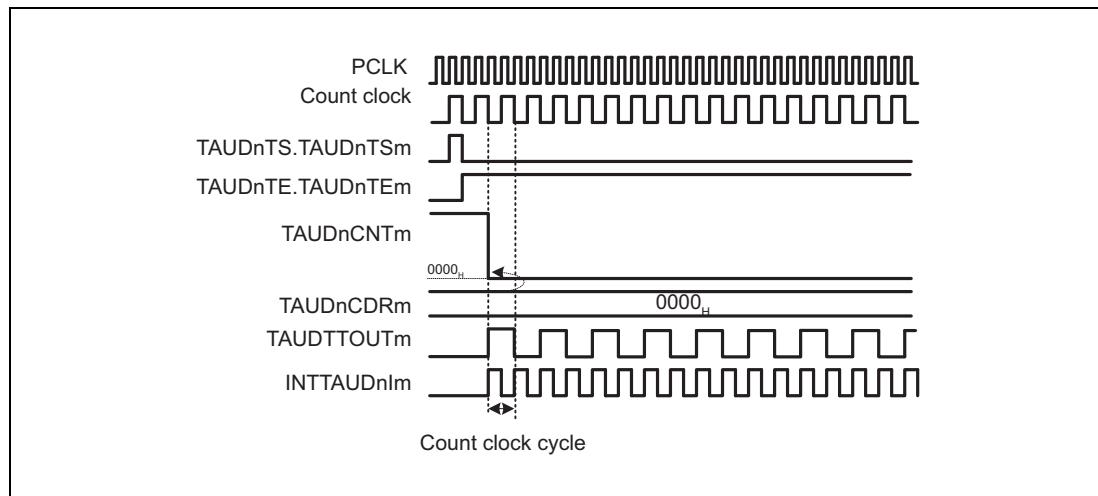
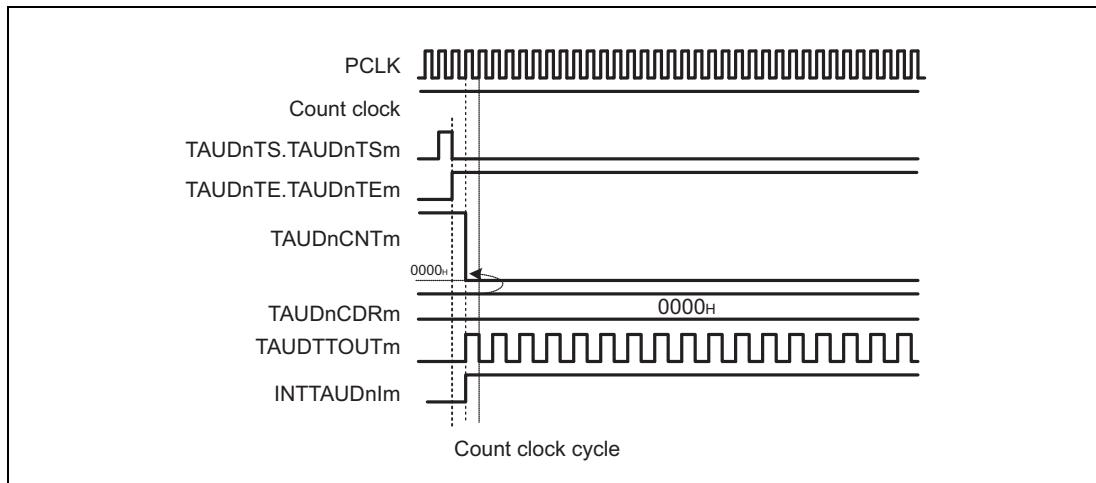


Figure 23.30 TAUDnCDRm = 0000<sub>H</sub>, Count Clock = PCLK/2

- If TAUDnCDRm = 0000<sub>H</sub> and the count clock = PCLK/2, the TAUDnCDRm value is loaded into TAUDnCNTm every count clock, meaning that TAUDnCNTm is always 0000<sub>H</sub>.
- INTTAUDnIm is generated every count clock, resulting in TAUDTOUTm toggling every count clock.

(2) TAUDnCDRm = 0000<sub>H</sub>, count clock = PCLKFigure 23.31 TAUDnCDRm = 0000<sub>H</sub>, Count Clock = PCLK

- If TAUDnCDRm = 0000<sub>H</sub> and the count clock = PCLK, the TAUDnCDRm value is loaded into TAUDnCNTm every PCLK clock, meaning that TAUDnCNTm is always 0000<sub>H</sub>.
  - INTTAUDnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
- TAUDTTOUTm is toggled every PCLK clock.

## (3) Operation stop and restart

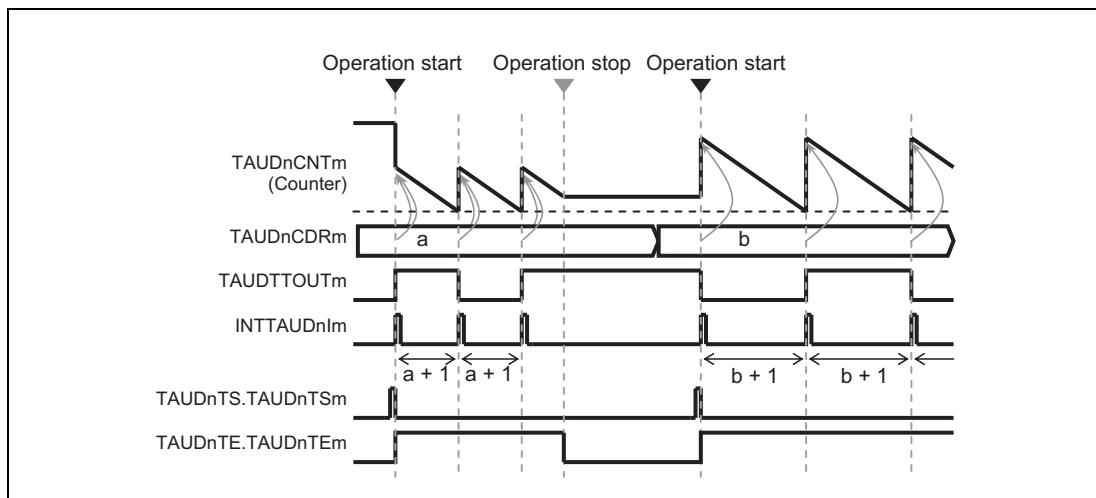


Figure 23.32 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEM to 0.
- TAUDnCNTm and TAUDTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

## (4) Forced restart (TAUDnCMORm.TAUDnMD0 = 1)

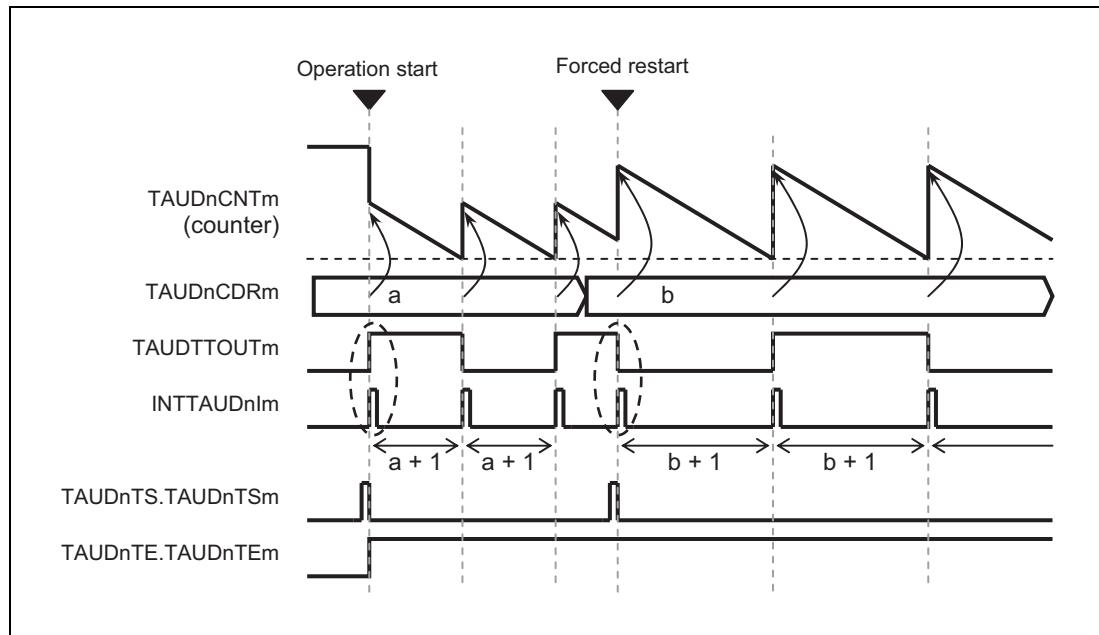
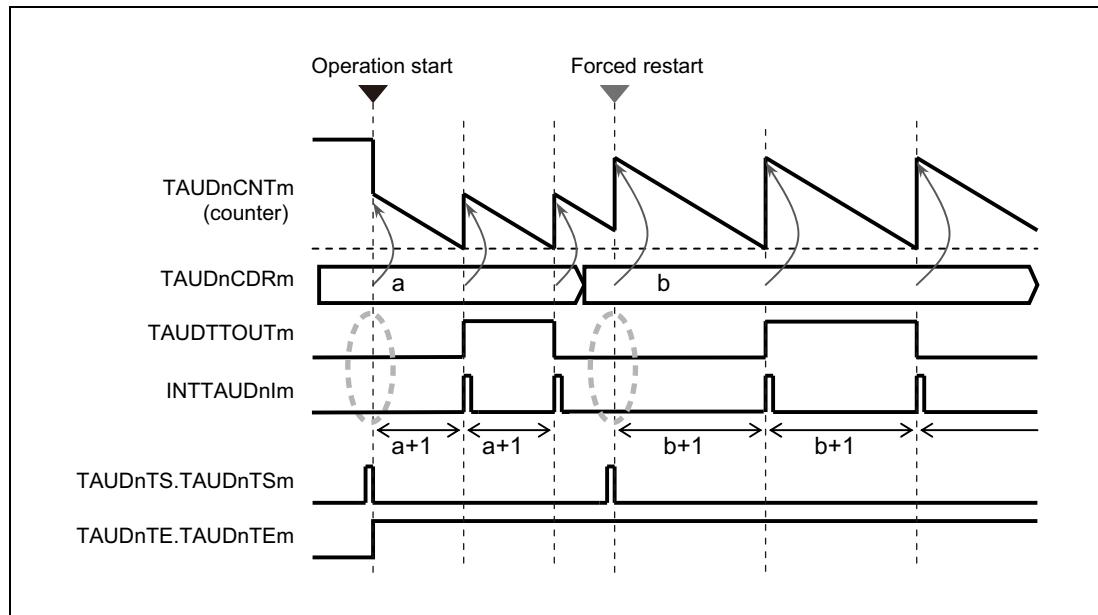


Figure 23.33 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUDnCDRm value is reflected to TAUDnCNTm and counting starts. Execute a forced restart to reflect the changed TAUDnCDRm value immediately.
- When a forced restart is made, an interrupt (INTTAUDnIm) is generated and TAUDTTOUTm is inverted.

**(5) Forced restart (TAUDnCMORm.TAUDnMD0 = 0)****Figure 23.34 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 0)**

- When a forced restart is made, an interrupt (INTTAUDnIm) is not generated and TAUDTTOUTm is not inverted.

## 23.12.2 TAUDTTINm Input Interval Timer Function

### 23.12.2.1 Overview

#### Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals or when a valid TAUDTTINm input edge is detected. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

#### Prerequisites

- The operating mode should be set to interval timer mode. See **Table 23.53, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function**.
- The channel output mode should be set to independent channel output mode 1. See **Section 23.7, Channel Output Modes**.

#### Functional description

This function operates in an identical manner to the interval timer function (see **Section 23.12.1, Interval Timer Function**) except that this function is restarted by a valid TAUDTTINm input edge. The type of edge used as a trigger is specified using the TAUDnCMURm.TAUDnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

### 23.12.2.2 Equations

$$\text{INTTAUDnIm cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1)$$

$$\text{TAUDTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm} + 1) \times 2$$

### 23.12.2.3 Block Diagram and General Timing Diagram

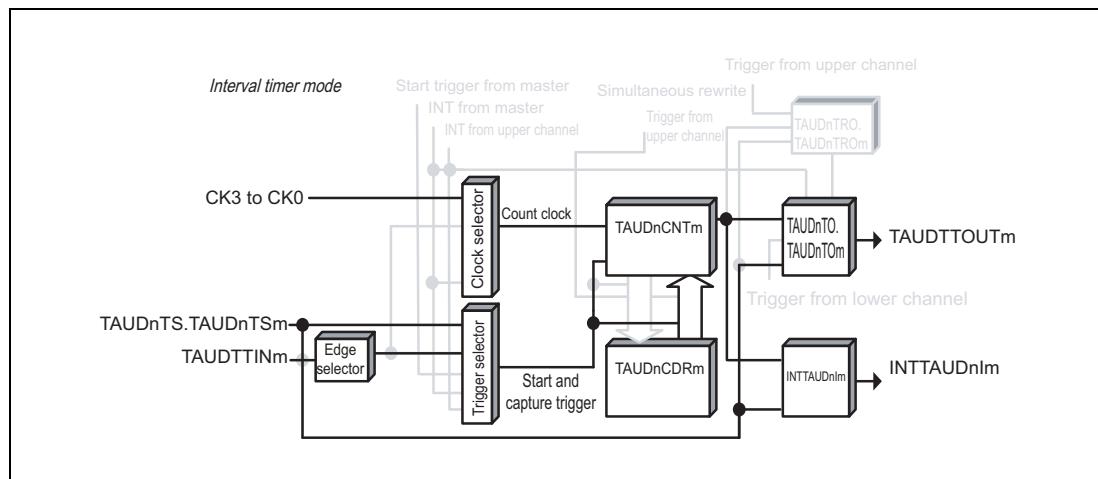


Figure 23.35 Block Diagram of TAUDTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).
- Rising edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>)

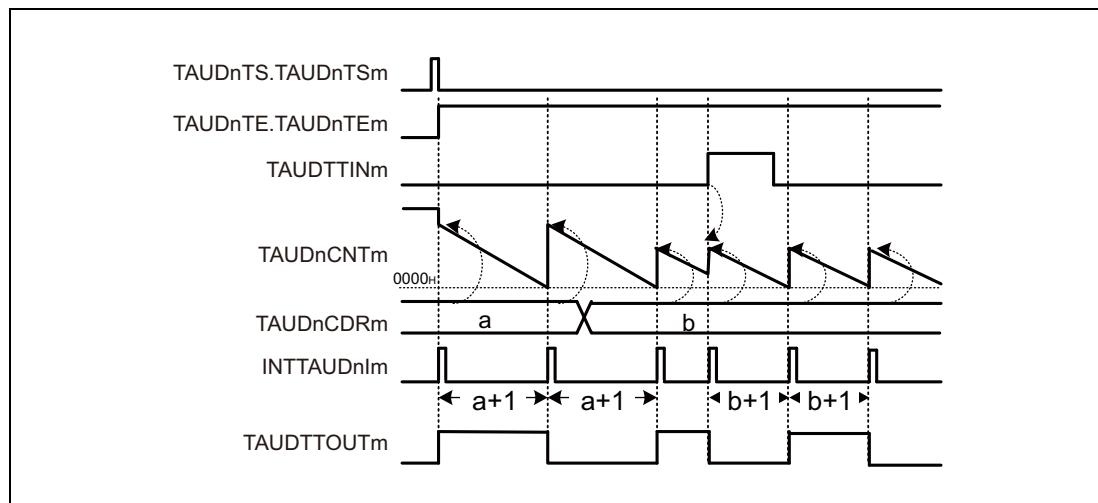


Figure 23.36 General Timing Diagram of TAUDTTINm Input Interval Timer Function

### 23.12.2.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]	TAUDnCCS [1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS [1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.53 Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W

Table 23.54 Contents of the TAUDnCMURm Register for TAUDTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges

### (3) Channel output mode

Table 23.55 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMM	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMM = 0)
TAUDnTOL.TAUDnTOLM	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMM	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLM	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set these bits to 0
TAUDnTRC.TAUDnTRCM	
TAUDnTME.TAUDnTMEM	0: Disables modulation

#### NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Section 23.7, Channel Output Modes**.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 23.56 Simultaneous Rewrite Settings for TAUDTTINm Input Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSM	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMM	
TAUDnRDC.TAUDnRDCM	

### 23.12.2.5 Operating Procedure for TAUDTTINm Input Interval Timer Function

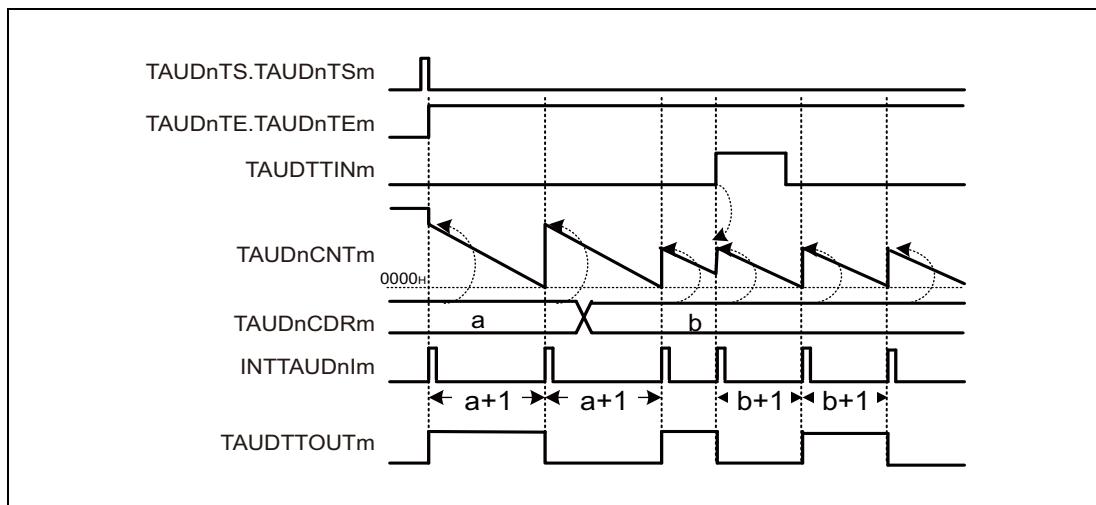
Table 23.57 Operating Procedure for TAUDTTINm Input Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.53, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function</b>, and <b>Table 23.54, Contents of the TAUDnCMURm Register for TAUDTTINm Input Interval Timer Function</b>.</p> <p>Set the value of TAUDnCDRm register.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.55, Control Bit Settings in Independent Channel Output Mode 1</b>.</p>	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnlm is generated and TAUDTTOUTm toggles.
During Operation	<p>The values of the TAUDnCMURm.TAUDnTIS[1:0] and the TAUDnCDRm register are changeable at any time. The TAUDnCNTm register can be read at all times.</p> <p>Detection of TAUDTTINm edge</p>	<p>TAUDnCNTm counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues.</li> <li>INTTAUDnlm is generated and TAUDTTOUTm toggles.</li> </ul> <p>When a TAUDTTINm input valid edge is detected during count operation, the TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. Afterwards, this procedure is repeated.</p>
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.



### 23.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 23.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUDTTINm input edge.



**Figure 23.37 Counter Triggered by Rising TAUDTTINm Input Edge**  
 $(TAUDnCMURm.TAUDnTIS[1:0] = 01_B)$ ,  $TAUDnCMORm.TAUDnMD0 = 1$

- If a valid TAUDTTINm input edge is detected, an interrupt is generated which causes TAUDTTOUTm to toggle. In this example, the valid edge is a rising edge ( $TAUDnCMURm.TAUDnTIS[1:0] = 01_B$ )

### 23.12.3 Clock Divide Function

#### 23.12.3.1 Overview

##### Summary

This function is used as a frequency divider. The frequency of the input signal TAUDTTINm is divided by a factor related to TAUDnCDRm, and the resulting signal is output to TAUDTTOUTm.

##### Prerequisites

- TAUDTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See **Table 23.58, Contents of the TAUDnCMORm Register for Clock Divide Function.**)
- The channel output mode should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes.**)

##### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value, using TAUDTTINm as a count clock.

When the counter value reaches 0000<sub>H</sub>, INTTAUDnIm occurs and TAUDTTOUTm signal is toggled. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The value of TAUDnCDRm can be rewritten at any time. The changed value of TAUDnCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEM = 0. TAUDnCNTm and TAUDTTOUTm stop but retain their values. The function can be restarted by setting TAUDnTS.TAUDnTSm = 1. The counter can also be forcibly restarted without making a stop by setting TAUDnTS.TAUDnTSm = 1 during operation (forced restart).

##### Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDTTOUTm does not toggle. This results in a negative TAUDTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details, see **Section 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

##### NOTE

TAUDTTINm input signals are sampled at the frequency of the operation clock set by TAUDnCMORm.TAUDnCKS[1:0] bits. Therefore, the TAUDTTOUTm output clock cycle has an error of  $\pm 1$  operation clock cycle.

### 23.12.3.2 Equations

- When rising edge detection is selected:  
 $\text{TAUDTTOUTm frequency} = \text{TAUDTTINm frequency}/[(\text{TAUDnCDRm} + 1) \times 2]$
- When falling edge detection is selected:  
 $\text{TAUDTTOUTm frequency} = \text{TAUDTTINm frequency}/[(\text{TAUDnCDRm} + 1) \times 2]$
- When falling and rising edge detection is selected:  
 $\text{TAUDTTOUTm frequency} = \text{TAUDTTINm frequency}/(\text{TAUDnCDRm} + 1)$

### 23.12.3.3 Block Diagram and General Timing Diagram

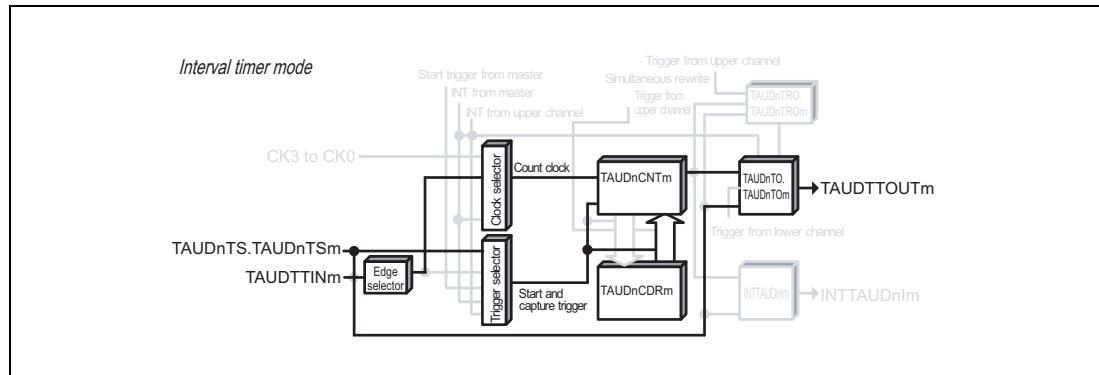


Figure 23.38 Block Diagram of Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. ( $\text{TAUDnCMORm.TAUDnMD0} = 1$ )
- Detection of rising edge ( $\text{TAUDnCMURm.TAUDnTIS}[1:0] = 01_B$ )

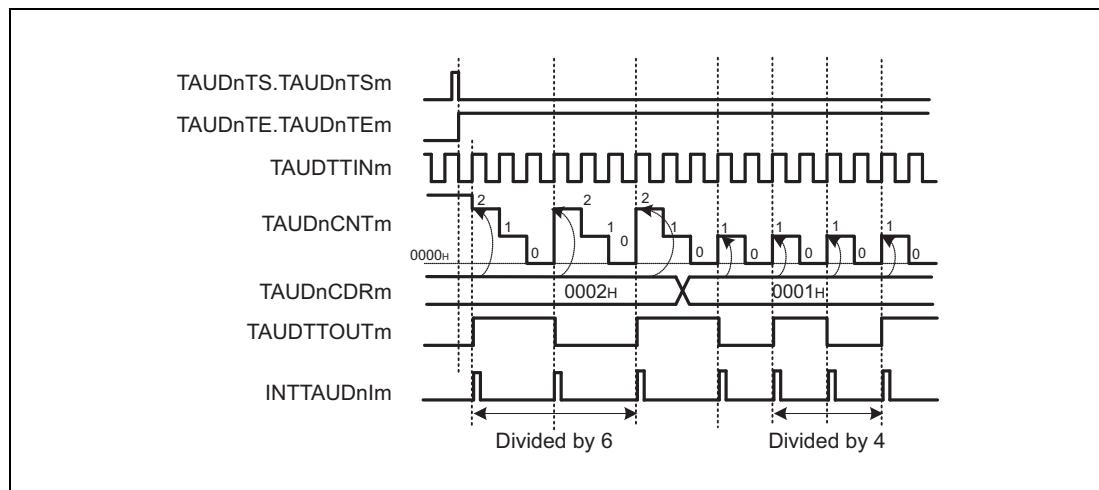


Figure 23.39 General Timing Diagram of Clock Divide Function

### 23.12.3.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]	TAUDnCCS [1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS [1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.58 Contents of the TAUDnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.59 Contents of the TAUDnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges

### (3) Channel output mode

**Table 23.60 Control Bit Settings in Independent Channel Output Mode 1**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set these bits to 0
TAUDnTRC.TAUDnTRCM	
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the clock divide function. Therefore, these registers should be set to 0.

**Table 23.61 Simultaneous Rewrite Settings for Clock Divide Function**

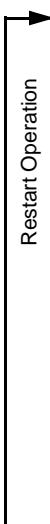
Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.12.3.5 Operating Procedure for Clock Divide Function

Table 23.62 Operating Procedure for Clock Divide Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.58, Contents of the TAUDnCMORm Register for Clock Divide Function</b>, and <b>Table 23.59, Contents of the TAUDnCMURm Register for Clock Divide Function</b>.</p> <p>Set the value of TAUDnCDRm register.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.60, Control Bit Settings in Independent Channel Output Mode 1</b>.</p>	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value. If TAUDnCMORm.TAUDnMDO is set to 1, INTTAUDnIm is generated and TAUDTTOUTm is toggled.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down each time TAUDTTINm input edge is detected. When the counter reaches 0000H: <ul style="list-style-type: none"> <li>• TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues.</li> <li>• INTTAUDnIm is generated.</li> <li>• TAUDTTOUTm is toggled.</li> </ul> Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDTTOUTm retain their current values.

Restart Operation



### 23.12.3.6 Specific Timing Diagrams

#### (1) TAUDnCDRm = 0000<sub>H</sub>

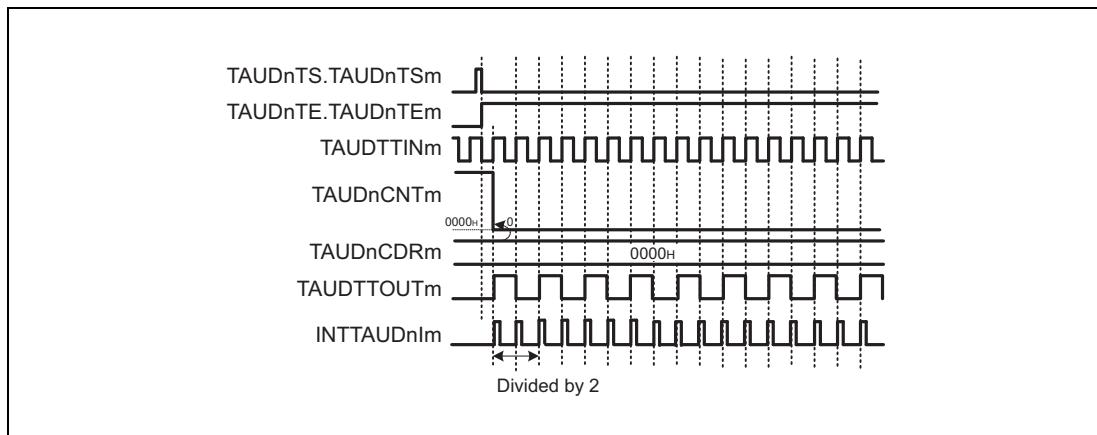


Figure 23.40 TAUDnCDRm = 0000<sub>H</sub>, TAUDnCMORm.TAUDnMD0 = 1,  
TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>

- If TAUDnCDRm is 0000<sub>H</sub>, TAUDnCNTm is always 0000<sub>H</sub>.
- INTTAUDnIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

**Figure 23.40** shows an operation timing example. Actually, there is a delay from TINm detection until TOUTm output because of the delay time of a noise filter or synchronization circuit placed between the TAUDnIm pin and TAUDn.

#### (2) Operation restart

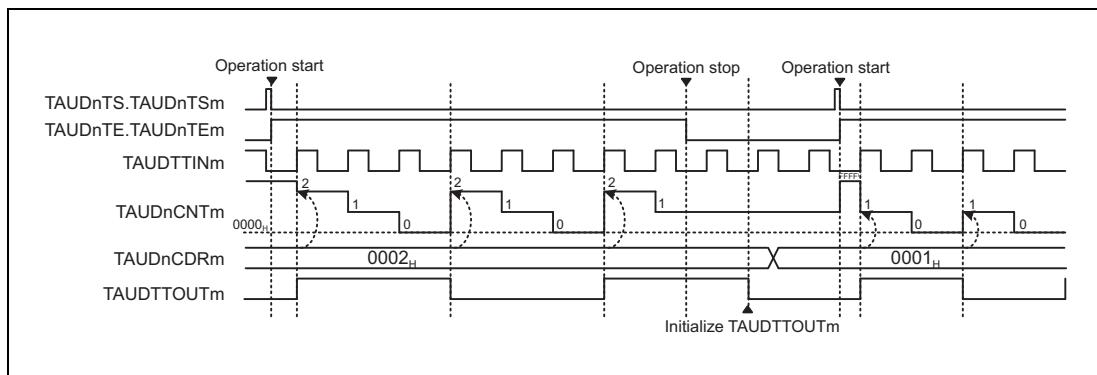


Figure 23.41 Operation Restart  
(TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>)

To reset the value of TAUDTTOUTm:

- Set TAUDnTOE.TAUDnTOEm = 0 when the counter is stopped (TAUDnTE.TAUDnTEM = 0).
- Then, write either 0 or 1 to TAUDnTO.TAUDnTOm to set the new start value of TAUDTTOUTm.

## (3) Forced restart

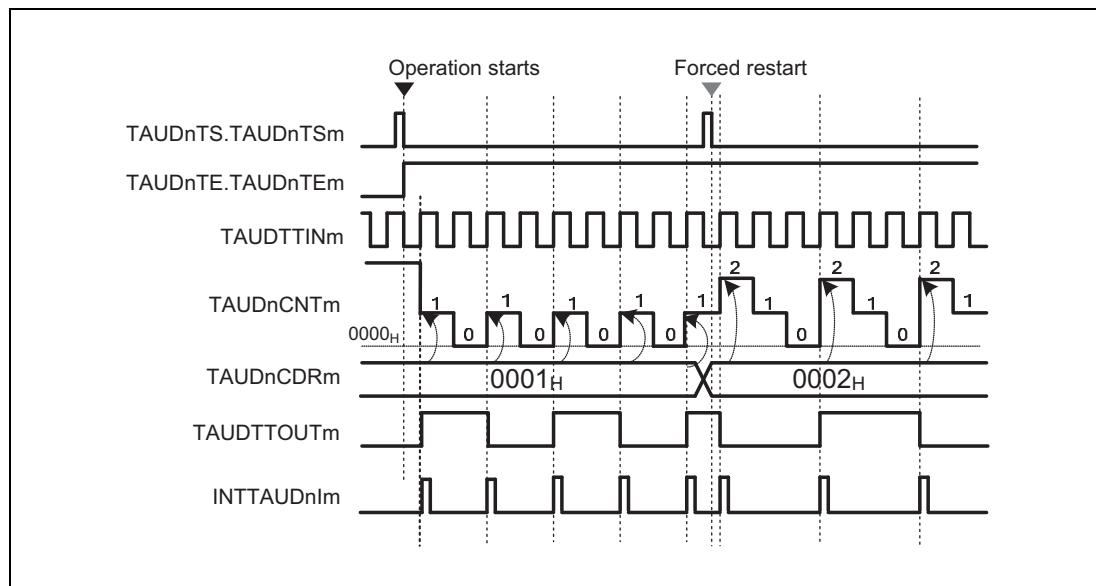


Figure 23.42 Forced Restart Operation  
 $(\text{TAUDnCMORm.TAUDnMD0} = 1, \text{TAUDnCMURm.TAUDnTIS}[1:0] = 01_B)$

- The counter can be forcibly restarted (without stopping it first) by setting  $\text{TAUDnTS.TAUDnTSm} = 1$  during operation.
- The value of  $\text{TAUDnCDRm}$  is written to  $\text{TAUDnCNTm}$  and the count operation restarts.
- $\text{TAUDTTOUTm}$  restarts at the same level as before the forced restart.

## 23.12.4 External Event Count Function

### 23.12.4.1 Overview

#### Summary

This function is used as an event timer, which generates an interrupt (INTTAUDnIm) when a specific number of valid TAUDTTINm input edges are detected.

#### Prerequisites

- The operating mode should be set to the event count mode. (See **Table 23.63, Contents of the TAUDnCMORm Register for External Event Count Function.**)
- TAUDTOUTm is not used with this function.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. When the counter starts, the current value of TAUDnCDRm is loaded into TAUDnCNTm.

When a valid TAUDTTINm input edge is detected, the value of TAUDnCNTm decrements by 1. TAUDnCNTm retains this value until a valid TAUDTTINm input edge is detected or the counter is restarted.

When the valid edge is detected for the (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEM to 0. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUDnTS.TAUDnTSm to 1 during operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

#### Conditions

An edge type used as a trigger is specified by TAUDnCMURm.TAUDnTIS[1:0] bits.

- When TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>, falling edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>, rising edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 10<sub>B</sub>, both edges are counted.

### 23.12.4.2 Equations

Number of valid edges detected before INTTAUDnIm generation = TAUDnCDRm + 1

### 23.12.4.3 Block Diagram and General Timing Diagram

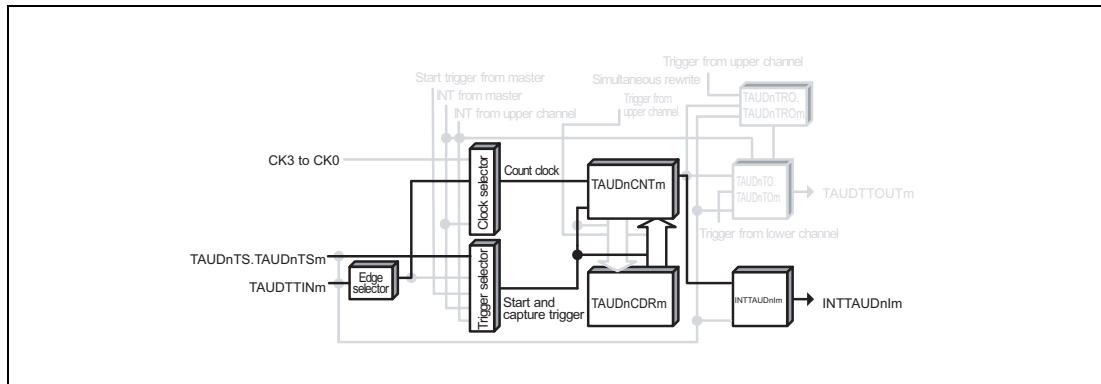


Figure 23.43 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge ( $\text{TAUDnCMURm.TAUDnTIS[1:0]} = 01_B$ )

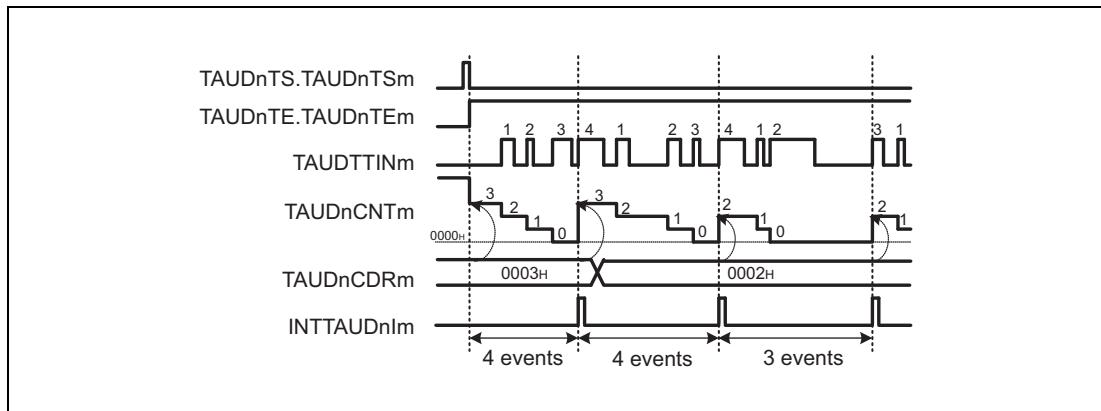


Figure 23.44 General Timing Diagram of External Event Count Function

### 23.12.4.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]	—	TAUDnMD[4:1]			TAUDnMDO		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.63 Contents of the TAUDnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—		—		—		—	
Value after reset	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W

Table 23.64 Contents of the TAUDnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected.

#### (3) Channel output mode

The channel output mode is not used by this function.

#### (4) Simultaneous rewrite

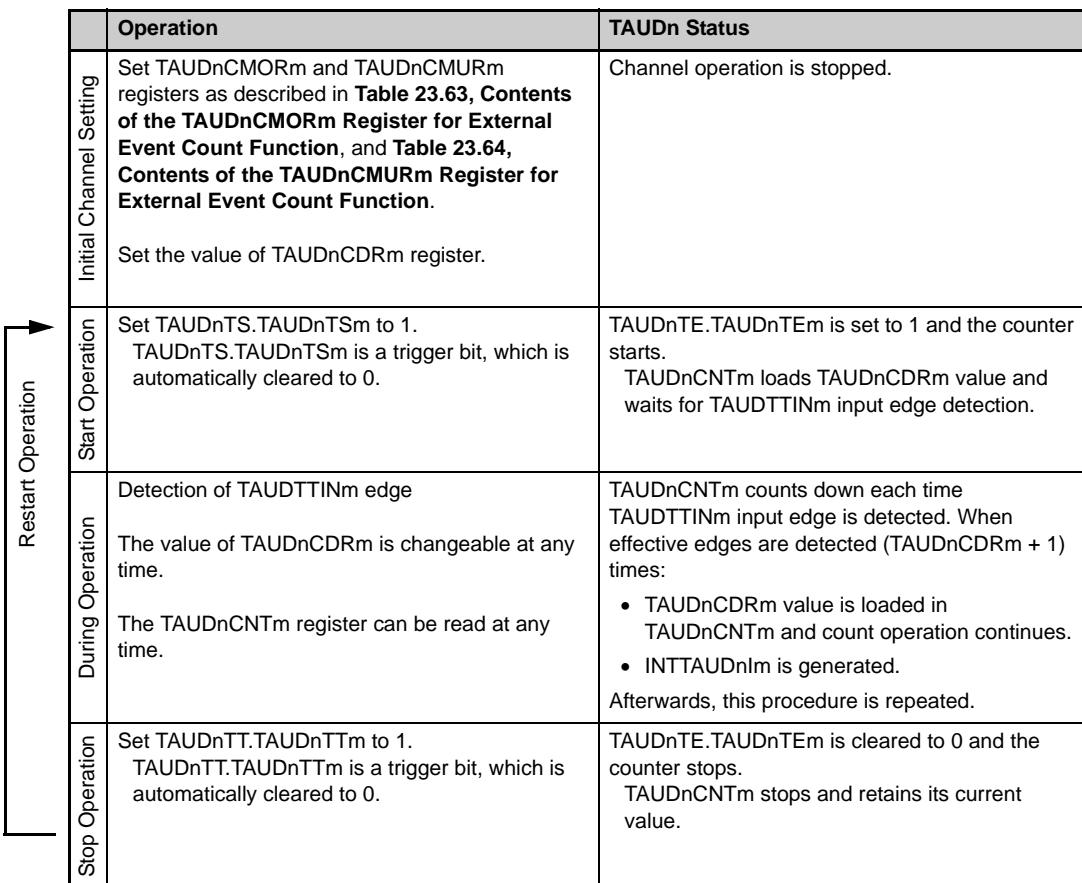
Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the external event count function. Therefore, these registers should be set to 0.

**Table 23.65 Simultaneous Rewrite Settings for External Event Count Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.4.5 Operating Procedure for External Event Count Function

**Table 23.66 Operating Procedure for External Event Count Function**



	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.63, Contents of the TAUDnCMORm Register for External Event Count Function</b> , and <b>Table 23.64, Contents of the TAUDnCMURm Register for External Event Count Function</b> .  Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value and waits for TAUDTTINm input edge detection.
During Operation	Detection of TAUDTTINm edge  The value of TAUDnCDRm is changeable at any time.  The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down each time TAUDTTINm input edge is detected. When effective edges are detected (TAUDnCDRm + 1) times: <ul style="list-style-type: none"> <li>• TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues.</li> <li>• INTTAUDnlm is generated.</li> </ul> Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

### 23.12.4.6 Specific Timing Diagrams

#### (1) TAUDnCDRm = 0000<sub>H</sub>

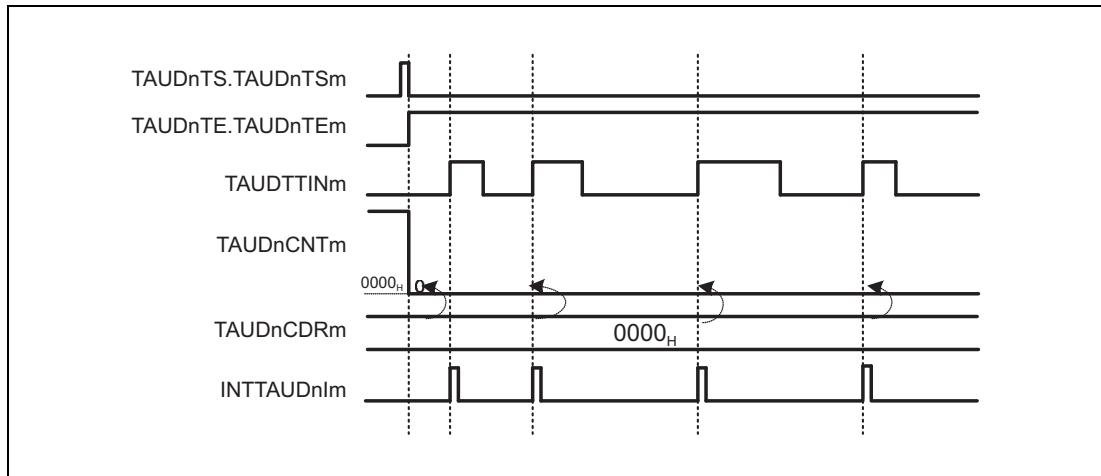


Figure 23.45 TAUDnCDRm = 0000<sub>H</sub>, TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>

- If 0000<sub>H</sub> = TAUDnCDRm, 0000<sub>H</sub> is loaded into TAUDnCNTm each time a valid TAUDTTINm input edge is detected.  
In other words, INTTAUDnIm is generated each time a valid TAUDTTINm input edge is detected.

#### (2) Operation stop and restart

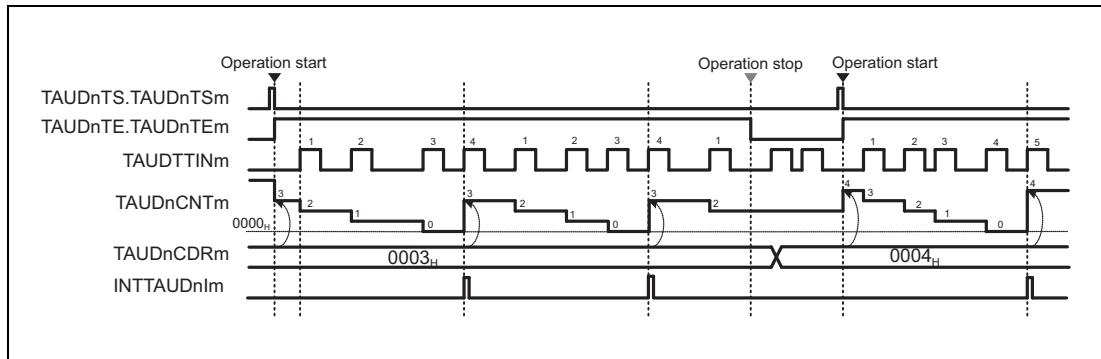


Figure 23.46 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEM to 0.
- TAUDnCNTm stops and retains its current value. TAUDTTINm continues and TAUDnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm loads the TAUDnCDRm value and restarts count operation.

### (3) Forced restart

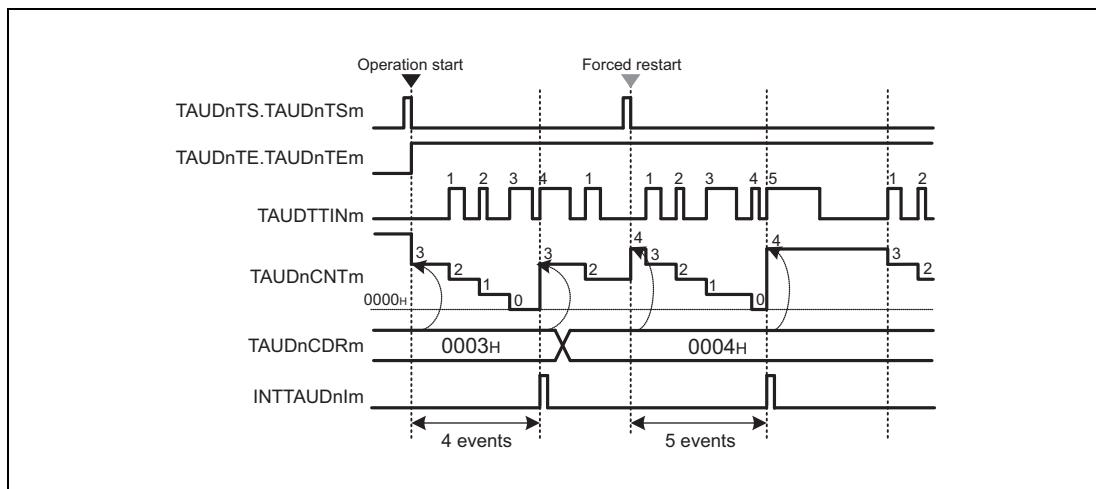


Figure 23.47 Forced Restart Operation (TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>)

Once a forced restart is made, the changed TAUDnCDRm value is applied to TAUDnCNTm immediately.

- The counter can be restarted without making a stop by setting TAUDnTS.TAUDnTSm to 1 during operation.
- The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter awaits the next valid TAUDTTINm input edge.

## 23.12.5 Delay Count Function

### 23.12.5.1 Overview

#### Summary

This function generates interrupts (INTTAUDnIm), which have a defined delay to the TAUDTTINm input signal. TAUDTTINm input signal pulses that occur within the delay period are ignored.

#### Prerequisites

- The operating mode should be set to one-count mode. See **Table 23.67, Contents of the TAUDnCMORm Register for Delay Count Function**.
- TAUDTOUTm is not used with this function.
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches  $0000_H$ , an interrupt is generated. The counter returns to  $FFFF_H$  and awaits the next valid TAUDTTINm input edge.

When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

#### Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] =  $00_B$ , falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] =  $01_B$ , rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] =  $10_B$ , rising and falling edges trigger the counter.

### 23.12.5.2 Equations

Delay between TAUDTTINm and INTTAUDnIm = count clock cycle  $\times$  (TAUDnCDRm + 1)

### 23.12.5.3 Block Diagram and General Timing Diagram

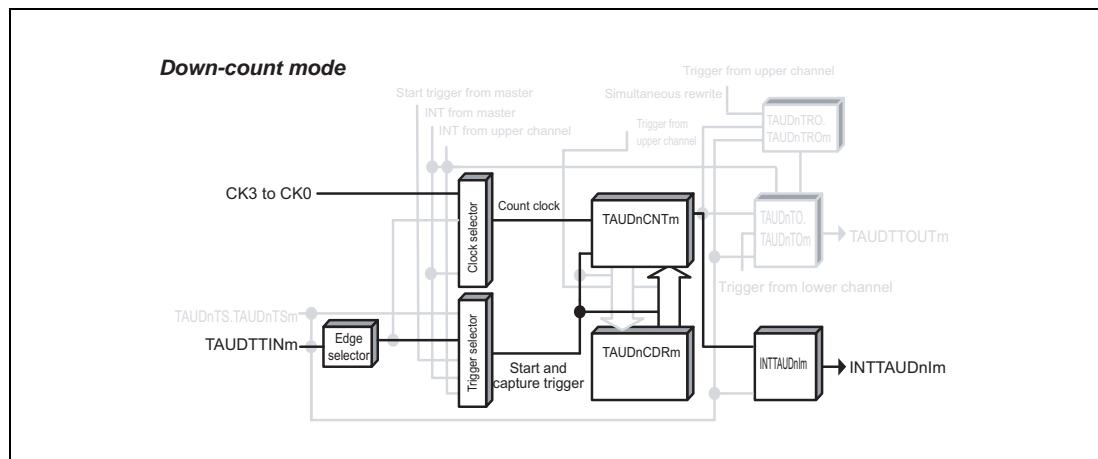


Figure 23.48 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

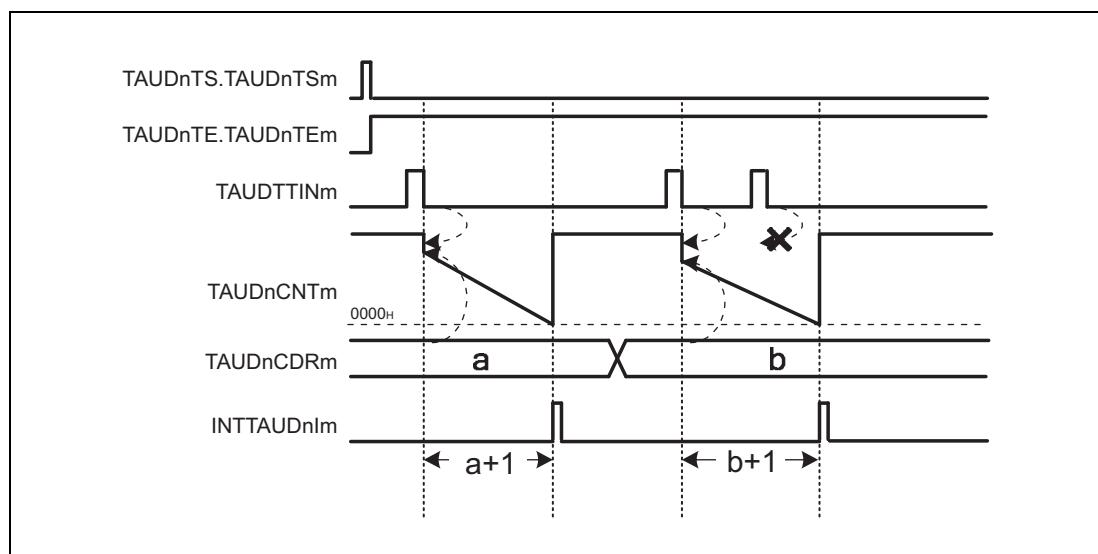


Figure 23.49 General Timing Diagram of Delay Count Function

### 23.12.5.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.67 Contents of the TAUDnCMORm Register for Delay Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	0: Disables a start trigger during operation

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.68 Contents of the TAUDnCMURm Register for Delay Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

#### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

#### (4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the delay count function. Therefore, these registers should be set to 0.

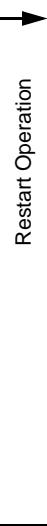
**Table 23.69 Simultaneous Rewrite Settings for Delay Count Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.5.5 Operating Procedure for Delay Count Function

**Table 23.70 Operating Procedure for Delay Count Function**

Operation	TAUDn Status	
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.67, Contents of the TAUDnCMORm Register for Delay Count Function</b> , and <b>Table 23.68, Contents of the TAUDnCMURm Register for Delay Count Function</b> .  Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.  Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEM is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.  When a start edge is detected, the TAUDnCDRm value is loaded in TAUDnCNTm.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000 <sub>H</sub> , INTTAUDnIm is generated. TAUDnCNTm stops counting, returns FFFF <sub>H</sub> , and waits for a trigger.  If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its value.



## 23.12.6 One-Pulse Output Function

### 23.12.6.1 Overview

#### Summary

This function generates an interrupt (INTTAUDnIm) when a valid TAUDTTINm input edge is detected and at a defined interval afterward. TAUDTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

#### Prerequisites

- The operation mode should be set to pulse one-count mode. (See **Table 23.71, Contents of the TAUDnCMORm Register for One-Pulse Output Function.**)
- The channel output mode should be set to independent channel output mode 2. (See **Section 23.7, Channel Output Modes.**)
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. An interrupt is generated and TAUDTTOUTm is set to active level.

When the counter reaches  $0001_H$ , an interrupt is generated and TAUDTTOUTm is set to the inactive level. The counter stops at  $0000_H$  and awaits the next valid TAUDTTINm input edge.

When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

#### Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] =  $00_B$ , falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] =  $01_B$ , rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] =  $10_B$ , rising and falling edges trigger the counter.

### 23.12.6.2 Equations

Interval between TAUDTTINm and INTTAUDnIm = TAUDTTOUTm (timer output) width = count clock cycle  $\times$  TAUDnCDRm

### 23.12.6.3 Block Diagram and General Timing Diagram

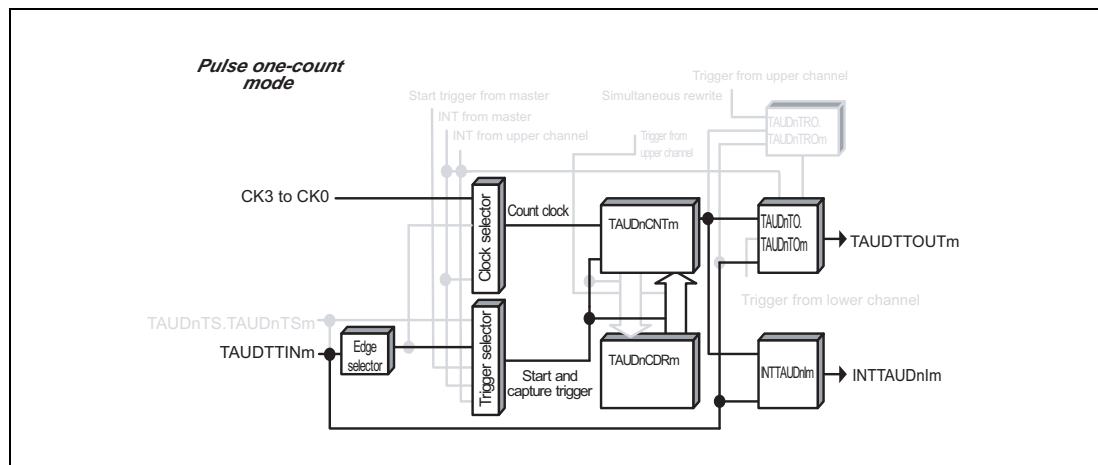


Figure 23.50 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

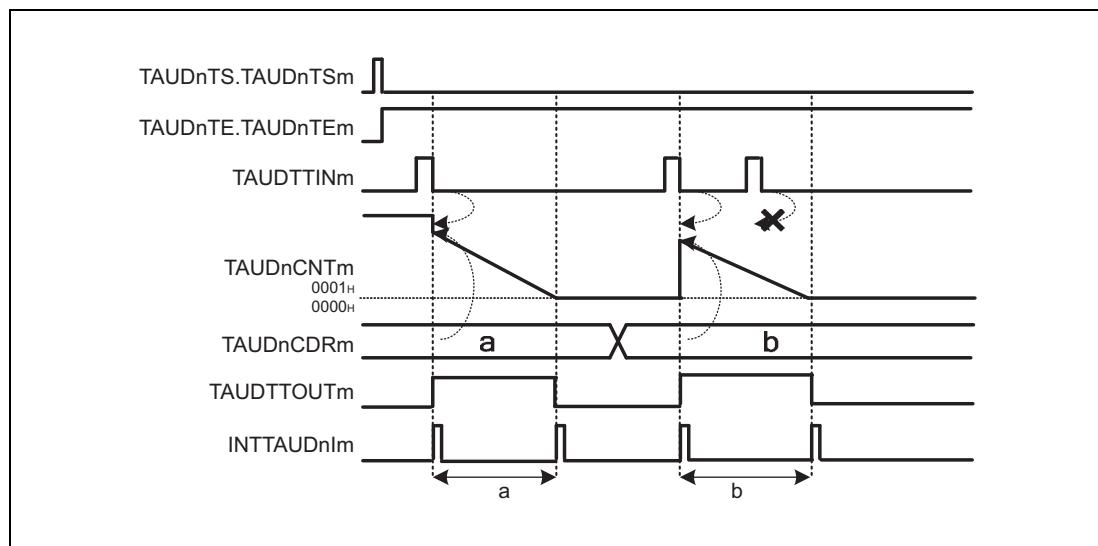


Figure 23.51 General Timing Diagram of One-Pulse Output Function

### 23.12.6.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.71 Contents of the TAUDnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMDO	0: Disables a start trigger during operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.72 Contents of the TAUDnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

### (3) Channel output mode

**Table 23.73 Control Bit Settings in Independent Channel Output Mode 2**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode controlled by software.
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEM	0: Disables modulation

#### NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTOUTm can then be controlled independently of the interrupts. For details, see **Table 23.47, Channel Output Modes**.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers should be set to 0.

**Table 23.74 Simultaneous Rewrite Settings for One-Pulse Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.12.6.5 Operating Procedure for One-Pulse Output Function

Table 23.75 Operating Procedure for One-Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.71, Contents of the TAUDnCMORm Register for One-Pulse Output Function</b>, and <b>Table 23.72, Contents of the TAUDnCMURm Register for One-Pulse Output Function</b>.</p> <p>Set the value of TAUDnCDRm register.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.73, Control Bit Settings in Independent Channel Output Mode 2</b>.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p> <p>Detection of TAUDTTINm start edge</p>	<p>TAUDnTE.TAUDnTEM is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.</p> <p>When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value.</p>
During Operation	<p>The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.</p>	<p>INTTAUDnIm is generated when TAUDnCNTm starts and TAUDTTOUTm is set to its active level. TAUDnCNTm counts down. When the counter reaches <math>0001_H</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>TAUDTTOUTm is set to its inactive level.</li> </ul> <p>TAUDnCNTm stops counting and waits for a trigger.</p> <p>If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored.</p>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops.</p> <p>TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>



## 23.12.7 TAUDTTINm Input Pulse Interval Measurement Function

### 23.12.7.1 Overview

#### Summary

This function captures the count value and uses this value and the overflow bit TAUDnCSRm.TAUDnOVF to measure the interval of the TAUDTTINm input signals.

#### Prerequisites

- The operating mode should be set to capture mode. See **Table 23.77, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function.**
- TAUDTTOUTm is not used with this function.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The counter TAUDnCNTm starts to count up from  $0000_H$ . When a valid TAUDTTINm edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter resets to  $0000_H$  and subsequently continues operation.

If the counter reaches  $FFFF_H$  before a valid TAUDTTINm edge is detected, it overflows to  $0000_H$ . The counter is reset to  $0000_H$  and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

**Table 23.76 Effects of Overflow**

TAUDnCMORm. TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDTTINm Input is Detected	
	TAUDnCDRm	TAUDnCSRm. TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm. TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm set to 0, TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, the overflow bit (TAUDnCSRm.TAUDnOVF) can be cleared only by setting TAUDnCSCm.TAUDnCLOV = 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the interval of the TAUDTTINm signal. However, if an overflow occurs multiple times before a valid TAUDTTINm input is detected, the overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

The function can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEM = 0. TAUDnCNTm stops but retains its value. While the function is stopped, valid TAUDTTINm input edge detection and TAUDnCNTm capture are not performed.

The counter is reset to  $0000_H$  and subsequently continues operation.

### Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 23.9, TAUDTOUTm Output and INTTAUDnlm Generation when Counter Starts or Restarts.**

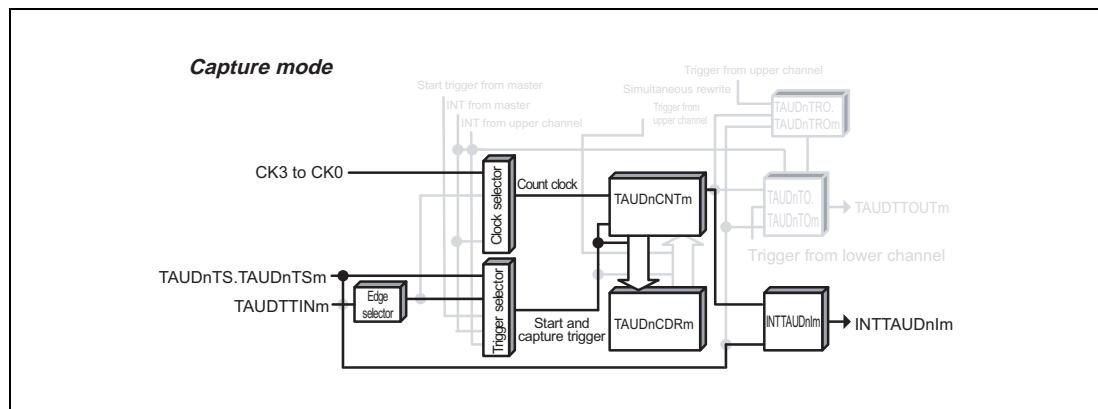
#### NOTE

When TAUDnCMORm.TAUDnCOS[1:0] = 10<sub>B</sub> or 11<sub>B</sub>, the value of TAUDnCNTm is not loaded into TAUDnCDRm when the first valid TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

### 23.12.7.2 Equations

$$\begin{aligned} \text{TAUDTTINm input pulse interval} &= \text{count clock cycle} \times [(\text{TAUDnCSRm.TAUDnOVF} \times (\text{FFFF}_H + 1)) \\ &+ \text{TAUDnCDRm capture value} + 1] \end{aligned}$$

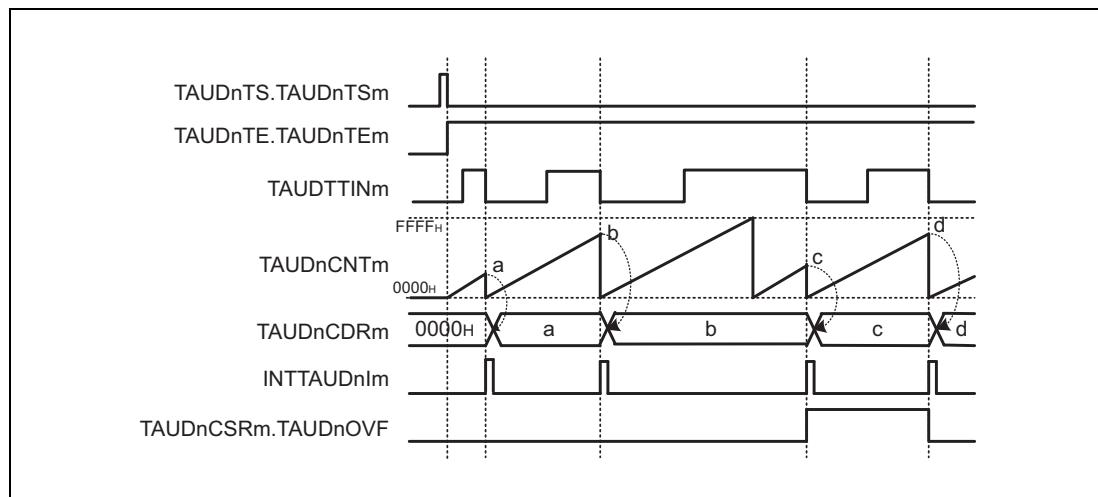
### 23.12.7.3 Block Diagram and General Timing Diagram



**Figure 23.52 Block Diagram of TAUDTTINm Input Pulse Interval Measurement Function**

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 0).
- Falling edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)
- When a valid TAUDTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRM.TAUDnOVF is set to 1 (TAUDnCMORm.TAUDnCOS[1:0] = 00<sub>B</sub>).



**Figure 23.53 General Timing Diagram of TAUDTTINm Input Pulse Interval Measurement Function**

### 23.12.7.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.77 Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as the external capture trigger.
7, 6	TAUDnCOS[1:0]	See Table 23.76, Effects of Overflow.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.78 Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges

### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input pulse interval measurement function. Therefore, these registers should be set to 0.

**Table 23.79 Simultaneous Rewrite Settings for TAUDTTINm Input Pulse Interval Measurement Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.12.7.5 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

**Table 23.80 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function**



	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.77, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function</b> , and <b>Table 23.78, Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Measurement Function</b> .  The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts.  TAUDnCNTm is cleared to 0000H. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	Detection of TAUDTTINm edge  The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time.  TAUDnCSCm.TAUDnCLOV can be written to 1. (TAUDnCSRm.TAUDnOVF bit is cleared to 0.)	TAUDnCNTm starts to count up from 0000H. When a TAUDTTINm valid edge is detected: <ul style="list-style-type: none"> <li>• TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and returns to 0000H.</li> <li>• INTTAUDnIm is then generated.</li> </ul> Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops.  TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

### 23.12.7.6 Specific Timing Diagrams: Overflow Operation

#### (1) TAUDnCMORm.TAUDnCOS[1:0] = 00<sub>B</sub>

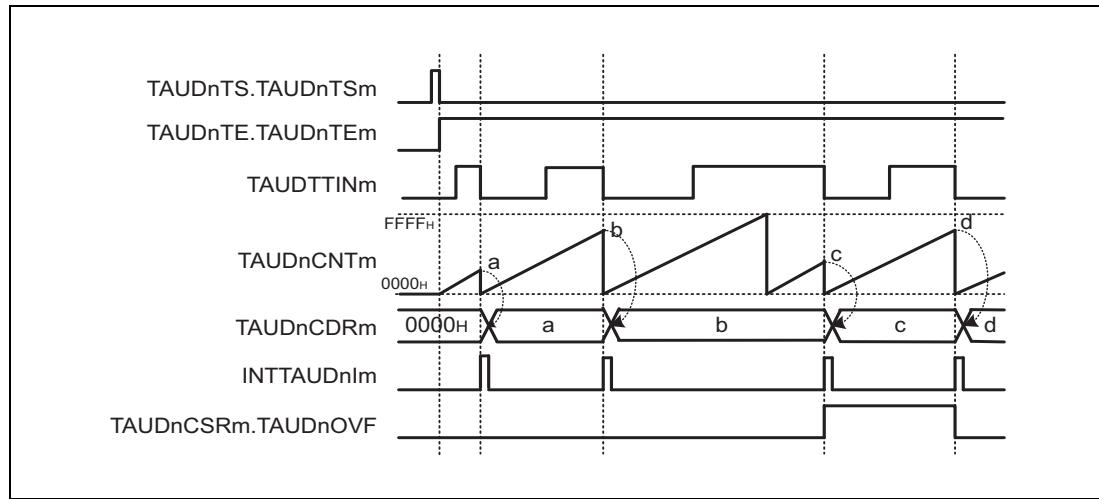


Figure 23.54 TAUDnCMORm.TAUDnCOS[1:0] = 00<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

#### (2) TAUDnCMORm.TAUDnCOS[1:0] = 01<sub>B</sub>

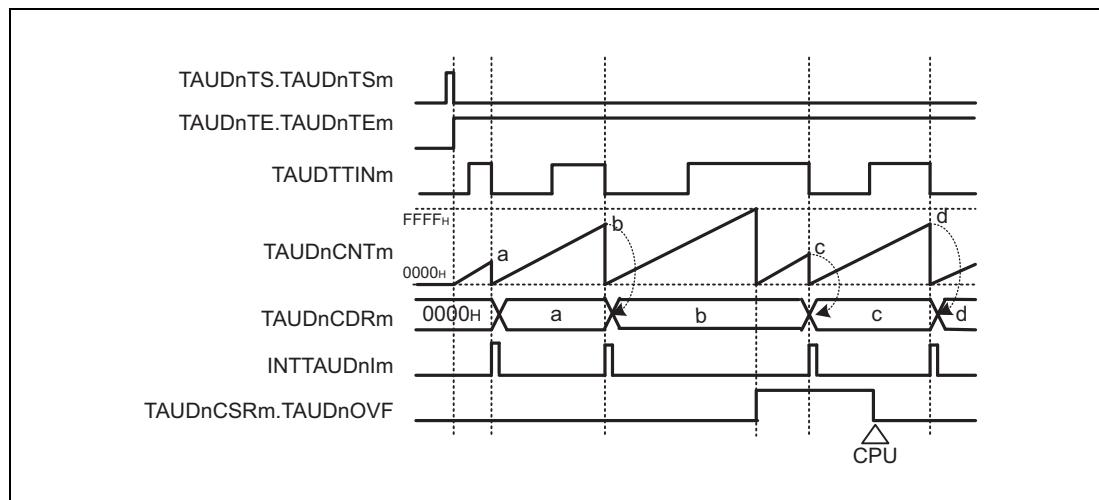


Figure 23.55 TAUDnCMORm.TAUDnCOS[1:0] = 01<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.

- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

**(3) TAUDnCMORm.TAUDnCOS[1:0] = 10<sub>B</sub>**

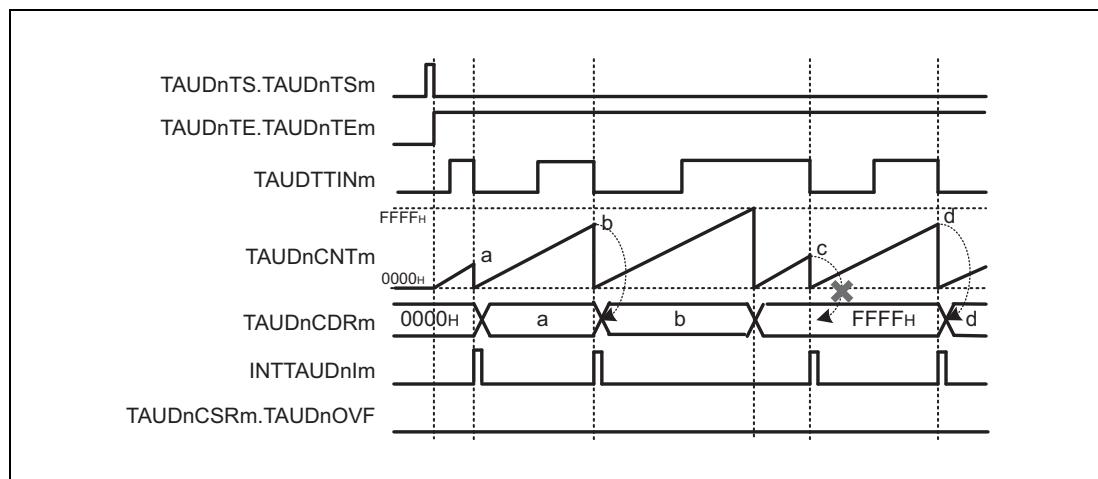


Figure 23.56 TAUDnCMORm.TAUDnCOS[1:0] = 10<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, TAUDnCDRm is set to FFFF<sub>H</sub> and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.

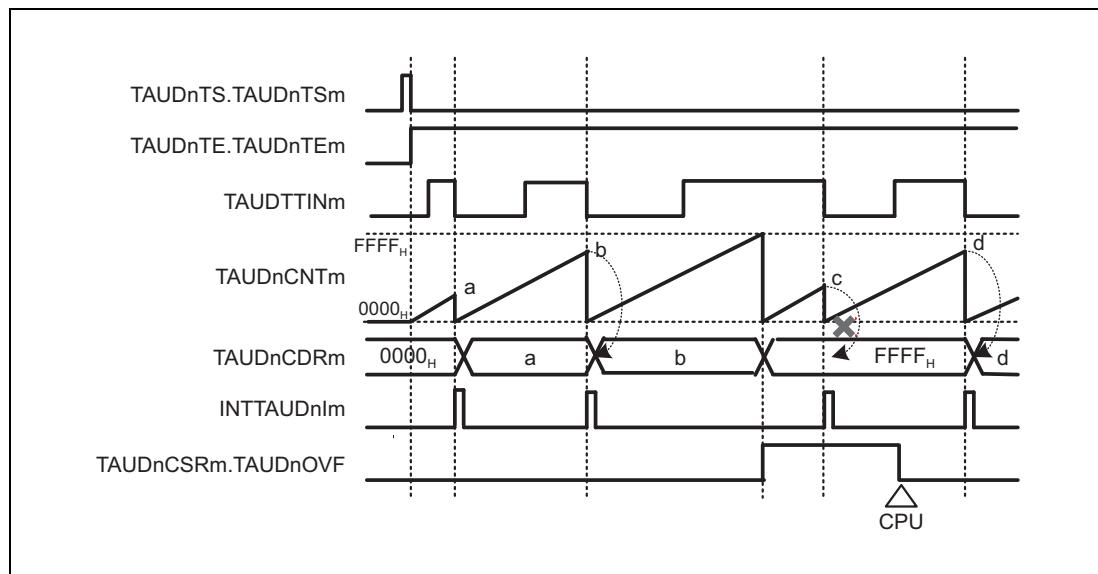
(4) TAUDnCMORm.TAUDnCOS[1:0] = 11<sub>B</sub>

Figure 23.57 TAUDnCMORm.TAUDnCOS[1:0] = 11<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, TAUDnCDRm is set to FFFF<sub>H</sub> and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

## 23.12.8 TAUDTTINm Input Signal Width Measurement Function

### 23.12.8.1 Overview

#### Summary

This function measures the width of a TAUDTTINm signal, by starting the count at one edge of TAUDTTINm and capturing the count value at the other edge.

#### Prerequisites

- The operating mode should be set to capture and one-count mode. See **Table 23.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function.**
- TAUDTTOUTm is not used with this function.
- TAUDnCMORm.TAUDnMD0 should be set to 0.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. When a valid TAUDTTINm start edge is detected, the counter TAUDnCNTm starts to count up from  $0000_H$ . When a valid TAUDTTINm stop edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter retains its value (TAUDnCDRm + 1) and awaits the next valid TAUDTTINm input start edge.

If the counter reaches  $FFFF_H$  before a valid TAUDTTINm stop edge is detected, it overflows. The counter is reset to  $0000_H$  and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

**Table 23.81 Effects of Overflow**

TAUDnCMORm. TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDTTINm Input Stop Edge is Detected	
	TAUDnCDRm	TAUDnCSRm. TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm. TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm stops counting TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, overflow bit TAUDnCSRm.TAUDnOVF can be cleared only by setting TAUDnCSCm.TAUDnCLOV to 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the width of the TAUDTTINm signal. However, if an overflow occurs multiple times before a valid TAUDTTINm input is detected, overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

This function cannot be forcibly restarted.

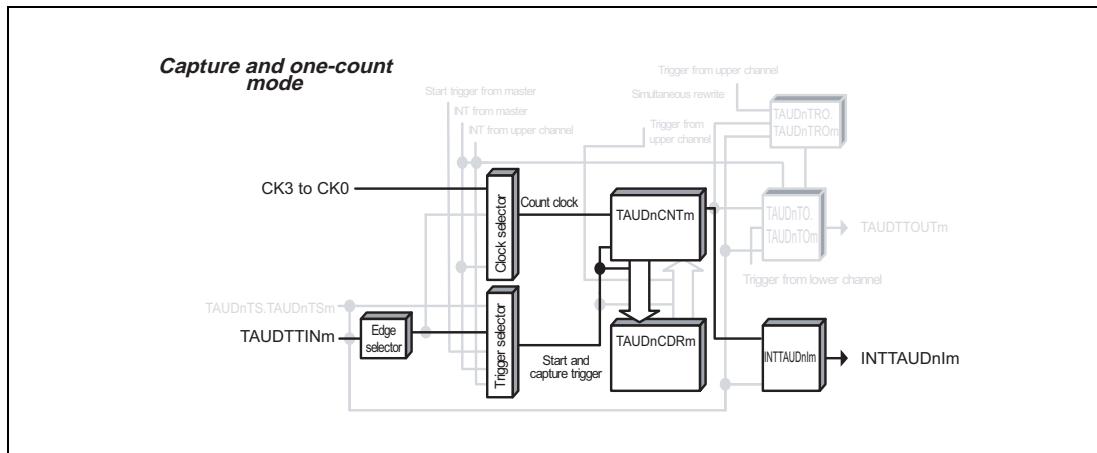
**NOTE**

When TAUDnCMORm.TAUDnCOS[1] = 1, the value of TAUDnCNTm is not loaded to TAUDnCDRm when the first valid TAUDTTInm input edge occurs after an overflow. However, an interrupt is generated.

### 23.12.8.2 Equations

$$\text{TAUDTTInm input signal width} = \text{count clock cycle} \times [(\text{TAUDnCSRm.TAUDnOVF} \times (\text{FFFF}_H + 1)) + \text{TAUDnCDRm capture value} + 1]$$

### 23.12.8.3 Block Diagram and General Timing Diagram



**Figure 23.58 Block Diagram of TAUDTTInm Input Signal Width Measurement Function**

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>)
- When a valid TAUDTTInm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1. (TAUDnCMORm.TAUDnCOS[1:0] = 00<sub>B</sub>)

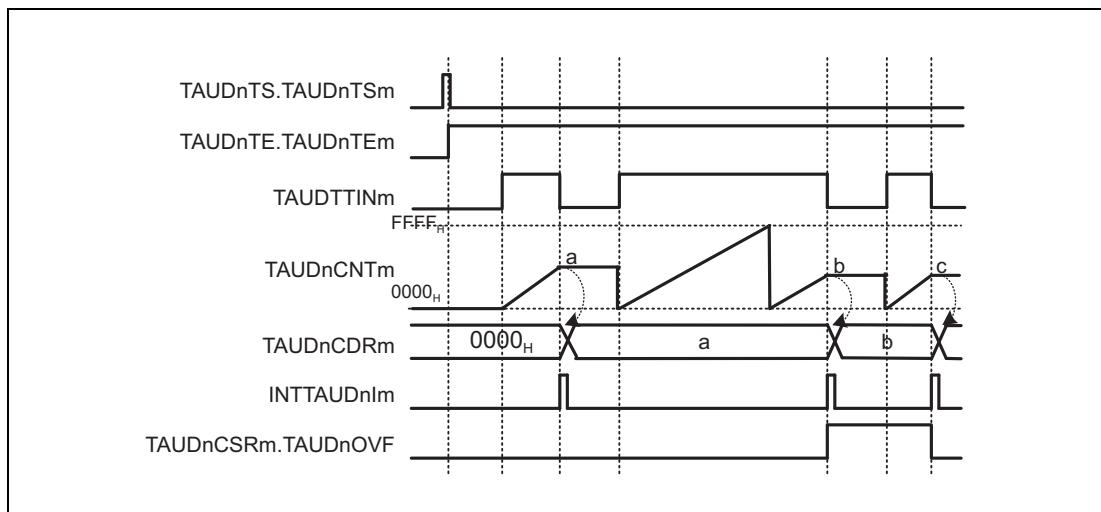


Figure 23.59 General Timing Diagram of TAUDTTINm Input Signal Width Measurement Function

### 23.12.8.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.82 Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	See Table 23.81, Effects of Overflow.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0110: Capture and one-count mode
0	TAUDnMDO	0: Disables the start trigger during operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.83 Contents of the TAUDnCMURm Register For TAUDTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input signal width measurement function. Therefore, these registers should be set to 0.

**Table 23.84 Simultaneous Rewrite Settings for TAUDTTINm Input Signal Width Measurement Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

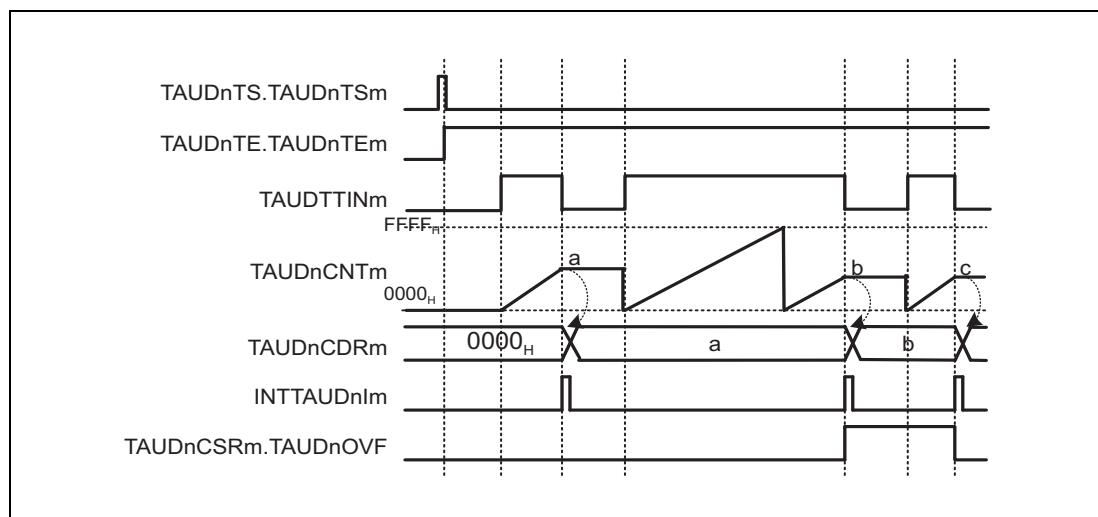
### 23.12.8.5 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function

**Table 23.85 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function**

Initial Channel Setting	Operation	TAUDn Status
	<p>Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function, and Table 23.83, Contents of the TAUDnCMURm Register For TAUDTTINm Input Signal Width Measurement Function.</b></p> <p>The TAUDnCDRm register functions as a capture register.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a TAUDTTINm start edge is detected, TAUDnCNTm starts to count up.</p>
During Operation	<p>TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV bit can be set to 1.</p>	<p>TAUDnCNTm starts to count up from 0000H. When TAUDTTINm valid edge is detected:</p> <ul style="list-style-type: none"> <li>• TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and retains its value. INTTAUDnIm is then generated.</li> </ul> <p>Counting stops at the “value that transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDTTINm start edge. Afterwards, this procedure is repeated.</p>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.</p>

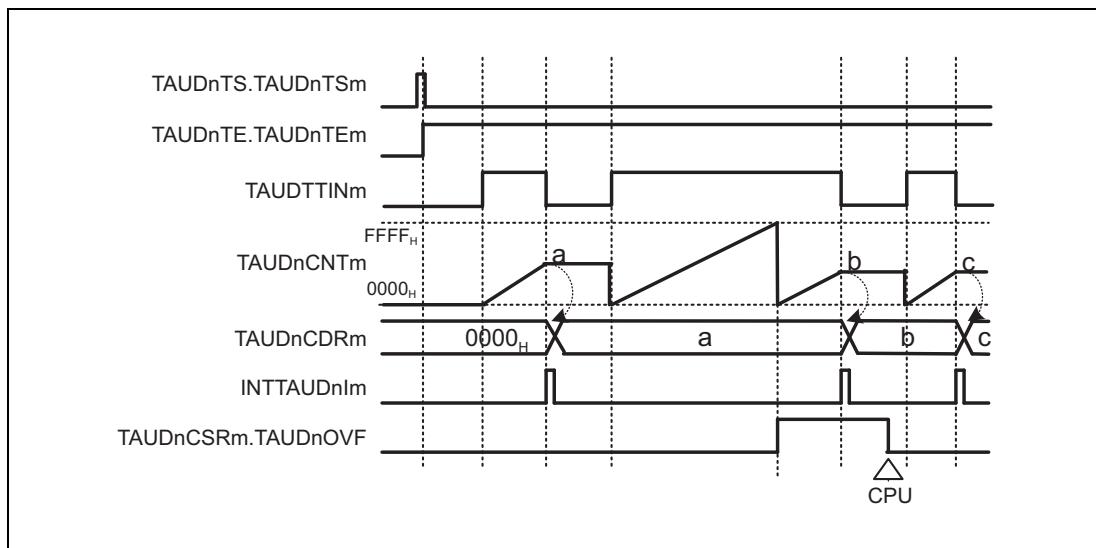
### 23.12.8.6 Specific Timing Diagrams: Overflow Operation

#### (1) TAUDnCMORm.TAUDnCOS[1:0] = 00<sub>B</sub>



**Figure 23.60** TAUDnCMORm.TAUDnCOS[1:0] = 00<sub>B</sub>, TAUDnCMORm.TAUDnMDO = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01<sub>B</sub>

**Figure 23.61** TAUDnCMORm.TAUDnCOS[1:0] = 01<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

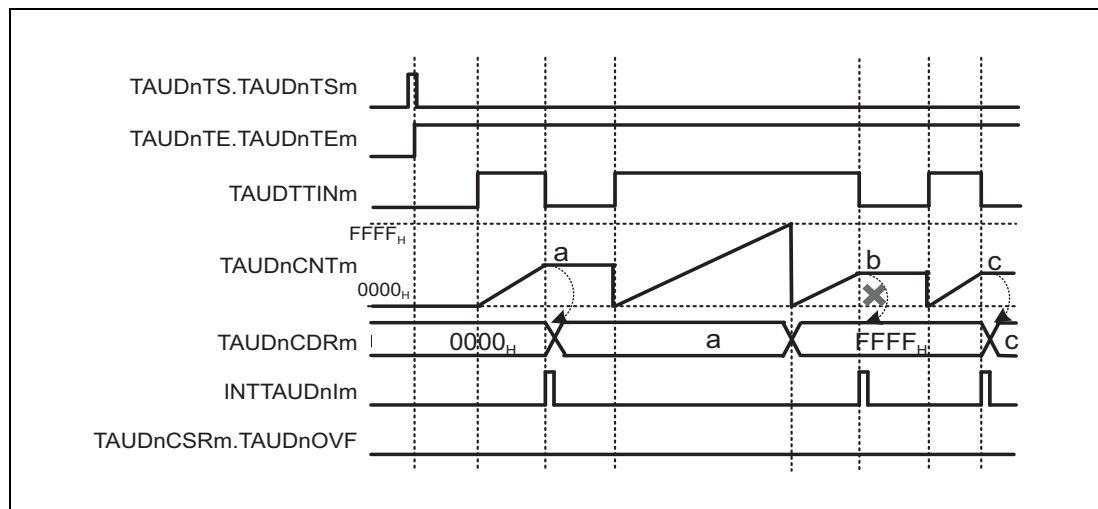
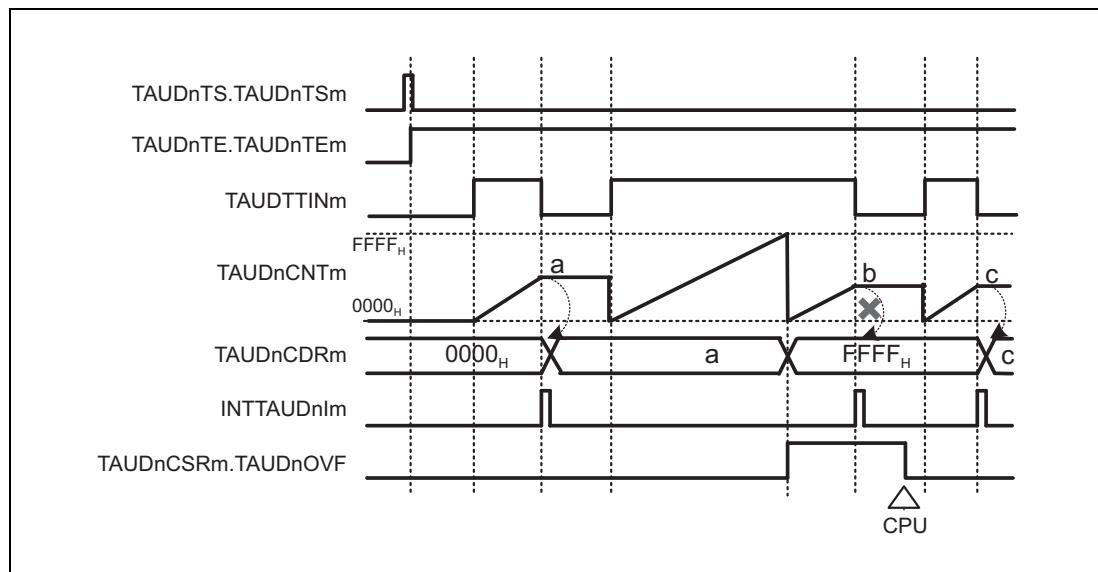
(3) TAUDnCMORm.TAUDnCOS[1:0] = 10<sub>B</sub>

Figure 23.62 TAUDnCMORm.TAUDnCOS[1:0] = 10<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, TAUDnCDRm is set to FFFF<sub>H</sub> and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11<sub>B</sub>

**Figure 23.63** TAUDnCMORm.TAUDnCOS[1:0] = 11<sub>B</sub>, TAUDnCMORm.TAUDnMD0 = 0,  
TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, TAUDnCDRm is set to FFFF<sub>H</sub> and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

## 23.12.9 TAUDTTINm Input Position Detection Function

### 23.12.9.1 Overview

#### Summary

This function measures the input signal duration by capturing the count value at the valid edge of TAUDTTINm.

#### Prerequisites

- The operating mode should be set to count capture mode. (See **Table 23.86, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function.**)
- TAUDTOUTm is not used with this function.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The counter starts counting from  $0000_H$ . When a valid TAUDTTINm input edge is detected, the current value of TAUDnCNTm is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The count operation continues.

When the counter reaches  $FFFF_H$ , the counter restarts from  $0000_H$ .

#### NOTE

The TAUDTTINm input signal is sampled at the frequency of the operation clock, specified by TAUDnCMORm.TAUDnCKS[1:0] bits. As a result, the output cycle of TAUDTOUTm has an error of  $\pm 1$  operation clock cycle.

#### Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see **Section 23.9, TAUDTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

### 23.12.9.2 Equations

Functional duration at a TAUDTTINm input pulse =  
count clock cycle  $\times$  (TAUDnCDRm capture value + 1)

### 23.12.9.3 Block Diagram and General Timing Diagram

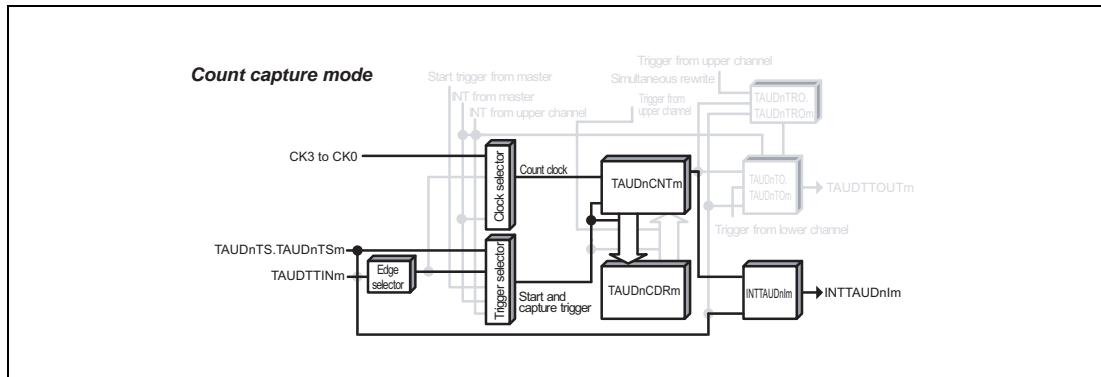


Figure 23.64 Block Diagram of TAUDTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

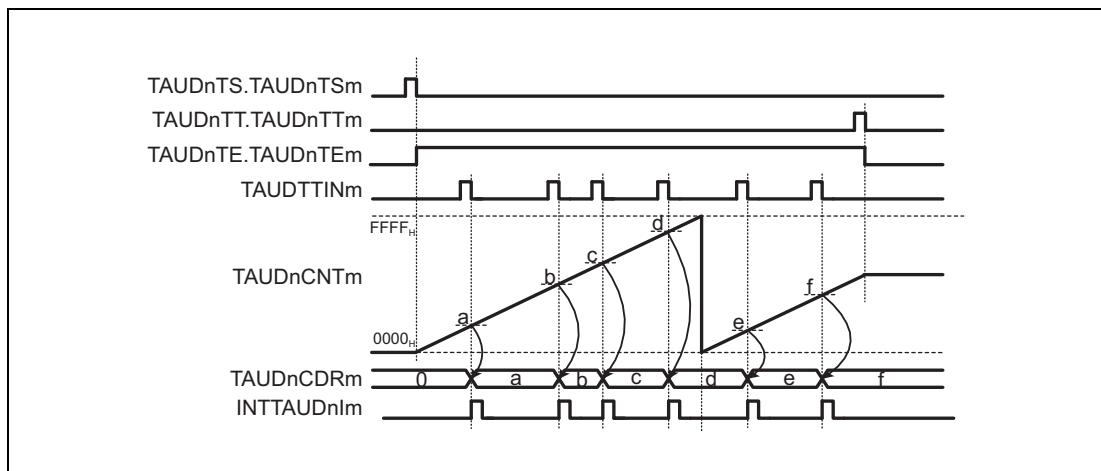


Figure 23.65 General Timing Diagram of TAUDTTINm Input Position Detection Function

### 23.12.9.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.86 Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external capture trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1011: Count capture mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.87 Contents of the TAUDnCMURm Register for TAUDTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges

#### (3) Channel output mode

The channel output mode is not used by this function.

#### (4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input position detection function. Therefore, these registers should be set to 0.

**Table 23.88 Simultaneous Rewrite Settings for TAUDTTINm Input Position Detection Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

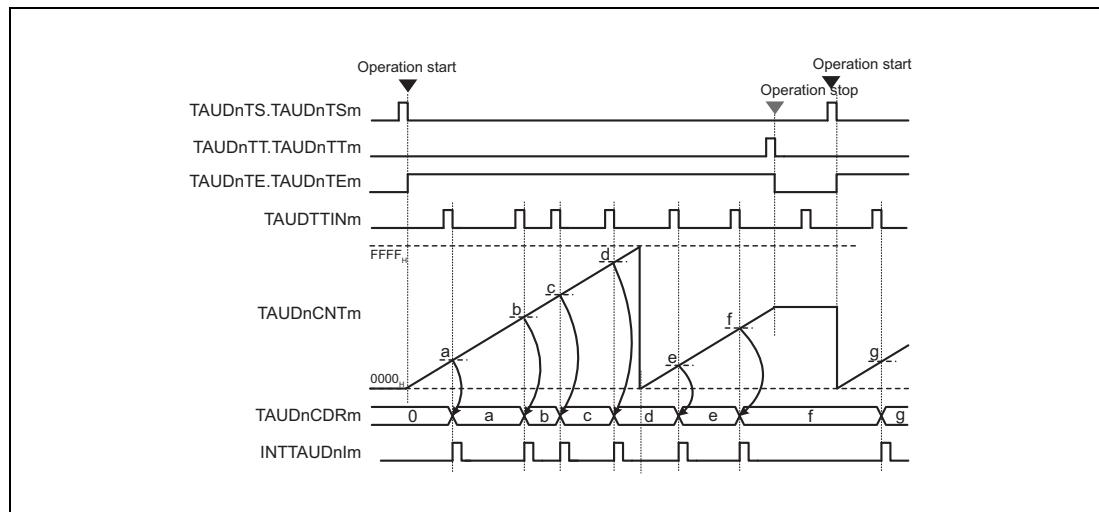
#### 23.12.9.5 Operating Procedure for TAUDTTINm Input Position Detection Function

**Table 23.89 Operating Procedure for TAUDTTINm Input Position Detection Function**

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.86, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function, and Table 23.87, Contents of the TAUDnCMURm Register for TAUDTTINm Input Position Detection Function</b> .  The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs.
During Operation	The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time.	TAUDnCNTm starts to count up from 0000 <sub>H</sub> . When a valid TAUDTTINm edge is detected: <ul style="list-style-type: none"><li>• TAUDnCNTm transfers (captures) its own value to TAUDnCDRm.</li><li>• Outputs INTTAUDnIm.</li><li>• The counter is not cleared to 0000<sub>H</sub> and TAUDnCNTm continues counting.</li></ul> Afterwards, this procedure is repeated. When TAUDnCNTm reaches FFFF <sub>H</sub> , the counter restarts from 0000 <sub>H</sub> .
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

### 23.12.9.6 Specific Timing Diagrams

#### (1) Operation stop and restart



**Figure 23.66 Operation Stop and Restart**  
 $(TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B)$

- The counter can stop operating by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEM to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter stops operating, valid TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from  $0000_H$ .

## 23.12.10 TAUDTTINm Input Period Count Detection Function

### 23.12.10.1 Overview

#### Summary

This function measures the cumulative width of a TAUDTTINm input signal.

#### Prerequisites

- The operating mode should be set to capture and gate count mode. (See **Table 23.90, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function.**)
- TAUDTTOUTm is not used with this function.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The counter awaits a valid TAUDTTINm input edge.

When a valid TAUDTTINm input start edge is detected, the counter starts to count from 0000<sub>H</sub>.

When a valid TAUDTTINm input stop edge is detected, the current TAUDnCNTm value is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The counter stops and retains its value (TAUDnCDRm + 1) until the next valid TAUDTTINm input start edge is detected.

When the next valid TAUDTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.

If the counter reaches FFFF<sub>H</sub>, the counter restarts from 0000<sub>H</sub>.

#### NOTES

1. TAUDTTINm input signal is sampled at the frequency of an operation clock set by the TAUDnCMORm.TAUDnCKS[1:0] bits.
2. As this function is to measure the TAUDTTINm input signal width, setting TAUDnTS.TAUDnTSm to 1 is disabled while TAUDnTE.TAUDnTEM = 1.

#### Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10<sub>B</sub>, the TAUDTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>, the TAUDTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

### 23.12.10.2 Equations

$$\text{Cumulative TAUDTTINm input width} = \text{count clock cycle} \times (\text{TAUDnCDRm capture value} + 1)$$

### 23.12.10.3 Block Diagram and General Timing Diagram

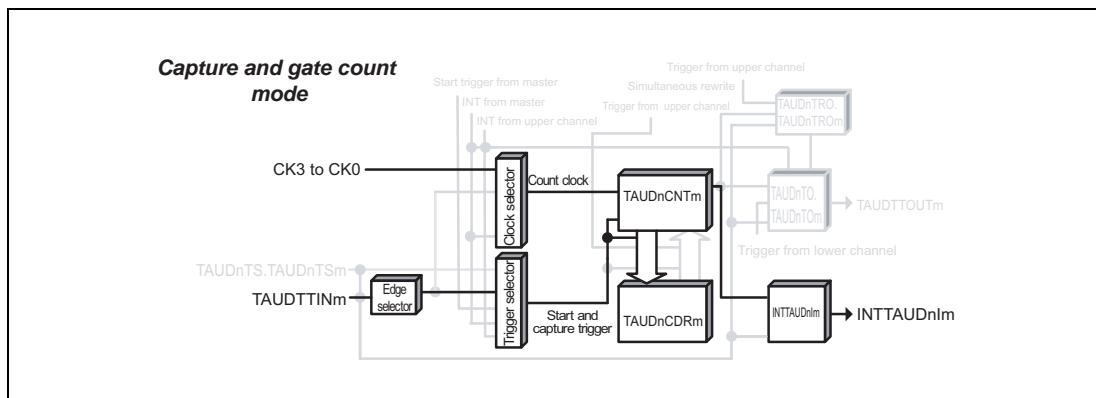


Figure 23.67 Block Diagram of TAUDTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement  
(TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>)

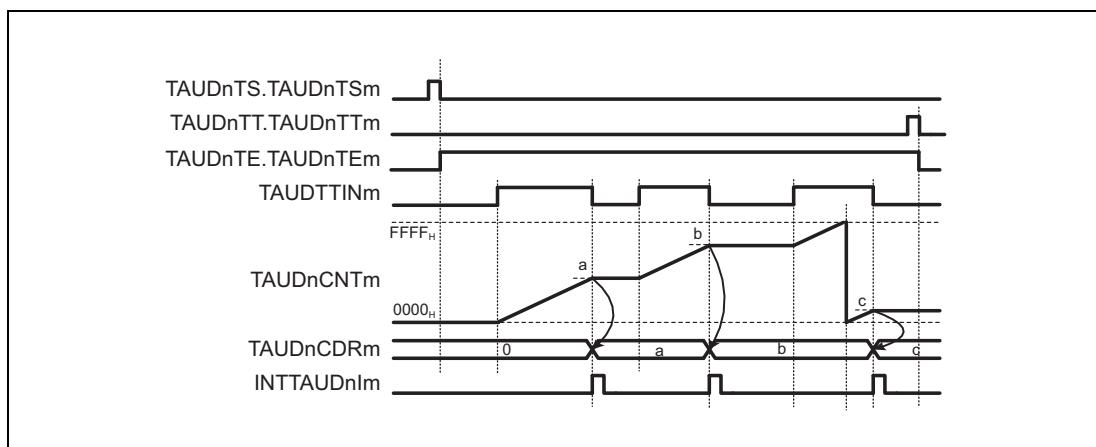


Figure 23.68 General Timing Diagram of TAUDTTINm Input Period Count Detection Function

### 23.12.10.4 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.90 Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1101: Capture and gate count mode
0	TAUDnMDO	0: Disables the start trigger during operation.

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.91 Contents of the TAUDnCMURm Register for TAUDTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

#### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input period count detection function. Therefore, these registers should be set to 0.

**Table 23.92 Simultaneous Rewrite Settings for TAUDTTINm Input Period Count Detection Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.10.5 Operating Procedure for TAUDTTINm Input Period Count Detection Function

**Table 23.93 Operating Procedure for TAUDTTINm Input Period Count Detection Function**

Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.90, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function, and Table 23.91, Contents of the TAUDnCMURm Register for TAUDTTINm Input Period Count Detection Function</b> .  The TAUDnCDRm register functions as a capture register.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.
During Operation	Detection of TAUDTTINm edge  The TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time.  When a TAUDTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUDnCNTm starts counting up from the stop value. When TAUDnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUDnCDRm and INTTAUDnIm is generated. Counting stops at the “value transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When TAUDnCNTm reaches FFFF <sub>H</sub> , the counter restarts from 0000 <sub>H</sub> . Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.

### 23.12.10.6 Specific Timing Diagrams

#### (1) Operation stop and restart

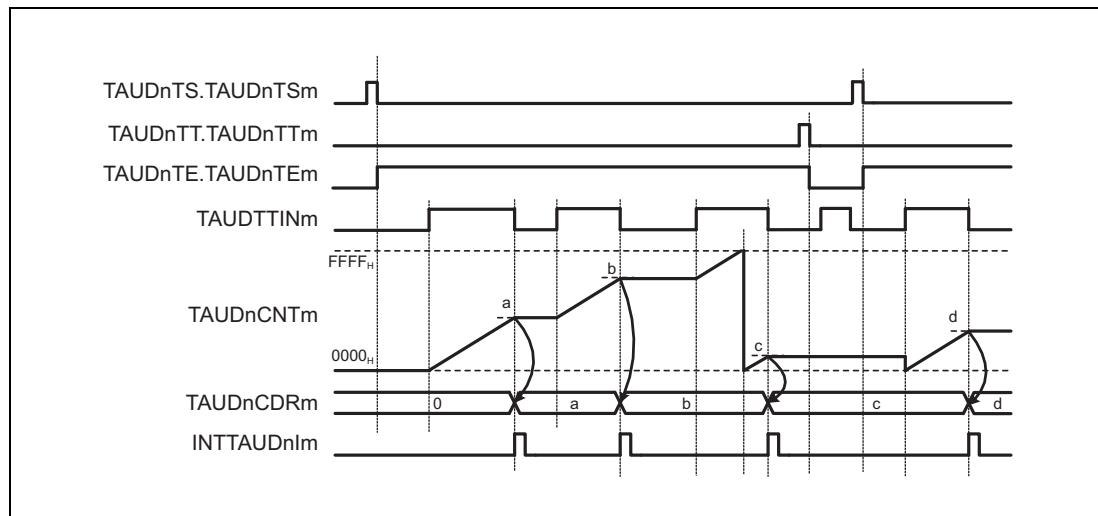


Figure 23.69 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEM to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter is stopped, valid TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000<sub>H</sub>.

## 23.12.11 TAUDTTINm Input Pulse Interval Judgment Function

### 23.12.11.1 Overview

#### Summary

This function outputs the result of a comparison between the count value (TAUDnCNTm) and the value in the channel data register (TAUDnCDRm) when a TAUDTTINm input pulse occurs. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true.

#### Prerequisites

- The operating mode should be set to judge mode. See **Table 23.94, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function.**
- TAUDTTOUTm is not used with this function.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDTTINm valid edge is detected or TAUDnTS.TAUDnTSm is set to 1, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. TAUDnCNTm reloads the value of TAUDnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches  $0000_H$  before a TAUDTTINm valid edge is detected, TAUDnCNTm overflows and is set to  $FFFF_H$ . It then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

#### Conditions

The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:

- If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when  $TAUDnCNTm \leq TAUDnCDRm$ .
- If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when  $TAUDnCNTm > TAUDnCDRm$ .

### 23.12.11.2 Block Diagram and General Timing Diagram

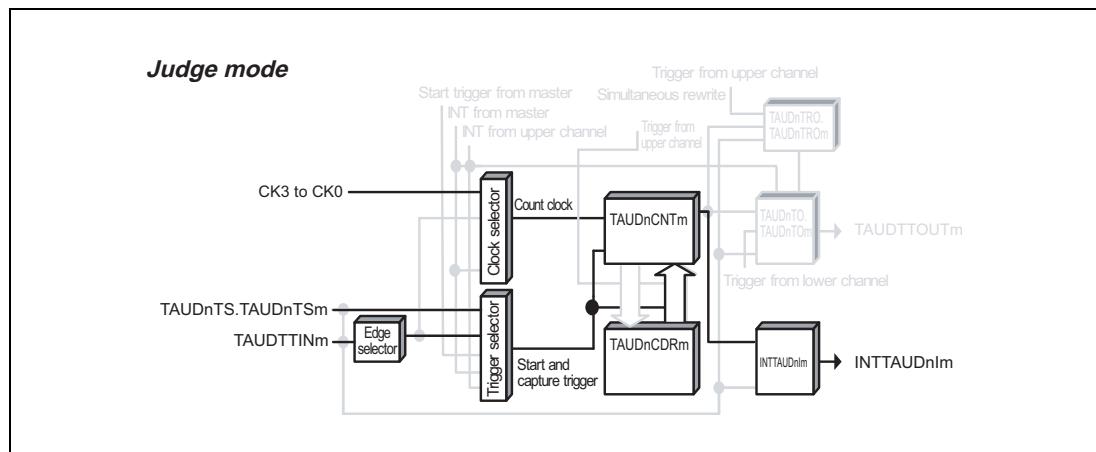


Figure 23.70 Block Diagram of TAUDTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

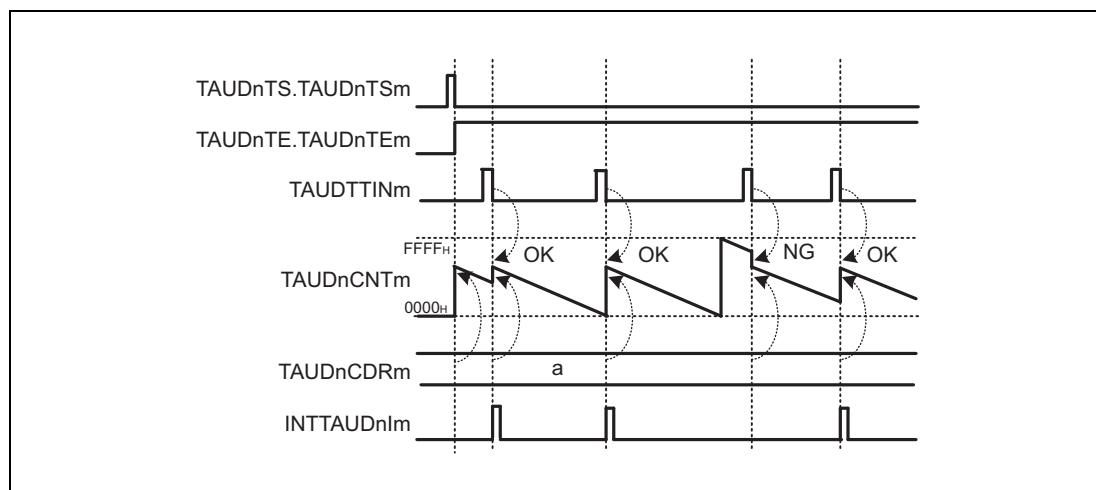


Figure 23.71 General Timing Diagram of TAUDTTINm Input Pulse Interval Judgment Function

### 23.12.11.3 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.94 Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0001: Judge mode
0	TAUDnMDO	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.95 Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

### (4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input pulse interval judgment function. Therefore, these registers should be set to 0.

**Table 23.96 Simultaneous Rewrite Settings for TAUDTTINm Input Pulse Interval Judgment Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.11.4 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

**Table 23.97 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function**

Operation	TAUDn Status
Initial Channel Setting Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.94, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function, and Table 23.95, Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Judgment Function.</b> Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm.
During Operation The following register can be changed at any time: <ul style="list-style-type: none"><li>• TAUDnCDRm register</li></ul>	When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDTTINm input edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDTTINm input edge is detected, INTTAUDnIm is generated. If a TAUDTTINm input edge is detected, then TAUDnCNTm starts to count down from the value of TAUDnCDRm. Afterwards, this procedure is repeated.
Stop Operation Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

## 23.12.12 TAUDTTINm Input Signal Width Judgment Function

### 23.12.12.1 Overview

#### Summary

This function compares the count value (TAUDnCNTm) for the high or low level width of a TAUDTTINm input signal and the TAUDnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUDnIm.

#### Prerequisites

- The operating mode should be set to judge and one-count mode. (See **Table 23.98, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function.**)
- TAUDTTOUTm is not used with this function.

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation. When a valid TAUDTTINm input start edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDTTINm valid stop edge is detected, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. The counter TAUDnCNTm retains its value until the next valid TAUDTTINm start edge is detected, regardless of the result of the comparison.

If the counter reaches  $0000_H$  before a valid TAUDTTINm stop edge is detected, TAUDnCNTm overflows and is set to  $FFFF_H$ . The counter then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

#### Conditions

- The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:
  - If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when  $TAUDnCNTm \leq TAUDnCDRm$ .
  - If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when  $TAUDnCNTm > TAUDnCDRm$ .
- The TAUDnCMURm.TAUDnTIS[1:0] bits specify a type of width measurement:
  - For high width measurement (TAUDnCMURm.TAUDnTIS[1:0] =  $11_B$ ), TAUDTTINm rising edge is used as a start edge and TAUDTTINm falling edge as a stop edge.
  - For low width measurement (TAUDnCMURm.TAUDnTIS[1:0] =  $10_B$ ), TAUDTTINm falling edge is used as a start edge and TAUDTTINm rising edge as a stop edge.
- This function cannot make a forced restart.

### 23.12.12.2 Block Diagram and General Timing Diagram

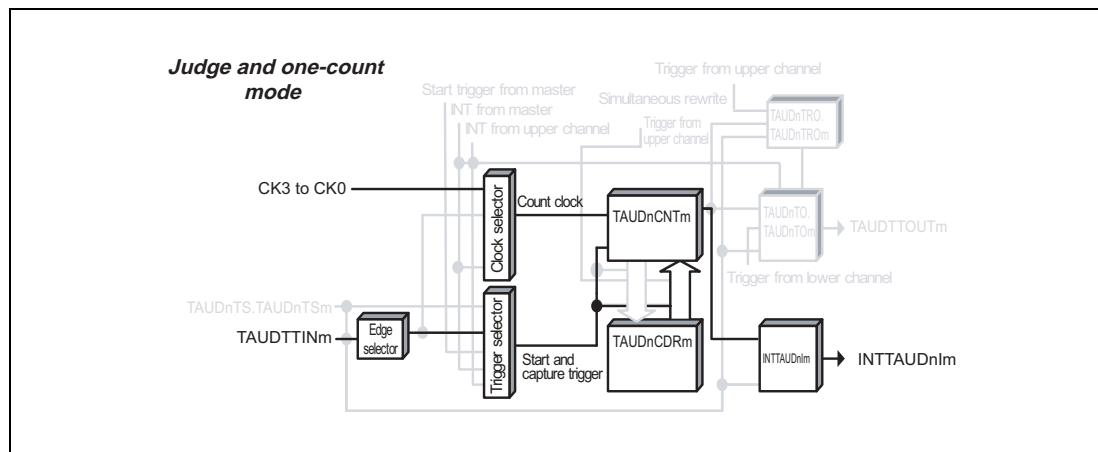


Figure 23.72 Block Diagram of TAUDTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated when  $TAUDnCNTm \leq TAUDnCDRm$  ( $TAUDnCMORm.TAUDnMD0 = 0$ ).
- TAUDTTINm valid start edge = rising edge, TAUDTTINm valid stop edge = falling edge ( $TAUDnCMURm.TAUDnTIS[1:0] = 11_B$ )

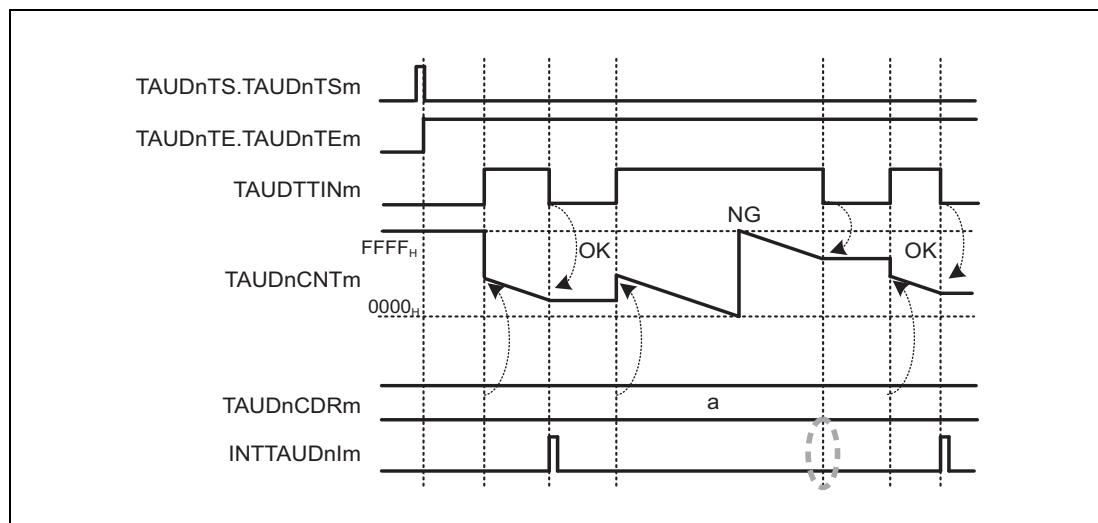


Figure 23.73 General Timing Diagram of TAUDTTINm Input Signal Width Judgment Function

### 23.12.12.3 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.98 Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0111: Judge and one-count mode
0	TAUDnMDO	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.99 Contents of the TAUDnCMURm Register for TAUDTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

#### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input signal width judgment function. Therefore, these registers should be set to 0.

**Table 23.100 Simultaneous Rewrite Settings for TAUDTTINm Input Signal Width Judgment Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.12.4 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

**Table 23.101 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function**

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.98, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function, and Table 23.99, Contents of the TAUDnCMURm Register for TAUDTTINm Input Signal Width Judgment Function.</b></p> <p>Set the value of TAUDnCDRm register.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.
During Operation	<p>The following register can be changed at any time:</p> <ul style="list-style-type: none"> <li>• TAUDnCDRm register</li> </ul>	<p>Upon detection of a TAUDTTINm start edge, TAUDnCNTm starts count down from the value of TAUDnCDRm.</p> <p>When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDTTINm input stop edge is detected, INTTAUDnIm is generated.</p> <p>When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm &gt; TAUDnCDRm when a TAUDTTINm input stop edge is detected, INTTAUDnIm is generated.</p> <p>Afterwards, this procedure is repeated.</p>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.</p>

### 23.12.13 Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

#### 23.12.13.1 Overview

##### Summary

This function measures the width of an individual TAUDTTINm input signal. An interrupt is generated if the TAUDTTINm input width is longer than  $\text{FFFF}_H + 1$ .

##### Prerequisites

- The operation mode must be set to One-Count Mode (see **Table 23.102, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**).
- TAUDTTOUTm is not used for this function.
- The value of TAUDnCDRm must be set to  $\text{FFFF}_H$ .

##### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected.  $\text{FFFF}_H$  is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUDTTINm input start edge is detected, TAUDnCNTm loads  $\text{FFFF}_H$  and starts to count down.

If the counter reaches  $0000_H$  before a stop edge is detected, an interrupt is generated.

##### Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] =  $10_B$ , the TAUDTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] =  $11_B$ , the TAUDTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

##### NOTE

The counter cannot be restarted during operation.

### 23.12.13.2 Block Diagram and General Timing Diagram

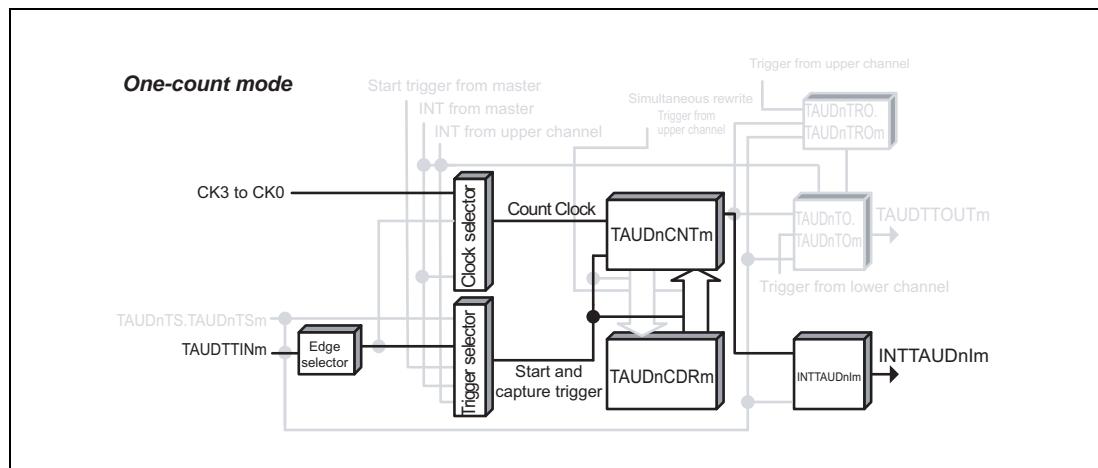


Figure 23.74 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement  
(TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>)

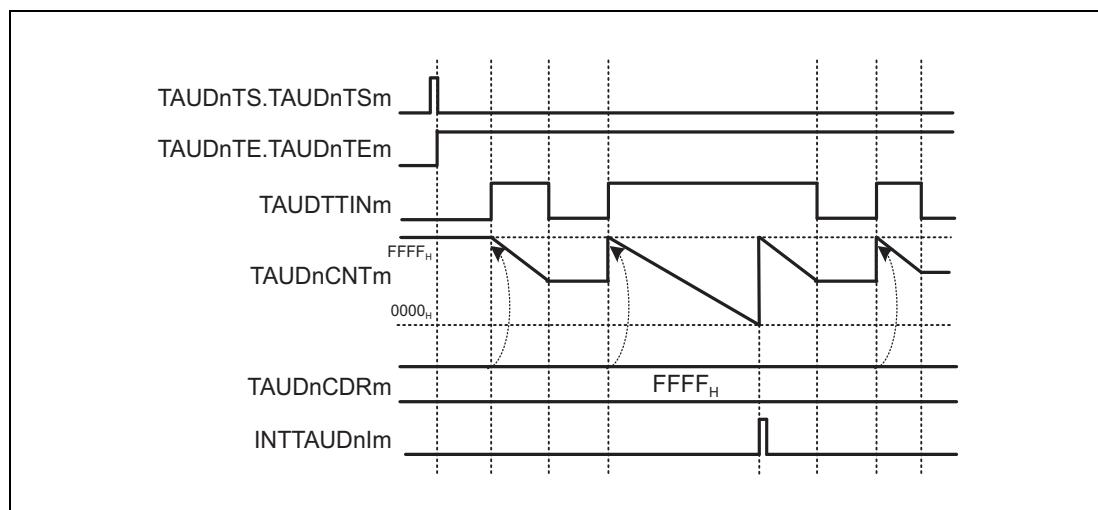


Figure 23.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

### 23.12.13.3 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.102 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	0: Disables the start trigger during operation

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.103 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

#### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used by this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDTTINm Width Measurement). Therefore, these registers must be set to 0.

**Table 23.104 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

**Table 23.105 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.102, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)</b> , and <b>Table 23.103, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)</b> .  Set the value of TAUDnCDRm register to FFFF <sub>H</sub> .	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.  Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge.  When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value (FFFF <sub>H</sub> ).
During Operation	The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000 <sub>H</sub> : <ul style="list-style-type: none"><li>• INTTAUDnIm is generated.</li></ul> When TAUDTTINm input stop edge is detected during count operation: <ul style="list-style-type: none"><li>• TAUDnCNTm stops and retains its current value.</li></ul> When TAUDTTINm input start edge is detected while the counter is stopped: <ul style="list-style-type: none"><li>• TAUDnCNTm loads the TAUDnCDRm value (FFFF<sub>H</sub>) again, and continues to count down.</li></ul> Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

## 23.12.14 Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

### 23.12.14.1 Overview

#### Summary

This function measures the cumulative width of a TAUDTTINm input signal. If the cumulative TAUDTTINm input width is longer than  $FFFF_H$ , an interrupt is generated and an overflow interrupt can be output.

#### Prerequisites

- The operation mode must be set to Gate Count Mode, (see **Table 23.106, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**).
- TAUDTTOUTm is not used with this function.
- The value of TAUDnCDRm must be set to  $FFFF_H$ .

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected.  $FFFF_H$  is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUDTTINm input start edge and then continues to count down from the current value.

When the counter reaches  $0000_H$  an interrupt is generated.  $FFFF_H$  is loaded to TAUDnCNTm and the counter continues to count down until a TAUDTTINm input stop edge is detected.

#### Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] =  $10_B$ , the TAUDTTINm input low width is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] =  $11_B$ , the TAUDTTINm input high width is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

#### NOTE

The counter cannot be restarted during operation.

### 23.12.14.2 Block Diagram and General Timing Diagram

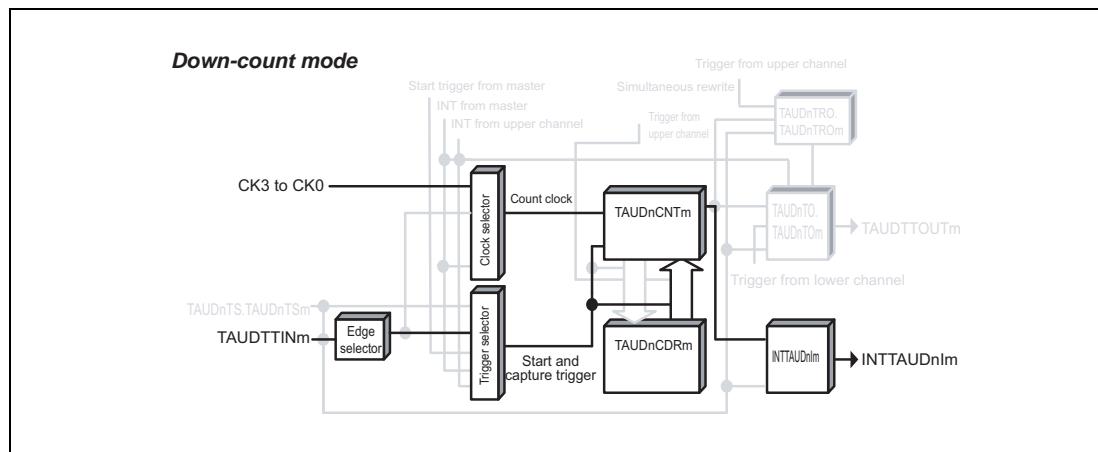


Figure 23.76 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement  
(TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>)

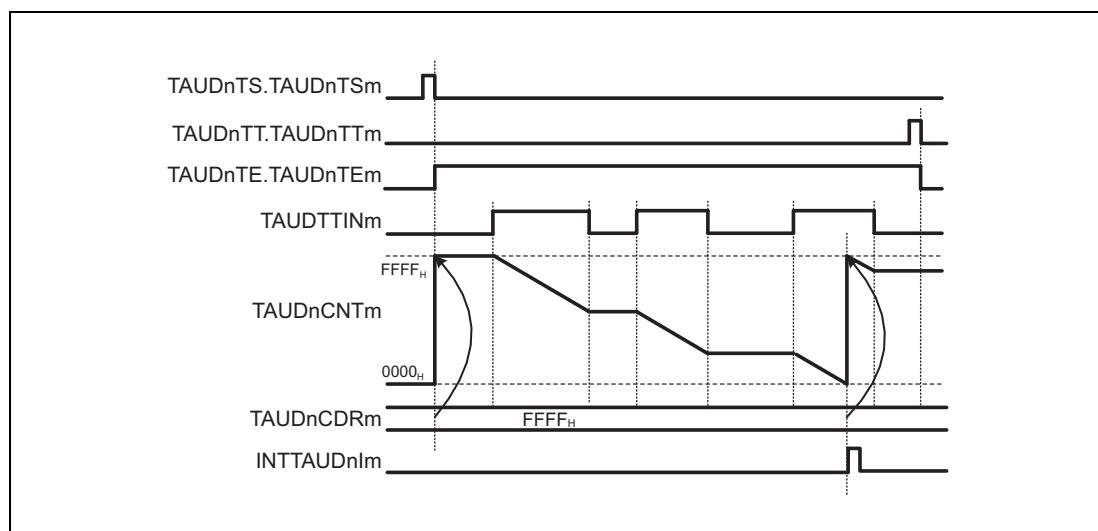


Figure 23.77 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

### 23.12.14.3 Register Settings

#### (1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.106 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1100: Gate count mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation

#### (2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.107 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing , write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

#### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

**Table 23.108 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

#### 23.12.14.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

**Table 23.109 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**

Initial Channel Setting	Operation	TAUDn Status
	Set TAUDnCMORm and TAUDnCMURm registers as described in <b>Table 23.106, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)</b> , and <b>Table 23.107, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)</b> .  Set the value of TAUDnCDRm register to FFFF <sub>H</sub> .	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.  Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge.  When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value (FFFF <sub>H</sub> ).
During Operation	The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000 <sub>H</sub> : <ul style="list-style-type: none"><li>• INTTAUDnIm is generated.</li><li>• TAUDnCNTm loads the TAUDnCDRm value (FFFF<sub>H</sub>) and continues to count down.</li></ul> When TAUDTTINm input stop edge is detected during count operation: <ul style="list-style-type: none"><li>• TAUDnCNTm stops and retains the current value.</li></ul> When TAUDTTINm input start edge is detected while the counter is stopped: <ul style="list-style-type: none"><li>• TAUDnCNTm counts down from the stop value.</li></ul> Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

## 23.12.15 One-Phase PWM Output Function

### 23.12.15.1 Overview

#### Summary

This function adds dead time to a TAUDTTINm input signal. The resulting PWM signal is output via TAUDTTOUTm of the channel and TAUDTTOUTm of upper channels.

#### Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUDnTDE.TAUDnTDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See **Table 23.111, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function.**)
- Any operating mode can be set to upper channels.
- Channel output mode for upper and lower channels should be set to synchronous channel output mode 2 with one-phase PWM output. (See **Section 23.7, Channel Output Modes.**)

#### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000<sub>H</sub>, an interrupt occurs. The counter is reset to FFFF<sub>H</sub> and waits for the next valid TAUDTTINm input start edge.

**Table 23.110 TAUDTTOUTm to which Dead Time is Added and State of TAUDTTINm**

TAUDnCMURm. TAUDnTIS[1:0]	TAUDnTOL. TAUDnTOLm	TAUDTTOUTm to which Dead Time is Added	TAUDnTDL. TAUDnTDLm	TAUDTTINm State when Added
10	0	TAUDTTOUTm low	0	High
			1	Low
11	0	TAUDTTOUTm high	0	High
			1	Low
	1	TAUDTTOUTm low	0	Low
			1	High
	1	TAUDTTOUTm high	0	Low
			1	High

### Conditions

- TAUDnCMURm.TAUDnTIS[1:0] bits specify the type of width measurement:
  - TAUDnCMURm.TAUDnTIS[1:0] = 10<sub>B</sub>: Uses both edges as valid edges for detection (Low width measurement).
  - TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>: Uses both edges as valid edges for detection (High width measurement).
- The TAUDnTDL.TAUDnTDLm bit specifies the operation of TAUDTTOUTm for each channel when an interrupt or valid TAUDTTINm edge is detected on the lower channel:
  - If TAUDnTDL.TAUDnTDLm = 0, an interrupt is used as a TAUDTTOUTm set trigger and a valid TAUDTTINm edge as a TAUDTTOUTm reset trigger.
  - If TAUDnTDL.TAUDnTDLm = 1, a valid TAUDTTINm edge is used as a TAUDTTOUTm set trigger and an interrupt as a TAUDTTOUTm reset trigger.
- This function cannot make a forced restart.

#### 23.12.15.2 Block Diagram and General Timing Diagram

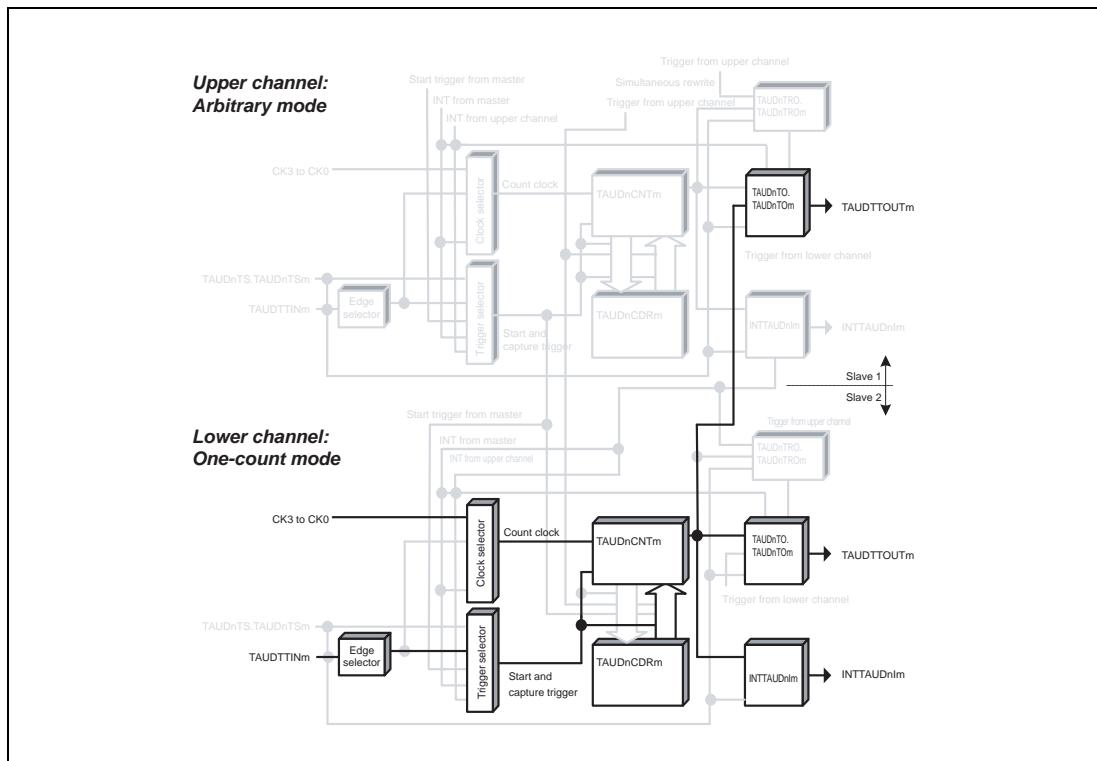
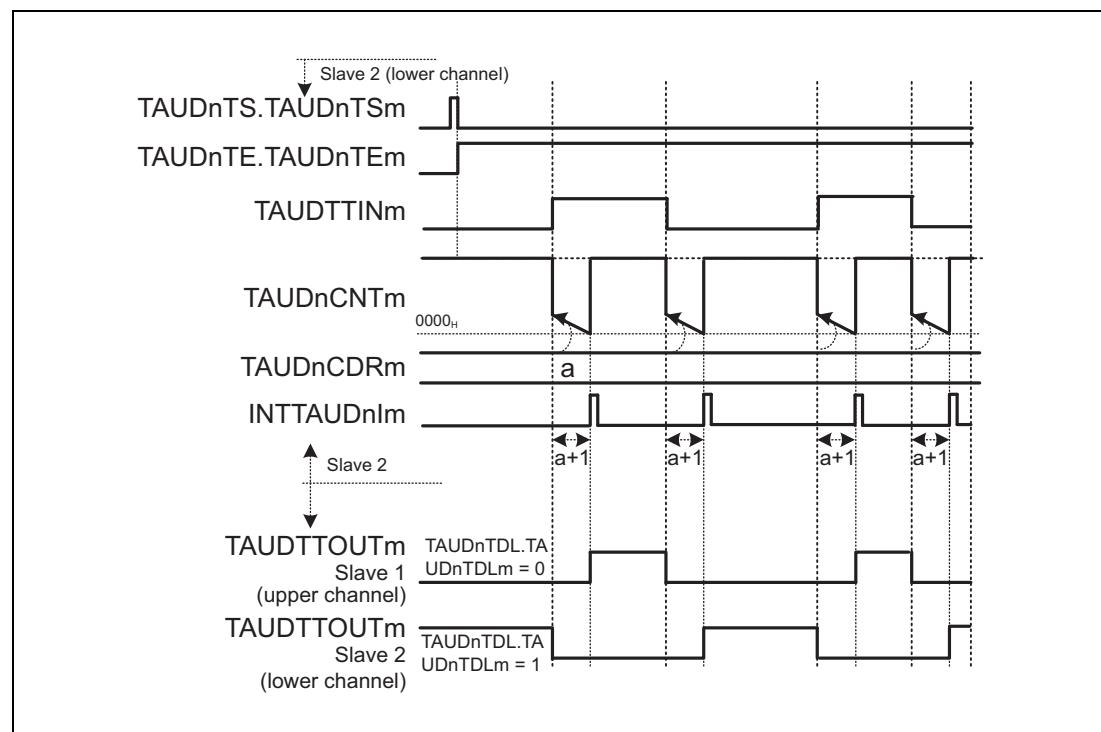


Figure 23.78 Block Diagram of One-Phase PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement  
(TAUDnCMURm.TAUDnTIS[1:0] = 11<sub>B</sub>)

This setting considers a duty as an active high.



**Figure 23.79 General Timing Diagram of One-Phase PWM Output Function**

### 23.12.15.3 Register Settings for Lower Channels

#### (1) TAUDnCMORm for lower channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.111 Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Enables start trigger detection while counting.

#### (2) TAUDnCMURm for lower channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.112 Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

### (3) Channel output mode for lower channels

**Table 23.113 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation
<b>CAUTION</b>	
Set TAUDnTDL.TAUDnTDLm exclusively from upper channels.	

### (4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the one-phase PWM output function. Therefore, these registers should be set to 0.

**Table 23.114 Simultaneous Rewrite Settings for One-Phase PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.12.15.4 Register Settings for Upper Channels

#### (1) TAUDnCMORm for upper channels

TAUDnCMORm register for upper channels can be set arbitrarily.

#### (2) TAUDnCMURm for upper channels

TAUDnCMURm register for upper channels can be set arbitrarily.

#### (3) Channel output mode for upper channels

**Table 23.115 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation
<b>CAUTION</b>	
Set TAUDnTDL.TAUDnTDLm exclusively from lower channels.	

#### (4) Simultaneous rewrite for upper channels

Simultaneous rewrite register for upper channels can be set arbitrarily.

### 23.12.15.5 Operating Procedure for One-phase PWM Output Function

Table 23.116 Operating Procedure for One-phase PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in <b>Table 23.111, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function, and Table 23.112, Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function.</b></p> <p>Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in <b>Section 23.12.15.4, Register Settings for Upper Channels.</b></p> <p>Set the value of TAUDnCDRm register.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.113, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output.</b></p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTOE.TAUDnTOEm (slave channels 1 and 2) to 1 (at restart time only).</p> <p>Set TAUDnTS.TAUDnTSm = 1 for slave channel 2.</p> <p>TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p> <p>Detection of TAUDTTINm start edge</p>	<p>TAUDnTE.TAUDnTEm is set to 1 (slave channel 2) and TAUDnCNTm waits for detection of TAUDTTINm start edge.</p> <p>TAUDnCNTm loads TAUDnCDRm value.</p>
During Operation	<p>The TAUDnCDRm register value can be changed at any time.</p> <p>The TAUDnCNTm register can be read at any time.</p>	<p>TAUDnCNTm of slave channel 2 counts down. When the counter reaches <math>0000_H</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>TAUDnCNTm stops counting.</li> </ul> <p>TAUDTTOUTm is changed by a TAUDTTINm edge detection signal and slave channel 2 INTTAUDnIm signal to output one-phase PWM waveform with dead time.</p> <p>Afterwards, this operation is repeated.</p>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm = 1 for slave channel 2.</p> <p>TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops.</p> <p>TAUDnCNTm stops. TAUDnCNTm and TAUDTTOUTm retain their current values.</p>



## 23.13 Independent Channel Real-Time Functions

This section describes functions that output the value of the TAUDnTRO.TAUDnTROm bit in real time.

### 23.13.1 Real-Time Output Function Type 1

#### 23.13.1.1 Overview

##### Summary

This function outputs a value of the TAUDnTRO.TAUDnTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

##### Prerequisites

- Channels should use the TAUDTTOUTm control of other channels.
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 23.117, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1.**)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 23.7, Channel Output Modes.**)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

##### Functional description

The counter of the upper channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM to 1, enabling count operation. The current value of the data register of the upper channel (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000<sub>H</sub>, INTTAUDnIm is generated and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1). TAUDnCNTm then reloads the TAUDnCDRm value to continue operation subsequently.

The TAUDTTOUTm signal changes only when an interrupt is generated, and when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.

##### Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the

channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.

- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

### 23.13.1.2 Equations

$$\text{INTTAUDnIm generation cycle} = \text{count clock cycle} \times (\text{TAUDnCDRm value} + 1)$$

### 23.13.1.3 Block Diagram and General Timing Diagram

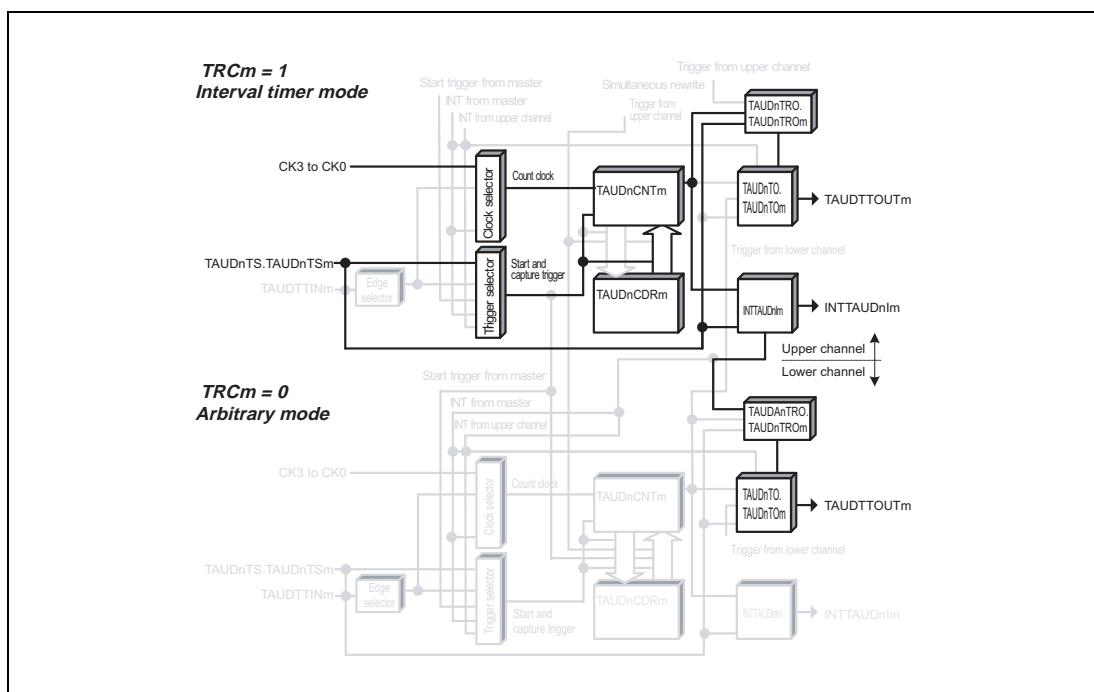


Figure 23.80 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

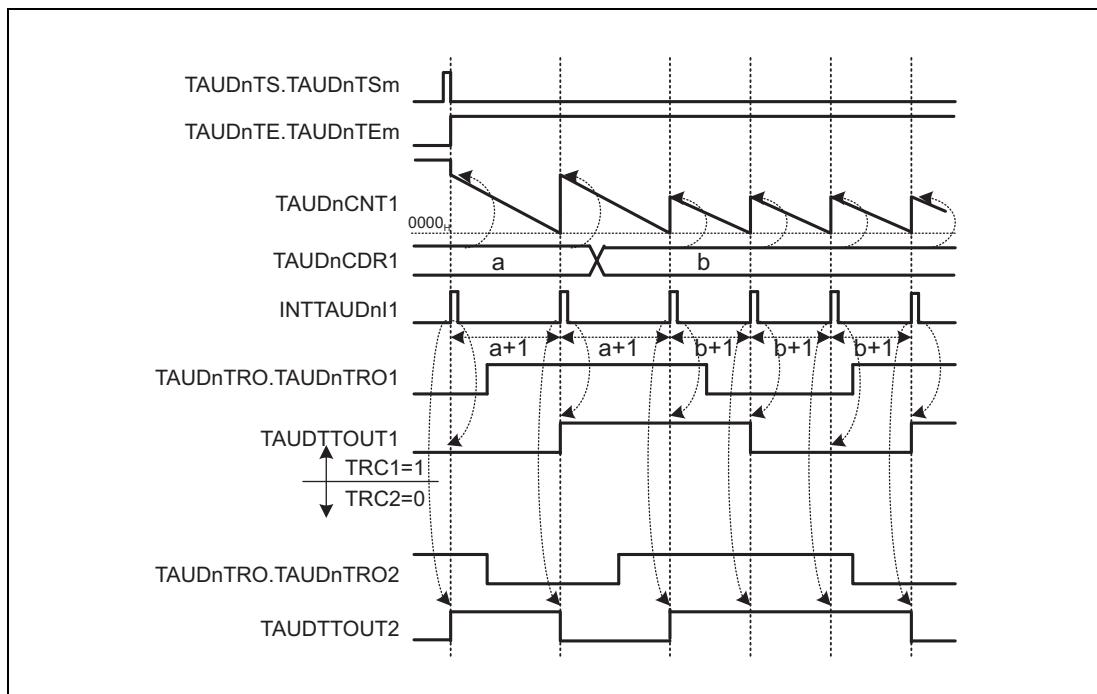


Figure 23.81 General Timing Diagram of Real-Time Output Function Type 1

### 23.13.1.4 Register Settings for Upper Channels

#### (1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.117 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS [1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for upper channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.118 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for upper channels

**Table 23.119 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

### (4) Simultaneous rewrite for upper channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 1. Therefore, these registers should be set to 0.

**Table 23.120 Simultaneous Rewrite Settings for Real-Time Output Function Type 1**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.13.1.5 Register Settings for Lower Channels

#### (1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

#### (2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

#### (3) Channel output mode for lower channels

**Table 23.121 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set this bit to 0
TAUDnTRE.TAUDnTREM	1: Enables real-time output
TAUDnTRO.TAUDnTROM	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

#### (4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers for lower channels are available for any setting.

### 23.13.1.6 Operating Procedure for Real-Time Output Function Type 1

Table 23.122 Operating Procedure for Real-Time Output Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in <b>Table 23.117, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1</b>, and <b>Table 23.118, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1</b>.</p> <p>Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in <b>Section 23.13.1.5, Register Settings for Lower Channels</b>.</p> <p>Set the value of TAUDnCDRm register (only channels with TAUDnTRC.TAUDnTRCm = 1)</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.119, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output</b>.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.121, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output</b>.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1.</p> <p>TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
During Operation	<p>TAUDnCDRm and TAUDnTRO.TAUDnTROm can be changed at any time.</p> <p>The TAUDnCNTm register can be read at any time.</p>	<p>TAUDnCNTm counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues.</li> <li>INTTAUDnIm is generated.</li> <li>TAUDTTOUTm outputs the current value of the real-time output bit TAUDnTRO.TAUDnTROm.</li> </ul> <p>Afterwards, this procedure is repeated.</p>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm to 1.</p> <p>TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops.</p> <p>TAUDnCNTm stops. Both TAUDnCNTm and TAUDTTOUTm retain their current values.</p>



### 23.13.1.7 Specific Timing Diagrams

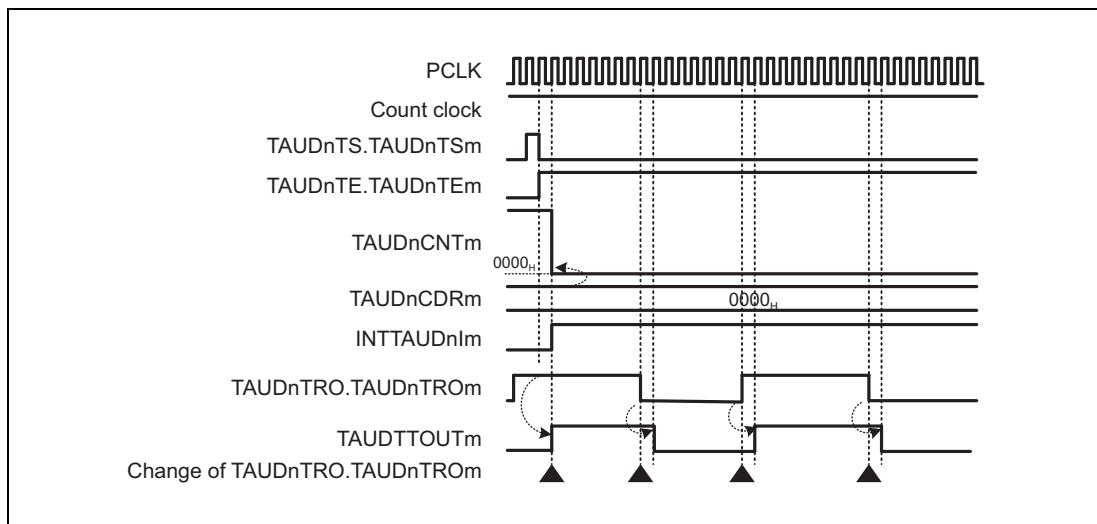


Figure 23.82 TAUDnCDRm =  $0000_{\text{H}}$ , TAUDnCMORm.TAUDnMD0 = 1

- The value of TAUDTTOUTm changes according to the setting of TAUDnTRO.TAUDnTROm with a delay of one PCLK cycle.

## 23.13.2 Real-Time Output Function Type 2

### 23.13.2.1 Overview

#### Summary

This function outputs the value of TAUDnTRO.TAUDnTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated when a valid TAUDTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

#### Prerequisites

- Channels should use the TAUDTTOUTm control of the other channels.
- The operating mode for the upper channel should be set to capture mode. (See **Table 23.123, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2.**)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 23.7, Channel Output Modes.**)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

#### Functional description

The counter for upper channels is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM to 1, enabling count operation. The counter starts to count up.

When a valid TAUDTTINm input edge is generated on one of upper channels, an interrupt occurs and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1).

The TAUDTTOUTm signal changes only when an interrupt is generated, and when TAUDTTOUTm value is different from the current value of TAUDnTRO.TAUDnTROm during the occurrence of the interrupt.

#### Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not

output. For details, see **Section 23.9, TAUDTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

### 23.13.2.2 Block Diagram and General Timing Diagram

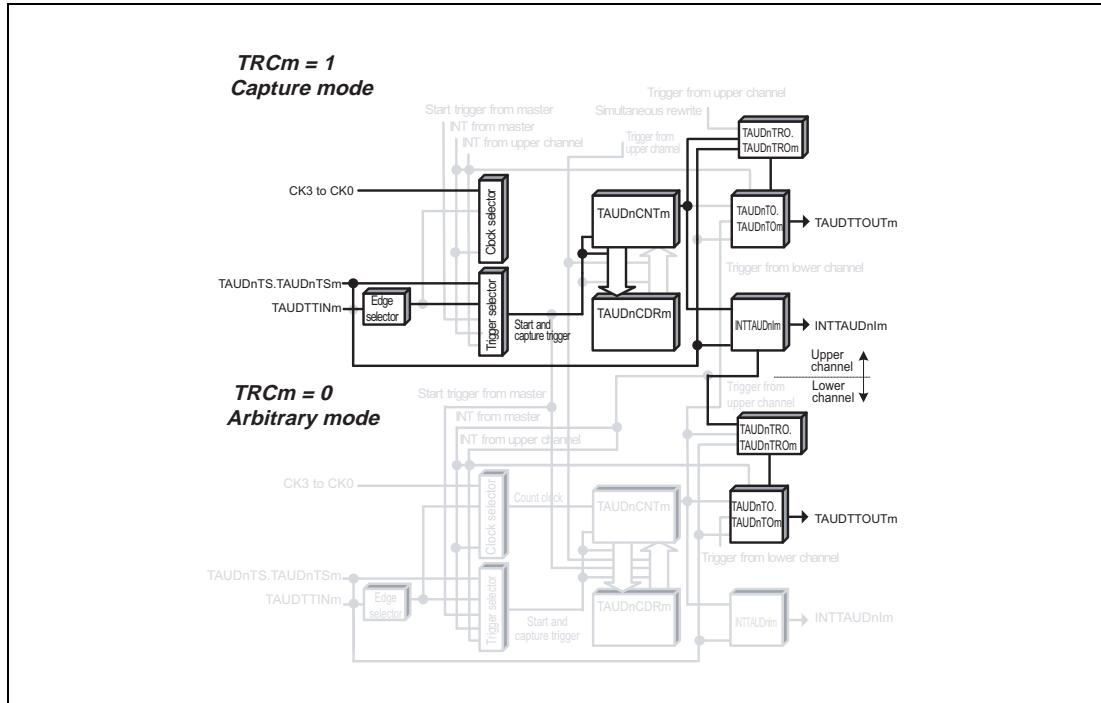


Figure 23.83 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)

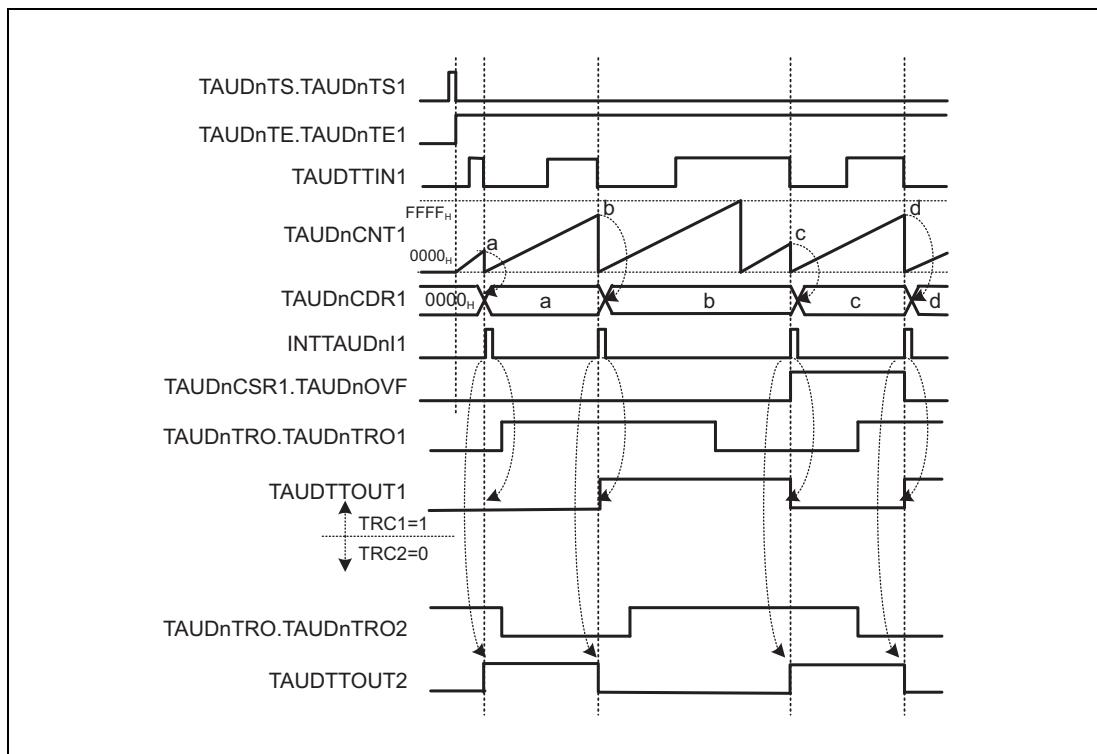


Figure 23.84 General Timing Diagram of Real-Time Output Function Type 2

### 23.13.2.3 Register Settings for Upper Channels

#### (1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.123 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for upper channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.124 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

### (3) Channel output mode for upper channels

**Table 23.125 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

### (4) Simultaneous rewrite for upper channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 2. Therefore, these registers should be set to 0.

**Table 23.126 Simultaneous Rewrite Settings for Real-Time Output Function Type 2**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.13.2.4 Register Settings for Lower Channels

#### (1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

#### (2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

#### (3) Channel output mode for lower channels

**Table 23.127 Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	1: Enables real-time output.
TAUDnTRO.TAUDnTROM	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

#### (4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

### 23.13.2.5 Operating Procedure for Real-Time Output Function Type 2

Table 23.128 Operating Procedure for Real-Time Output Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in <b>Table 23.123, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2</b>, and <b>Table 23.124, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2</b>.</p> <p>Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in <b>Section 23.13.2.4, Register Settings for Lower Channels</b>.</p> <p>The TAUDnCDRm register functions as a capture register (only channels with TAUDnTRC.TAUnTRCm = 1).</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.125, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output</b>.</p> <p>Set channel output mode by setting the control bits as described in <b>Table 23.127, Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output</b>.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1.</p> <p>TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEM is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000H. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnlm is generated.
During Operation	TAUDnTRO.TAUDnTROm can be changed at any time.	<p>TAUDnCNTm starts to count up from 0000H. When a valid TAUDTTINm input edge is detected:</p> <ul style="list-style-type: none"> <li>• TAUDnCNTm captures the TAUDnCDRm value, and the counter is cleared to 0000H.</li> <li>• INTTAUDnlm is generated.</li> <li>• When the TAUDTTINm input valid edge is detected immediately after the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is set to 1. When detected before the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is cleared to 0.</li> </ul> <p>TAUDTTOUTm outputs the current value of real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated.</p>
Stop Operation	Set TAUDnTT.TAUDnTTm to 1.	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops.</p> <p>TAUDnCNTm stops. TAUDnCNTm, TAUDnCSRm.TAUDnOVF, and TAUDTTOUTm retain their current values.</p>

Restart Operation  
→

### 23.13.2.6 Specific Timing Diagrams

#### (1) Operation start and stop

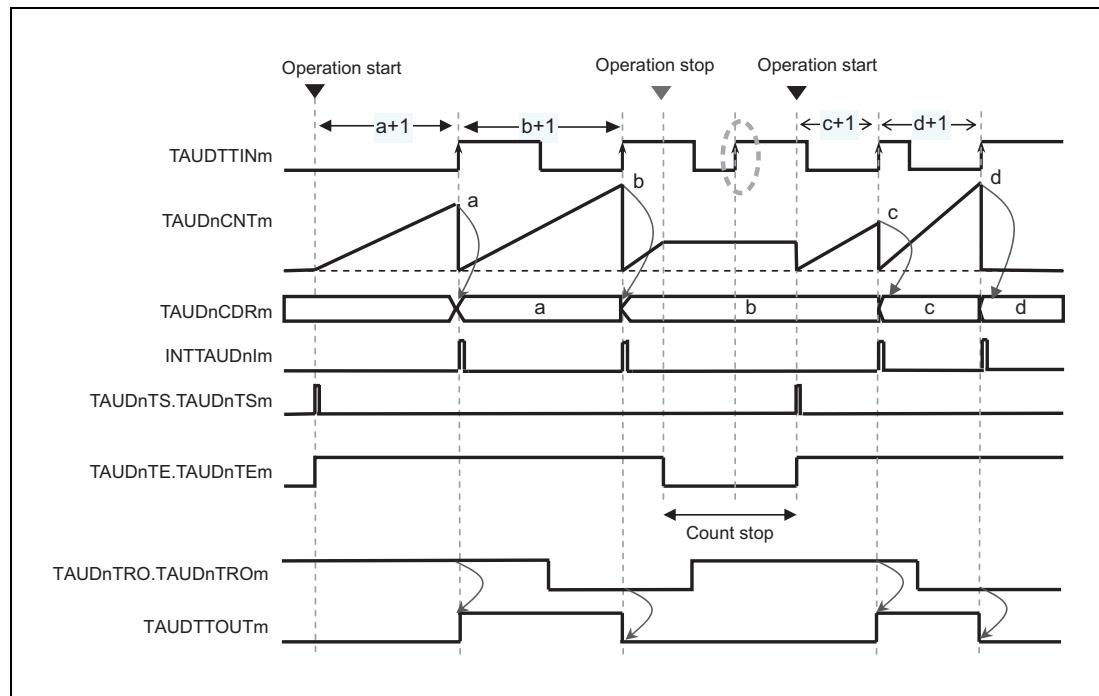


Figure 23.85 Operation Start and Stop ( $\text{TAUDnCMORm} \cdot \text{TAUDnMD0} = 0$ )

- When  $\text{TAUDnTS} \cdot \text{TAUDnTSm}$  is set to 1, the counter starts counting up.
- When a valid input edge is detected, the current value of the counter is written to the data register ( $\text{TAUDnCDRm}$ ) and an interrupt is generated.
- $\text{TAUDTTOUTm}$  outputs the current value of the real-time output bit ( $\text{TAUDnTRO} \cdot \text{TAUDnTROM}$ ) and the counter resets and starts to count up again.
- The  $\text{TAUDTTOUTm}$  signal only changes when an interrupt is generated, and then only when its value is different from the current value of  $\text{TAUDnTRO} \cdot \text{TAUDnTROM}$  at the moment that the interrupt is generated.
- If the counter is stopped ( $\text{TAUDnTE} \cdot \text{TAUDnTEM} = 0$ ), valid input edges are ignored and no interrupt is generated.

## 23.14 Independent Channel Simultaneous Rewrite Functions

This section describes functions that carry out simultaneous rewrite.

### 23.14.1 Simultaneous Rewrite Trigger Generation Function Type 1

#### 23.14.1.1 Overview

##### Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

##### Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 23.129, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1.**)
- For the operating mode that can be set for lower channels, see **Table 23.46, Channel Functions and the Methods They Use for Simultaneous Rewrite.**
- TAUDTTOUTm is not used for any channel in this function.

##### Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUDnCDRm buf) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value. The counter for lower channels start to count according to the selected operating mode.

Once the counter reaches 0000H, an interrupt occurs on the channel. The current value of the corresponding TAUDnCDRm buffer is loaded into TAUDnCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous rewrite (TAUDnRDC.TAUDnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

##### Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm

bit should be set to 0 for all other channels in which simultaneous rewrite should take place.

- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 23.9, TAUDTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts.**

### 23.14.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle  $\times$  (TAUDnCDRm + 1)

To control simultaneous rewrite, the following condition should be satisfied:

[PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1)  $\times$  number of interrupts] – 1

[Triangle PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1)  $\times$  2  $\times$  number of interrupts] – 1

That is, the ratio of TAUDnCDRm + 1 and TAUDnCDRm\_master + 1 should be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

### 23.14.1.3 Block Diagram and General Timing Diagram

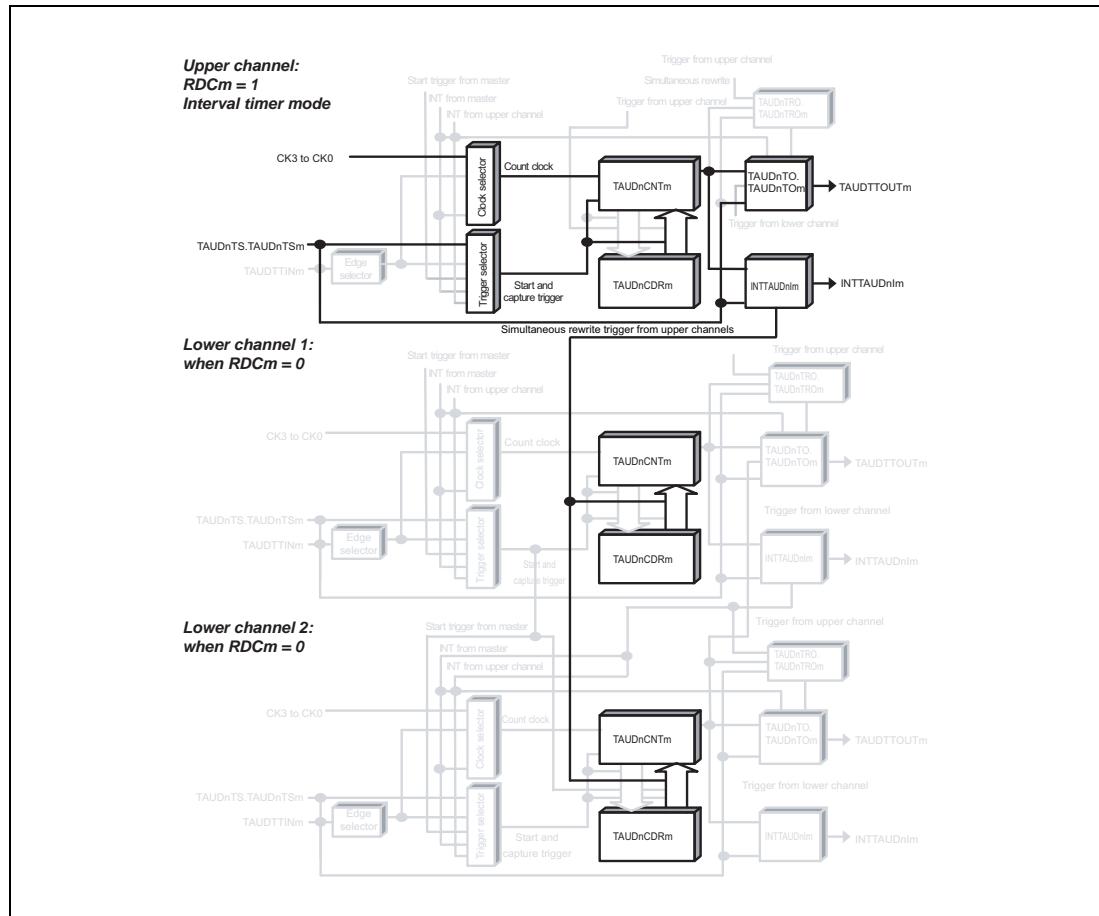


Figure 23.86 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

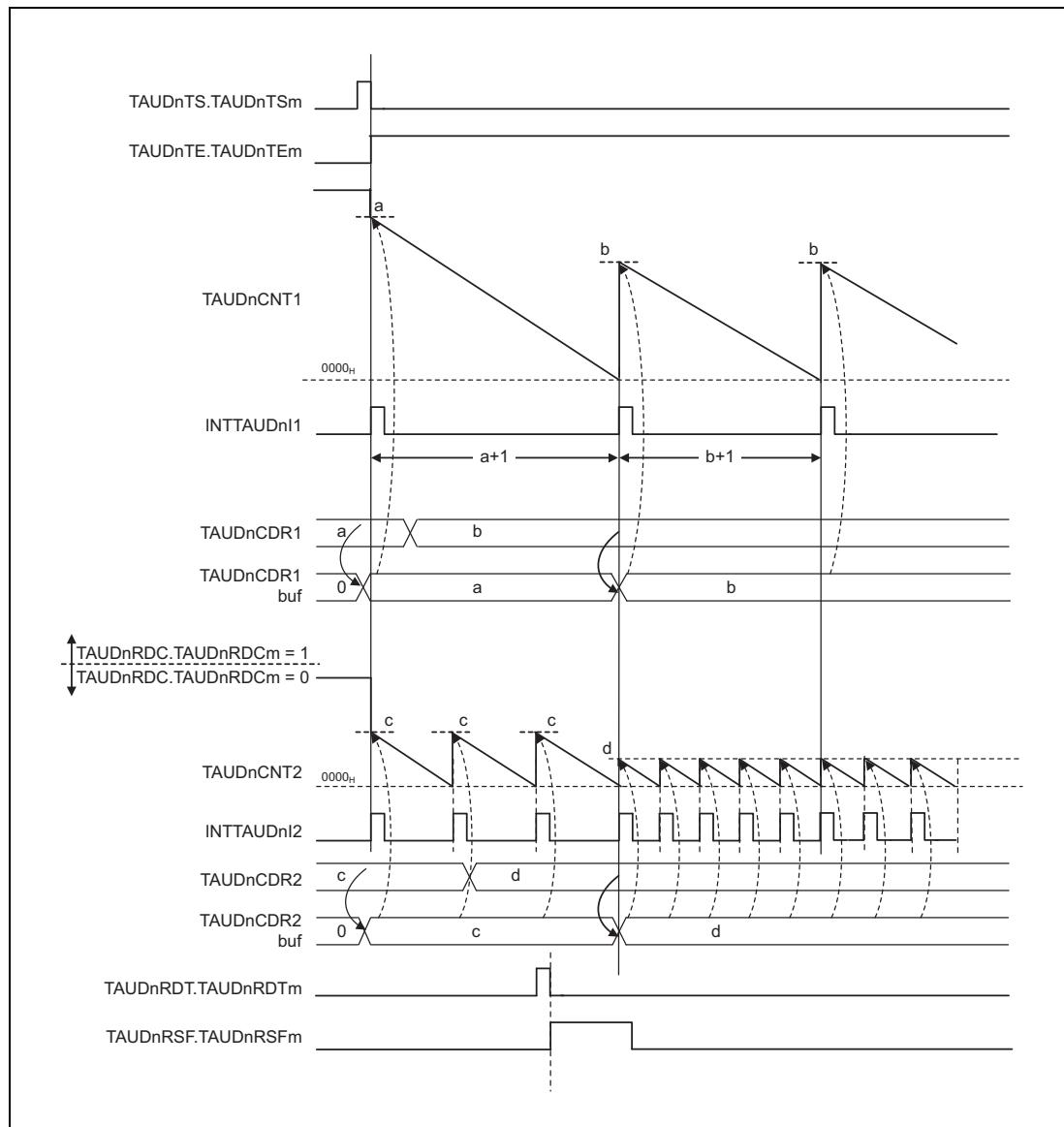


Figure 23.87 General Timing Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

### 23.14.1.4 Register Settings for Upper Channels

#### (1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.129 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for upper channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.130 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

#### (3) Channel output mode for upper channels

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

#### (4) Simultaneous rewrite for upper channels

**Table 23.131 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnlm signal which triggers a simultaneous rewrite on the channel.

### 23.14.1.5 Register Settings for Lower Channels

#### (1) TAUDnCMORm for lower channels

TAUDnCMORm register for lower channels must follow the TAUDnCMORm register settings in the operating mode which can be set. (See **Table 23.46, Channel Functions and the Methods They Use for Simultaneous Rewrite.**)

#### (2) TAUDnCMURm for lower channels

TAUDnCMURm register for lower channels must follow the TAUDnCMURm register settings in the operating mode which can be set. (See **Table 23.46, Channel Functions and the Methods They Use for Simultaneous Rewrite.**)

#### (3) Channel output mode for lower channels

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 23.45, Simultaneous Rewrite Methods and when They are Triggered.**

#### (4) Simultaneous rewrite for lower channels

**Table 23.132 Simultaneous Rewrite Settings for Lower Channels in Simultaneous Rewrite Trigger Generation Function Type 1**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.14.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

**Table 23.133 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1**

	Operation	TAUDn Status
Initial Channel Setting	<p>Set TAUDnCMORM and TAUDnCMURm registers for the upper channel as described in <b>Table 23.129, Contents of the TAUDnCMORM Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1</b>, and <b>Table 23.130, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1</b>.</p> <p>Set TAUDnCMORM and TAUDnCMURm registers for lower channels as described in <b>Section 23.14.1.5, Register Settings for Lower Channels</b>.</p> <p>Set the value of TAUDnCDRm register.</p>	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. If TAUDnCMORM.TAUDnMD0 = 1, INTTAUDnIm is generated.
During Operation	TAUDnRDT.TAUDnRDTm and TAUDnCDR.TAUDnCDRm are changeable. TAUDnRSF.TAUDnRSFm can be always read.	TAUDnCNTm counts down. When the counter reaches 0000H: <ul style="list-style-type: none"> <li>The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues.</li> <li>INTTAUDnIm is generated.</li> </ul> If INTTAUDnIm is generated on the channel where TAUDnRDC.TAUDnRDCm is set to 1, simultaneous rewrite is controlled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.



## 23.14.2 Simultaneous Rewrite Trigger Generation Function Type 2

### 23.14.2.1 Overview

#### Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. In this function, the interrupt is generated when a valid TAUDTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

#### Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operation mode of the upper channel must be set to Capture Mode (see **Table 23.134, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**).
- For the operation mode that can be set for a lower channel, see **Table 23.46, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

#### Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The counter for the upper channel starts to count up, and then the counter for lower channels start to count according to the selected operating mode.

When a TAUDTTINm input edge occurs on the upper channel, an interrupt is generated. The trigger is detected by the lower channel(s), which then also generate an interrupt.

When TAUDnRDC.TAUDnRDCm = 1 on the upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

#### Conditions

- The channel which is monitored for INTTAUDnIm is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, an interrupt is generated when the function starts. For details see **Section 23.9, TAUDTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

### 23.14.2.2 Block Diagram and General Timing Diagram

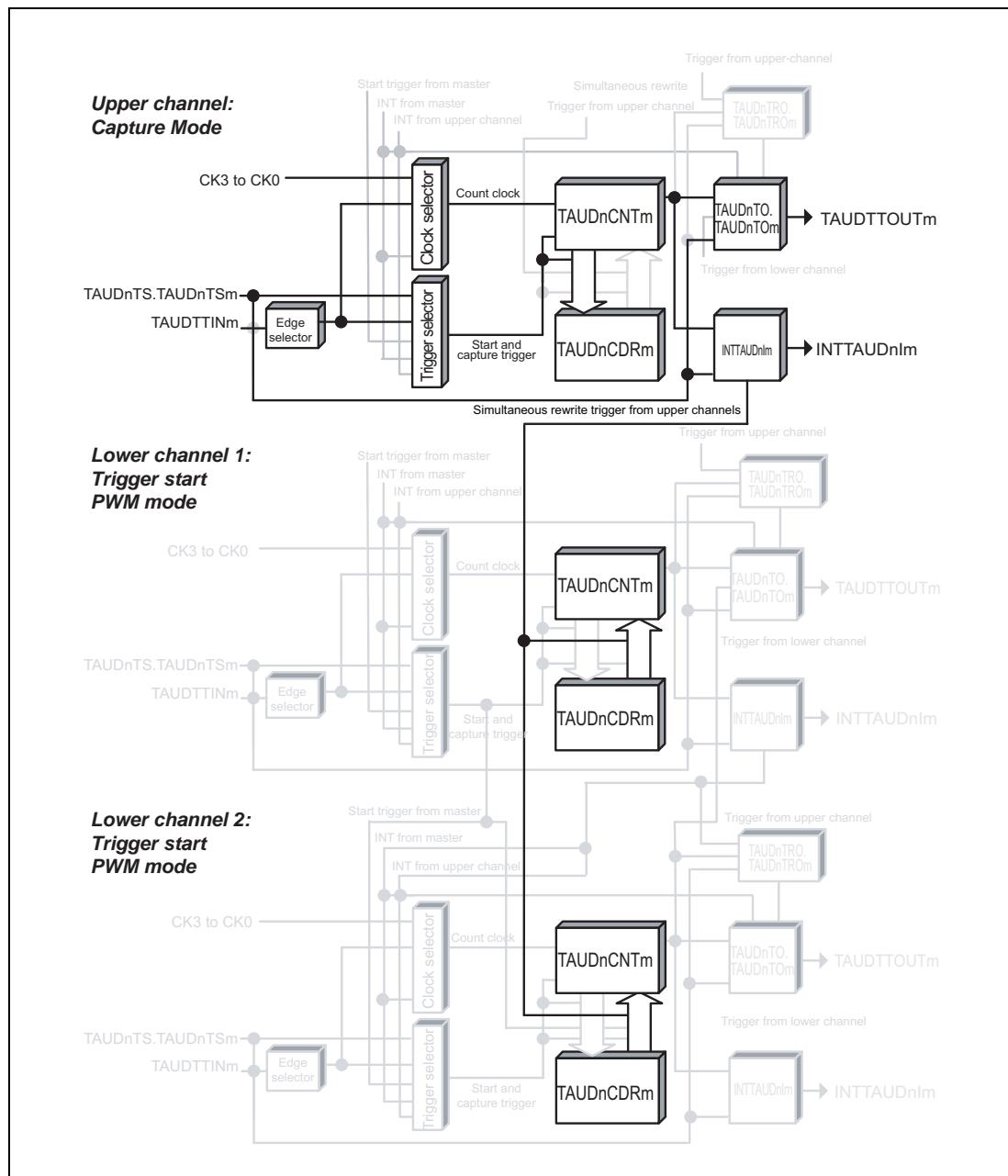
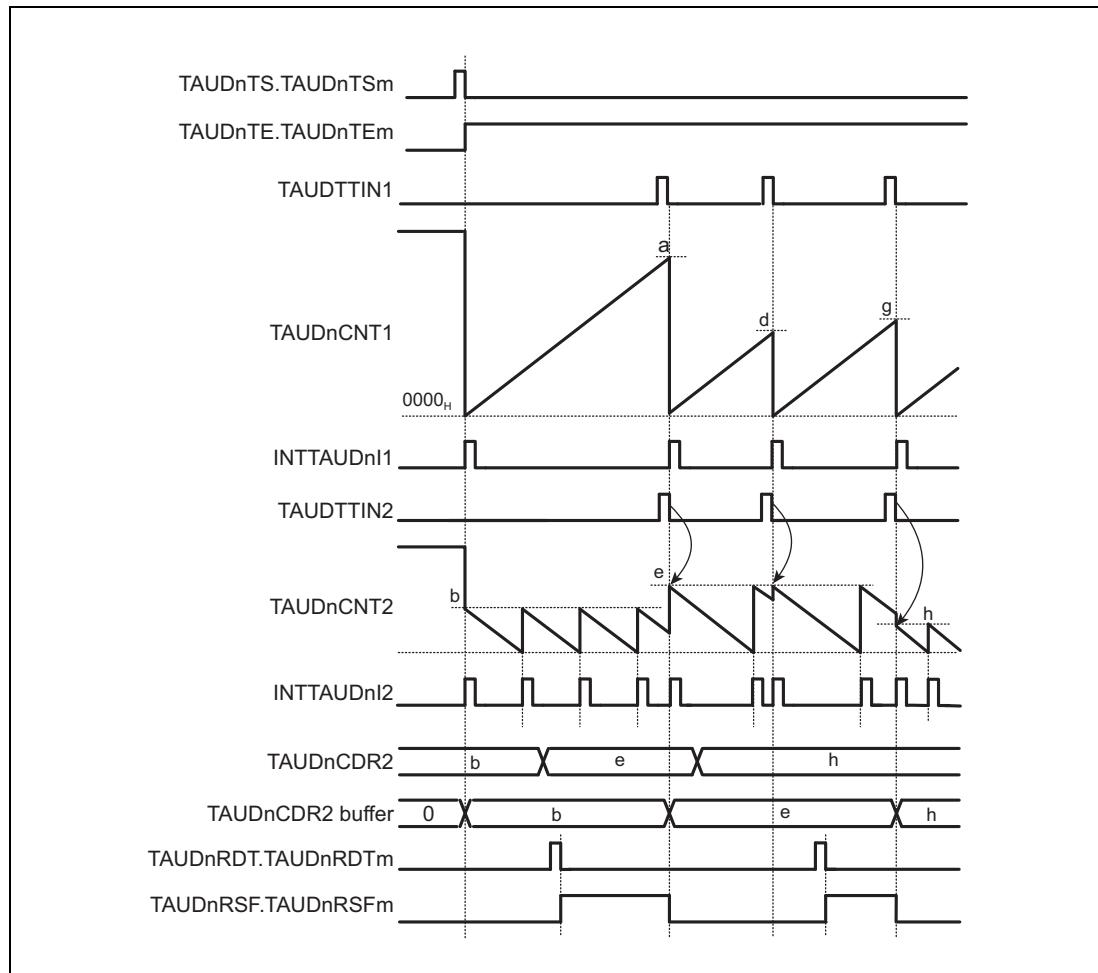


Figure 23.88 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)
- Upper channel (CH1) generates simultaneous rewrite trigger.



**Figure 23.89 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 2**

### 23.14.2.3 Register Settings for Upper Channels

#### (1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.134 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as the external capture trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for upper channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.135 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

**(3) Channel output mode for upper channels**

The channel output mode is not used by this function.

**(4) Simultaneous rewrite for upper channels**

**Table 23.136 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 2**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

### 23.14.2.4 Register Settings for Lower Channels

#### (1) TAUDnCMORm for lower channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.137 Contents of the TAUDnCMORm Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for lower channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.138 Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

**(3) Channel output mode for lower channels**

Output can be made according to the trigger start PWM mode setting.

**(4) Simultaneous rewrite for lower channels**

**Table 23.139 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 2**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.14.2.5 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

**Table 23.140 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2**

Operation	TAUDn Status	
Initial Channel Setting	Set the TAUDnCMORM and TAUDnCMURm registers for the upper channel as described in <b>Table 23.134, Contents of the TAUDnCMORM Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2</b> and <b>Table 23.135, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2</b> .  Set the TAUDnCMORM and TAUDnCMURm registers for the lower channel as described in <b>Table 23.137, Contents of the TAUDnCMORM Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2</b> and <b>Table 23.138, Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2</b> .  The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to $0000_H$ . INTTAUDnIm is generated when TAUDnCMORM.TAUDnMDO is set to 1.
During Operation	TAUDnRDT.TAUDnRDTm can be set at any time. TAUDnRSF.TAUDnRSFm can be read at any time.	TAUDnCNTm counts up from $0000_H$ . When a TAUDTTINm valid edge is detected: <ul style="list-style-type: none"><li>• TAUDnCNTm transfers (captures) its value to TAUDnCDRm and returns to <math>0000_H</math>.</li><li>• INTTAUDnIm is generated.</li></ul> Simultaneous rewrite is controlled when INTTAUDnIm is generated from the channel where TAUDnRDC.TAUDnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and it retains its current value.



## 23.15 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the timer array unit D.

For a general overview of synchronous channel operation, see [Section 23.2, Overview](#)

This section describes functions that generate PWM signals at regular intervals.

### 23.15.1 PWM Output Function

#### 23.15.1.1 Overview

##### Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUDTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

##### Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See [Table 23.141, Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function](#).)
- The operating mode for the slave channels should be set to one-count mode. (See [Table 23.144, Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function](#).)
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channels should be set to Synchronous Channel Output Mode 1. (See [Section 23.7, Channel Output Modes](#).)

##### Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm, and the counter starts counting down from the TAUDnCDRm value. If an INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set/reset, PWM output is made.

- Master channel:  
When the master channel counter reaches  $0000_H$  and the pulse cycle time has passed, INTTAUDnIm is generated. The counter loads TAUDnCDRm value into TAUDnCNTm and counts down.
- Slave channel:  
When INTTAUDnIm is generated on the master channel, the counter operation of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts counting down from the TAUDnCDRm value. TAUDTTOUTm signal is set to the active level.  
When the counter reaches to  $0000_H$  (duty time has elapsed), INTTAUDnIm is generated and a TAUDTTOUTm signal is set to an inactive level. The counter is reset to  $FFFF_H$  and waits for the next INTTAUDnIm (start of the next pulse cycle) of the master channel.

The counter can stop operating by setting the TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

### Conditions

Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

#### 23.15.1.2 Equations

$$\text{Pulse cycle} = (\text{TAUDnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$\text{Duty cycle [%]} = (\text{TAUDnCDRm (slave)}) / (\text{TAUDnCDRm (master)} + 1) \times 100$$

- Duty cycle = 0%  
TAUDnCDRm (slave) = 0000H
- Duty cycle = 100%  
TAUDnCDRm (slave)  $\geq$  TAUDnCDRm (master) + 1

#### 23.15.1.3 Block Diagram and General Timing Diagram

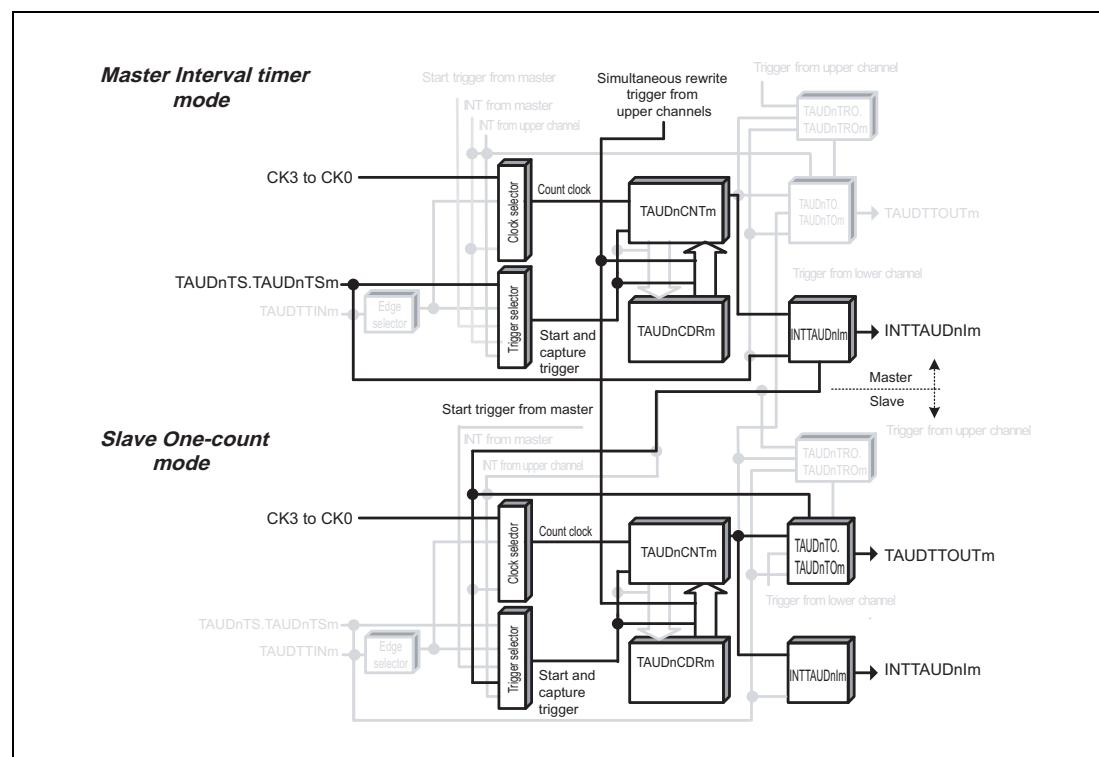


Figure 23.90 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

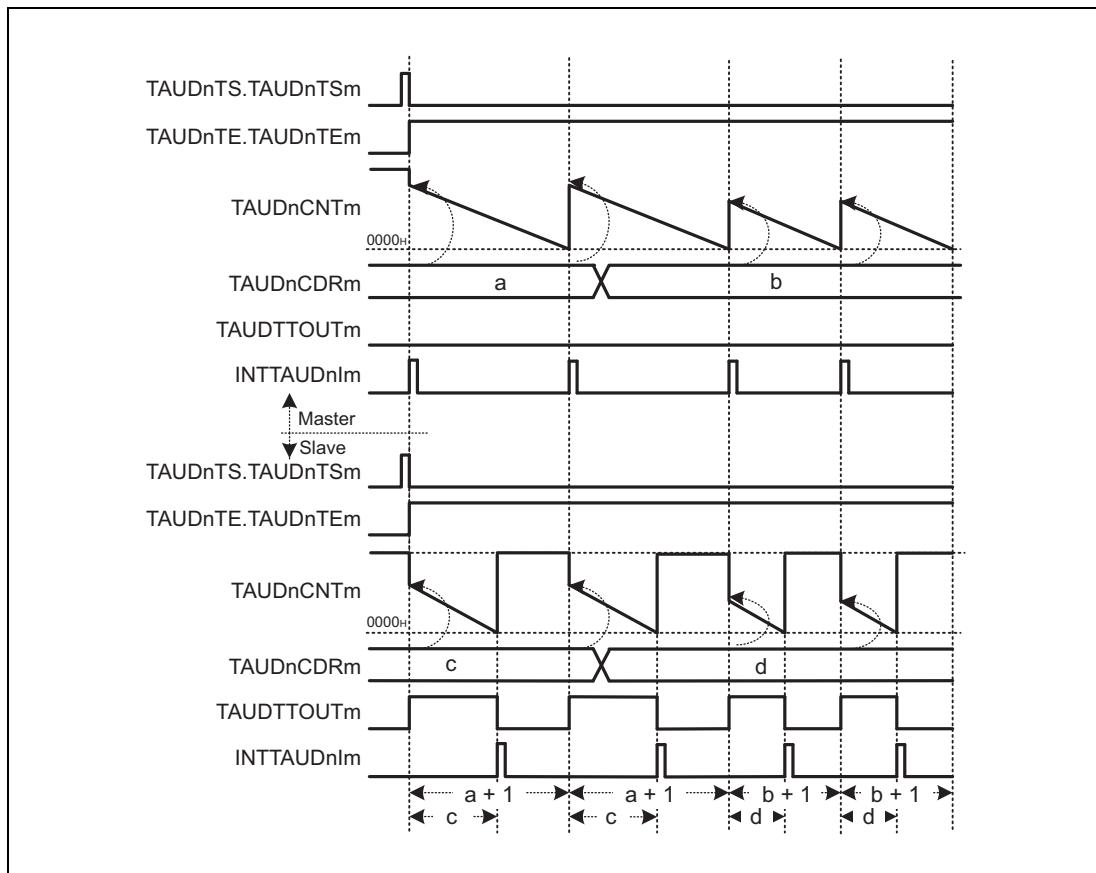


Figure 23.91 General Timing Diagram of PWM Output Function

#### NOTES

1. The interval between the starting to count and an interrupt being generated is the value of corresponding TAUDnCDRm + 1.
2. TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnlm of the master channel.

### 23.15.1.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.141 Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.142 Contents of the TAUDnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

The channel output mode is not used with this function.

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.143 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

#### NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires a channel higher than the master channel that operates with **Section 23.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions:

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1  
TAUDnCDRm settings for this channel are as follows:  
 $= ((\text{TAUDnCDR setting for the master channel targeted for simultaneous rewrite} + 1) \times \text{interrupt count}) - 1$
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

If TAUDnCDRm (slave) setting > TAUDnCDRm (master) setting + 1, the duty value (which exceeds 100%) is aggregated to be 100% output.

### 23.15.1.5 Register Settings for Slave Channels

#### (1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.144 Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Start trigger during operation is valid.

#### (2) TAUDnCMURm for slave channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 23.145 Contents of the TAUDnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channels

Table 23.146 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMM	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLM	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMM	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLM	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set these bits to 0
TAUDnTRC.TAUDnTRCM	
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.147 Simultaneous Rewrite Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSM	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMM	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCM	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.1.6 Operating Procedure for PWM Output Function

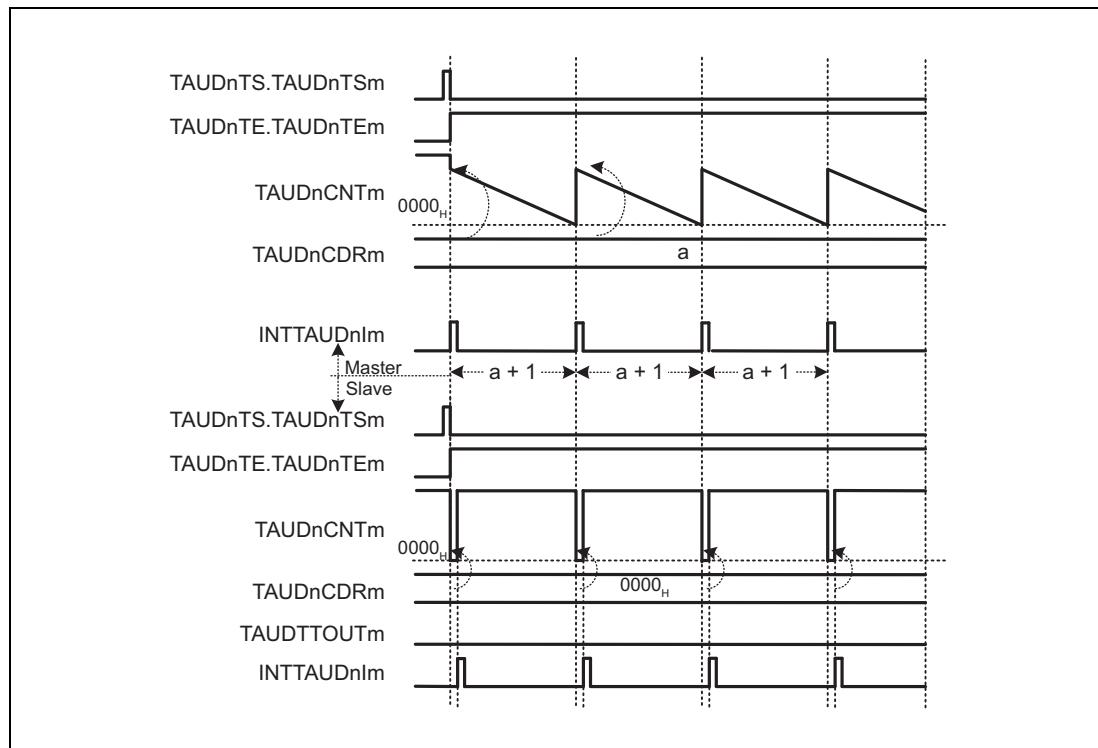
Table 23.148 Operating Procedure for PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.1.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.1.5, Register Settings for Slave Channels</b>.</p> <p>Set the value of TAUDnCDRm register of every channel.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set.</p>
During operation	<p>TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (master) is generated.</li> <li>TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation.</li> <li>TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down.</li> <li>TAUDTTOUTm (slave) is set to the active level.</li> </ul> <p>If TAUDnCNTm (slave) reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (slave) is generated.</li> <li>TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

Restart Operation

### 23.15.1.7 Specific Timing Diagrams

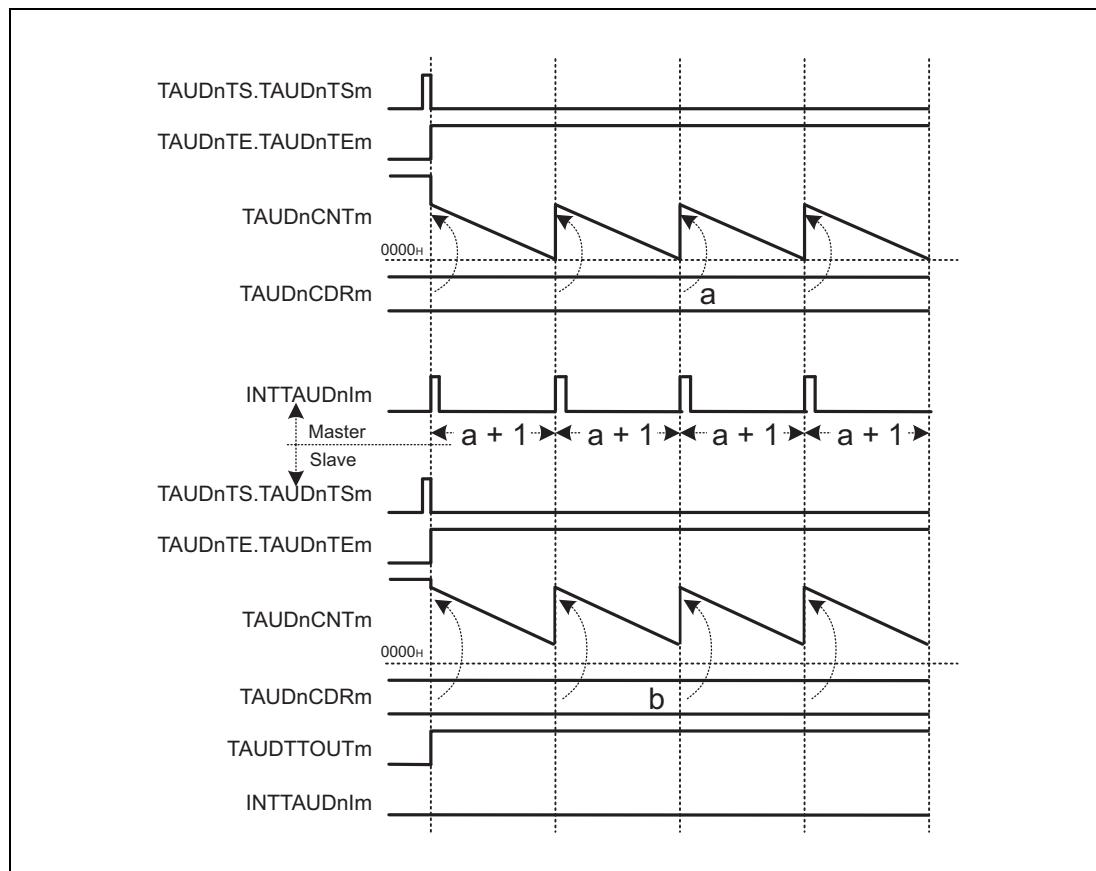
#### (1) Duty cycle = 0%



**Figure 23.92 TAUDnCDRm (Slave) = 0000<sub>H</sub>,  
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)**

- Every time the master channel generates an interrupt (INTTAUDnlm), 0000<sub>H</sub> is loaded into TAUDnCNTm (slave). As a result, a slave channel interrupt (INTTAUDnlm) is generated at the same time and TAUDTTOUTm remains inactive.
- TAUDnCDRm value is loaded into TAUDnCNTm (slave) to generate an interrupt.

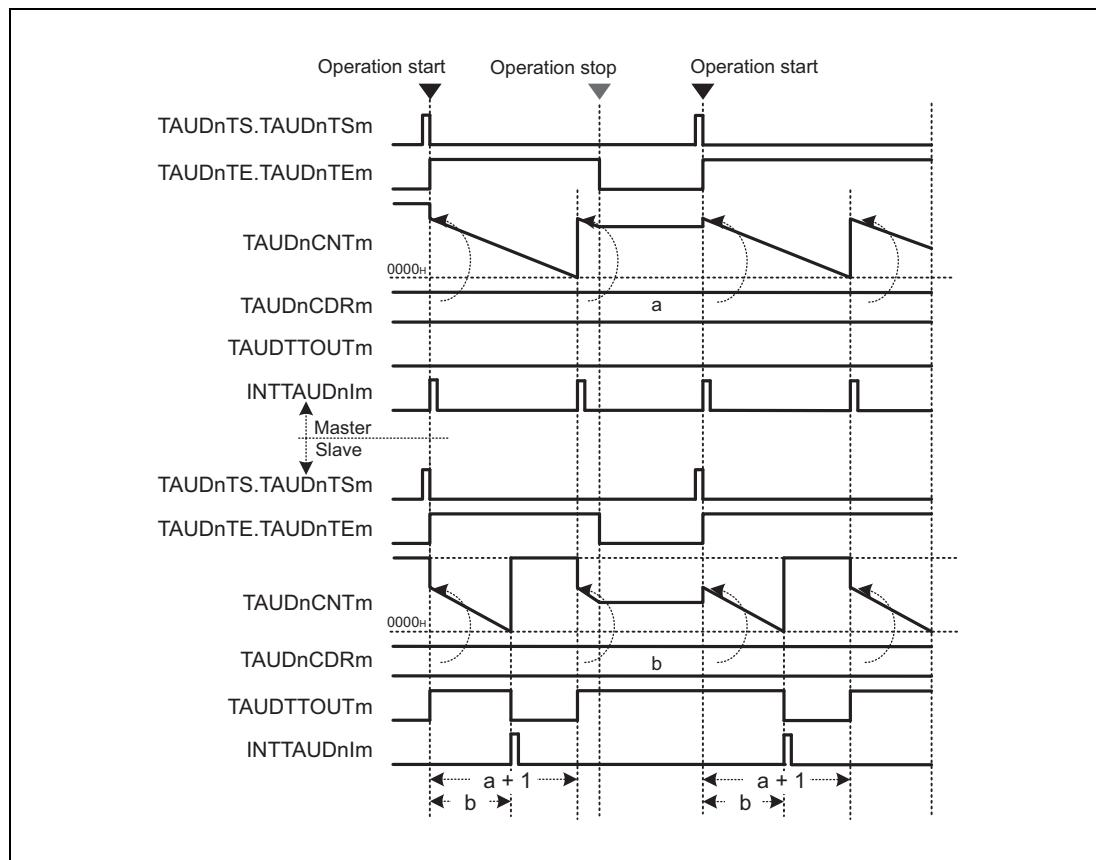
## (2) Duty cycle = 100%



**Figure 23.93 TAUDnCDRm (Slave)  $\geq$  TAUDnCDRm (Master) + 1  
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)**

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value, the slave channel counter does not reach  $0000_H$  and consequently, no interrupt occurs. TAUDTTOUTm remains active.

### (3) Operation stop and restart



**Figure 23.94 Operation Stop and Restart  
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)**

- The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm of master and slave channels to 1. TAUDnCDRm values of the master and slave channels are loaded to TAUDnCNTm and start to count down from these values.

## 23.15.2 One-Shot Pulse Output Function

### 23.15.2.1 Overview

#### Summary

This function outputs a signal pulse with a specific pulse width and delay time (both defined relative to an external input signal pulse) by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

#### Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See **Table 23.149, Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function.**)
- The operating mode for slave channels should be set to pulse one-count mode. (See **Table 23.152, Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function.**)
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channel should be set to independent channel output mode 2. (See **Section 23.7, Channel Output Modes.**)
- TAUDTTINm (master) has to be detected while TAUDnCNTm (master) and TAUDnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUDTTINm (slave).

#### Functional description

The counters are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1 for the master and slave channels. This sets TAUDnTE.TAUDnTEM to 1, enabling count operation.

- Master channel:

When the next valid TAUDTTINm input edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm. The counter starts to count down from this value. If TAUDnCMORm.TAUDnMD0 = 0, a trigger (TAUDTTINm) which is detected within the delay time is ignored.

When the counter of master channel reaches 0000<sub>H</sub>, INTTAUDnIm is generated. The counter is reset to FFFF<sub>H</sub> and waits for the next valid TAUDTTINm input edge.

- Slave channel:

INTTAUDnIm generated on master channel triggers the counter operation of slave channel. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUDTTOUTm signal is set.

When the counter reaches 0001<sub>H</sub>, INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter stops at 0000<sub>H</sub> and waits for the next INTTAUDnIm of master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Setting TAUDnTS.TAUDnTSm to 1 while counting allows the counter to restart counting of master channel without making a stop (forced restart).

### Conditions

- If TAUDnCMORn.TAUDnMD0 of master channel is set to 0, TAUDTTINm input edges detected during counting are ignored.
- Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

### 23.15.2.2 Equations

Delay from trigger input to pulse output  
 $= (\text{TAUDnCDRm (master)} + 1) \times \text{count clock cycle}$

Pulse width =  $(\text{TAUDnCDRm (slave)}) \times \text{count clock cycle}$

### 23.15.2.3 Block Diagram and General Timing Diagram

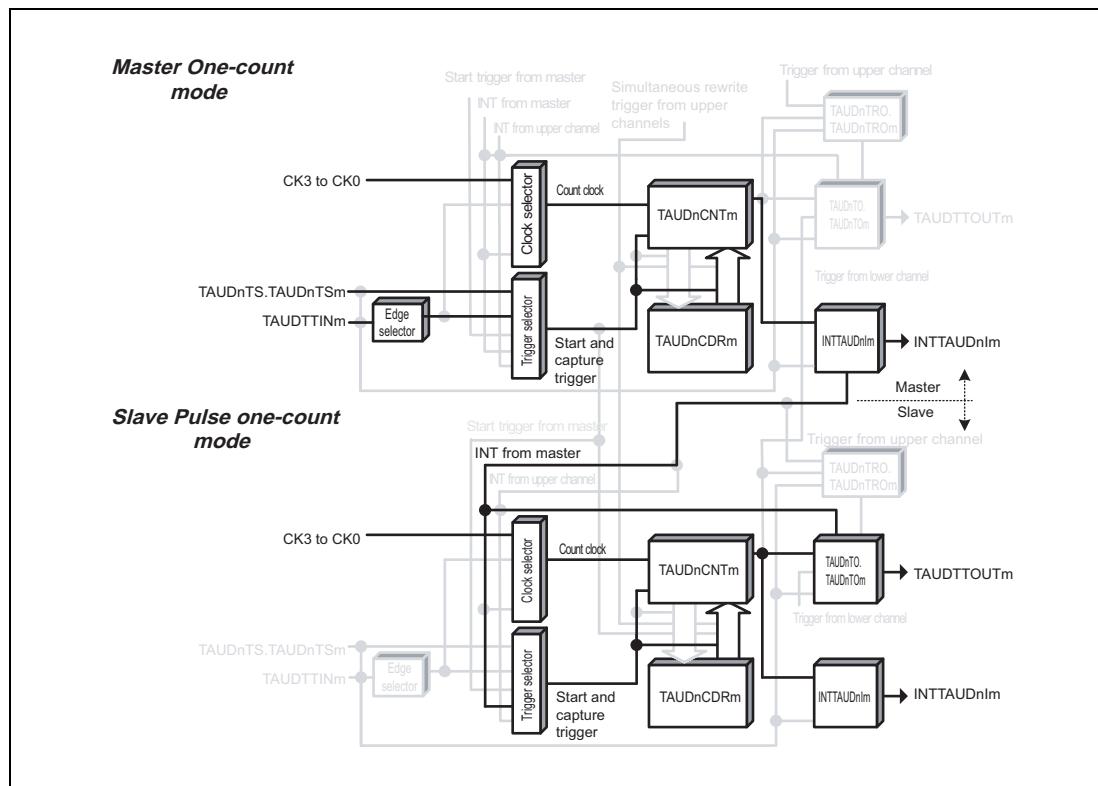


Figure 23.95 Block Diagram of One-Shot Pulse Output Function

The following settings apply to the general timing diagram.

- Start trigger detection is disabled during counting ( $\text{TAUDnCMORm.TAUDnMD0} = 0$ ).
- Detection of falling edge ( $\text{TAUDnCMURm.TAUDnTIS[1:0]} = 00_B$ )

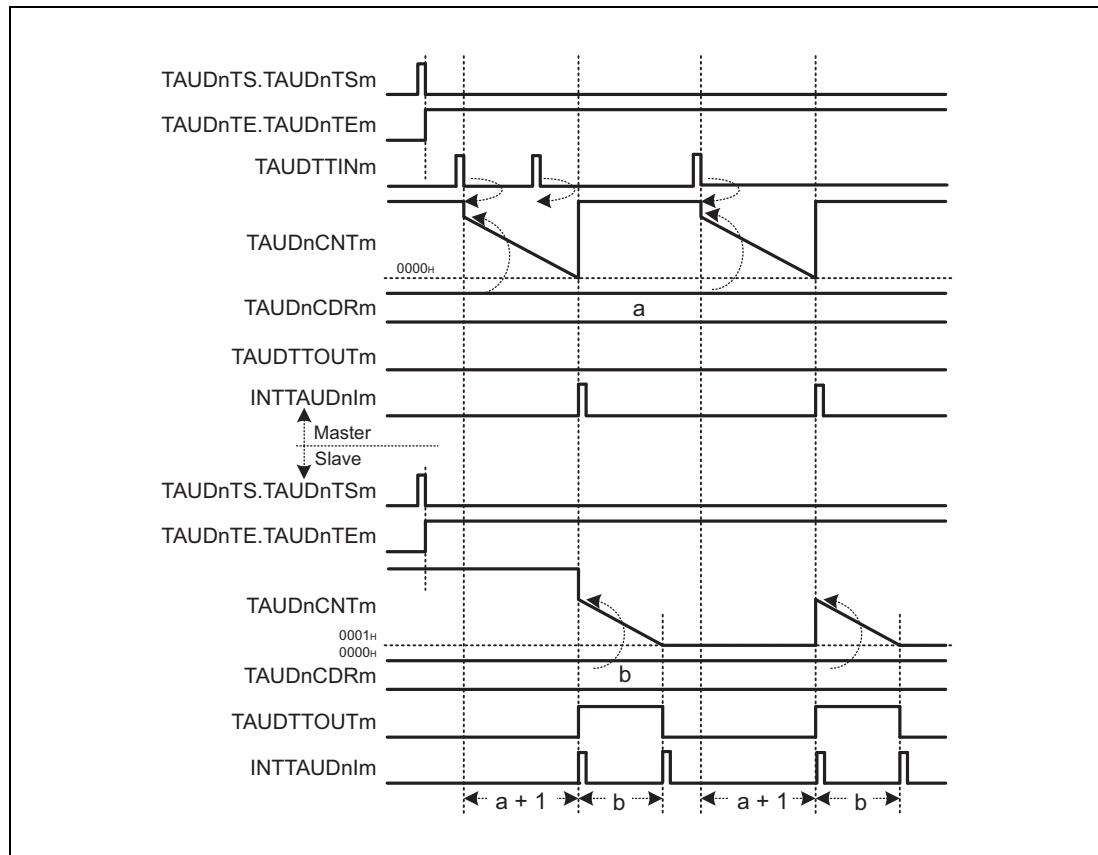


Figure 23.96 General Timing Diagram of One-Shot Pulse Output Function

### 23.15.2.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.149 Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. The value of the MDO bits of the master and slave channels must be identical.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.150 Contents of the TAUDnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

**(3) Channel output mode for the master channel**

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

**(4) Simultaneous rewrite for the master channel**

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.151 Simultaneous Rewrite Settings for the Master Channel of One-Shot Pulse Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.2.5 Register Settings for Slave Channels

#### (1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.152 Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMDO	0: Disables detection of start trigger during count operation. 1: Enables start trigger detection while counting. The MDO bits of the master and slave channels must be identical.

#### (2) TAUDnCMURm for slave channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.153 Contents of the TAUDnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the slave channel

Table 23.154 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

### (4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.155 Simultaneous Rewrite Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 23.156 Operating Procedure for One-Shot Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.2.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in <b>Section 23.15.2.5, Register Settings for Slave Channels</b>.</p> <p>Set the value of TAUDnCDRm register of every channel.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	TAUDnTE.TAUDnTEM (master and slave channels) is set to 1 and the master channel awaits a TAUDTTINm input.
During Operation	<p>TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>When valid TAUDTTINm input edge is detected, TAUDnCDRm value of master channel is loaded into TAUDnCNTm to perform counting down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (master) is generated.</li> <li>TAUDnCNTm (master) is reset to FFFFH and waits for the next valid TAUDTTINm input edge.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down.</li> <li>INTTAUDnIm (slave) is generated.</li> <li>TAUDTTOUTm (slave) is set to the active level.</li> </ul> <p>When TAUDnCNTm (slave) reaches 0001H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (slave) is generated.</li> <li>TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

Restart Operation →

### 23.15.2.7 Specific Timing Diagrams

#### (1) TAUDnCDRm (master) = 0000<sub>H</sub>

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

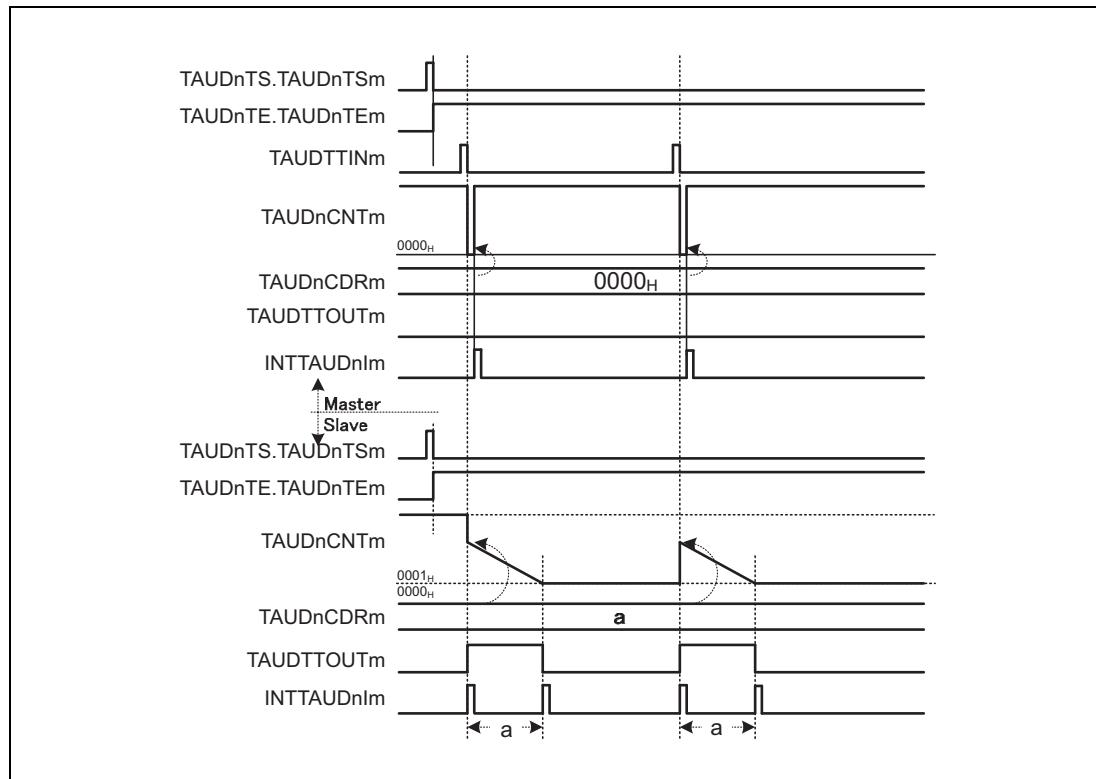


Figure 23.97 TAUDnCDRm (Master) = 0000<sub>H</sub>

- When a valid TAUDTTINm input edge is detected, the value 0000<sub>H</sub> is written to TAUDnCNTm (master). The counter is set to 0000<sub>H</sub> for one count and returns to FFFF<sub>H</sub>.  
Thus the slave channel starts to count down one count clock later than TAUDTTINm (master).

**(2) TAUDnCDRm (slave) = 0000<sub>H</sub>**

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

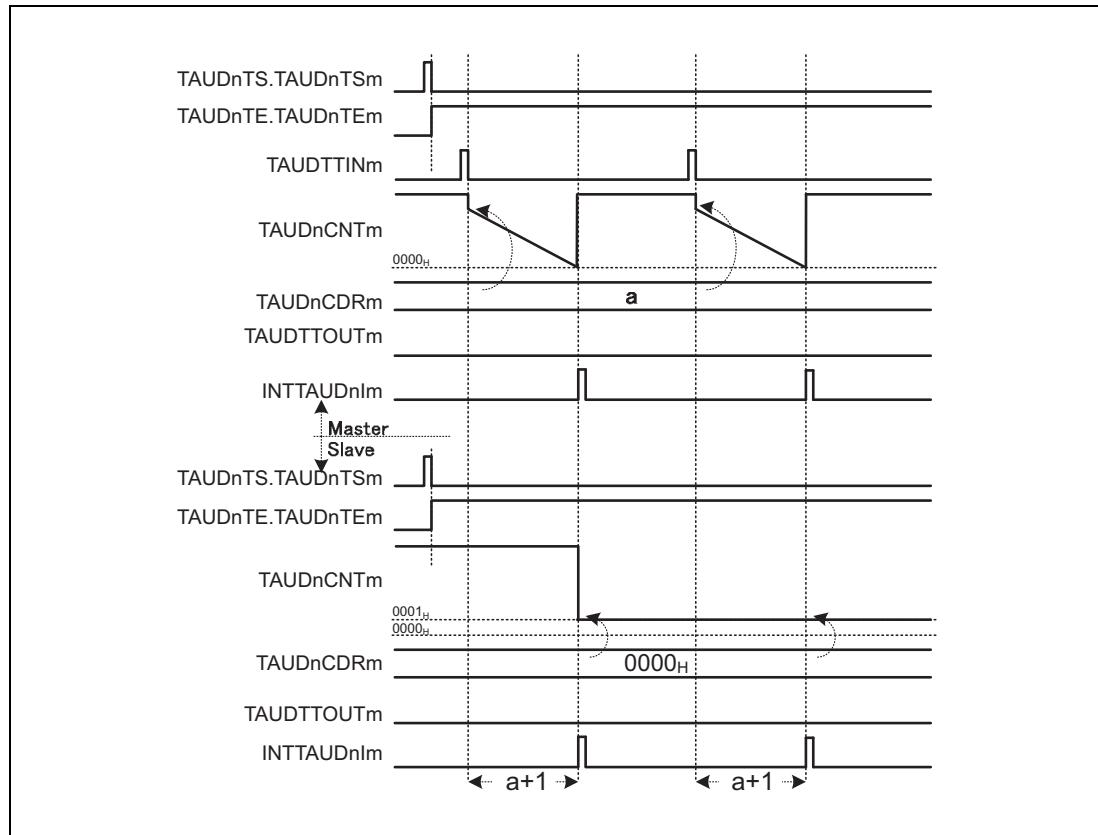


Figure 23.98 TAUDnCDRm (Slave) = 0000<sub>H</sub>

- TAUDTTOUT<sub>m</sub> remains inactive, because the pulse width is zero.

### (3) TAUDnCMORm.TAUDnMD0 = 1

The following settings apply to this diagram.

- Enables start trigger detection while counting. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

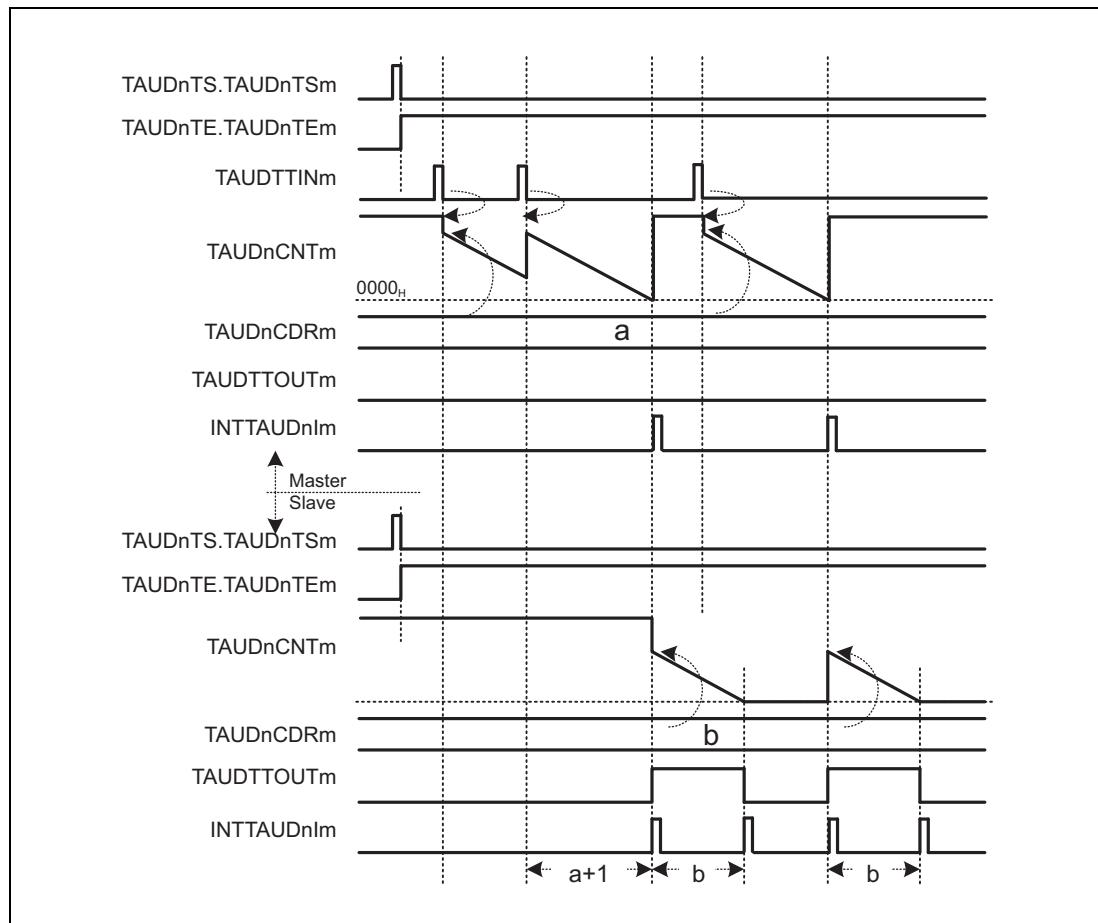


Figure 23.99 TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDTTINm input edge is detected while the counter of the master channel counts down, TAUDnCNTm reloads the value of TAUDnCDRm. The counter restarts to count down. This means the delay is extended by the value of TAUDnCNTm at the time when a valid TAUDTTINm input edge is detected.

#### (4) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

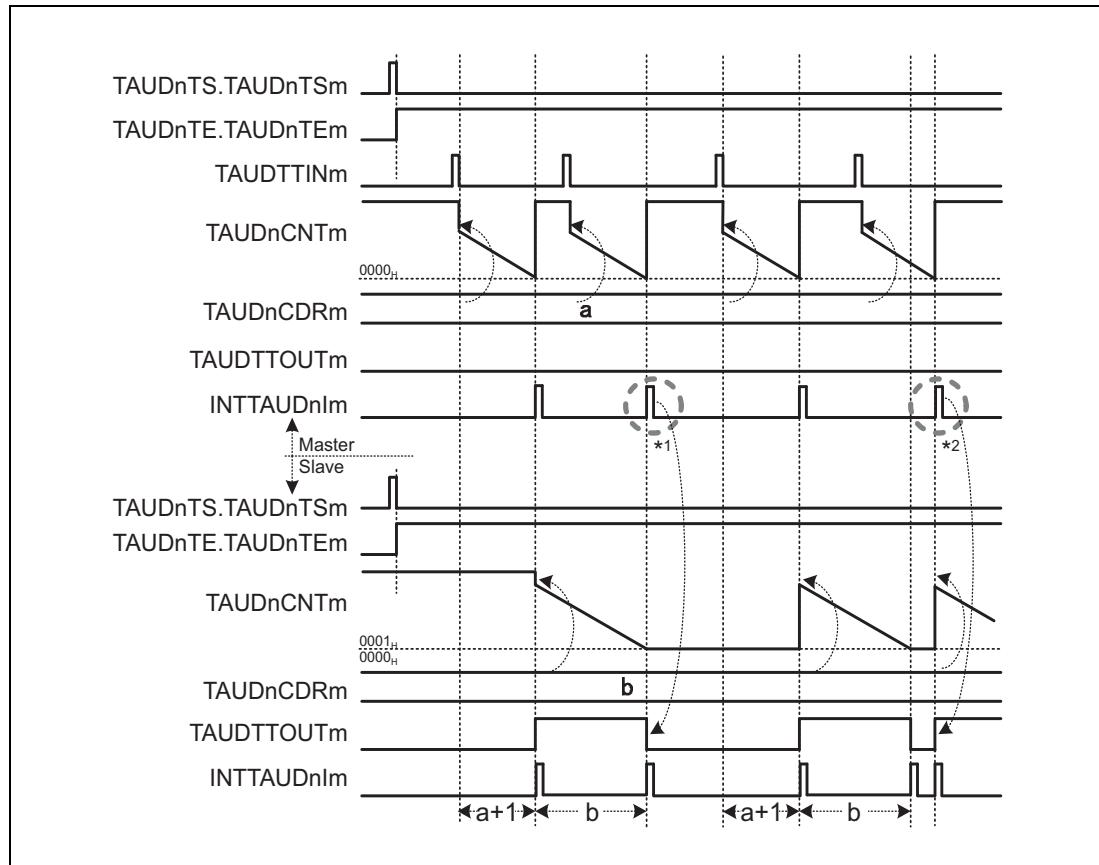


Figure 23.100 TAUDTTINm input interval  $\leq$  Delay Time + Pulse Width +1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001<sub>H</sub> or exactly when 0001<sub>H</sub> is reached<sup>\*1</sup>, the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUDnCDRm (slave) is reloaded. An interrupt is generated and TAUDTTOUTm toggles. If TAUDnCNTm (master) has started to count down while the TAUDnCNTm (slave) is still counting<sup>\*2</sup>, TAUDTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

### 23.15.3 Trigger Start PWM Output Function

#### 23.15.3.1 Overview

##### Summary

This function generates a PWM output using a master and a slave channel. It enables the pulse cycle (frequency) and the duty of the TAUDTTOUTm to be set. The pulse cycle is specified using the master channel. The duty is specified using the slave channel. The Trigger Start PWM Output Function is identical to PWM Output Function except that the master channel of this function can be reset by a valid TAUDTTINm input edge.

##### Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode (see **Table 23.157, Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function**).
- The operation mode of the slave channel must be set to One-Count Mode (see **Table 23.160, Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function**).
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 1 (see **Section 23.7, Channel Output Modes**).
- TAUDTTOUTm is not used with the master channel of this function.

##### Functional description

The counters (master and slave) are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEM to 1, enabling count operation. The current value of TAUDnCDRm is loaded to TAUDnCNTm, and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel, and a PWM output is realized by setting and resetting TAUDTTOUTm (slave).

- Master channel:

The current value of TAUDnCDRm is loaded to the counter (TAUDnCNTm), INTTAUDnIm is generated and the counter starts to count down from this value.

When the counter reaches  $0000_H$  and the pulse cycle time has elapsed, INTTAUDnIm is generated and the counters (master and slave) load the current TAUDnCDRm values.

If a valid TAUDTTINm input edge is detected, the counter of the master channel loads the current TAUDnCDRm value, restarts counting down and generates an interrupt.

- Slave channel:

When the slave detects an interrupt from the master channel, it starts to count down from the current value of TAUDnCDRm. The TAUDTTOUTm signal is set to the active level.

When the counter reaches  $0000_H$  (duty time has elapsed), INTTAUDnIm is generated and the TAUDTTOUTm signal is reset. The counter returns to  $FFFF_H$  and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channels, which in turn sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

### Conditions

Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

#### 23.15.3.2 Equations

$$\text{Pulse cycle} = (\text{TAUDnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$\text{Duty cycle [%]} = [\text{TAUDnCDRm (slave)} / (\text{TAUDnCDRm (master)} + 1)] \times 100$$

- Duty cycle = 0%  
TAUDnCDRm (slave) = 0000<sub>H</sub>
- Duty cycle = 100%  
TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

#### 23.15.3.3 Block Diagram and General Timing Diagram

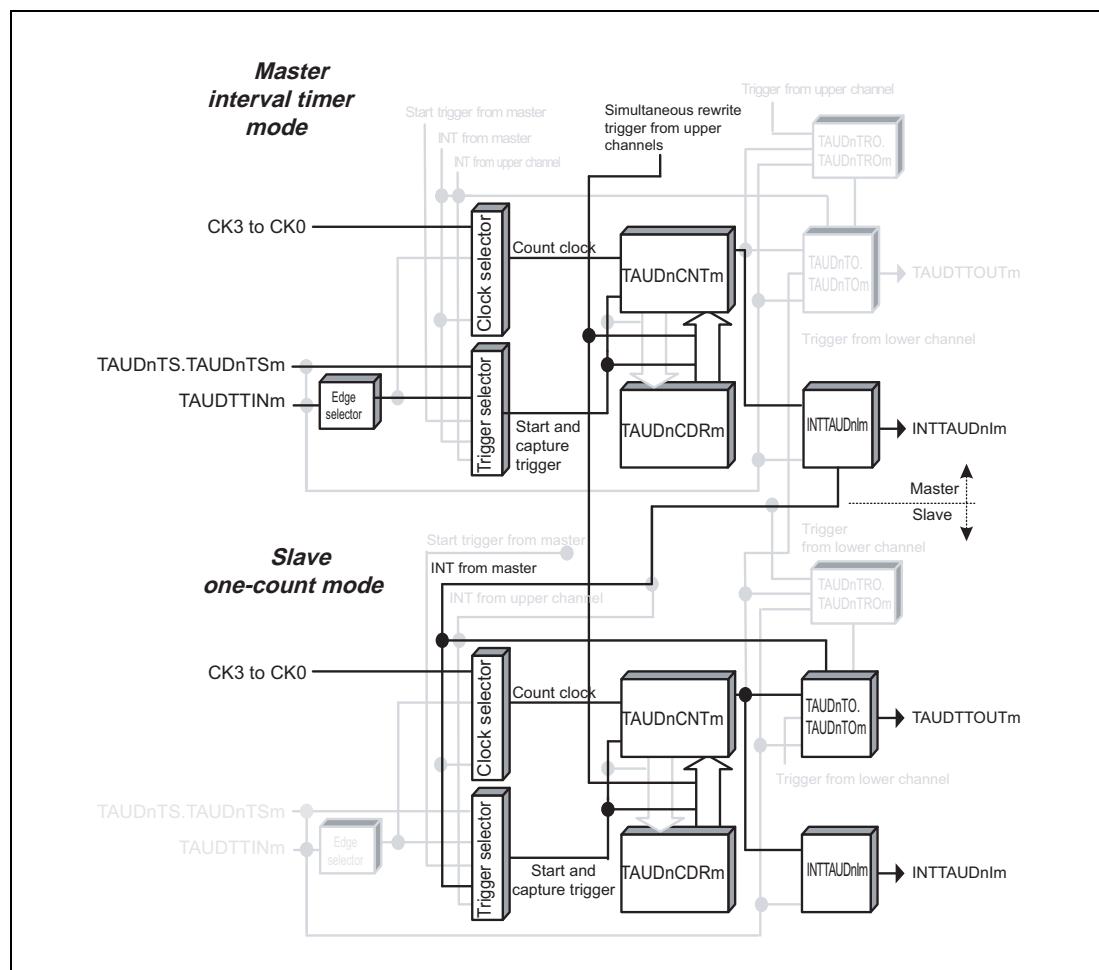


Figure 23.101 Block Diagram for Trigger Start PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01<sub>B</sub>)
- Positive logic (TAUDnTOL.TAUDnTOLm (slave) = 0)

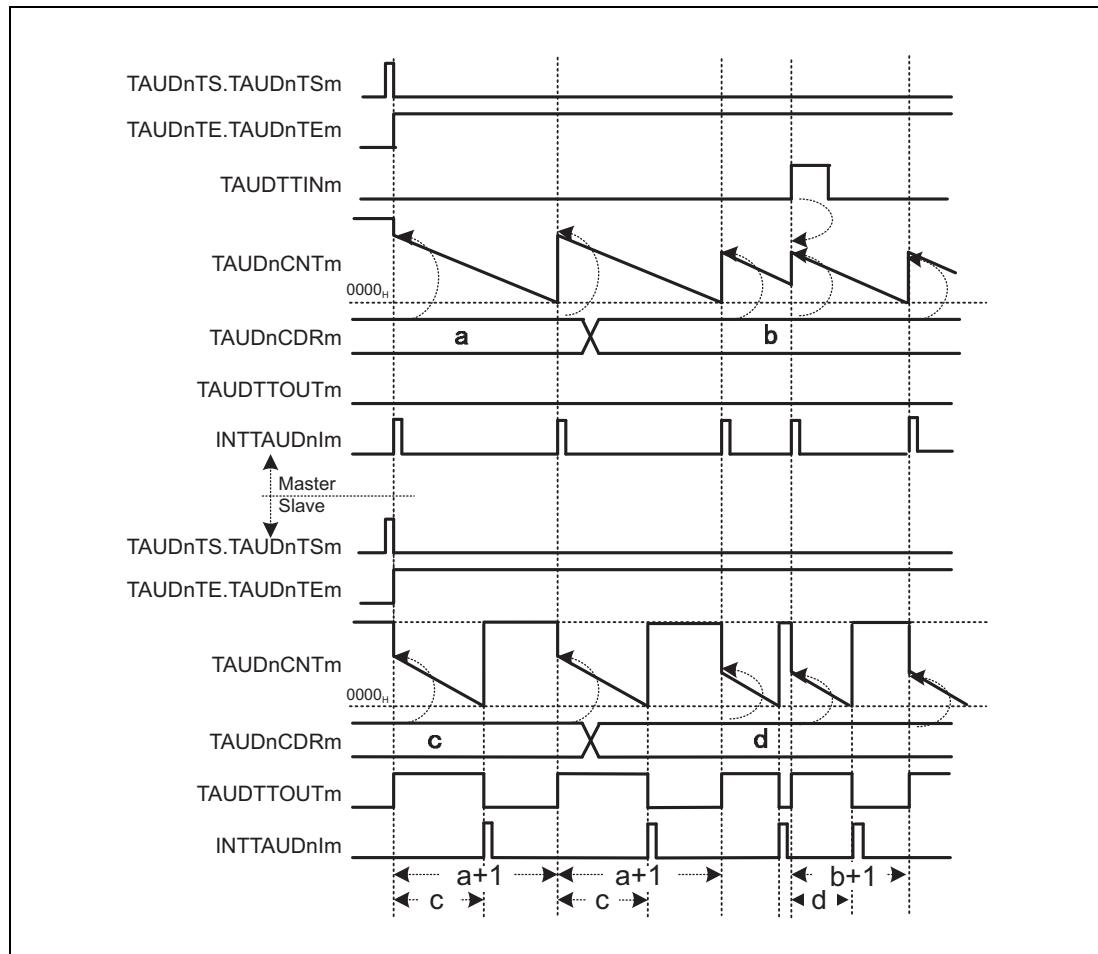


Figure 23.102 General Timing Diagram for Trigger Start PWM Output Function

#### NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

### 23.15.3.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.157 Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.158 Contents of the TAUDnCMURm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

**(3) Channel output mode for the master channel**

The channel output mode is not used by this function.

**(4) Simultaneous rewrite for the master channel**

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.159 Simultaneous Rewrite Settings for the Master Channel of the Trigger Start PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.3.5 Register Settings for Slave Channels

#### (1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.160 Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Start trigger during operation is valid. The value of the TAUDnMD[0] bit of the master and slave channels must be identical.

#### (2) TAUDnCMURm for slave channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.161 Contents of the TAUDnCMURm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the slave channel

**Table 23.162 Control Bit Settings in Independent Channel Output Mode 1**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMM	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLM	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMM	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLM	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCM	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.163 Simultaneous Rewrite Settings for the Slave Channel of the Trigger Start PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSM	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMM	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCM	0: Does not operate as a simultaneous rewrite trigger generation channel.

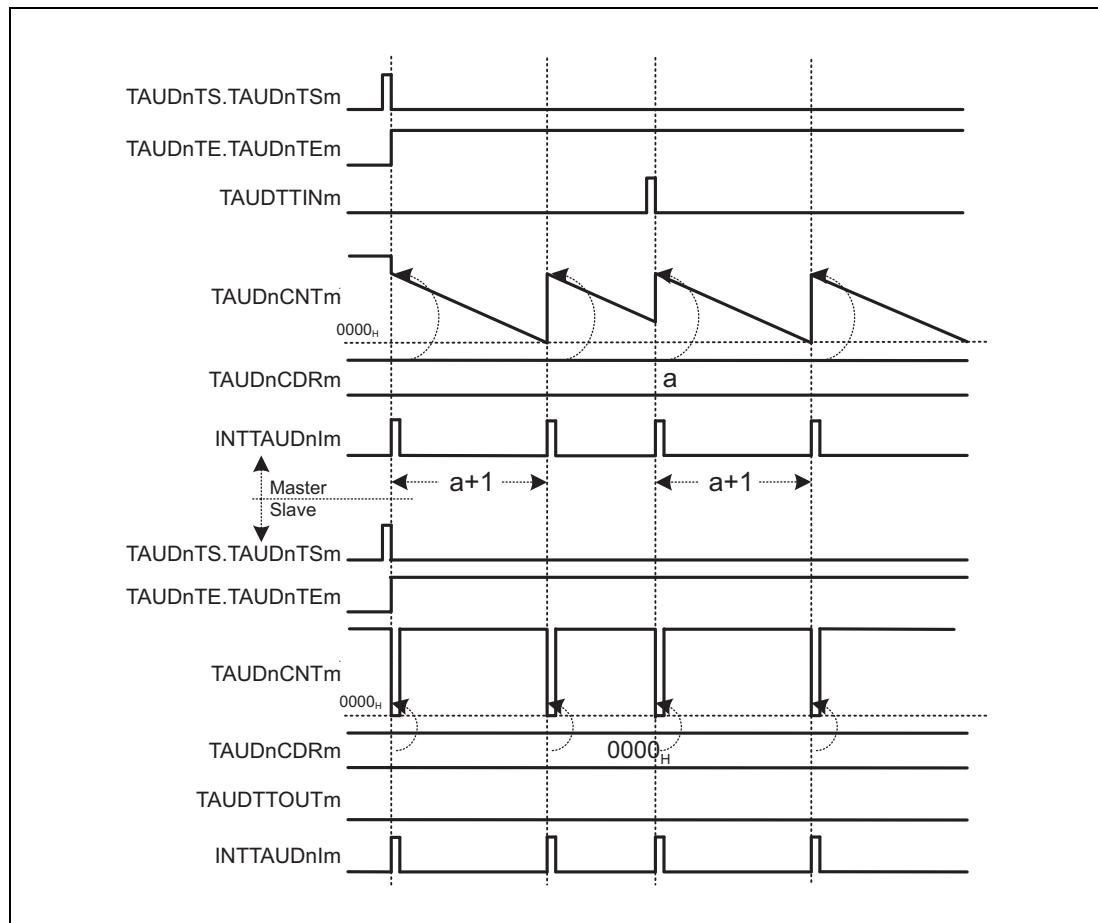
### 23.15.3.6 Operating Procedure for Trigger Start PWM Output Function

Table 23.164 Operating Procedure for Trigger Start PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.3.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.3.5, Register Settings for Slave Channels</b>.</p> <p>Set the value of TAUDnCDRm register of every channel.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.</p>
During Operation	<p>TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (master) is generated.</li> <li>TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation.</li> <li>TAUDnCNTm (slave) loads the TAUDnCDRm value and starts to count down</li> <li>TAUDTTOUTm (slave) is set</li> </ul> <p>When TAUDnCNTm of the slave = 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (slave) is generated.</li> <li>TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.</li> </ul> <p>If a TAUDTTINm input is detected on the master channel while TAUDnCNTm of the master channel is counting down:</p> <ul style="list-style-type: none"> <li>TAUDnCNTm (master and slave) loads the TAUDnCDRm value and counts down</li> <li>INTTAUDnIm (master) is generated.</li> <li>TAUDTTOUTm (slave) is set to the active level.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

### 23.15.3.7 Specific Timing Diagrams

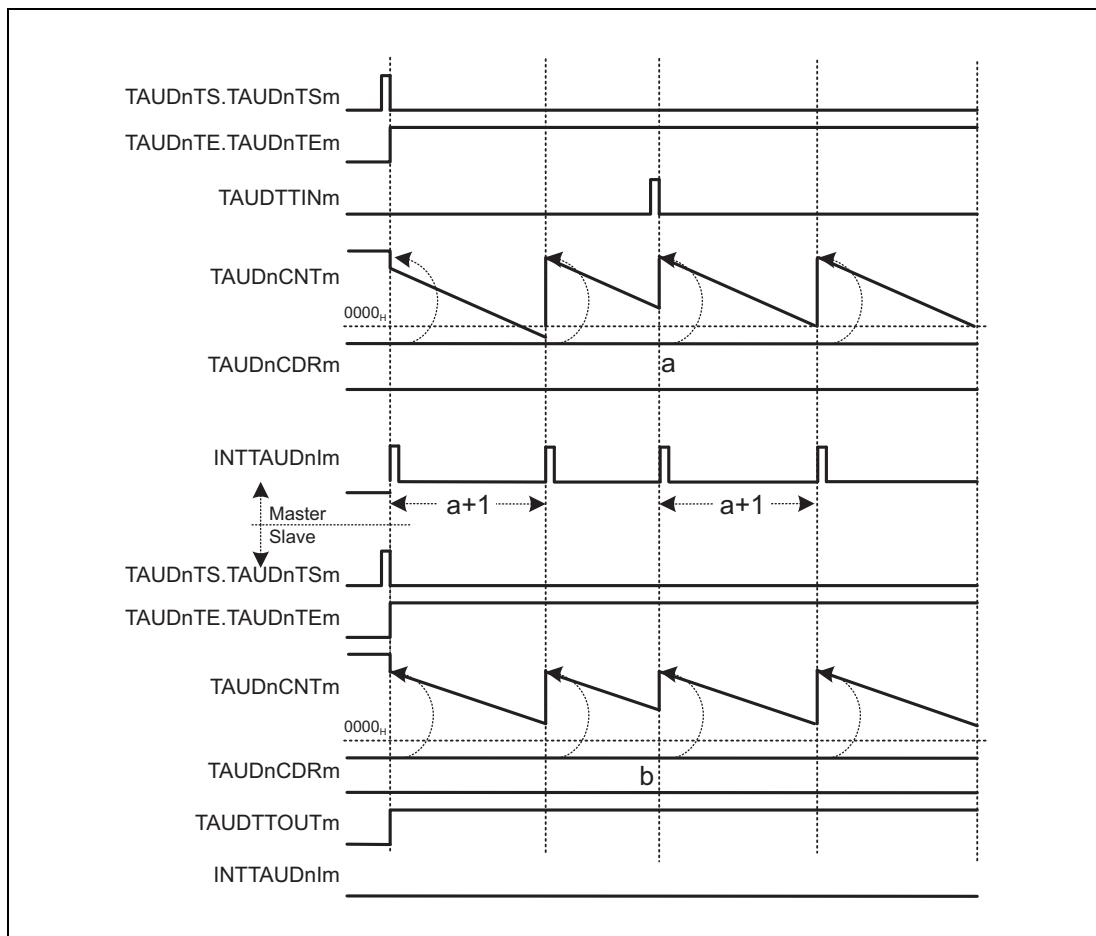
#### (1) Duty cycle = 0%



**Figure 23.103 TAUDnCDRm (Slave) = 0000<sub>H</sub>,  
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)  
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)**

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000<sub>H</sub> is written to TAUDnCNTm (slave). Therefore, TAUDnCNTm (slave) cannot start to count and TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

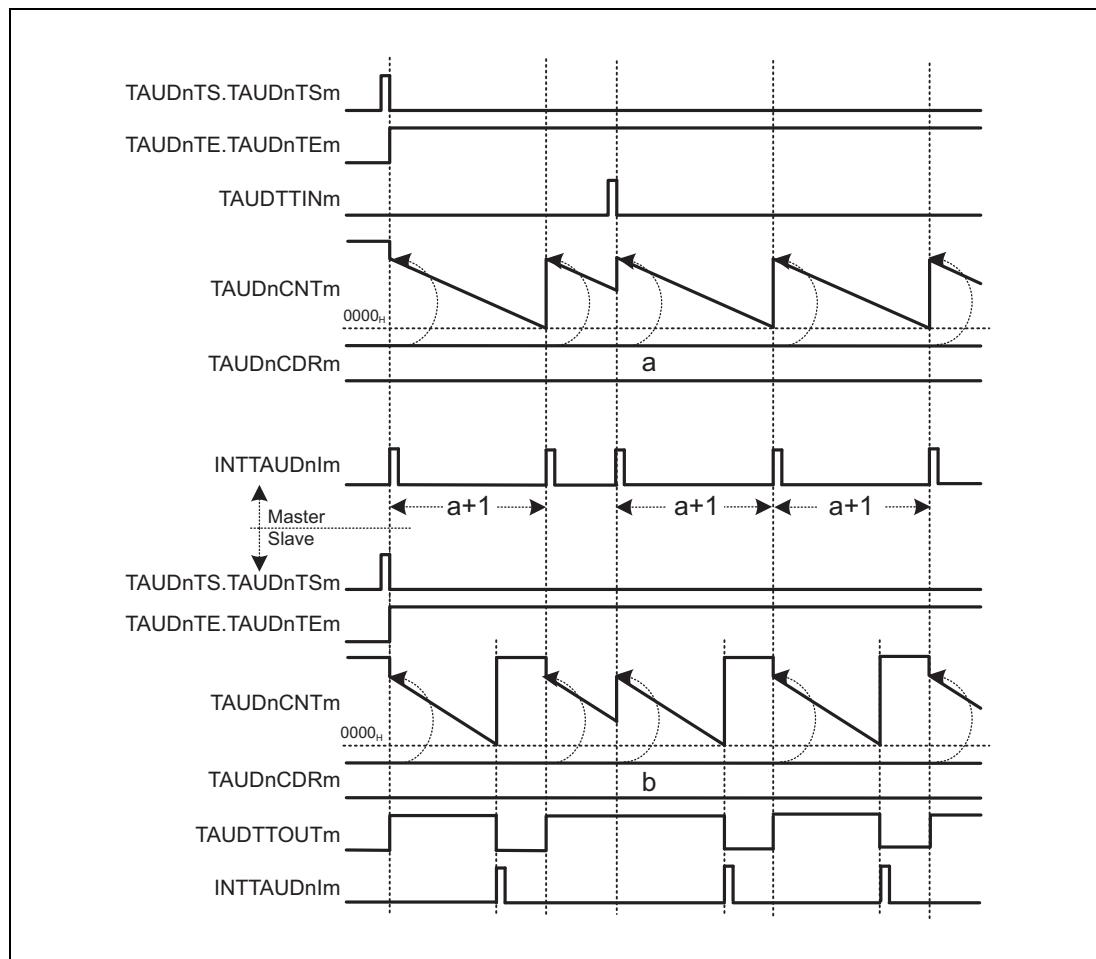
## (2) Duty cycle = 100%



**Figure 23.104 TAUDnCDRm (Slave)  $\geq$  TAUDnCDRm (Master) + 1,  
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)  
Falling Edge Detection (TAUDnCMURm.TIS[1:0] = 00<sub>B</sub>)**

- If the value TAUDnCDRm (slave) is higher than the value TAUDnCDRm (master), the counter of the slave channel cannot reach 0000<sub>H</sub> and cannot generate interrupts.  
The TAUDTTOUTm remains at active state.  
The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

## (3) TAUDTTINm detection and active slave counter



**Figure 23.105 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)  
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)**

- If TAUDnCNT<sub>m</sub> (slave) loads the value TAUDnCDR<sub>m</sub> (slave) while it is still counting down, TAUDTTOUT<sub>m</sub> remains unchanged and extends the duty.  
The duty does not correspond to the value of the slave's data register.

## 23.15.4 Delay Pulse Output Function

### 23.15.4.1 Overview

#### Summary

This function outputs two signals. The pulse width and pulse cycle of the reference signal are defined using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by the amount specified on slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified on slave channel 2.

#### Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 23.165, Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function.**)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See **Table 23.168, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function** and **Table 23.172, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function.**)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See **Table 23.175, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function.**)
- TAUDTTOUTm is not used with the master channel and slave channel 2.
- The channel output mode for slave channel 1 should be set to synchronous channel output mode 1. (See **Section 23.7, Channel Output Modes.**)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See **Section 23.7, Channel Output Modes.**)

#### Functional description

The counters of the channel group are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM to 1, enabling count operation.

- Master channel:  
The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel.  
When the counter value of master channel reaches  $0000_H$  and pulse cycle time has elapsed, INTTAUDnIm is generated. The TAUDnCDRm value is reloaded into the counter to perform counting down.
- Slave channels 1 and 2:  
Slave channels 1 and 2 start to count down from the current TAUDnCDRm value when detecting

an interrupt from the master channel. TAUDTTOUTm signal (slave 1) is set.

- Slave channel 1:

When the counter of slave channel 1 reaches  $0000_H$  (duty time has elapsed), INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter is reset to  $FFFF_H$  and waits for the next INTTAUDnIm of master channel.

- Slave channel 2:

When the counter of slave channel 2 reaches  $0000_H$  and delay time has elapsed, INTTAUDnIm is generated. The counter is reset to  $FFFF_H$  and waits for the next INTTAUDnIm of master channel.

Generating INTTAUDnIm (slave channel 2) triggers the counter of slave channel 3.

- Slave channel 3:

When slave channel 3 detects an interrupt from slave channel 2, its counter starts counting down from the current value of TAUDnCDRm. INTTAUDnIm is generated and the TAUDTTOUTm signal (slave channel 3) is set.

When the counter of slave channel 3 reaches  $0001_H$ , INTTAUDnIm is generated and the TAUDTTOUTm signal is reset.

The delayed PWM pulse is output from slave channel 3.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

### Conditions

Simultaneous rewrite can be used with this function. See **Section 23.6, Simultaneous Rewrite**.

#### 23.15.4.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUDnCDRm (slave 1)) × count clock cycle

Delay width = (TAUDnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUDnCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:

$0000_H \leq \text{TAUDnCDRm (slave 2)} < \text{TAUDnCDRm (master)}$

### NOTES

1. The waveform of TAUDTTOUTm (slave 3) becomes the waveform made by delaying the waveform of TAUDTTOUTm (slave 1) by the quantity generated by slave 2. It is impossible to make a delay longer than the pulse cycle.
2. If INTTAUD0Im of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the waveform of TAUDTTOUTm (slave 3) is retained on the active level. In this case, TAUDTTOUTm (Slave-CH-3) cannot output the waveform generated by delaying the basic pulse of TAUDTTOUTm (Slave-CH-1).

### 23.15.4.3 Block Diagram and General Timing Diagram

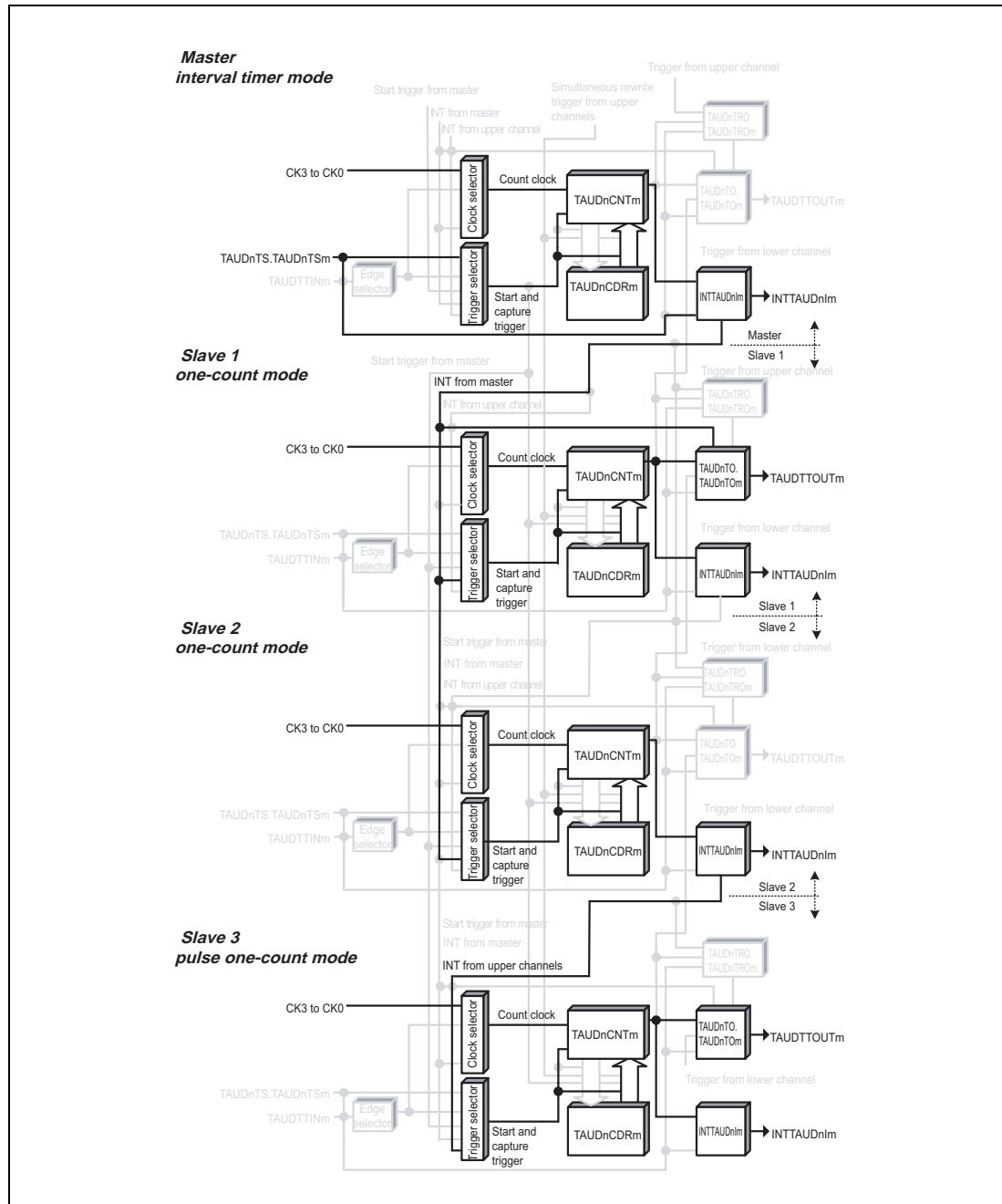


Figure 23.106 Block Diagram of Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUDnTOL.TAUDnTOLm = 0)
- Slave channel 3: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

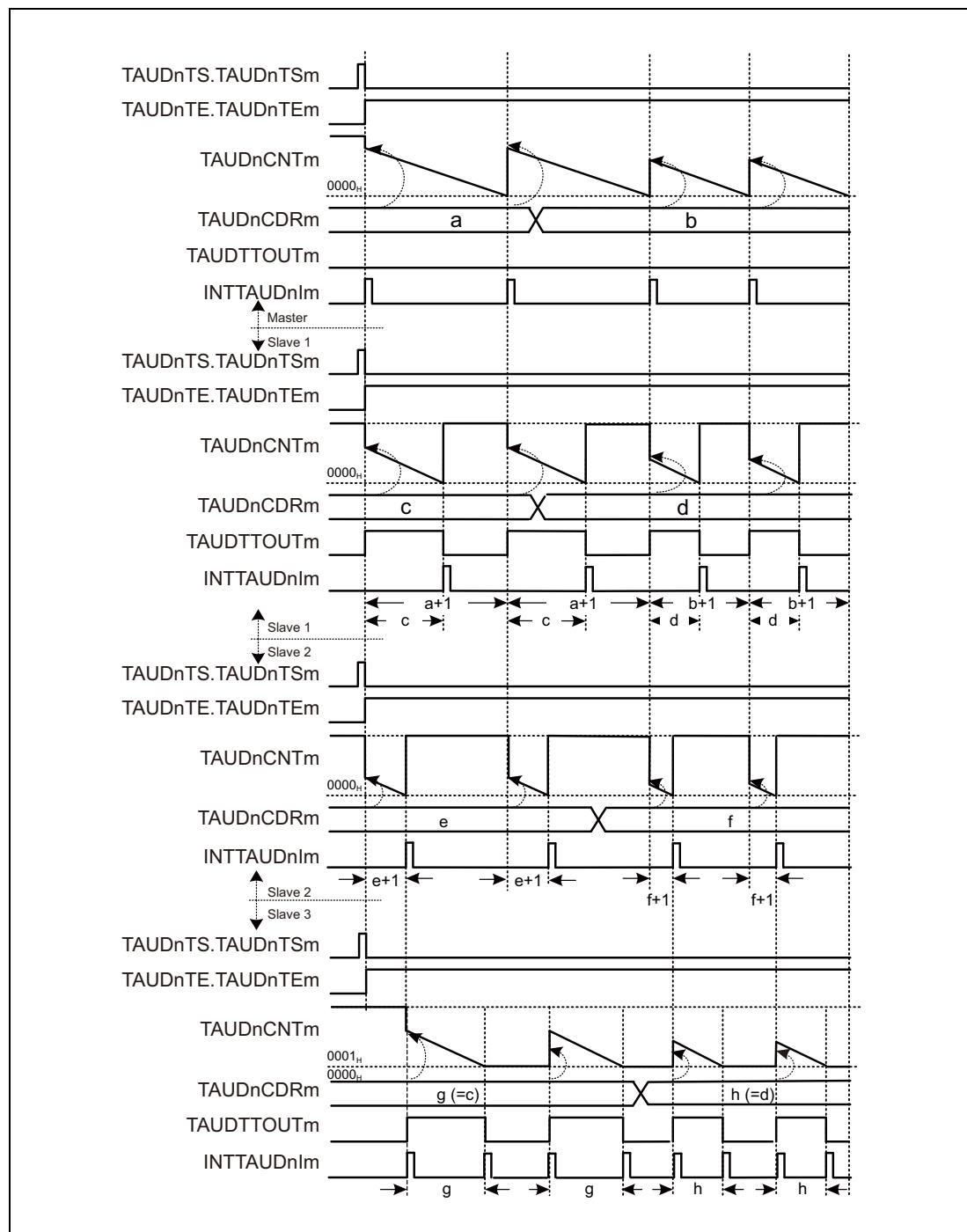


Figure 23.107 General Timing Diagram of Delay Pulse Output Function

#### NOTE

TAUDTTOUTm of slave channel 1 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

### 23.15.4.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.165 Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.166 Contents of the TAUDnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

**(3) Channel output mode for the master channel**

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used for the master channel with this function.

**(4) Simultaneous rewrite for the master channel**

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.167 Simultaneous Rewrite Settings for the Master Channel of Delay Pulse Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.4.5 Register Settings for Slave Channel 1

#### (1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.168 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Valid start trigger during operation

#### (2) TAUDnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.169 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channel 1

**Table 23.170 Control Bit Settings for Slave Channel 1 in Synchronous Channel Output Mode 1**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.171 Simultaneous Rewrite Settings for Slave Channel 1 of Delay Pulse Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.4.6 Register Settings for Slave Channel 2

#### (1) TAUDnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.172 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Valid start trigger during operation

#### (2) TAUDnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.173 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

**(3) Channel output mode for slave channel 2**

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

**(4) Simultaneous rewrite for slave channel 2**

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.174 Simultaneous Rewrite Settings for Slave Channel 2 of Delay Pulse Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.4.7 Register Settings for Slave Channel 3

#### (1) TAUDnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.175 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	101: INTTAUDnlm of upper channel (m - 1) is a start trigger regardless of master setting.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMDO	1: Valid start trigger during operation

#### (2) TAUDnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.176 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channel 3

Table 23.177 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for slave channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.178 Simultaneous Rewrite Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.4.8 Operating Procedure for Delay Pulse Output Function

Table 23.179 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.4.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.4.5, Register Settings for Slave Channel 1</b>.</p> <p>Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.4.6, Register Settings for Slave Channel 2</b>.</p> <p>Slave channel 3: Set the TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.4.7, Register Settings for Slave Channel 3</b>.</p> <p>Set the value of TAUDnCDRm register of every channel.</p>	Channel operation is stopped.

Table 23.179 Operating Procedure for Delay Pulse Output Function (2/2)

	Operation	TAUDn Status
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUDnlm is generated on the master channel and TAUDTTOUTm (slave channel 1) is set.</p>
During Operation	<p>TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnCDRm value of master channel and slave channels 1 and 2 is loaded to TAUDnCNTm and count down.</p> <p>When the counter of master channel reaches <math>0000_{\text{H}}</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnlm (master) is generated.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1/2) to count down.</li> <li>TAUDTTOUTm (slave 1) is set.</li> </ul> <p>When TAUDnCNTm (slave 1) reaches <math>0000_{\text{H}}</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnlm (slave 1) is generated.</li> <li>TAUDTTOUTm (slave 1) is reset.</li> </ul> <p>When TAUDnCNTm (slave 2) reaches <math>0000_{\text{H}}</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnlm (slave 2) is generated.</li> <li>INTTAUDnlm (slave 3) is generated.</li> <li>TAUDTTOUTm (slave 3) is set.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (slave 3) to count down operation.</li> </ul> <p>When TAUDnCNTm (slave 3) reaches <math>0001_{\text{H}}</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnlm (slave 3) is generated.</li> <li>TAUDTTOUTm (slave 3) is reset.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

### 23.15.4.9 Specific Timing Diagrams

#### (1) Duty cycle (slave 3) = 100%

The following values apply to **Figure 23.108**:

- TAUDnCDRm (master) =  $000A_H$
- TAUDnCDRm (slave 1) =  $000B_H$
- TAUDnCDRm (slave 2) =  $0000_H$
- TAUDnCDRm (slave 3) =  $000B_H$

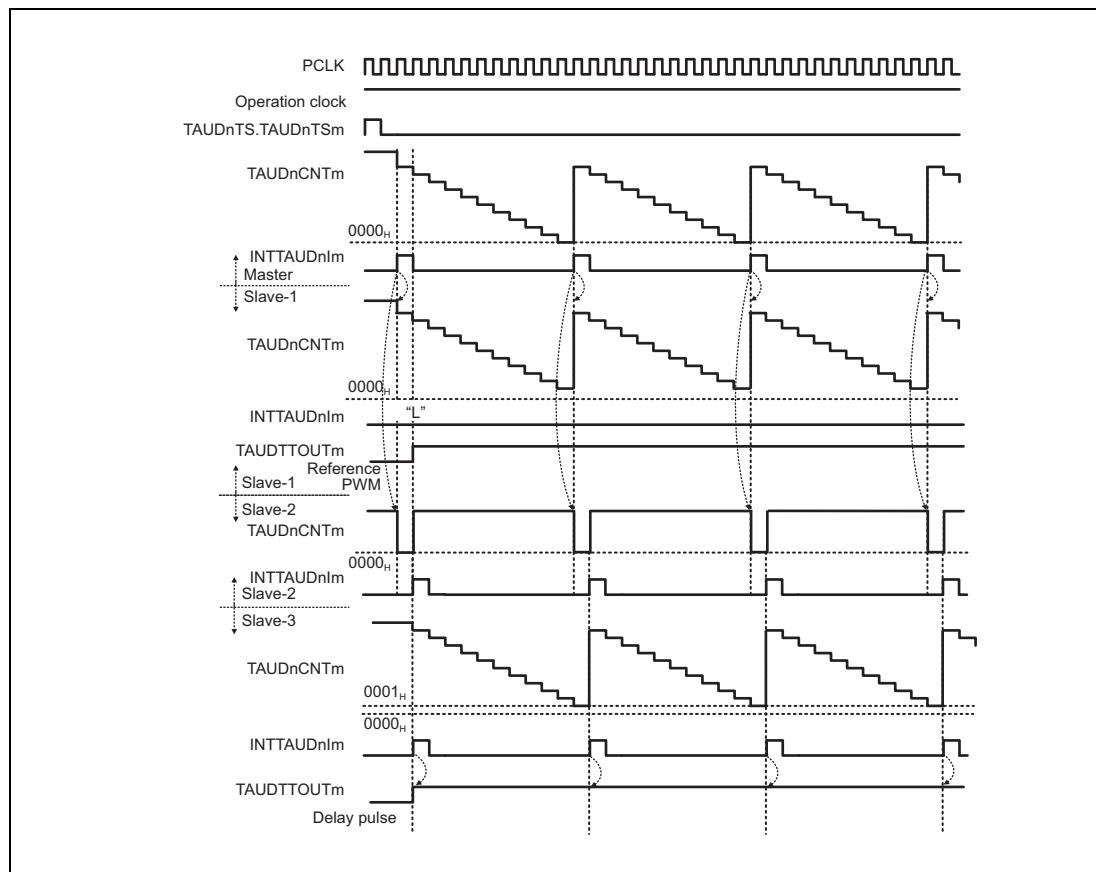


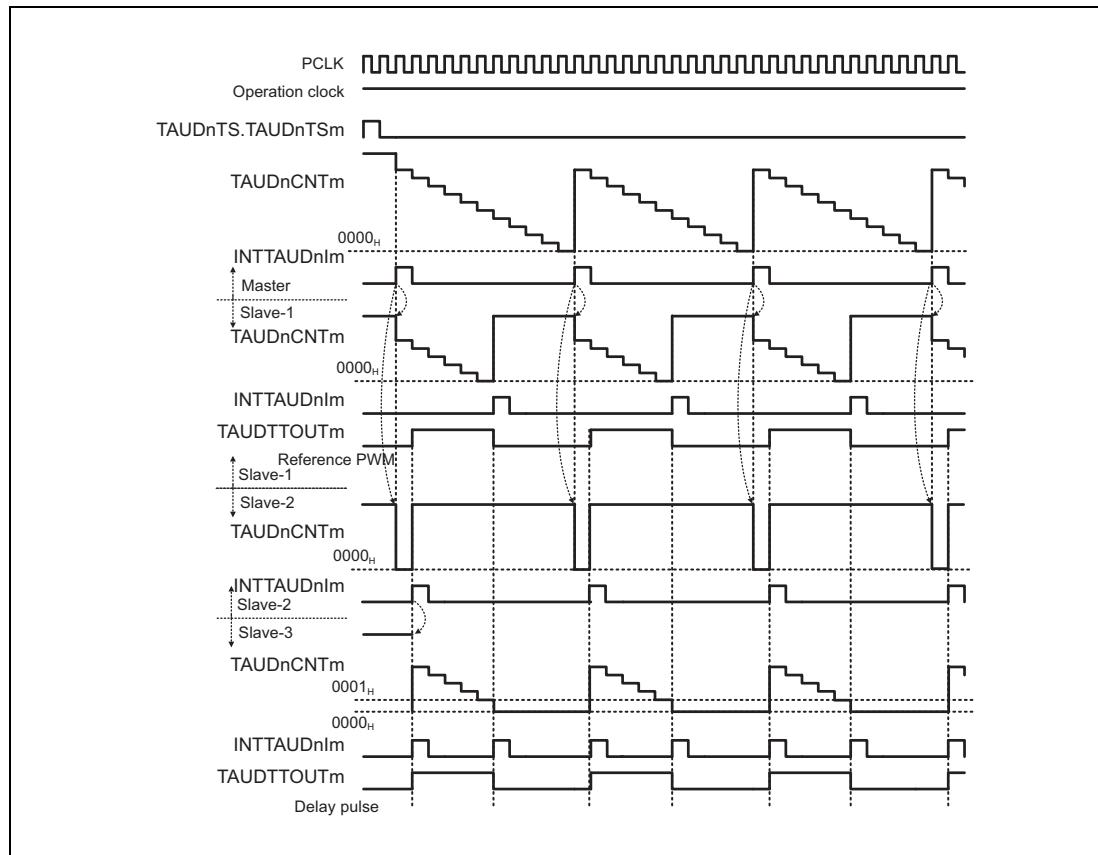
Figure 23.108 Duty Cycle (Slave 3) = 100%

- If the value of TAUDnCDRm (slaves 1 and 3) is higher than the value of TAUDnCDRm (master), the counter of the slave channel 1 cannot reach  $0000_H$  and cannot generate interrupts. TAUDTTOUTm of channels 1 and 3 remain in the active state.

## (2) TAUDTOUTm (slave 1) = TAUDTOUTm (slave 3)

The following values apply to **Figure 23.109**.

- TAUDnCDRm (master) =  $000A_H$
- TAUDnCDRm (slave 1) =  $0005_H$
- TAUDnCDRm (slave 2) =  $0000_H$
- TAUDnCDRm (slave 3) =  $0005_H$



**Figure 23.109 TAUDTOUTm (Slave 1) = TAUDTOUTm (Slave 3)**

- If  $\text{TAUDnCDRm}(\text{slave } 2) = 0000_H$ , the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

## 23.15.5 Offset Trigger Output Function

### 23.15.5.1 Overview

#### Summary

This function generates a PWM output using a master channel and a slave channel, enabling the pulse width (duration) of the TAUDTTOUTm to be set. The pulse cycle is set by detecting a valid input edge of master channel. The pulse width is specified on the slave channel.

#### Prerequisites

- Two channels
- The operating mode for the master channel should be set to capture mode. (See **Table 23.180, Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function.**)
- The operating mode for slave channels should be set to one-count mode. (See **Table 23.183, Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function.**)
- The output mode for slave channels should be set to synchronous channel output mode 1. (See **Section 23.7, Channel Output Modes.**)
- TAUDTTOUTm is not used with the master channel of this function.

#### Functional description

The counter can be enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEM = 1, enabling count operation. The master channel counter (TAUDnCNTm) starts to count up from  $0000_H$ .

- Master channel:  
When a valid TAUDTTINm input edge is detected, the current value of the counter (TAUDnCNTm) is loaded into the data register of master channel (TAUDnCDRm). INTTAUDnIm is generated and the counter restarts to count up from  $0000_H$ .
- Slave channel:  
If INTTAUDnIm is generated on the master channel, the TAUDTTOUTm (slave) signal is set and the counter of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts to count down from this value.  
When the counter reaches  $0000_H$  (duty time has elapsed), INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter returns to  $FFFF_H$  and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

### 23.15.5.2 Equations

$$\text{Pulse width} = (\text{TAUDnCDRm (slave)}) \times \text{count clock cycle}$$

$$\text{Duty cycle [%]} = [\text{TAUDnCDRm (slave)} / (\text{TAUDnCDRm (master)} + 1)] \times 100$$

- Duty cycle = 0%  
TAUDnCDRm (slave) = 0000<sub>H</sub>
- Duty cycle = 100%  
TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

### 23.15.5.3 Block Diagram and General Timing Diagram

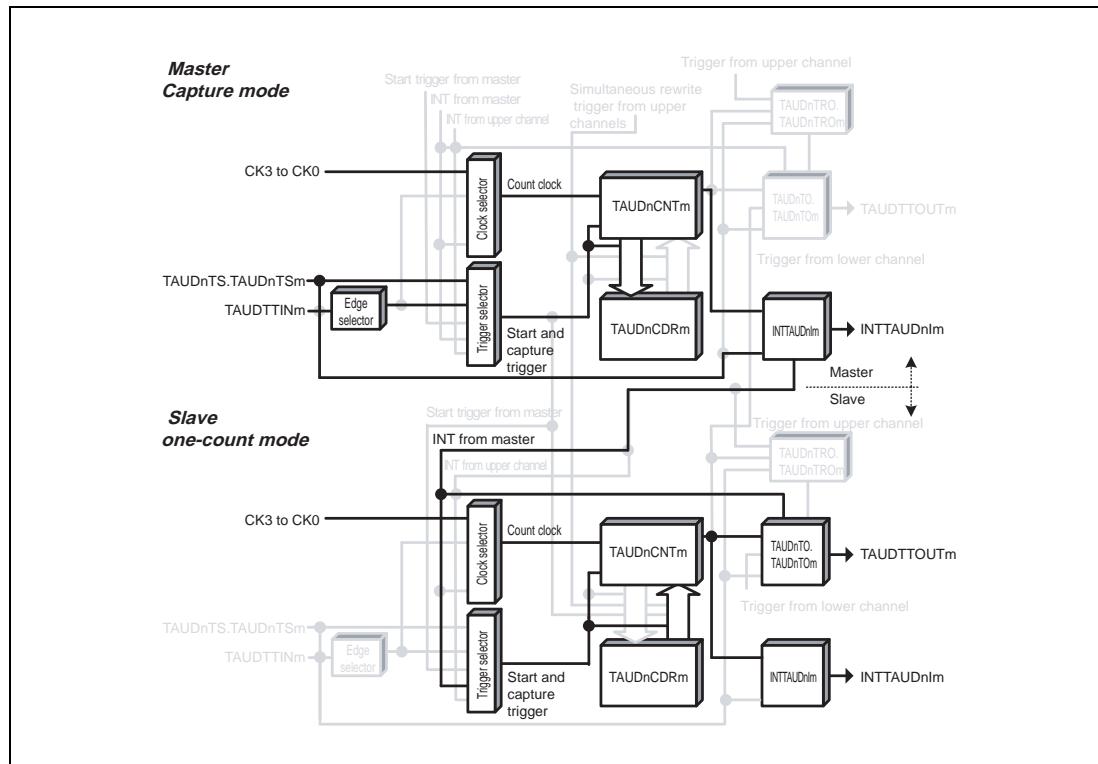


Figure 23.110 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

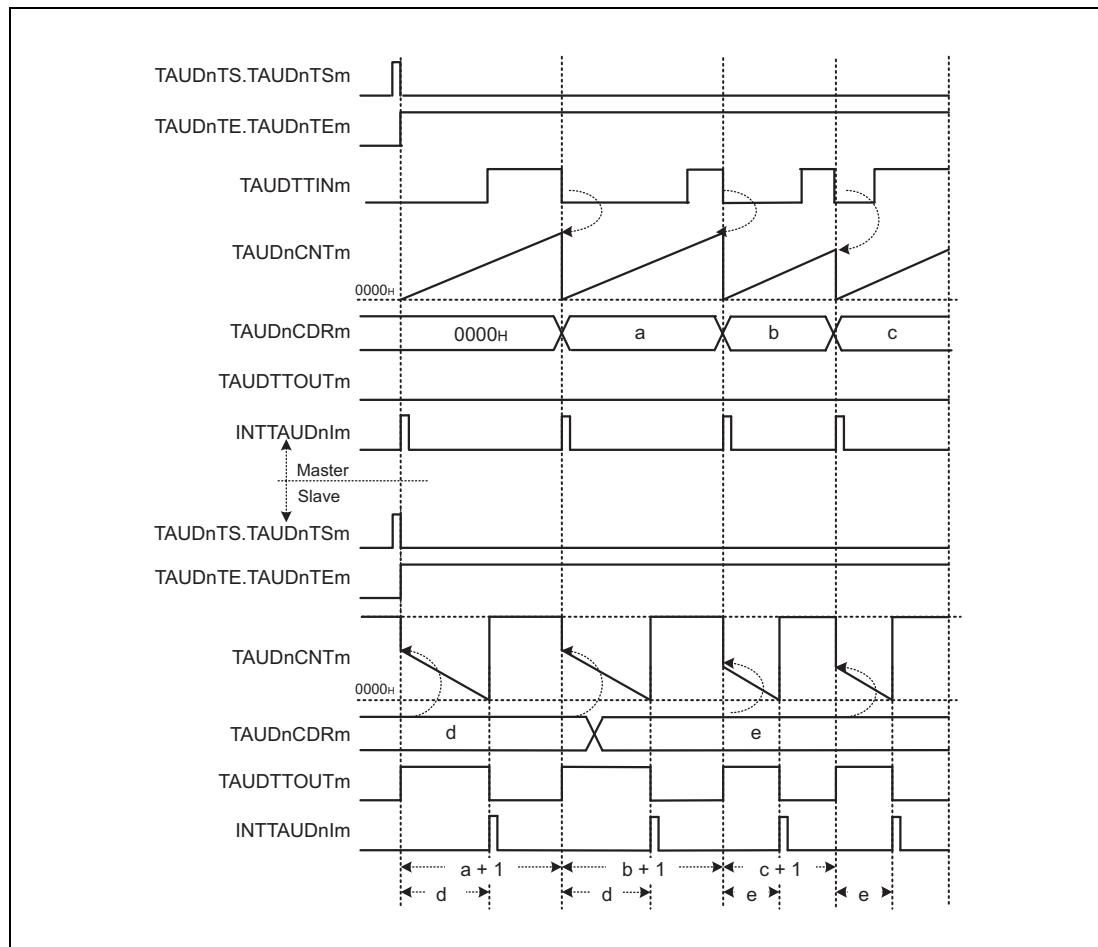


Figure 23.111 General Timing Diagram of Offset Trigger Output Function

#### NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

### 23.15.5.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.180 Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	11: Updated upon detection of a valid TAUDTTINm input edge or when a counter overflow occurs: – Detection of valid TAUDTTINm input edge: The counter value is written into TAUDnCDRm. – Occurrence of overflow: FFFF <sub>H</sub> is written into TAUDnCDRm. A valid TAUDTTINm input edge to be detected next is ignored. TAUDnCSRm.TAUDnOVF is set when a counter overflow occurs, and cleared by setting TAUDnCSCm.TAUDnCLOV = 1.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMDO	1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.181 Contents of the TAUDnCMURm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

**(3) Channel output mode for the master channel**

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

**(4) Simultaneous rewrite for the master channel**

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers should be set to 0.

**Table 23.182 Simultaneous Rewrite Settings for the Master Channel of Offset Trigger Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

### 23.15.5.5 Register Settings for Slave Channels

#### (1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.183 Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Enables start trigger detection while counting.

#### (2) TAUDnCMURm for slave channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W

Table 23.184 Contents of the TAUDnCMURm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channels

**Table 23.185 Control Bit Settings in Synchronous Channel Output Mode 1**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMM	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMM	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCM	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for slave channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers should be set to 0.

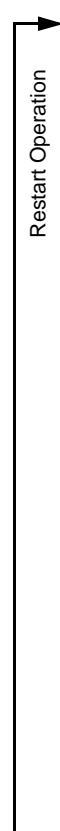
**Table 23.186 Simultaneous Rewrite Settings for Slave Channels of Offset Trigger Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSM	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEM = 0), set these bits to 0
TAUDnRDM.TAUDnRDMM	
TAUDnRDC.TAUDnRDCM	

### 23.15.5.6 Operating Procedure for Offset Trigger Output Function

Table 23.187 Operating Procedure for Offset Trigger Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.5.4, Register Settings for the Master Channel.</b></p> <p>Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in <b>Section 23.15.5.5, Register Settings for Slave Channels.</b></p> <p>The TAUDnCDRm register of master channel functions as a capture register. Set the value of TAUDnCDRm register of slave channel.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start:</p> <ul style="list-style-type: none"> <li>• TAUDnCNTm (master) counts up.</li> <li>• TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down.</li> </ul> <p>INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set.</p>
During Operation	<p>TAUDnCDRm can be changed at any time. TAUDnCSCm.TAUDnCLOV can be set to 1. TAUDnCDRm of slave channel can be changed after the generation of INTTAUDnIm (master). TAUDnCNT.TAUDnCNTm and TAUDnCSRm can be read at any time.</p>	<p>When TAUDnCNTm of the slave = 0000H:</p> <ul style="list-style-type: none"> <li>• INTTAUDnIm (slave) is generated.</li> <li>• TAUDTTOUTm (slave) is reset, and the counter of slave channel stops.</li> </ul> <p>When TAUDTTINm input edge is detected on the master channel:</p> <ul style="list-style-type: none"> <li>• INTTAUDnIm (master) is generated.</li> <li>• TAUDnCNTm (master) is reset to 0000H and then continues count operation subsequently.</li> <li>• TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down.</li> <li>• TAUDTTOUTm (slave) is set.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>



### 23.15.5.7 Specific Timing Diagrams

#### (1) Duty cycle = 0%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

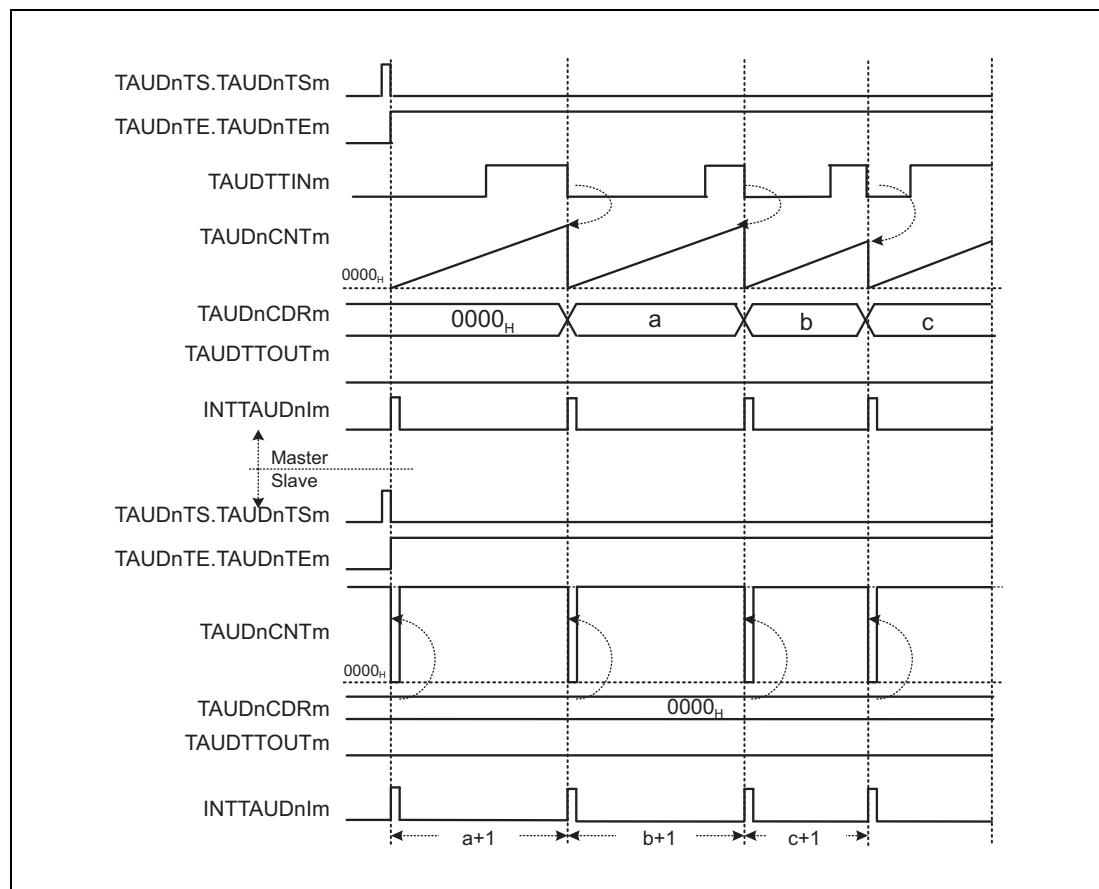


Figure 23.112 TAUDnCDRm (Slave) = 0000<sub>H</sub>

- When TAUDnCDRm (slave) = 0000<sub>H</sub>, 0000<sub>H</sub> is written to TAUDnCNTm every time the master channel generates an interrupt (INTTAUDnIm), and TAUDnCNTm cannot start to count. The TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The slave and the master channels generate interrupts in the same cycle.

## (2) Duty cycle = 100%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00<sub>B</sub>)

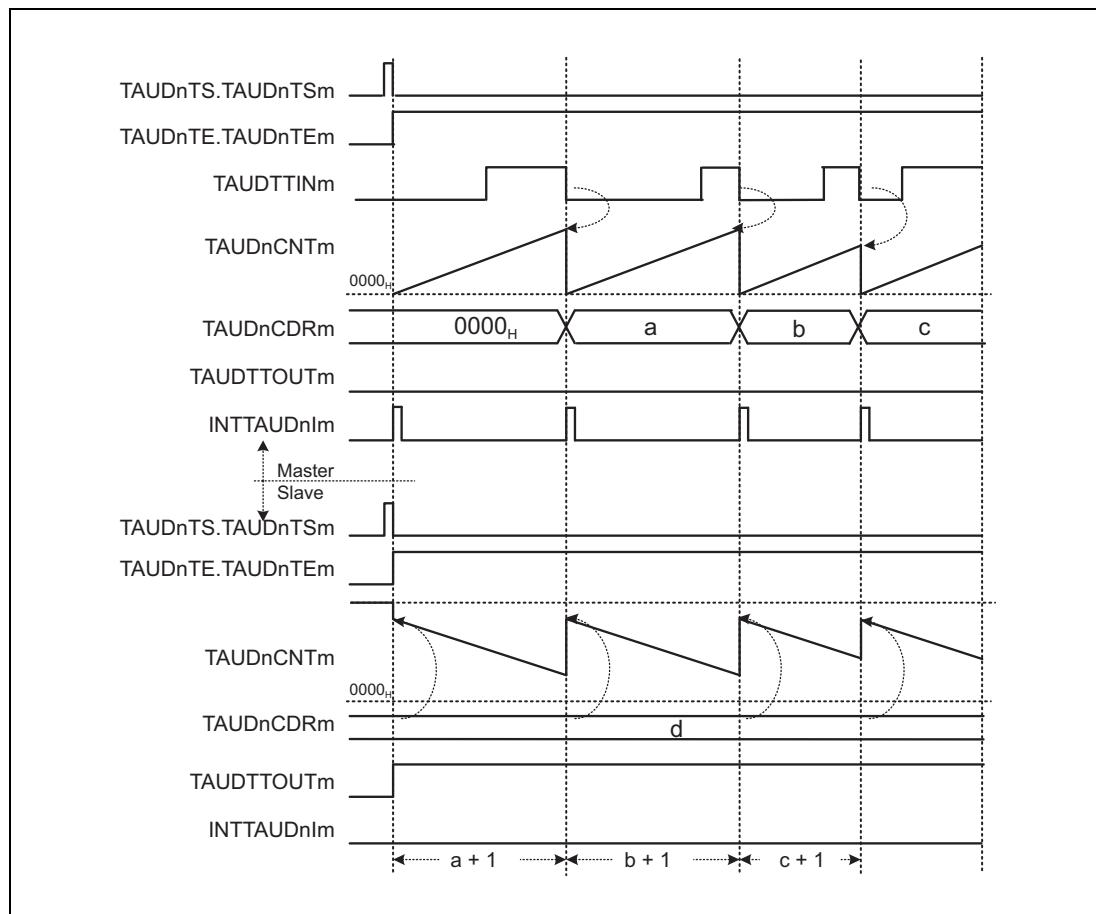


Figure 23.113 TAUDnCDRm (Slave)  $\geq$  TAUDnCDRm (Master) + 1

- If the value TAUDnCDRm (slave) is higher than the interval of valid input edges, the counter of the slave channel cannot reach 0000<sub>H</sub> and cannot generate interrupts. The TAUDTTOUTm remains at active state.

## 23.15.6 A/D Conversion Trigger Output Function Type 1

### 23.15.6.1 Overview

#### Summary

This function is identical to **Section 23.15.1, PWM Output Function**, except that TAUDTTOUTm is not output.

This is achieved by setting the channel output mode for the slave to independent channel output mode controlled by software.

### 23.15.6.2 Block Diagram and General Timing Diagram

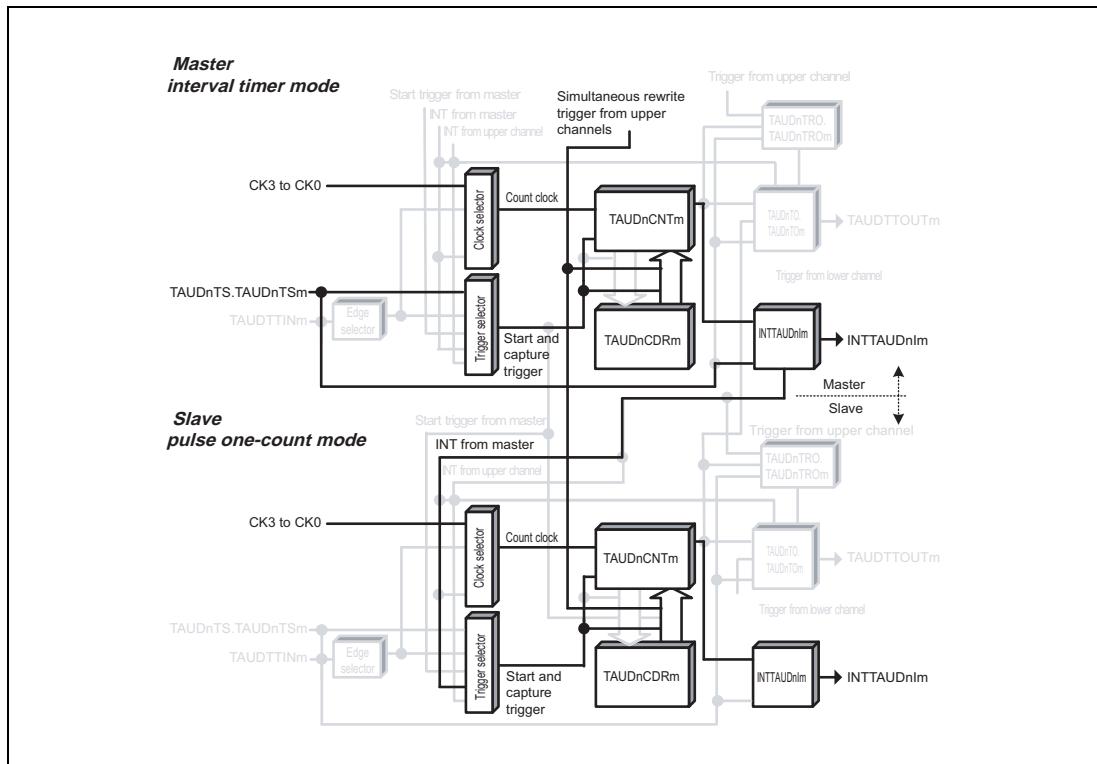


Figure 23.114 Block Diagram of A/D Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

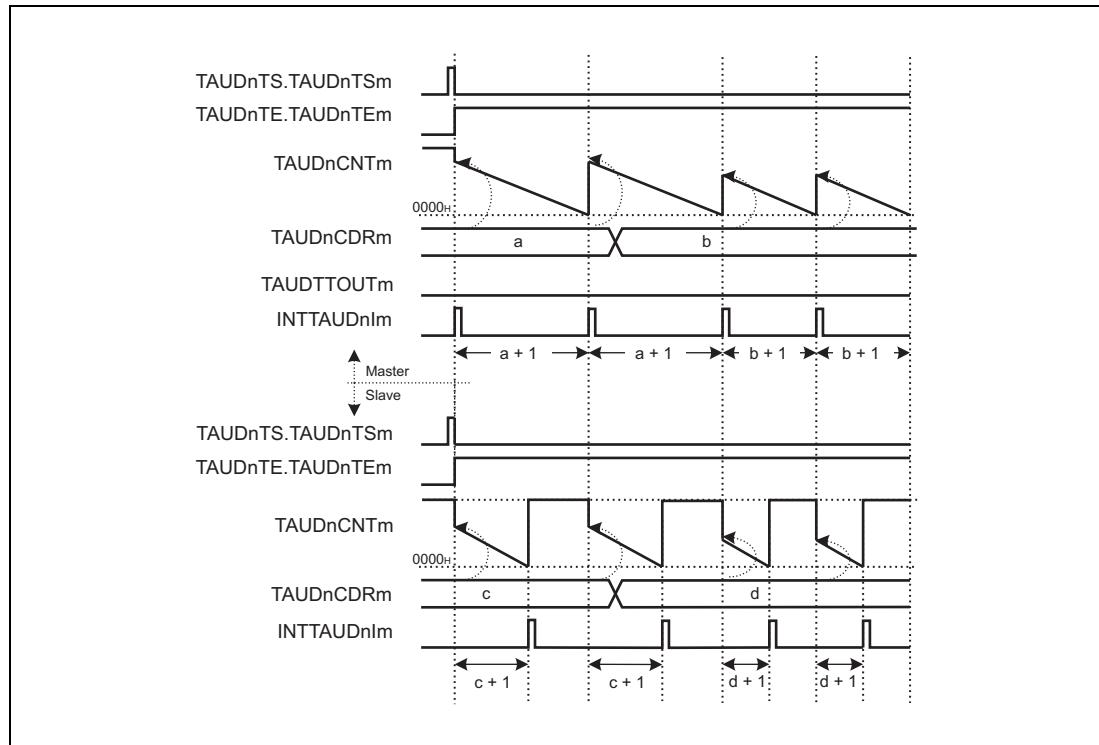


Figure 23.115 General Timing Diagram of A/D Conversion Trigger Output Function Type 1

## 23.15.7 Triangle PWM Output Function

### 23.15.7.1 Overview

#### Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUDTTOUTm to be set using the master and slave channels respectively.

The master channel generates a carrier cycle. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slave counter.

#### Prerequisites

- Two channels
- The operating mode for the master channels should be set to interval timer mode. (See **Table 23.188, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function.**)
- The operating mode for slave channels should be set to count-up/-down mode. (See **Table 23.192, Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function.**)
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes.**)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See **Section 23.7, Channel Output Modes.**)
- The following settings allow the TAUDTTOUTm signal to be at high level during the down status of a carrier cycle.
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOM should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOM should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

#### Functional description

The counters are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for every channel. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current values of TAUDnCDRm (master and slave) are loaded into TAUDnCNTm (master and slave) and the counters start counting down from these values. When the TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and TAUDTTOUTm signal of master toggles.

- Master channel:  
When the counter of master channel reaches  $0000_H$  (pulse cycle time has elapsed), INTTAUDnIm is generated and the TAUDTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and counts down.

- Slave channel:

If INTTAUDnIm is generated on the master slave, the counter of the slave channel is triggered.

- If the slave counter is counting down, the count direction changes.
- If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts to count down.

When the counter of the slave channel reaches  $0001_H$  while counting up or down, INTTAUDnIm is generated and the TAUDTTOUTm (slave) signal is set/reset.

The counter continues count-up/-down and waits for the next INTTAUDnIm of the master channel.

Setting TAUDnTOL.TAUDnTOLm allows TAUDTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM = 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

### Conditions

This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

#### 23.15.7.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq \text{TAUDnCDRm (master)} < FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle 100 [%] =

$[(\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave)}) / (\text{TAUDnCDRm (master)} + 1)] \times 100$

- Duty cycle = [%]

$\text{TAUDnCDRm (slave)} = 0000_H$

- Duty cycle = 0%

$\text{TAUDnCDRm (slave)} \geq \text{TAUDnCDRm (master)} + 1$

### 23.15.7.3 Block Diagram and General Timing Diagram

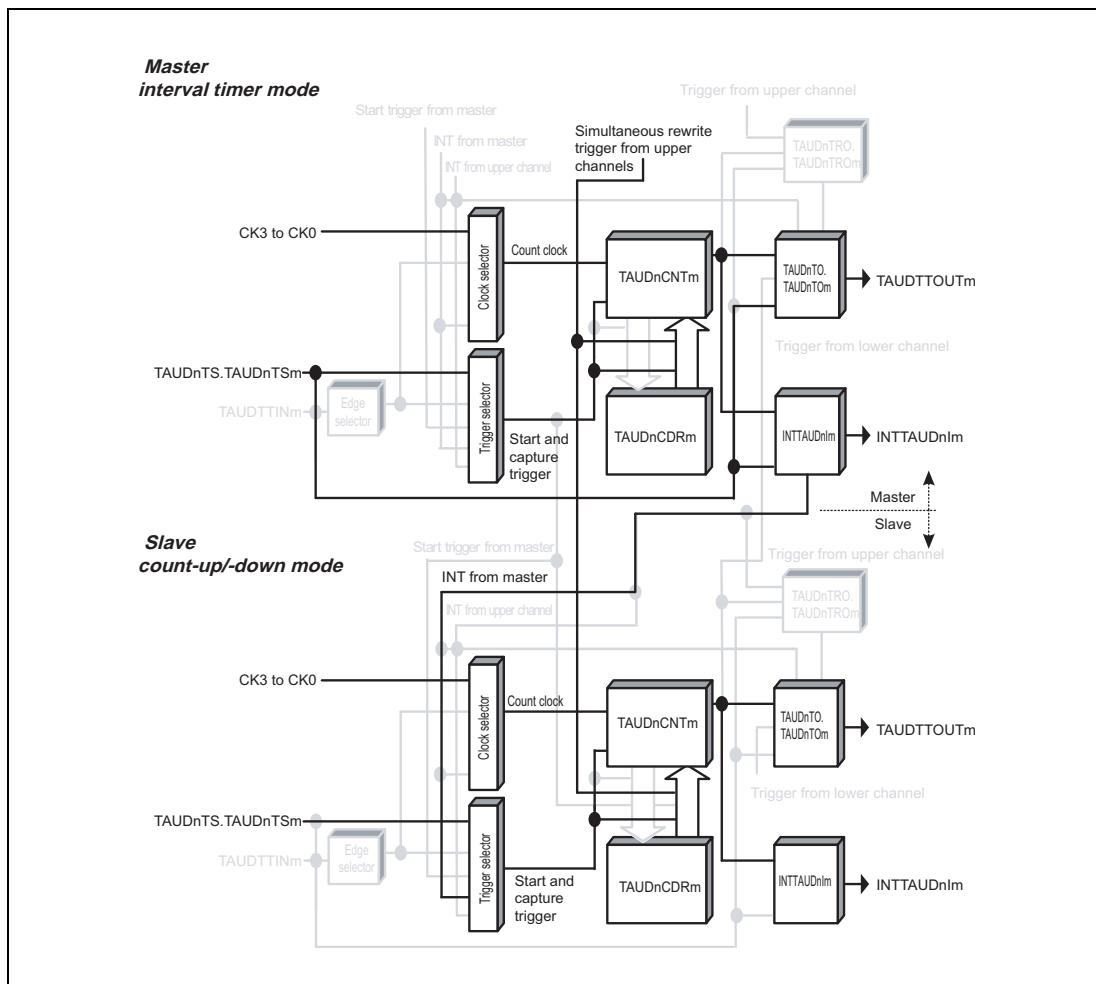


Figure 23.116 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 1)

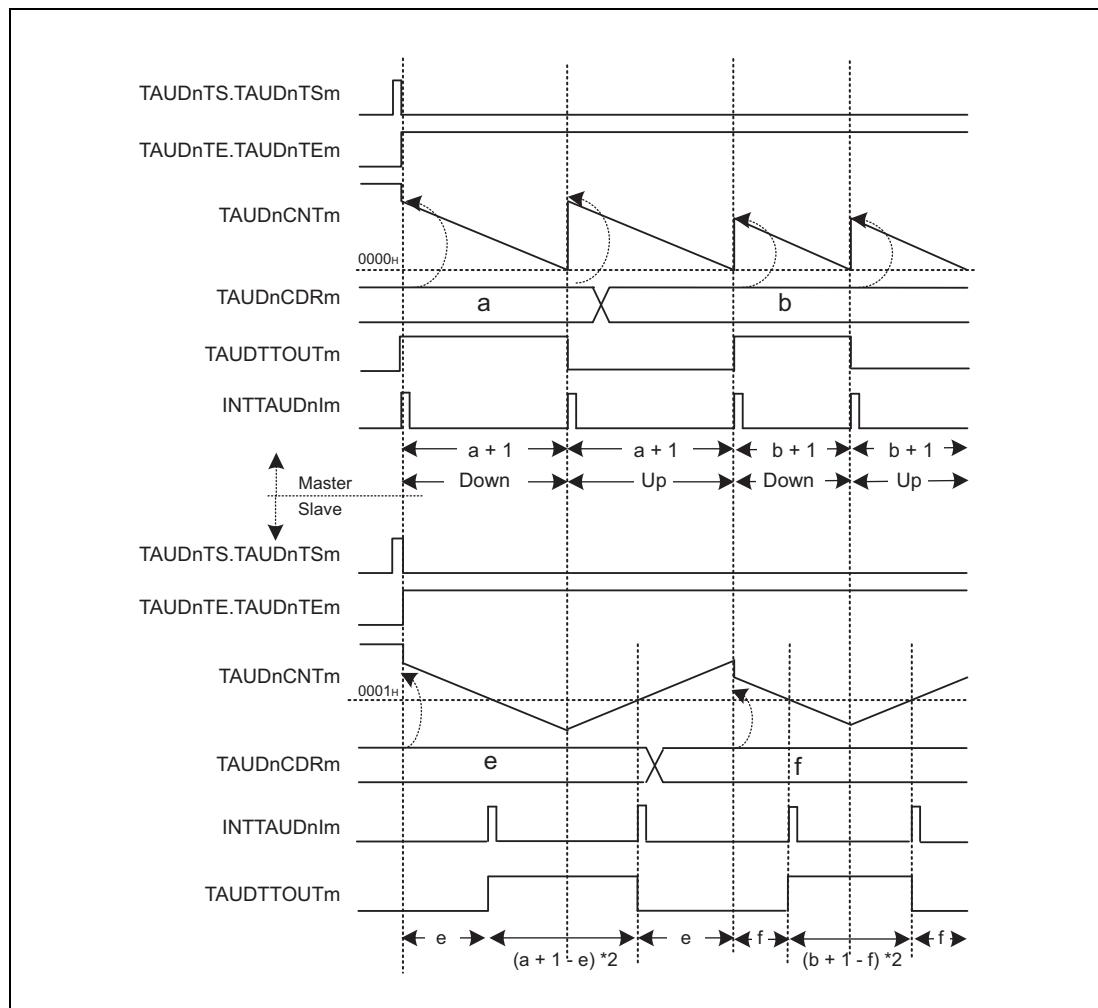


Figure 23.117 General Timing Diagram of Triangle PWM Output Function

### 23.15.7.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.188 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.189 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

Table 23.190 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCM	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.191 Simultaneous Rewrite Settings for the Master Channel of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

#### NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

### 23.15.7.5 Register Settings for Slave Channels

#### (1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

R/W	R	R/W	R/W	R/W	R/W	R/W										
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	-----	-----	-----	-----	-----

Table 23.192 Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation.

#### (2) TAUDnCMURm for slave channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0

R/W	R	R	R	R	R	R	R/W	R/W
-----	---	---	---	---	---	---	-----	-----

Table 23.193 Contents of the TAUDnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channels

**Table 23.194 Control Bit Settings in Synchronous Channel Output Mode 2**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMM	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMM	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.195 Simultaneous Rewrite Settings for Slave Channels of Triangle PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSM	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMM	1: A simultaneous rewrite trigger signal is generated when the master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCM	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.7.6 Operating Procedure for Triangle PWM Output Function

Table 23.196 Operating Procedure for Triangle PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.7.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.7.5, Register Settings for Slave Channels</b>.</p> <p>Set the value of TAUDnCDRm register of every channel.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.</p>
During Operation	<p>TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnCDRm value of master and slave channels is loaded into TAUDnCNTm to count down. When the counter of master channel reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (master) is generated.</li> <li>TAUDTTOUTm (master) is toggled.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (slave) or counting is started in opposite direction.</li> </ul> <p>When TAUDnCNTm of slave channel reaches 0001H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (slave) is generated.</li> <li>TAUDTTOUTm (slave) is set in the count-down status or reset in count-up status.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

Restart Operation

### 23.15.7.7 Specific Timing Diagrams

#### (1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm =  $a = 5_H$
- Slave channel:
  - TAUDnCDRm =  $6_H$

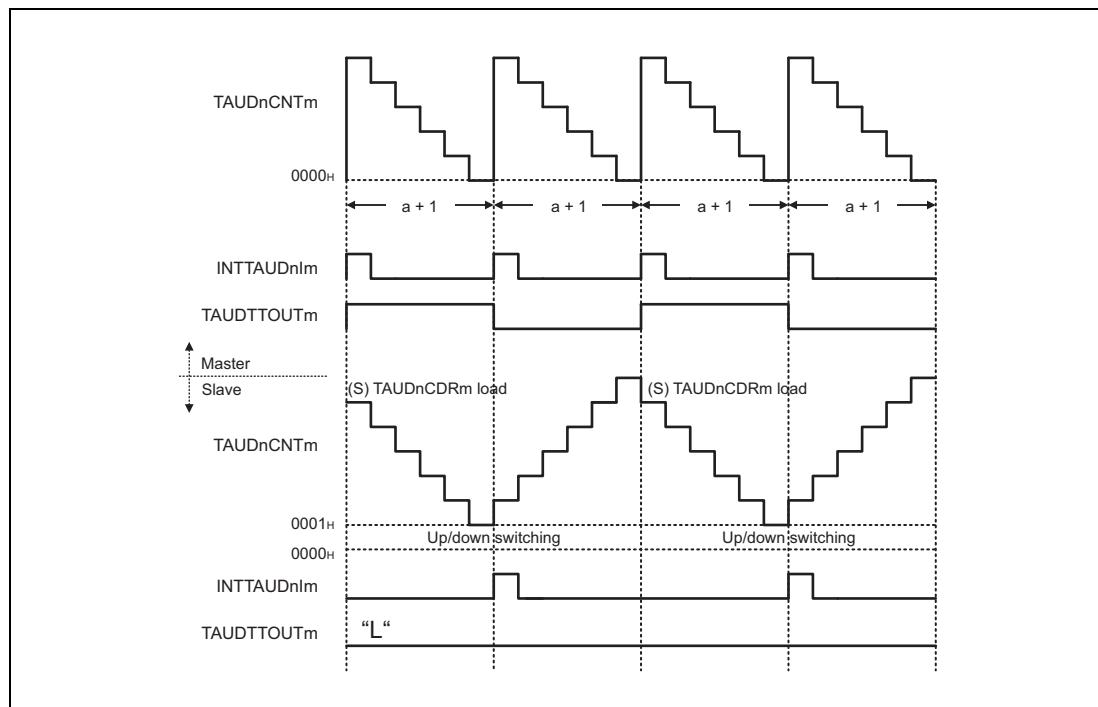


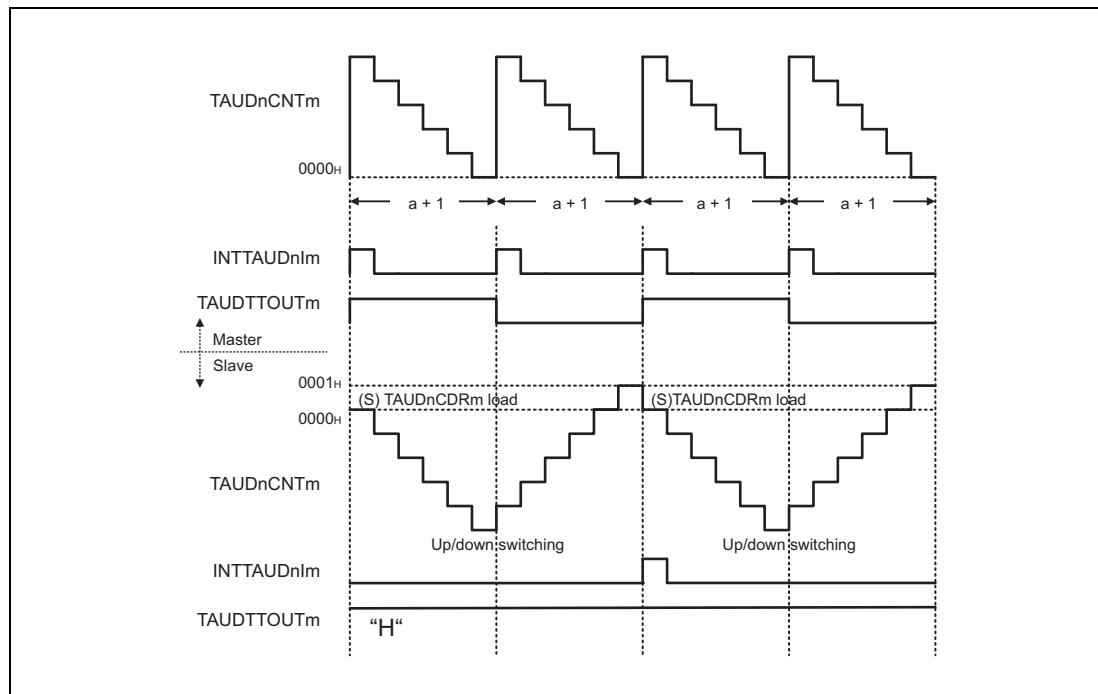
Figure 23.118 TAUDnCDRm (Slave)  $\geq$  TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave) value  $\geq$  TAUDnCDRm (master) value +1, INTTAUDnIm of the slave channel is not generated while counting down. TAUDTTOUTm remains low because there is no set signal to be detected.

## (2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm =  $a = 5_H$
- Slave channel:
  - TAUDnCDRm =  $0_H$



**Figure 23.119 TAUDnCDRm (Slave) = 0000<sub>H</sub>**

- If TAUDnCDRm (slave) = 0000<sub>H</sub>, INTTAUDnIm of the slave channel is not generated while counting up. TAUDTTOUTm remains high because there is no reset signal to be detected.

## 23.15.8 Triangle PWM Output Function with Dead Time

### 23.15.8.1 Overview

#### Summary

This function generates multiple triangle PWM outputs with a predefined dead time added by using a master and two or more slave channels. The resulting PWM signals with dead time are output via TAUDTTOUTm of the slave channels 2 and 3, enabling the pulse cycle (frequency) and the duty cycle of TAUDTTOUTm to be set using the master and slave channels.

Carrier cycles are generated on the master channel. The first pulse controls the down status of the slave counter and the second one controls the up status.

An interrupt on slave 2 causes TAUDTTOUTm of slave channels to be set/reset. Depending on the settings of TAUDnTDL.TAUDnTDLm, delay time is added to positive or negative logic side of the signal (i.e., whether TAUDTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

#### Prerequisites

- Three channels. For slave channels 2 and 3, select even channel CH (a) and odd channel CH (a + 1).
- The operating mode for the master channel should be set to interval timer mode. (See **Table 23.198, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**)
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even channel (a), and slave channel 3 is an odd channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operating mode for slave channel 2 should be set to count-up/-down mode (See **Table 23.202, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time**). Slave channel 2 should be an even channel.
- The operating mode for slave channel 3 should be set to one-count mode (See **Table 23.206, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time**). Slave channel 3 should be an odd channel.
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes**)
- The output mode for slave channels 2 and 3 should be set to synchronous channel output mode 2 with dead time output. (See **Section 23.7, Channel Output Modes**)
- The following settings make a TAUDTTOUTm signal at high level during the down status of the carrier cycle:

- If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOM should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
- If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOM should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

#### NOTE

The triangle PWM output function with dead time does not use slave channel 1. Slave channel 1 can be used as a separate timer (independent function).

#### Functional description

The counter starts by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEM = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and the master's TAUDTTOUTm signal is toggled.

- Master channel:

When the counter of the master channel reaches  $0000_H$ , an INTTAUDnIm is generated and the TAUDTTOUTm signal is toggled. The TAUDnCDRm value is reloaded to continue counting down.

- Slave channel 2:

If INTTAUDnIm is generated on the master channel, the counter of slave channel 2 is triggered.

- If the slave counter is counting down, the counting direction changes.
- If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts counting down.

The counter continues to count down/up and waits for the next INTTAUDnIm of the master channel.

When the counter value of slave channel 2 reaches  $0001_H$ , INTTAUDnIm is generated

- Slave channel 3:

If INTTAUDnIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUDnCDRm (slave 3) is loaded into TAUDnCNTm (slave 3) and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches  $0000_H$ , INTTAUDnIm occurs. The counter returns to  $FFFF_H$  and waits for the next INTTAUDnIm of slave channel 2.

As described in **Table 23.197, Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**, the set/reset timing (right after occurrence of an interrupt or after dead time has elapsed) depends on the TAUDnTDL.TAUDnTDLm setting of the corresponding channel.

The setting of TAUDnTOL.TAUDnTOLm also determines whether a high level signal (TAUDnTOL.TAUDnTOLm = 0) or a low level signal (TAUDnTOL.TAUDnTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

TAUDTTOUTm can be 100% output by setting the TAUDnCDRm value of slave channel 2 to  $0000_H$ .

### Conditions

This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

TAUDnTOL.TAUDnTOLm and TAUDnTDL.TAUDnTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUDnTDL.TAUDnTDLm.

**Table 23.197 Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**

TAUDnTDL. TAUDnTDLm	Count Direction of Slave Channel 2 upon Occurrence of Interrupt	TAUDTTOUTm Set/Reset Timing
0	Down	Set after elapse of dead time
	Up	Reset right after interrupt occurs
1	Down	Set right after interrupt occurs
	Up	Reset after elapse of dead time

### 23.15.8.2 Equations

$$\text{Pulse cycle} = (\text{TAUDnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$0000_{\text{H}} \leq \text{TAUDnCDRm (master)} < FFFF_{\text{H}}$$

$$\text{Carrier cycle (down/up)} = (\text{TAUDnCDRm (master)} + 1) \times 2 \times \text{count clock cycle}$$

$$\text{PWM signal width (normal phase)} = [(\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave 2)}) \times 2 - (\text{TAUDnCDRm (slave 3)} + 1)] \times \text{count clock cycle}$$

$$\text{PWM signal width (reverse phase)} = [(\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave 2)}) \times 2 + (\text{TAUDnCDRm (slave 3)} + 1)] \times \text{count clock cycle}$$

### 23.15.8.3 Block Diagram and General Timing Diagram

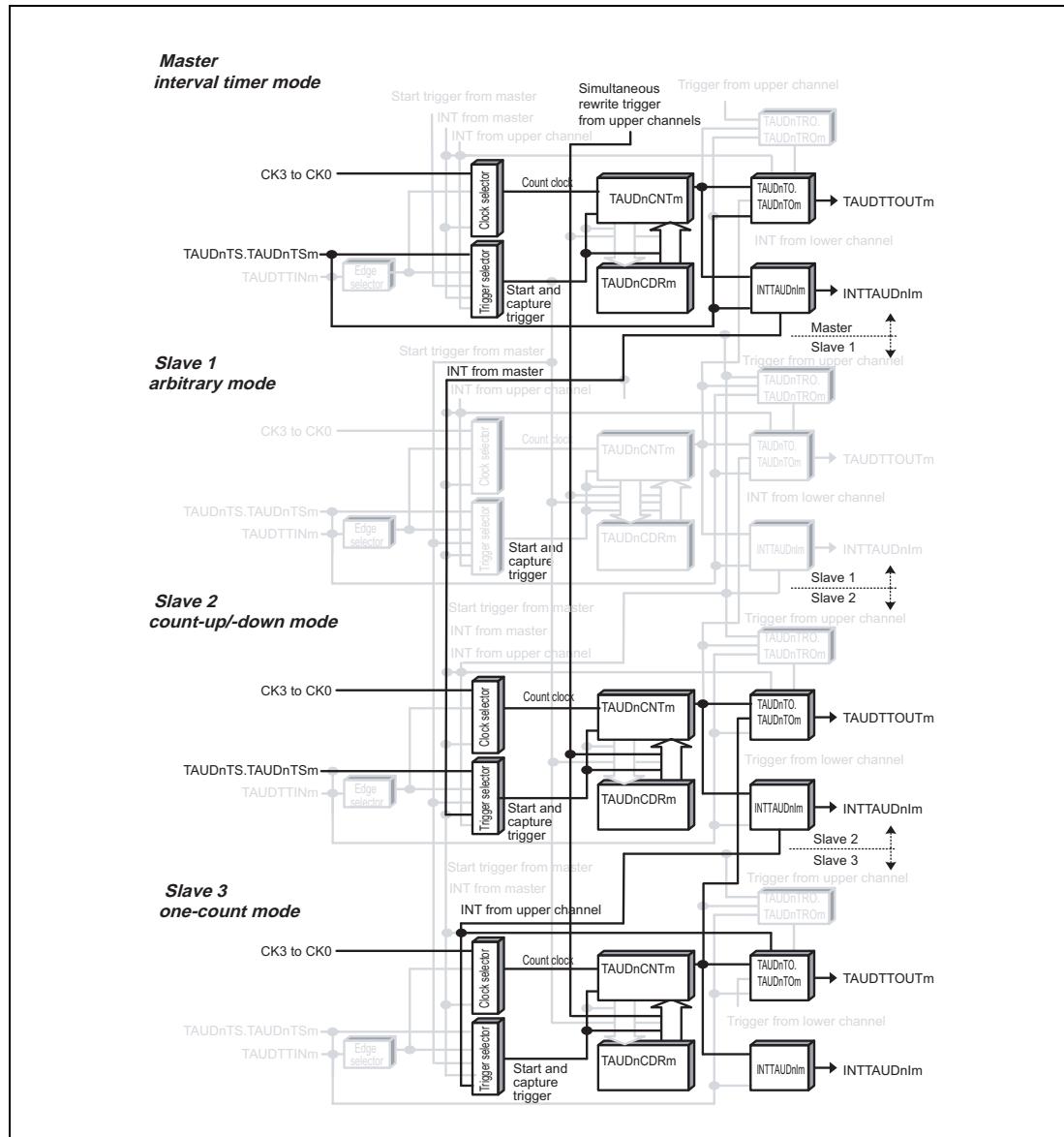


Figure 23.120 Block Diagram of Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 1)
- Slave channel 2:
  - INTTAUDnIm not generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 0)
  - TAUDnTDL.TAUDnTDLm = 0
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

- Slave channel 3:
  - Enables start trigger detection during counting (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnTDL.TAUDnTDLm = 1
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

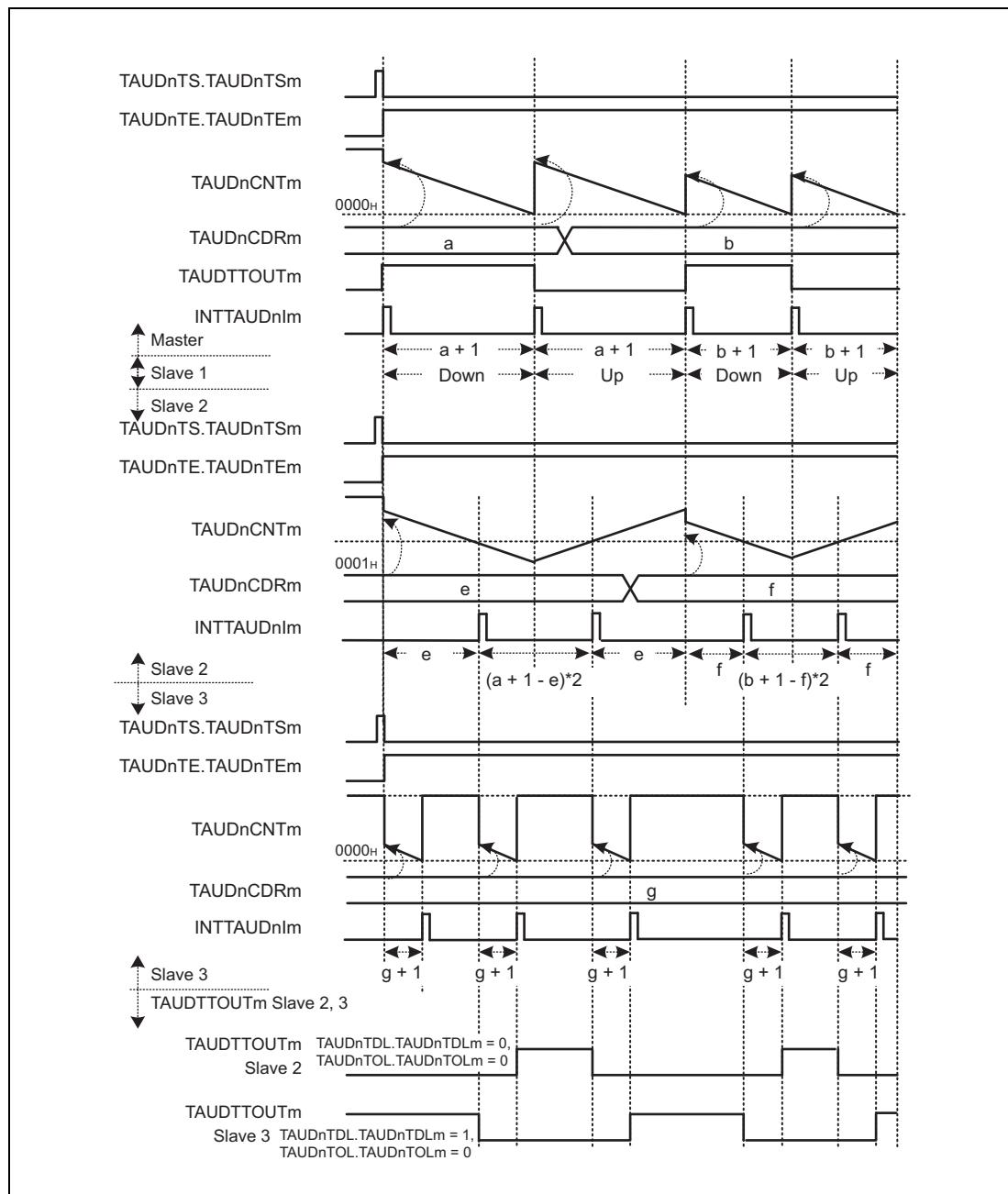


Figure 23.121 General Timing Diagram of Triangle PWM Output Function with Dead Time

### 23.15.8.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.198 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.199 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

Table 23.200 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCM	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.201 Simultaneous Rewrite Setting for the Master Channel of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

#### NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

### 23.15.8.5 Register Settings for Slave Channel 2

#### (1) TAUDnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.202 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation.

#### (2) TAUDnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.203 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channel 2

**Table 23.204 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation
<b>CAUTION</b>	
Set TAUDnTDL.TAUDnTDLm exclusively from odd channels.	

### (4) Simultaneous rewrite for slave channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.205 Simultaneous Rewrite Settings for Slave Channel 2 of Triangle PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.8.6 Register Settings for Slave Channel 3

#### (1) TAUDnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.206 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time output signal of the TAUDTTOUTm generation unit
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Enables start trigger detection while counting.

#### (2) TAUDnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.207 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channel 3

**Table 23.208 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation
<b>CAUTION</b>	
Set TAUDnTDL.TAUDnTDLm exclusively from even channels.	

### (4) Simultaneous rewrite for slave channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.209 Simultaneous Rewrite Settings for Slave Channel 3 of Triangle PWM Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.8.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 23.210 Operating Procedure for Triangle PWM Output Function with Dead Time

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.8.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.8.5, Register Settings for Slave Channel 2</b>.</p> <p>Slave channel 3: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.8.6, Register Settings for Slave Channel 3</b>.</p> <p>Set the value of TAUDnCDRm register of every channel.</p>	Channel operation is stopped.
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.</p>
During Operation	<p>TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnCDRm value of master channel and slave channel 2 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (master) is generated.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm (slave 2) or counting is started in opposite direction.</li> </ul> <p>When TAUDnCNTm of slave channel 2 reaches 0001H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm (slave 2) is generated.</li> <li>TAUDnCDRm value of slave channel 3 is loaded into TAUDnCNTm perform counting down.</li> </ul> <p>When TAUDnCNTm of slave channel 3 reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

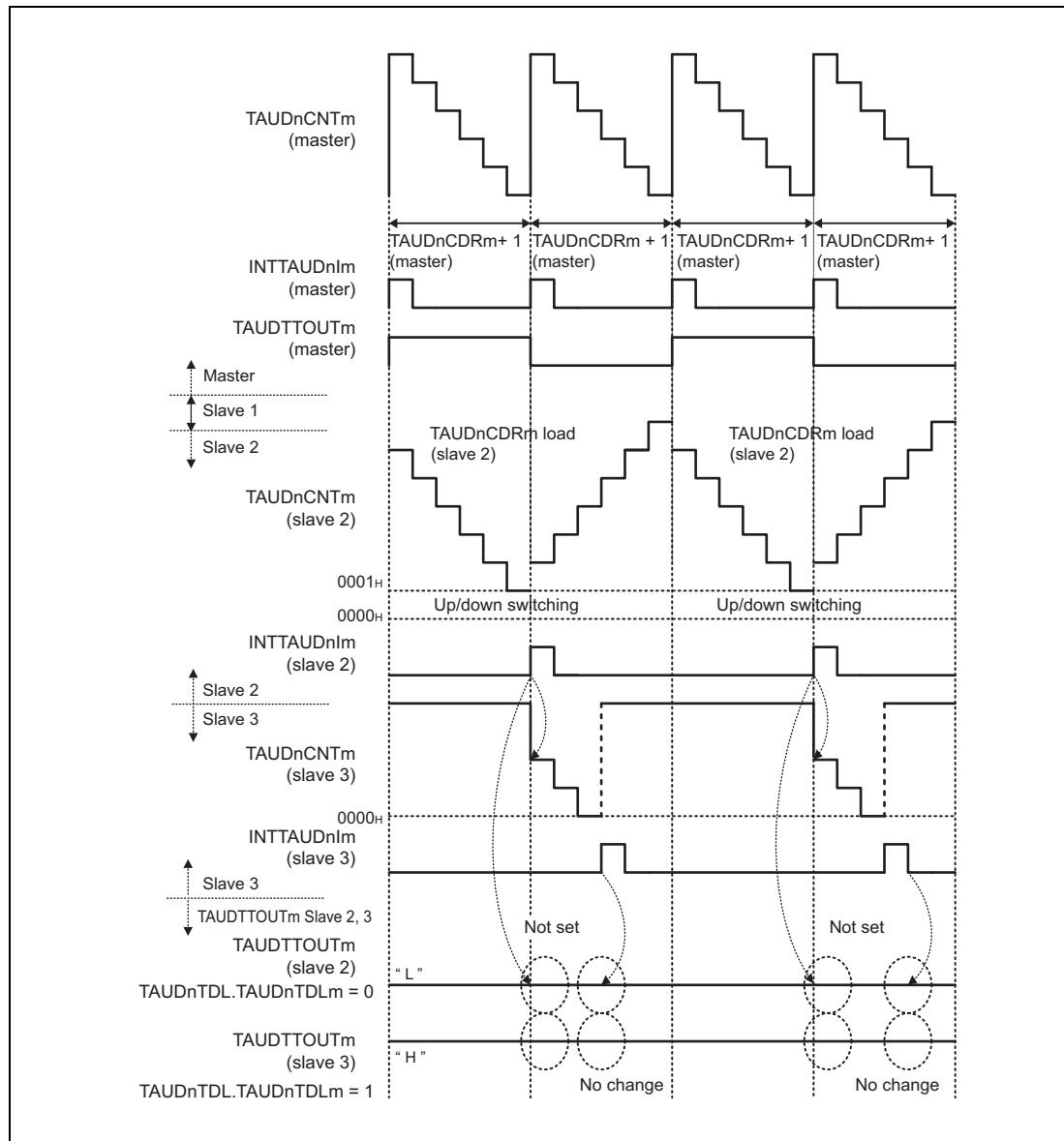


### 23.15.8.8 Specific Timing Diagrams

#### (1) Duty cycle = 0%

The following settings apply to the general timing diagram in **Figure 23.122**.

- Slave channel 2:
  - Positive logic ( $\text{TAUDnTDL} \cdot \text{TAUDnTDLm} = 0$ )
- Slave channel 3:
  - Negative logic ( $\text{TAUDnTDL} \cdot \text{TAUDnTDLm} = 1$ )



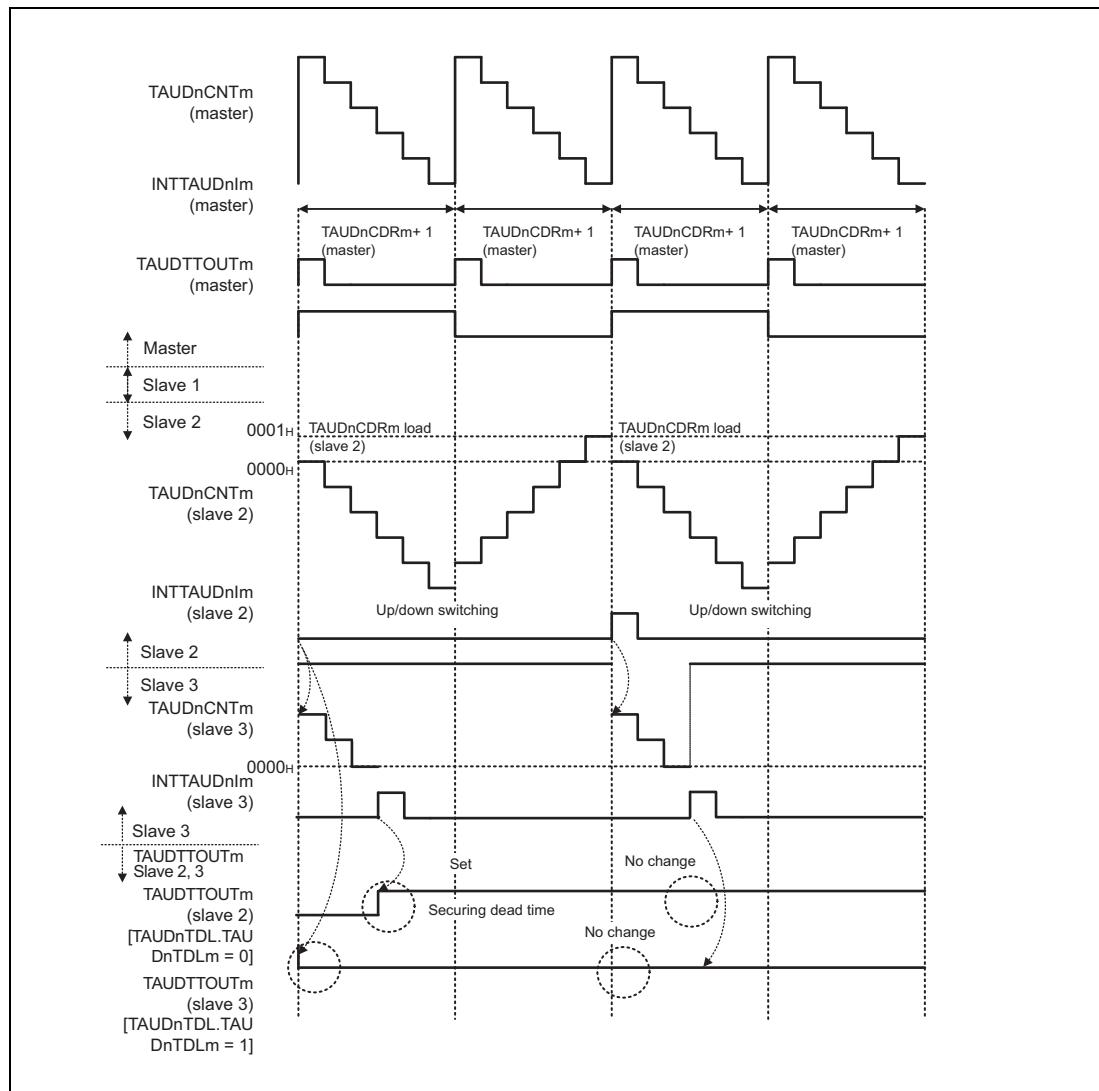
**Figure 23.122 TAUDnCDRm (Slave 2)  $\geq$  TAUDnCDRm (Master) + 1**

- If  $\text{TAUDnCDRm}$  (slave 2) is greater than  $\text{TAUDnCDRm}$  (master), the counter of slave channel does not reach  $0000_H$  while counting down. Therefore,  $\text{TAUDTTOUTm}$  signal is not set/reset and remains in the initial state. This signal becomes a reset signal because an interrupt occurs on slave channel 2 during count-up operation.

## (2) Duty cycle = 100%

The following settings apply to the general timing diagram in **Figure 23.123**.

- Slave channel 2:
  - Positive logic ( $\text{TAUDnTDL.TAUDnTDLm} = 0$ )
- Slave channel 3:
  - Negative logic ( $\text{TAUDnTDL.TAUDnTDLm} = 1$ )



**Figure 23.123** TAUDnCDRm (Slave) = 0000<sub>H</sub>

- If  $\text{TAUDnCDRm} (\text{slave } 2) = 0000_H$ , the slave channel counter does not reach  $0001_H$  while counting up. Therefore, no  $\text{INTTAUDnIm}$  is generated during count-up operation.
  - The set conditions for a channel with  $\text{TAUDnTDL.TAUDnTDLm} = 0$  are met after elapse of dead time.  $\text{TAUDTTOUTm}$  is left in a newly set state even if a set/reset is made because no reset conditions are satisfied on such a channel.
  - Slave channel 3 in the above diagram is set when the counter starts. However,  $\text{TAUDTTOUTm}$  is left in an initial state on the slave channel with  $\text{TAUDnTDL.TAUDnTDLm} = 1$  because no reset conditions are satisfied on that channel.

## 23.15.9 A/D Conversion Trigger Output Function Type 2

### 23.15.9.1 Overview

#### Summary

This function is identical to **Section 23.15.7, Triangle PWM Output Function**, except that TAUDTTOUTm is not output.

This function is enabled by setting channel output mode for the slave to independent channel output mode controlled by software.

### 23.15.9.2 Block Diagram and General Timing Diagram

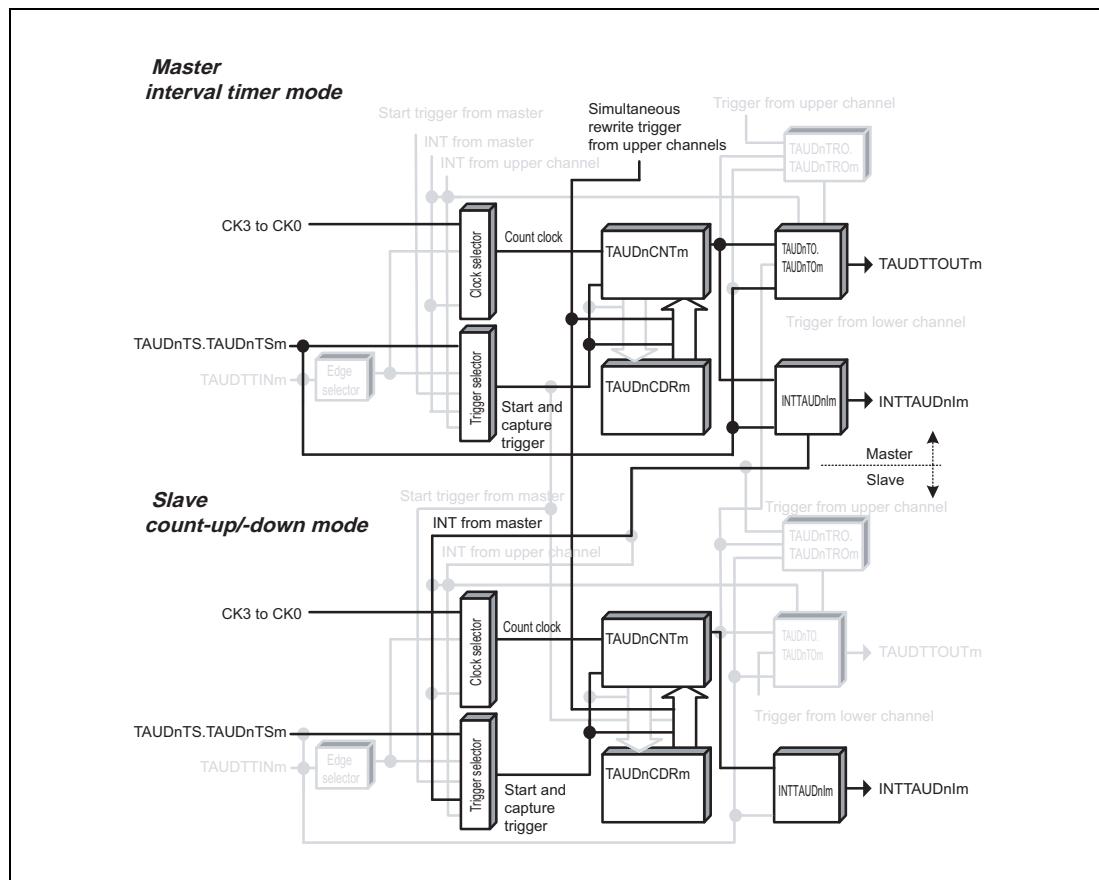


Figure 23.124 Block Diagram of A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 1)

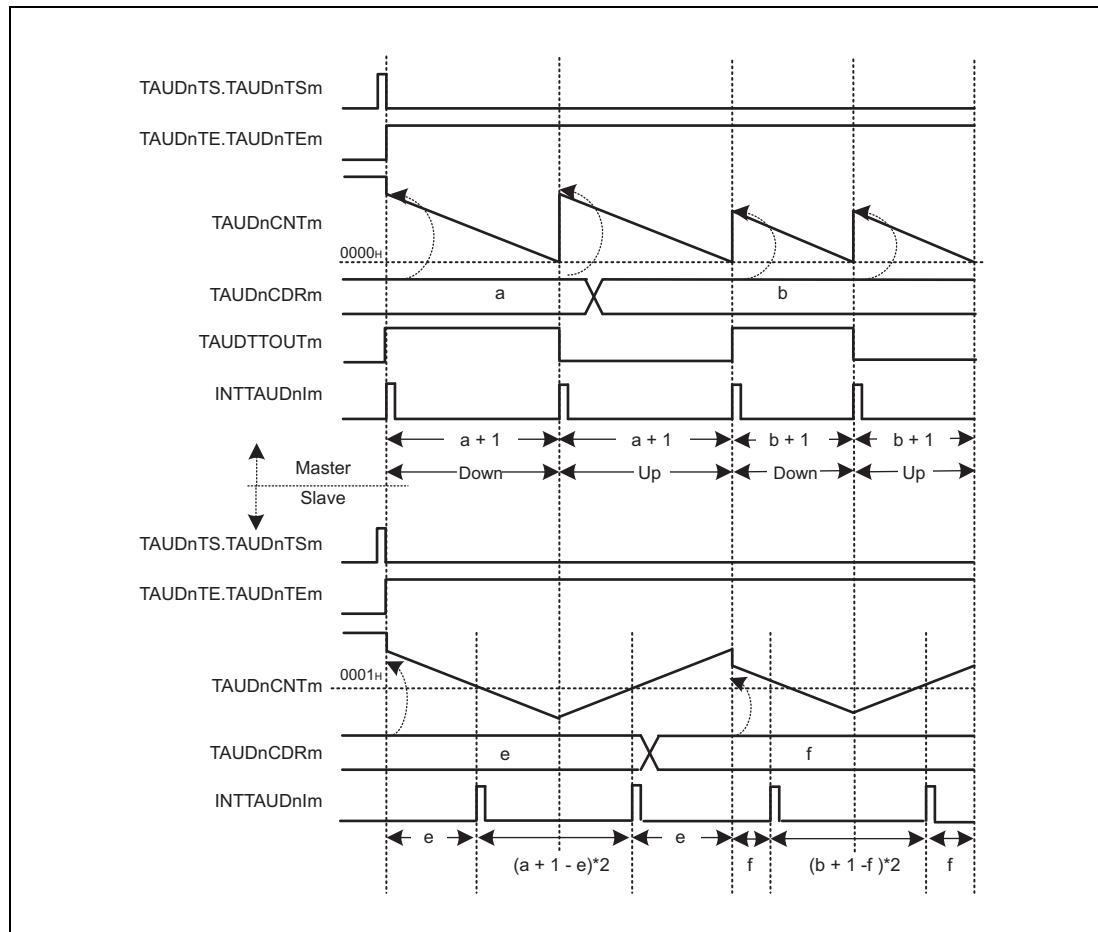


Figure 23.125 General Timing Diagram of A/D Conversion Trigger Output Function Type 2

## 23.15.10 Interrupt Request Signals Culling Function

### 23.15.10.1 Overview

#### Summary

This function divides the number of interrupts of the master channel by a specified value using a slave channel.

The interrupt request signals culling function is a sub function of the following functions:

- PWM Output Function (See **Section 23.15.1, PWM Output Function**)
- Triangle PWM Output Function (See **Section 23.15.7, Triangle PWM Output Function**)
- Triangle PWM Output Function with Dead Time  
(See **Section 23.15.8, Triangle PWM Output Function with Dead Time**)

#### Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 23.211, Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function**)
- The operation mode of the slave channel must be set to Event Count Mode. (See **Table 23.214, Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function**)
- This function does not use TAUDTOUTm.

#### Functional description

The counters (master and slave) are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for both channels. This in turn sets TAUDnTE.TAUDnTEM, enabling count operation. The current value of the data register of the master channel and slave channel (TAUDnCDRm) are written to the counter (TAUDnCNTm).

- Master channel:  
When the counter of the master channel reaches  $0000_H$ , INTTAUDnIm is generated and TAUDnCDRm value is reloaded to TAUDnCNTm.
- Slave channel:  
Every time the master channel generates an INTTAUDnIm, the counter of the slave channel decrements by one. When the counter reaches  $0000_H$ , it awaits the next interrupt from the master channel. This causes TAUDnCNTm (slave) to reload the value of TAUDnCDRm, and an INTTAUDnIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channels, which in turn sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm of master and slave channels stops but retains its value.

#### Conditions

This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

### 23.15.10.2 Equations

Interrupt division operator = TAUDnCDRm (slave channel)

- One INTTAUDnIm is generated for the INTTAUDnIm count of the master channel defined by TAUDnCDRm (slave channel) + 1.

### 23.15.10.3 Block Diagram and General Timing Diagram

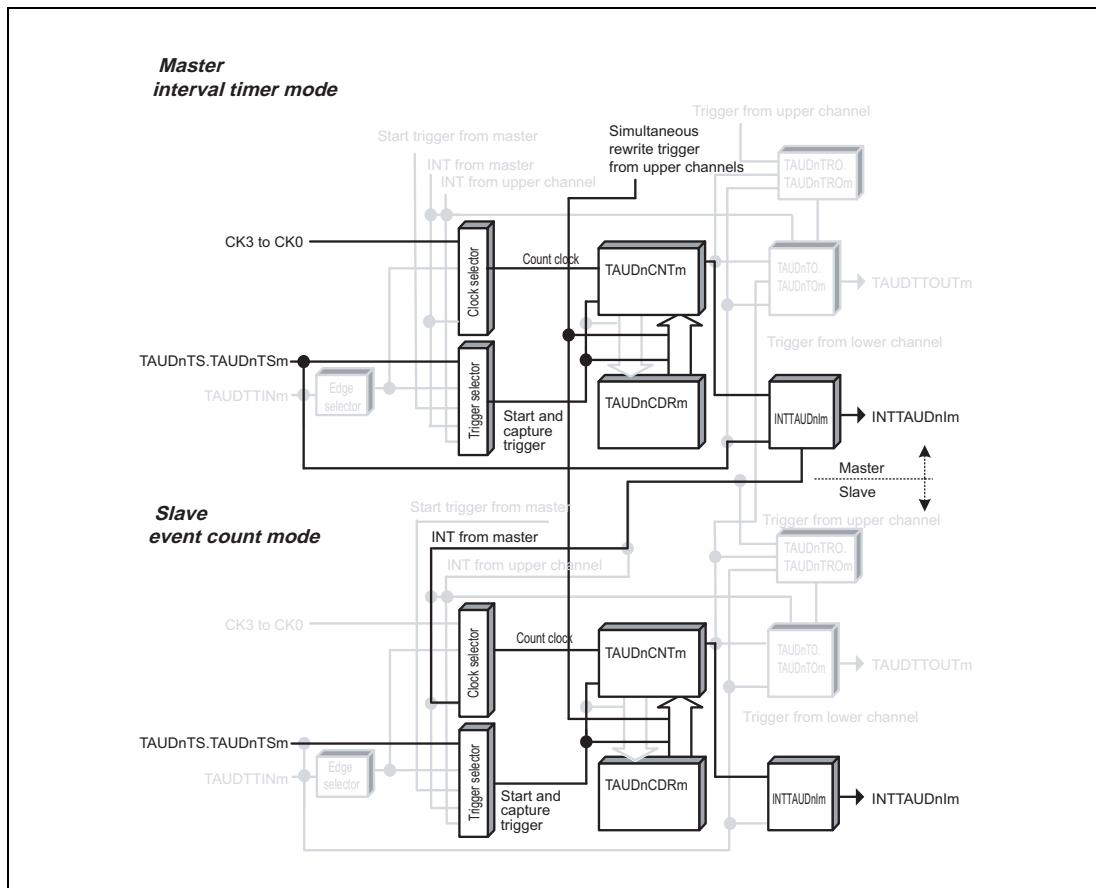


Figure 23.126 Block Diagram of Interrupt Request Signals Culling Function

The following settings apply to the general timing diagram.

Master channel:

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

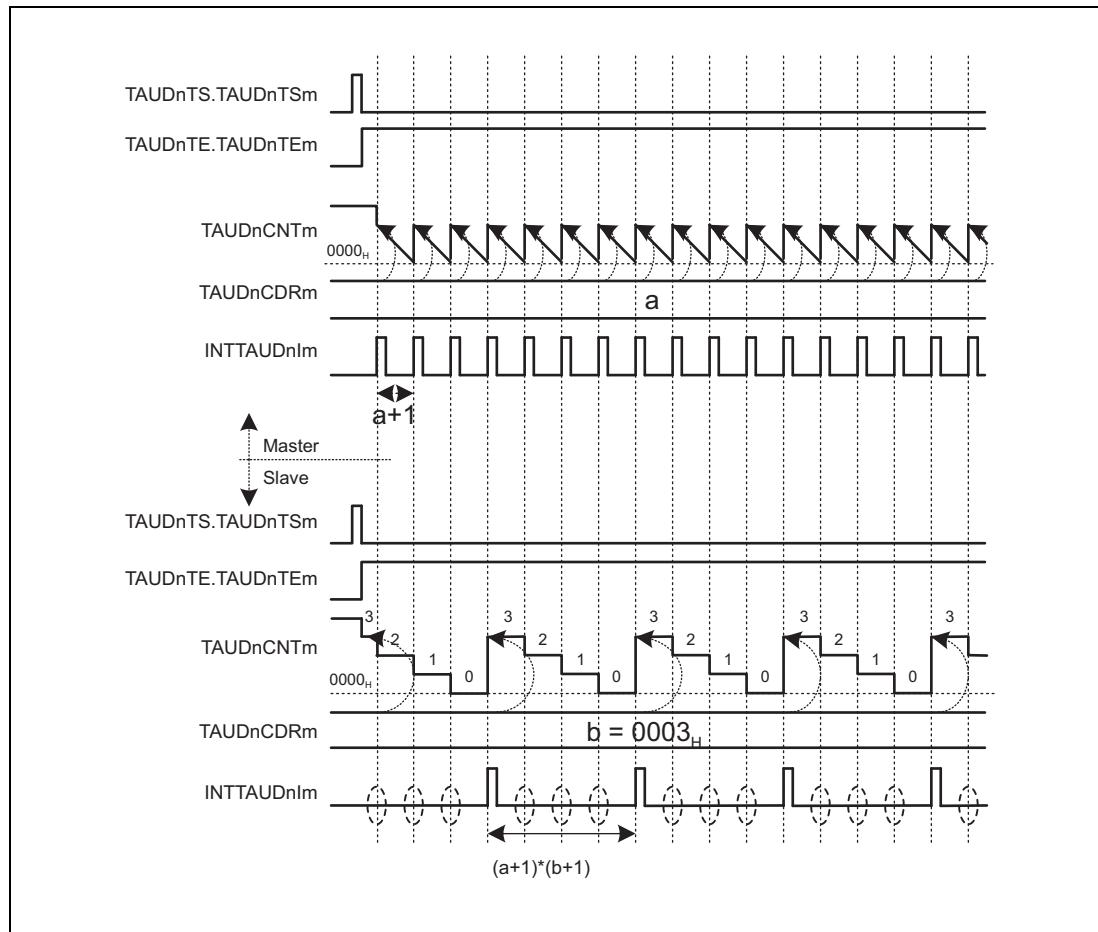


Figure 23.127 General Timing Diagram of Interrupt Request Signals Culling Function

### 23.15.10.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

R/W R/W

Table 23.211 Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0

R/W R R R R R R R/W R/W

Table 23.212 Contents of the TAUDnCMURm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.213 Simultaneous Rewrite Settings for the Master Channel of Interrupt Request Signals Culling Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

### 23.15.10.5 Register Settings for the Slave Channel

#### (1) TAUDnCMORm for the slave channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.214 Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMDO	0: INTTAUDnIm not generated at the beginning of operation.

#### (2) TAUDnCMURm for the slave channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.215 Contents of the TAUDnCMURm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

#### (3) Channel output mode for the slave channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite for the slave channel

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.216 Simultaneous Rewrite Settings for the Slave Channel of Interrupt Request Signals Culling Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

#### 23.15.10.6 Operating Procedure for Interrupt Request Signals Culling Function

**Table 23.217 Operating Procedure for Interrupt Request Signals Culling Function**

	Operation	TAUDn Status
Restart Operation	Initial Channel Setting  Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.10.4, Register Settings for the Master Channel</b> .  Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.15.10.5, Register Settings for the Slave Channel</b> .  Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
	Start Operation  Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
	During Operation  TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.  TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000H: <ul style="list-style-type: none"><li>• INTTAUDnIm (master) is generated.</li><li>• TAUDnCNTm (master) loads TAUDnCDRm value and continues count operation.</li><li>• TAUDnCNTm of slave channels counts down each time INTTAUDnIm of master channel is detected.</li></ul> When TAUDnCNTm of the slave = 0000H: <ul style="list-style-type: none"><li>• INTTAUDnIm (slave) is generated.</li><li>• The TAUDnCDRm value is loaded in TAUDnCNTm (slave) and count operation continues.</li></ul>
	Stop Operation  Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

### 23.15.10.7 Specific Timing Diagrams

#### (1) Interrupt count (master) = interrupt count (slave)

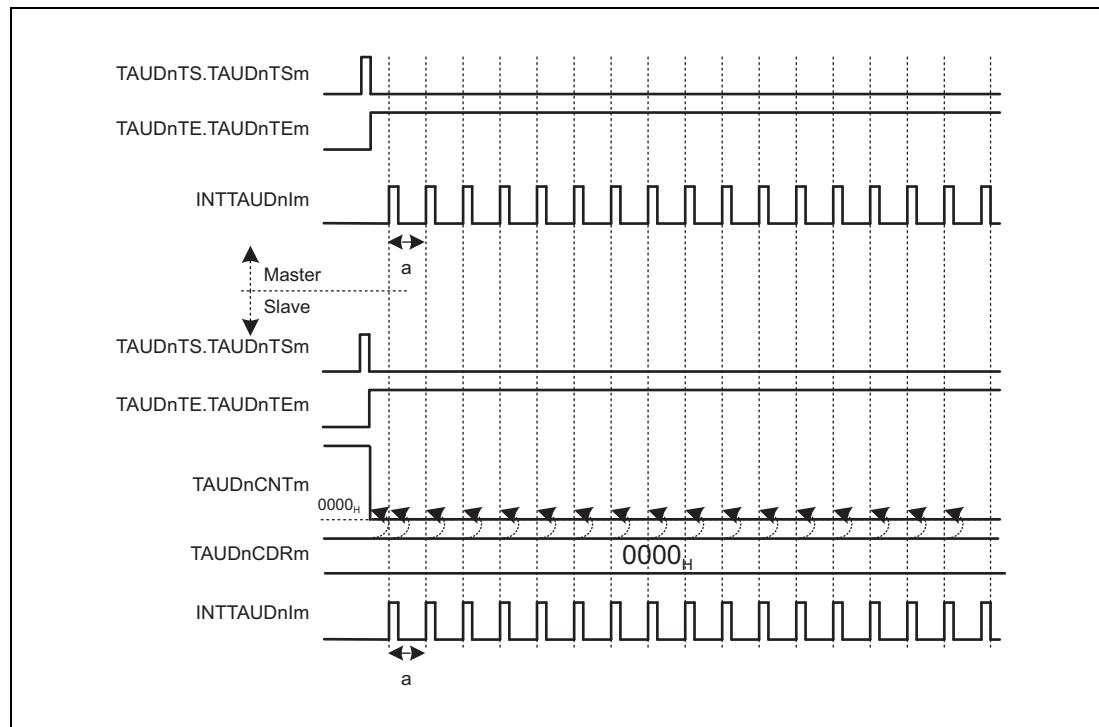


Figure 23.128 TAUDnCDRm (Slave) =  $0000_H$

- If  $\text{TAUDnCDRm} = 0000_H$ , the  $\text{TAUDnCDRm}$  value of the slave channel is loaded into  $\text{TAUDnCNTm}$  each time  $\text{INTTAUDnIm}$  of master channel is detected. In other words,  $\text{TAUDnCNTm}$  is always  $0000_H$ .
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

## 23.16 Synchronous Non-Complementary and Complementary Modulation Output Functions

This section describes functions that generate 6-phase PWM output or triangle PWM output using a master channel and seven slave channels.

### 23.16.1 Non-Complementary Modulation Output Function Type 1

#### 23.16.1.1 Overview

##### Summary

This function outputs a PWM signal, a high-level signal, or a low-level signal from TAUDTTOUTm depending on the values of the real-time output bits (TAUDnTRO.TAUDnTROm) and the modulation output enable bits (TAUDnTME.TAUDnTMem) of a pair of slave channels. Three pairs of channels are typically used.

##### Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 23.219, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1**).
- The operating mode for slave channels 1 to 7 should be set to one-count mode (See **Table 23.222, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1**, and **Table 23.225, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1**).
- TAUDTTOUTm is not used with the master channel of this function.
- TAUDTTOUTm of slave channel 1 is not used with this function, but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 23.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 1 with non-complementary modulation output (See **Section 23.7, Channel Output Modes**).
- TAUDnCDRm of slave channel 1 should be set to 0000<sub>H</sub>.

##### Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down. When the counter reaches 0000<sub>H</sub>, INTTAUDnIm is generated.

- Slave channel 1:  
Slave channel 1 is set as a channel that triggers real-time output (TAUDnTRC.TAUDnTRCm = 1). If an interrupt occurs on slave channel 1 (TAUDnCDRm is fixed to 0000<sub>H</sub>), the value of real-time output bit (TAUDnTRO.TAUDnTROm) of the channel that monitors the interrupt on slave channel 1 is reflected to the TAUDTTOUTm output. After that, the counter returns to FFFF<sub>H</sub> and waits for the next interrupt of master channel.

- Slave channel 2:

Slave channel 2 generates a PWM output. The master channel specifies a PWM output cycle and slave channel 2 specifies a duty cycle. After generating an interrupt, the counter returns to  $FFFF_H$  and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in **Table 23.218, TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)**, a signal output from TAUDTTOUTm depends on the value of the real-time output bit (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSM to 1.

### Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

**Table 23.218 TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)**

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.
- TAUDnCDRm value of slave channel 1 should be set to  $0000_H$  so that a real-time output is triggered at the same time with PWM generation on slave channels 2 to 7.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOM is set to 0 (low) before TAUDnTE.TAUDnTEM is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOM is set to 1 (high) before TAUDnTE.TAUDnTEM is set to 0.

### 23.16.1.2 Equations

Slave channels 2 to 7:

Pulse period = [TAUDnCDRm (master) + 1] × count clock cycle

Duty time = [TAUDnCDRm (slave)] × count clock cycle

### 23.16.1.3 Block Diagram and General Timing Diagram

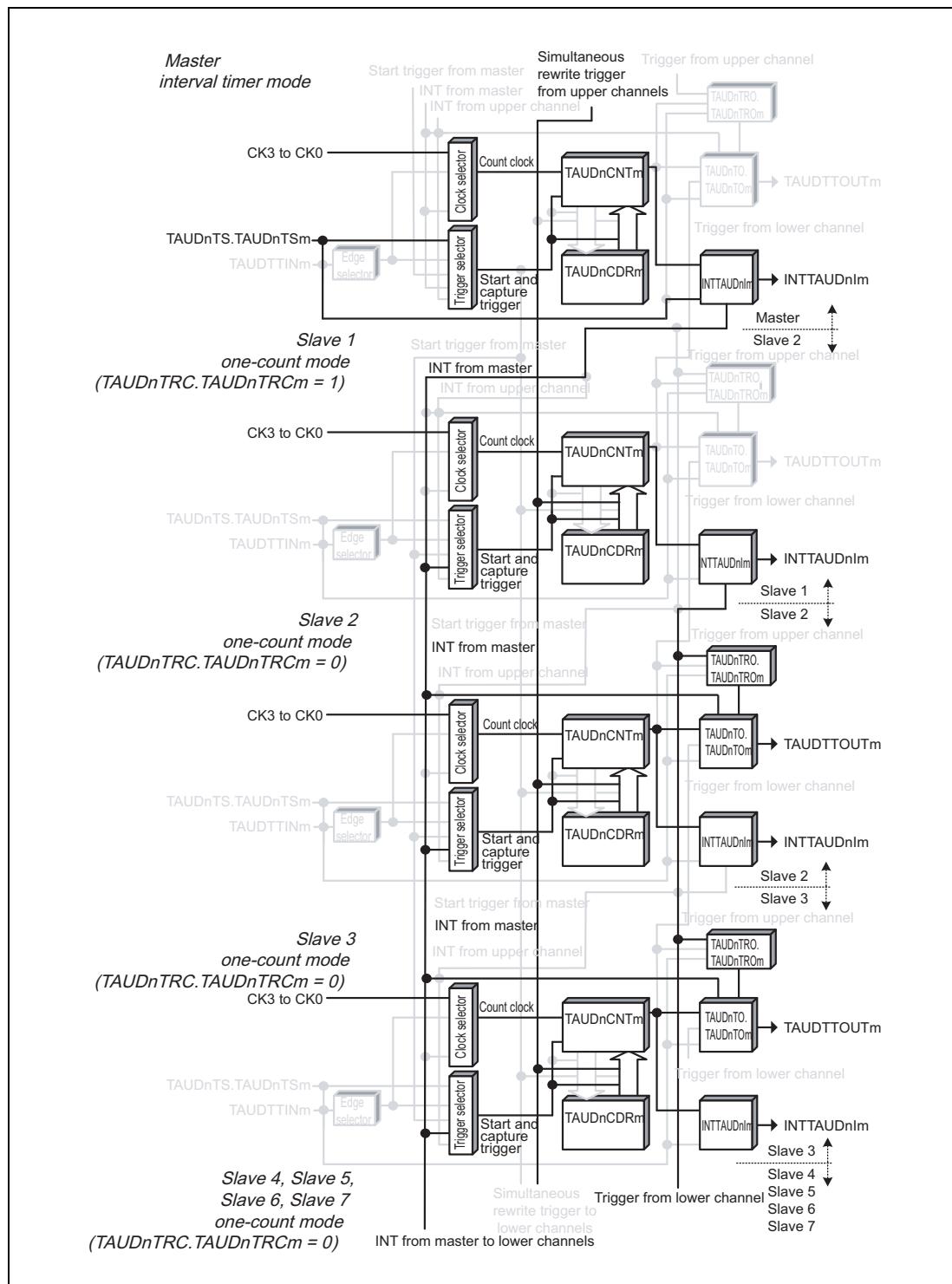
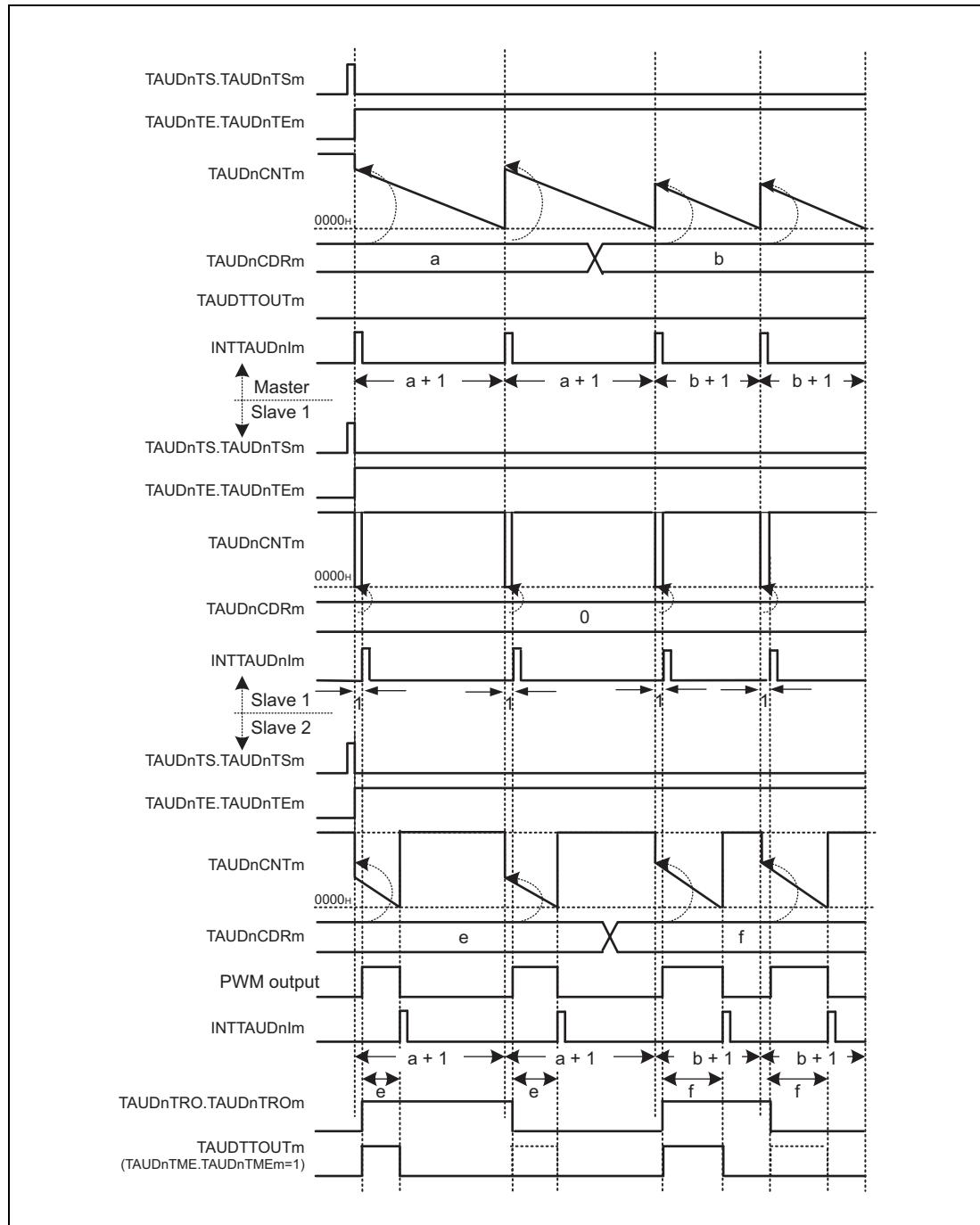


Figure 23.129 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)



**Figure 23.130 General Timing Diagram of Non-Complementary Modulation Output Function Type 1**

#### NOTE

TAUDTTOUTm of slave channel 2 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

### 23.16.1.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.219 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.220 Contents of the TAUDnCMURm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.221 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 1**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

#### NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires an upper channel higher than the master channel that operates with **Section 23.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions.

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1  
In addition, TAUDnCDRm settings for this channel are as follows.  

$$= ((\text{TAUDnCDR setting for the master channel targeted for simultaneous rewrite} + 1) \times \text{Interrupt count}) - 1$$
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

### 23.16.1.5 Register Settings for Slave Channel 1

#### (1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.222 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Start trigger during operation is valid.

#### (2) TAUDnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.223 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

#### **CAUTION**

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

### (4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.224 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.1.6 Register Settings for Slave Channels 2 to 7

#### (1) TAUDnCMORm for slave channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.225 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnlm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Start trigger during operation is valid.

#### (2) TAUDnCMURm for slave channels 2 to 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.226 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for slave channels 2 to 7

**Table 23.227 Control Bit Settings in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

### (4) Simultaneous rewrite of slave channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.228 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.1.7 Operating Procedure for Non-Complementary Modulation Output Function Type 1

**Table 23.229 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (1/2)**

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.1.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.1.5, Register Settings for Slave Channel 1</b>.</p> <p>Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.1.6, Register Settings for Slave Channels 2 to 7</b>.</p> <p>Set the value of TAUDnCDRm register of every channel. Set a pulse cycle with TAUDnCDRm of master channel, <math>0000_{\text{H}}</math> in TAUDnCDRm of slave channel 1, and duty width with TAUDnCDRm of slave channels 2 to 7.</p> <p>Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.</p>	Channel operation is stopped.

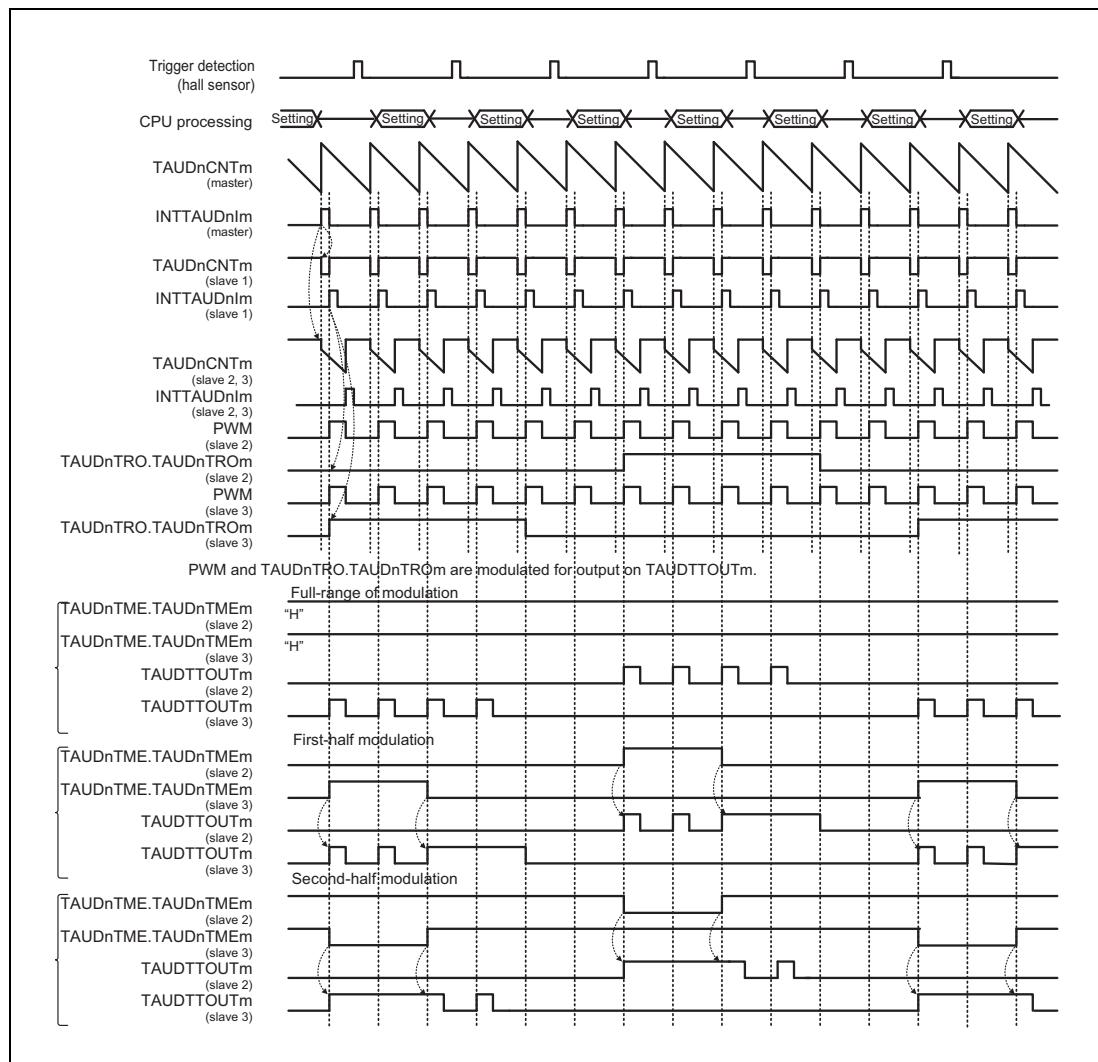
**Table 23.229 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (2/2)**

	<b>Operation</b>	<b>TAUDn Status</b>
Start Operation  →  During Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p> <p>TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnTE.TAUDnTEM of master and slave channels is set to 1 and the counter starts counting down.</p> <p>TAUDnCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>TAUDnCDRm value of master channel is reloaded into TAUDnCNTm to continue counting down.</li> <li>PWM output signals of slave channels 2 to 7 are set.</li> <li>TAUDnCDRm value of slave channel 1 is reloaded into TAUDnCNTm to perform counting down.</li> <li>TAUDnCDRm value of slave channels 2 to 7 is reloaded into TAUDnCNTm to perform counting down.</li> <li>When the counter of slave channel 1 reaches 0000H: <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output.</li> </ul> </li> <li>When the counter of slave channels 2 to 7 reaches 0000H: <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>PWM output signals of slave channels 2 to 7 are set.</li> </ul> </li> </ul> <p>TAUDTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of a pair of slave channels.</p>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.</p>

### 23.16.1.8 Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)



**Figure 23.131 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1**

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and master channel cycle. According to the modified setting, modulation waveforms are output from TAUDTTOUTm.

A TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

## 23.16.2 Non-Complementary Modulation Output Function Type 2

### 23.16.2.1 Overview

#### Summary

This function outputs a triangular PWM output signal, a high-level signal, or low-level signal from TAUDTTOUTm depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output enable bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

#### Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 23.231, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 23.235, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channels 2 to 7 should be set to count-up/-down mode (See **Table 23.238, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2**).
- The output mode for the master channel should be set to independent channel output mode 1. (See **Section 23.7, Channel Output Modes**.)
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 23.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output (See **Section 23.7, Channel Output Modes**).

#### Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm).

- Master channel:  
The counter of master channel starts to count down. When the counter reaches 0000<sub>H</sub>, INTTAUDnIm is generated.
- Slave channel 1:  
When slave channel 1 detects an interrupt from the master channel, the TAUDnCNTm value is decremented. When an interrupt from the master channel is detected (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, the TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.  
Since slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1), if an interrupt occurs on slave channel 1, the real-time output bit (TAUDnTRO.TAUDnTROm) of the channel which monitors an interrupt on the corresponding channel is reflected to the TAUDTTOUTm output.

- Slave channel 2:

Once detecting an interrupt from the master channel, TAUDnCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUDnCDRm value is reloaded and then the counter starts to count down.

If TAUDnCNTm = 0001<sub>H</sub>, an interrupt occurs and a PWM output signal is set/reset.

The combined use of the master channel and slave channel 2 generates a PWM output signal. The master channel generates a PWM output cycle and slave channel 2 generate a duty cycle.

Slave channels 3 to 7 operate like slave channel 2.

A signal that is output from TAUDTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm) and a modulation output bit value (TAUDnTME.TAUDnTMEm) of the slave channel, as described in **Table 23.230, TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)**.

This function cannot make a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSM to 1.

### Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

**Table 23.230 TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)**

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOM is set to 0 (low) before TAUDnTE.TAUDnTEM is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOM is set

to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

### 23.16.2.2 Equations

Slave channels 2 to 7:

Carrier cycle (down/up) = [TAUDnCDRm (master) + 1] × 2 × count clock cycle

Duty time = [TAUDnCDRm (master) + 1 - TAUDnCDRm (slave)] × 2 × count clock cycle

### 23.16.2.3 Block Diagram and General Timing Diagram

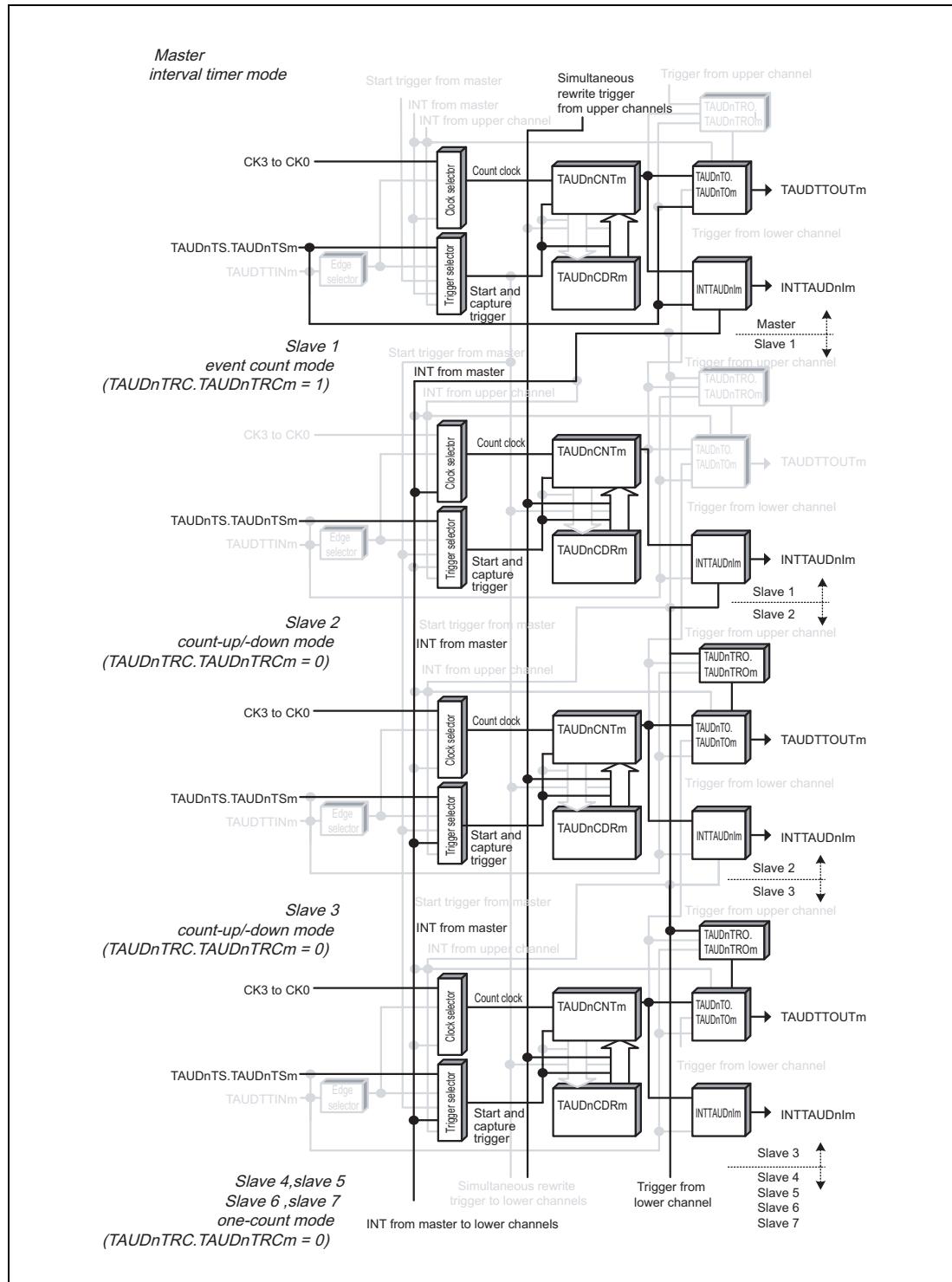


Figure 23.132 Block Diagram of Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

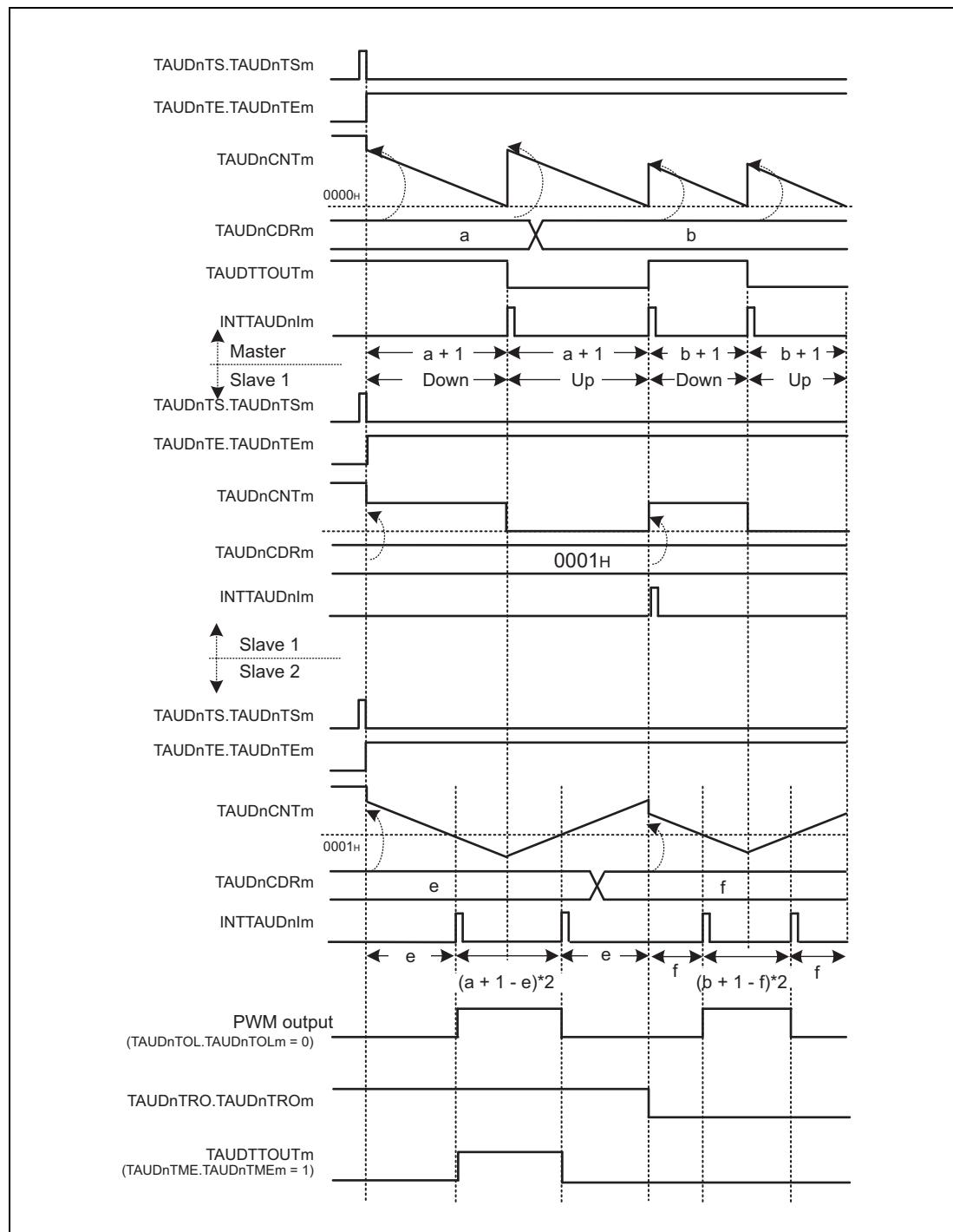


Figure 23.133 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

### 23.16.2.4 Register Settings the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.231 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.232 Contents of the TAUDnCMURm Register for the Master channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

**Table 23.233 Control Bit Settings for the Master Channel in Non-Complementary Modulation Output Function Type 2**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (toggle mode with TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.234 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 2**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

#### NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

### 23.16.2.5 Register Settings for Slave Channel 1

#### (1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.235 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMDO	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

#### (2) TAUDnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.236 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

#### **CAUTION**

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

### (4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.237 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.2.6 Register settings for slave channels 2 to 7

#### (1) TAUDnCMORm for slave channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.238 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: The up/down output trigger signal of the master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMDO	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

#### (2) TAUDnCMURm for slave channels 2 to 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.239 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Output mode for slave channels 2 to 7

**Table 23.240 Control Bit Settings in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: The upper channel generates the real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

### (4) Simultaneous rewrite for slave channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.241 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.2.7 Operating Procedure for Non-Complementary Modulation Output Function Type 2

**Table 23.242 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (1/2)**

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.2.4, Register Settings the Master Channel</b>.</p> <p>Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.2.5, Register Settings for Slave Channel 1</b>.</p> <p>Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.2.6, Register settings for slave channels 2 to 7</b>.</p> <p>Set the value of TAUDnCDRm register of every channel. Set pulse cycle in TAUDnCDRm of master channel, and in TAUDnCDRm of slave channel 1, set the number of interrupts from master channel to be ignored before slave channel 1 generates a real-time output trigger. Set duty width in TAUDnCDRm of slave channels 2 to 7.</p> <p>Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.</p>	Channel operation is stopped.

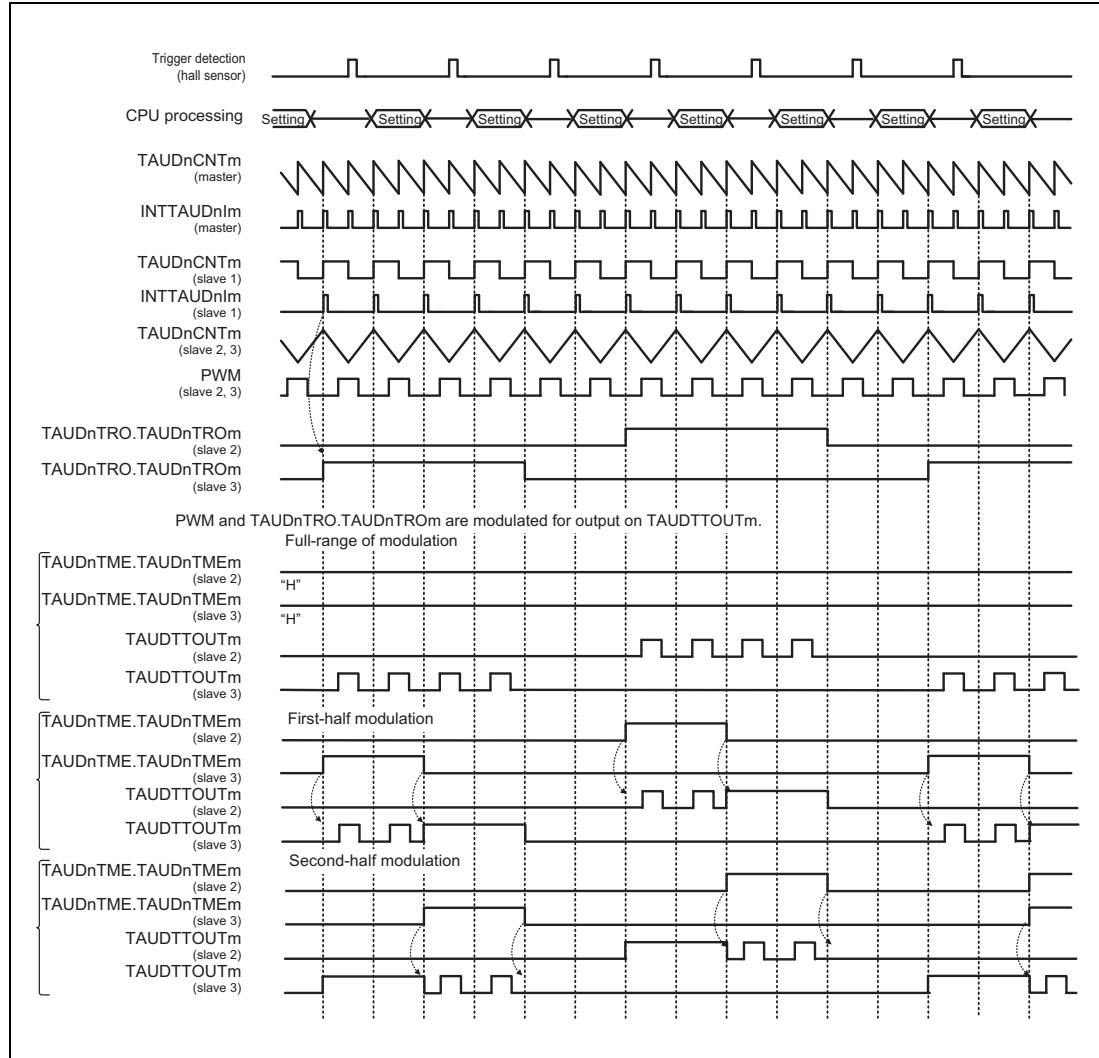
**Table 23.242 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (2/2)**

	<b>Operation</b>	<b>TAUDn Status</b>
Start Operation	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	TAUDnTE.TAUDnTEM of master and slave channels is set to 1 and the counter starts counting down.
During Operation	<p>TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>The TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. The TAUDnCDRm value of slave channel 1 is loaded and the counter waits for an interrupt from the master channel. When the counter of master channel reaches <math>0000_H</math>:</p> <ul style="list-style-type: none"> <li>• INTTAUDnIm is generated.</li> <li>• TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down.</li> <li>• The TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel.</li> <li>• TAUDnCNTm of slave channels 2 to 7 reloads the TAUDnCDRm value, or performs counting in opposite direction.</li> <li>• At the same timing when the TAUDnCDRm value is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDTOUTm output.</li> <li>• When slave channel 1 detects an interrupt from the master channel for the <math>(TAUDnCDRm + 1)</math> times: <ul style="list-style-type: none"> <li>– INTTAUDnIm is generated.</li> <li>– The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDTOUTm output.</li> </ul> </li> <li>• When the counter of slave channels 2 to 7 reaches <math>0001_H</math>: <ul style="list-style-type: none"> <li>– INTTAUDnIm is generated.</li> <li>– PWM output signals of slave channels 2 to 7 are set/reset.</li> </ul> </li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	TAUDnTE.TAUDnTEM is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTOUTm stop and retain their current values.

### 23.16.2.8 Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)



**Figure 23.134 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2**

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

### 23.16.3 Complementary Modulation Output Function

#### 23.16.3.1 Overview

##### Summary

This function outputs a triangle PWM output signal, a high-level signal, or low-level signal from TAUDTTOUTm with dead time added, depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels, and an output level bit value (TAUDnTDL.TAUDnTDLm). Three pairs of channels are typically used.

##### Prerequisites

- One master channel and seven slave channels
  - The operation mode of the master channel must be set to interval timer mode (See **Table 23.244, Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function**).
  - The operating mode for slave channel 1 should be set to event count mode (See **Table 23.248, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function**).
  - The operating mode for slave channels 2, 4 and 6 should be set to count-up/-down mode (See **Table 23.251, Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function**).
  - The operating mode for slave channels 3, 5 and 7 should be set to one-count mode (See **Table 23.255, Contents of the TAUDnCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function**).
- In addition, as the number of occurrences of an interrupt for slave channels 3, 5 and 7 within the carrier cycle is not uniquely determined, do not use the interrupt as an interrupt source.
- The output mode for master channels should be set to independent channel output mode 1 (See **23.7, Channel Output Modes**).
  - This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 23.7, Channel Output Modes**).
  - The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with complementary modulation output (See **Section 23.7, Channel Output Modes**).

##### Functional description

- Master channel:  
The counter of the master channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEM = 1, enabling count operation. The value of data register (TAUDnCDRm) of the master channel is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.  
When the counter of master channel reaches 0000<sub>H</sub>, INTTAUDnIm is generated. This decrements the counter value of slave channel 1 by 1 and the counter of slave channel 2 starts to count in the opposite direction.
- Slave channel 1:  
When the counter reaches 0000<sub>H</sub>, slave channel 1 waits for the next interrupt from the master

channel. And the TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1) and INTTAUDnIm is generated.

Slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1). The value of real-time output bit (TAUDnTRO.TAUDnTROm) of each channel is applied to the channel that detects the occurrence of an interrupt on slave channel 1 by an interrupt. The real-time output bit value can be changed in any timing by application software but a new value is not applied until an interrupt occurs on slave channel 1.

- Slave channel 2:

When the slave channel 2 counter reaches  $0001_H$ , the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches  $0000_H$ , an interrupt occurs.

- Slave channels 2 and 3:

The combined use of the master channel and slave channels 2 and 3 generates a PWM output signal. The master channel generates a PWM output cycle, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time.

- Slave channels 4 to 7:

Slave channels 4 and 6 operate like slave channel 2. Slave channels 5 and 7 operate like slave channel 3.

A signal that is output from TAUDTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm), a modulation output bit value (TAUDnTME.TAUDnTMEm), and an output level bit value (TAUDnTDL.TAUDnTDLm) of the slave channel, as described in **Table 23.243, TAUDTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function.**

It is, however, prohibited that a high-level signal is output from both channel 2 and channel 3 (in order to prevent a motor driver short circuit).

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEM to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSM to 1.

### Conditions

- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 1 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm of one channel is set to 1, TAUDTTOUTm outputs the corresponding PWM of the channel.
  - If TAUDnTRO.TAUDnTROm of both channels is set to 0, TAUDTTOUTm of a pair outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 0 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm of the channel outputs a high-level signal.
  - If TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm of the channel outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic.

**Table 23.243 TAUDTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function**

TAUDnTME.T AUDnTME2	TAUDnTME.T AUDnTME3	TAUDnTRO.T AUDnTRO2	TAUDnTRO.T AUDnTRO3	TAUDnTDL.T AUDnTDL2	TAUDnTDL.T AUDnTDL3	TAUDTOUT2 Output	TAUDTOUT3 Output
0	0	0	0	X	X	Low level	Low level
		0	1	1	0	Low level	High level
		1	0	0	1	High level	Low level
		1	1	X	X	Setting prohibited	Setting prohibited
1	1	0	0	X	X	Low level	Low level
		0	1	1	0	~PWM	PWM
		1	0	0	1	PWM	~PWM
		1	1	X	X	Setting prohibited	Setting prohibited

#### NOTES

- In the above table, PWM indicates a positive PWM signal and ~PWM indicates an inverted PWM signal (positive logic). PWM and ~PWM are set by TAUDnTDL.TAUDnTDLm.
- Any settings not listed above are prohibited.

- If TAUDnTME.TAUDnTMEm is continuously set to 1 while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the first half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the second half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, second-half modulation is applied.
- Whether dead time is added to a normal or reverse phase PWM signal when two channels become high-level signal outputs simultaneously depends on a TAUDnTDL.TAUDnTDLm bit value.
  - If TAUDnTDL.TAUDnTDLm = 0, dead time is added to a normal phase PWM signal.
  - If TAUDnTDL.TAUDnTDLm = 1, dead time is added to a reverse phase PWM signal.
  - The operation defined by a TAUDnTDL.TAUDnTDLm bit value should be conducted by application software during operation. To modify TAUDnTDL.TAUDnTDLm, rewrite it during the period when TAUDnTRO.TAUDnTROm is 00<sub>B</sub>.
- The TAUDnCDRm value of slave channel 1 should be set to the value to generate INTTAUDnIm of slave channel 1 at a carrier cycle (peak interrupt timing).
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOM is set to 0 (low) before TAUDnTE.TAUDnTEM is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOM is set to 1 (high) before TAUDnTE.TAUDnTEM is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOM is set to 1 (high) before TAUDnTE.TAUDnTEM is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOM is set to 0 (low) before TAUDnTE.TAUDnTEM is set to 0.

- This function enables simultaneous rewrite. See **Section 23.6, Simultaneous Rewrite**.

### 23.16.3.2 Equations

Pulse period =  $(\text{TAUDnCDRm (master)} + 1) \times \text{count clock cycle}$

$0000_{\text{H}} \leq \text{TAUDnCDRm (master)} < FFFF_{\text{H}}$

Carrier cycle (down/up) =  $(\text{TAUDnCDRm (master)} + 1) \times 2 \times \text{count clock cycle}$

For slave channels 2 and 3:

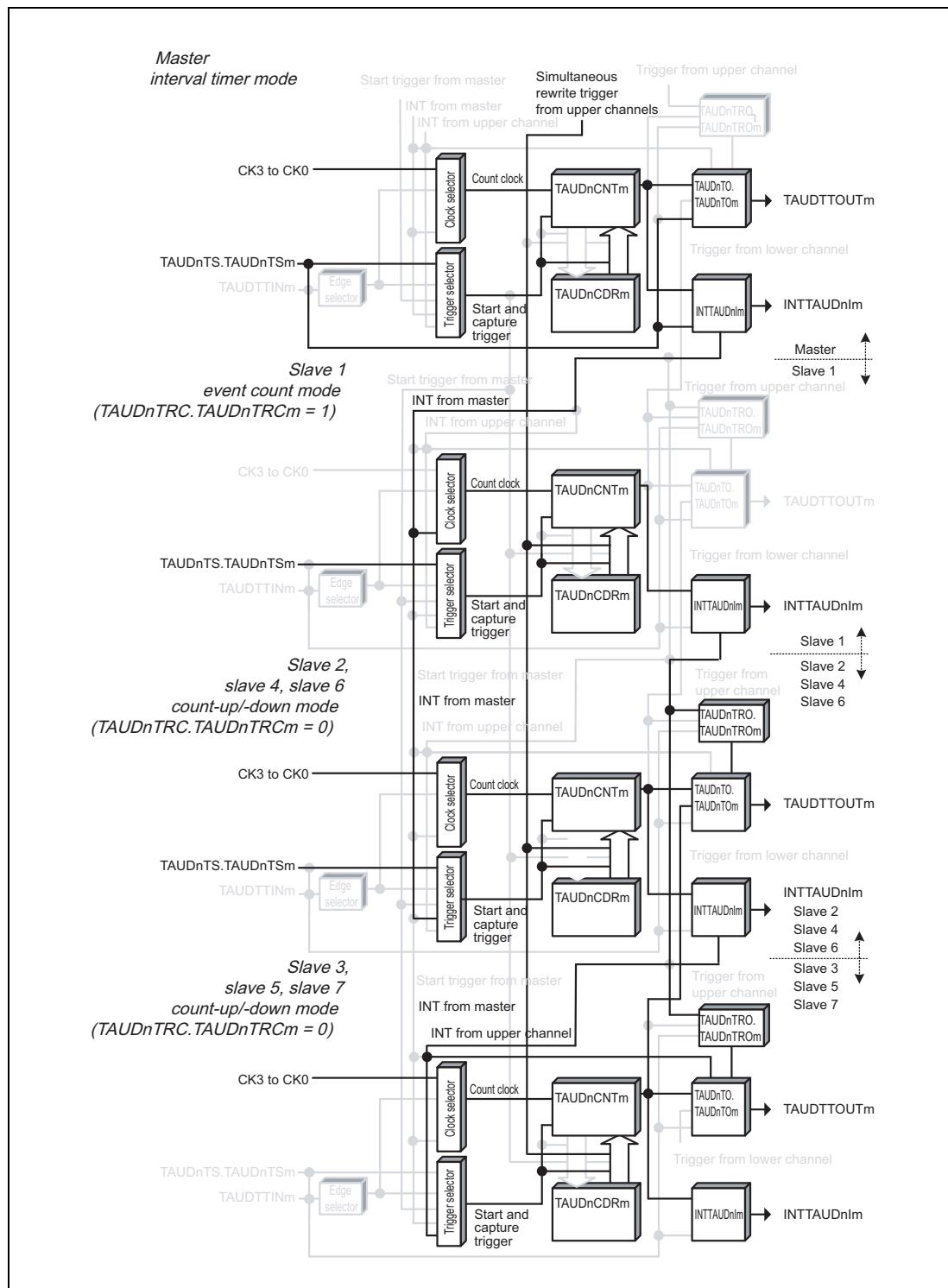
PWM signal width (positive phase) =  $[(\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave 2)} \times 2) - (\text{TAUDnCDRm (slave 3)} + 1)] \times \text{count clock cycle}$

PWM signal width (negative phase) =  $[(\text{TAUDnCDRm (master)} + 1 - \text{TAUDnCDRm (slave 2)} \times 2) + (\text{TAUDnCDRm (slave 3)} + 1)] \times \text{count clock cycle}$

For slave channels 4 to 7:

Slave channels 4 and 6 are calculated in the same way as slave channel 2, whereas slave channels 5 and 7 are calculated as slave channel 3.

### 23.16.3.3 Block Diagram and General Timing Diagram



**Figure 23.135** Block Diagram of Complementary Modulation Output Function

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 0)
  - Slave channel 1: TAUDnCDRm = 0001<sub>H</sub>

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

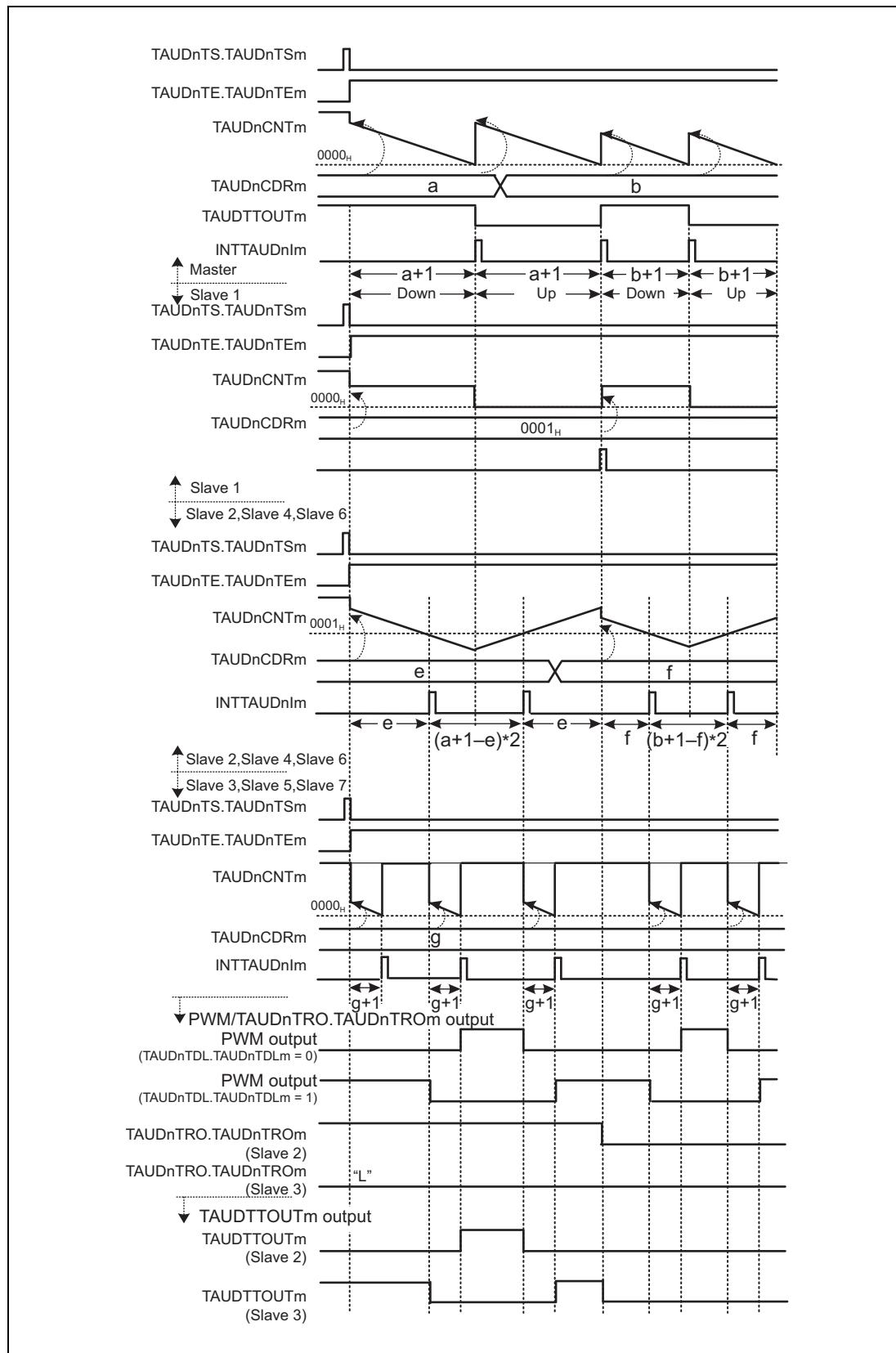


Figure 23.136 General Timing Diagram of Complementary Modulation Output Function

### 23.16.3.4 Register Settings for the Master Channel

#### (1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.244 Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMDO	0: INTTAUDnIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

#### (2) TAUDnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.245 Contents of the TAUDnCMURm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode for the master channel

Table 23.246 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEM	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEM = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREM	0: Disables real-time output
TAUDnTRO.TAUDnTROM	0: When real-time output is disabled (TAUDnTRE.TAUDnTREM = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEM	0: Disables modulation

### (4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.247 Simultaneous Rewrite Settings for the Master Channel of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

#### NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

### 23.16.3.5 Register Settings for Slave Channel 1

#### (1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.248 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMDO	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

#### (2) TAUDnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.249 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

#### **CAUTION**

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

### (4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.250 Simultaneous Rewrite Settings for Slave Channel 1 of Complementary Modulation Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.3.6 Register settings for slave channels 2, 4, and 6

#### (1) TAUDnCMORm for slave channels 2, 4, and 6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.251 Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMDO	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

#### (2) TAUDnCMURm for slave channels 2, 4, and 6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.252 Contents of the TAUDnCMURm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Output mode for slave channels 2, 4, and 6

**Table 23.253 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

#### CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

### (4) Simultaneous rewrite for slave channels 2, 4, and 6

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.254 Simultaneous Rewrite Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.3.7 Register settings for slave channels 3, 5, and 7

#### (1) TAUDnCMORm for slave channels 3, 5, and 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]	TAUDnCCS[1:0]	TAUDnMAS		TAUDnSTS[2:0]	TAUDnCOS[1:0]	—		TAUDnMD[4:1]		TAUDnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.255 Contents of the TAUDnCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMDO	1: Enables start trigger detection while counting.

#### (2) TAUDnCMURm for slave channels 3, 5, and 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.256 Contents of the TAUDnCMURm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

### (3) Output mode for slave channels 3, 5, and 7

**Table 23.257 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output**

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEM	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

#### CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from even channels.

### (4) Simultaneous rewrite for slave channels 3, 5, and 7

Both the master and slave channels should have the same simultaneous rewrite settings.

**Table 23.258 Simultaneous Rewrite Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function**

Bit Name	Setting
TAUDnRDE.TAUDnRDEM	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

### 23.16.3.8 Operating Procedure for Complementary Modulation Output Function

Table 23.259 Operating Procedure for Complementary Modulation Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	<p>Master channel: Set TAUDnCMORM and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.3.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel 1: Set TAUDnCMORM and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.3.5, Register Settings for Slave Channel 1</b>.</p> <p>Slave channels 2, 4, and 6: Set TAUDnCMORM and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.3.6, Register settings for slave channels 2, 4, and 6</b>.</p> <p>Slave channels 3, 5, and 7: Set TAUDnCMORM and TAUDnCMURm registers and the channel output mode as described in <b>Section 23.16.3.7, Register settings for slave channels 3, 5, and 7</b>.</p> <p>Set the value of TAUDnCDRm register of every channel. Set a pulse cycle using TAUDnCDRm of master channel, and an interrupt count of master channel to be ignored using TAUDnCDRm of slave channel 1. Also set a duty width in TAUDnCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7.</p> <p>Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.</p>	Channel operation is stopped.

Table 23.259 Operating Procedure for Complementary Modulation Output Function (2/2)

	Operation	TAUDn Status
Start Operation →	<p>Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.</p>	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	<p>TAUDnCDRm, TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time.</p> <p>TAUDnRDT.TAUDnRDTm can be changed during operation.</p>	<p>TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. TAUDnCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches <math>0000_H</math>:</p> <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down.</li> <li>TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for the next master channel interrupt.</li> <li>TAUDnCNTm of slave channels 2, 4, and 6 reloads the TAUDnCDRm value, or performs counting in opposite direction.</li> <li>At the same timing when the TAUDnCDRm value of slave channels 2, 4, and 6 is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDTOUTm output.</li> <li>The counter of slave channel 1 waits for the next interrupt from the master channel when reaching <math>0000_H</math>. When the interrupt is detected: <ul style="list-style-type: none"> <li>TAUDnCDRm value is reloaded into TAUDnCNTm and the counter waits for the next master channel interrupt.</li> <li>INTTAUDnIm is generated.</li> <li>TAUDnTRO.TAUDnTROm is changeable.</li> </ul> </li> <li>When the counter of slave channels 2, 4, and 6 reaches <math>0001_H</math>: <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched).</li> <li>TAUDnCDRm value of slave channels 3, 5, and 7 is loaded into TAUDnCNTm to perform counting down.</li> </ul> </li> <li>When the counter of slave channels 3, 5, and 7 reaches <math>0000_H</math>: <ul style="list-style-type: none"> <li>INTTAUDnIm is generated.</li> <li>PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched).</li> </ul> </li> </ul>
Stop Operation	<p>Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.</p>	<p>TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops.</p> <p>TAUDnCNTm and TAUDTOUTm stop and retain their current values.</p>

### 23.16.3.9 Specific Timing Diagrams

The following settings apply to the timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation.  
(TAUDnCMORm.TAUDnMD0 = 0)
- Slave channel 1: TAUDnCDRm = 0001<sub>H</sub>
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

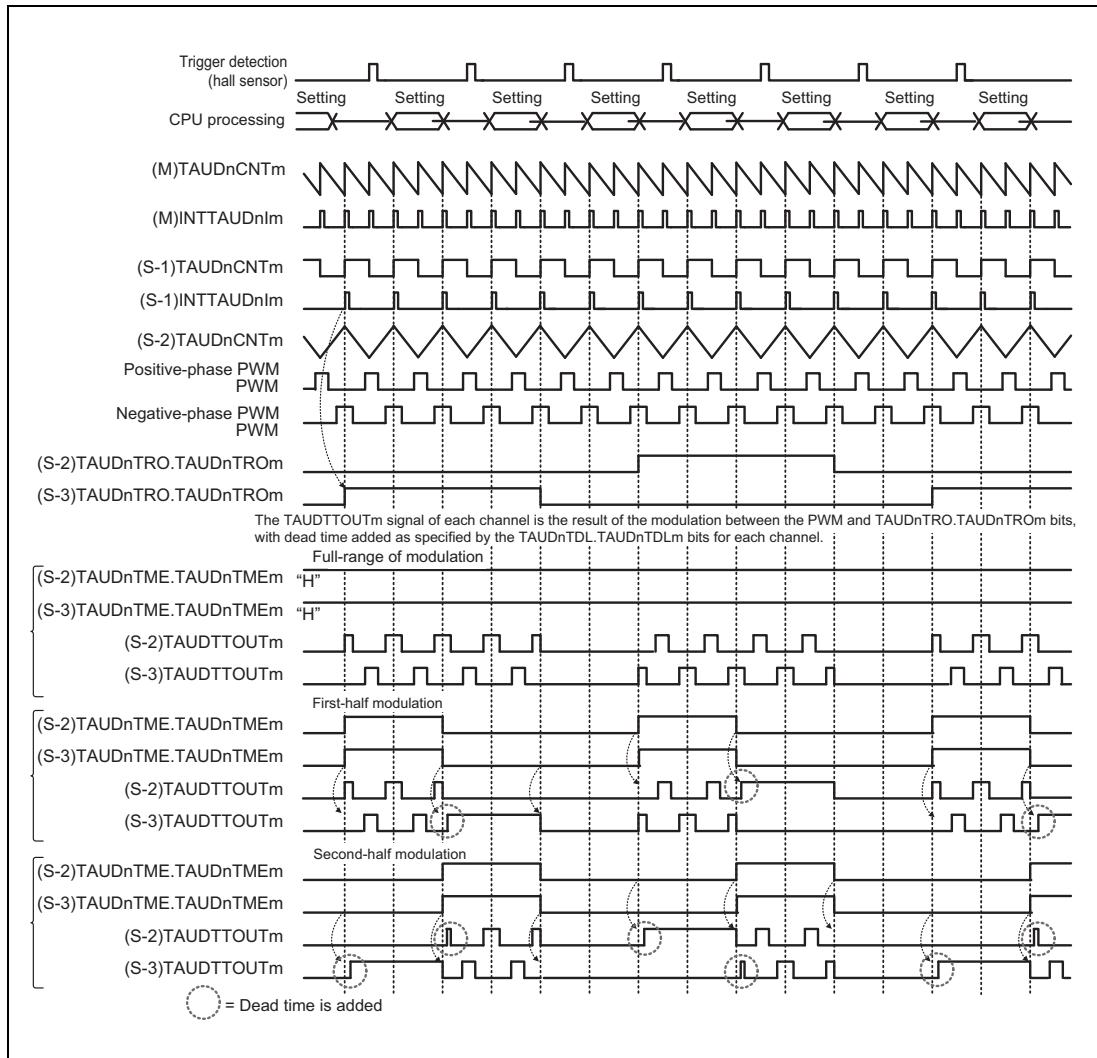


Figure 23.137 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUDnTRO.TAUDnTROm bit value are output from slave channels 2 and 3.

TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm settings are reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

**NOTE**

---

Dead time is added to suppress simultaneous change of PWM edges of normal and reverse phases.

---

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, TAUDnTRO.TAUDnTROm, and TAUDnTDL.TAUDnTDLm can be changed.

## Section 24 Timer Array Unit J (TAUJ)

This section contains a generic description of the timer array unit J (TAUJ).

The first part of this section describes the RH850/F1L specific features such as the number of units and the register base addresses. The remainder of the section describes the functions and registers of the TAUJ.

### 24.1 Features of RH850/F1L TAUJ

#### 24.1.1 Number of Units

This microcontroller has the following number of TAUJ units.

**Table 24.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	2	2	2
Name	TAUJ <sub>n</sub> (n = 0)	TAUJ <sub>n</sub> (n = 0)	TAUJ <sub>n</sub> (n = 0)	TAUJ <sub>n</sub> (n = 0, 1)	TAUJ <sub>n</sub> (n = 0, 1)	TAUJ <sub>n</sub> (n = 0, 1)

TAUJ<sub>n</sub> has the following number of channels of timers.

**Table 24.2 TAUJ<sub>n</sub> Unit Configurations and Channels**

Unit Name (Channel Name) TAUJ <sub>n</sub>	Number of Channels per Unit	RH850/F1L 48 pins (4 ch)	RH850/F1L 64 pins (4 ch)	RH850/F1L 80 pins (4 ch)	RH850/F1L 100 pins (8 ch)	RH850/F1L 144 pins (8 ch)	RH850/F1L 176 pins (8 ch)
TAUJ0	4	✓	✓	✓	✓	✓	✓
TAUJ1	4	—	—	—	✓	✓	✓

**Table 24.3 Indices**

Index	Description
n	Throughout this section, the individual TAUJ units are identified by the index "n"; for example, TAUJ <sub>n</sub> TOM is the TAUJ <sub>n</sub> channel output mode register.
m	The TAUJ has 4 channels. Throughout this section, the individual channels are identified by the index "m" (m = 0 to 3), thus a certain channel is denoted as CH <sub>m</sub> . The even numbered channels (m = 0, 2) are denoted as CH <sub>m</sub> _even. The odd numbered channels (m = 1, 3) are denoted as CH <sub>m</sub> _odd.

#### 24.1.2 Register Base Address

TAUJ<sub>n</sub> base addresses are listed in the following table.

TAUJ<sub>n</sub> register addresses are given as offsets from the base addresses.

**Table 24.4 Register Base Addresses**

Base Address Name	Base Address
<TAUJ0_base>	FFE5 0000 <sub>H</sub>
<TAUJ1_base>	FFE5 1000 <sub>H</sub>

### 24.1.3 Clock Supply

The TAUJn clock supply is shown in the following table.

**Table 24.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
TAUJ0	PCLK	CKSCLK_ATAUJ
	Register access clock	CKSCLK_ATAUJ
TAUJ1	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

### 24.1.4 Interrupt Requests

TAUJn interrupt requests are listed in the following table.

**Table 24.6 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>TAUJ0</b>			
INTTAUJ0I0	Channel 0 interrupt	72	21 (Channels 0 to 7)
INTTAUJ0I1	Channel 1 interrupt	73	16 (Channels 8 to 15)
INTTAUJ0I2	Channel 2 interrupt	74	17 (Channels 8 to 15)
INTTAUJ0I3	Channel 3 interrupt	75	22 (Channels 0 to 7)
<b>TAUJ1</b>			
INTTAUJ1I0	Channel 0 interrupt	160	46 (Channels 0 to 7)
INTTAUJ1I1	Channel 1 interrupt	161	36 (Channels 8 to 15)
INTTAUJ1I2	Channel 2 interrupt	162	47 (Channels 0 to 7)
INTTAUJ1I3	Channel 3 interrupt	163	37 (Channels 8 to 15)

### 24.1.5 Reset Sources

TAUJn reset sources are listed in the following table. TAUJn is initialized by these reset sources.

**Table 24.7 Reset Sources**

Unit Name	Reset Source
TAUJ0	All reset sources except the transition to DeepSTOP mode (AWORES)
TAUJ1	All reset sources (ISORES)

### 24.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

**Table 24.8 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal Name
<b>TAUJ0</b>		
TAUJTTIN0, TAUJTTIN1	Channel 0, 1 input	TAUJ0I0, TAUJ0I1
TAUJTTIN2	Channel 2 input	TAUJ0I2 or RTCA0OUT <sup>*1</sup>
TAUJTTIN3	Channel 3 input	TAUJ0I3 or RTCA0OUT <sup>*1</sup>
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ0O0 to TAUJ0O3
<b>TAUJ1</b>		
TAUJTTIN0 to TAUJTTIN3	Channel 0 to 3 input	TAUJ1I0 to TAUJ1I3
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ1O0 to TAUJ1O3

Note 1. For details, see **Section 24.1.8, TAUJ0 Input Selection**.

### 24.1.7 Internal Input/Output Signals

The internal input/output signals of TAUJn are listed below.

**Table 24.9 Internal Input/Output Signals**

Unit Signal Name	Description	Connected to
TAUJnTSSTM <sup>*1</sup>	Simultaneous channel start trigger input	PIC

Note 1. n = 1 only. TAUJ0TSSTM is not connected to PIC.

### 24.1.8 TAUJ0 Input Selection

The 1-Hz pulse output (RTCA0OUT) from RTCA0 can be input to TAUJTTIN2 and TAUJTTIN3 as shown in the following figure.

**NOTE**

This function is only available in the 144 pin and 176 pin devices.

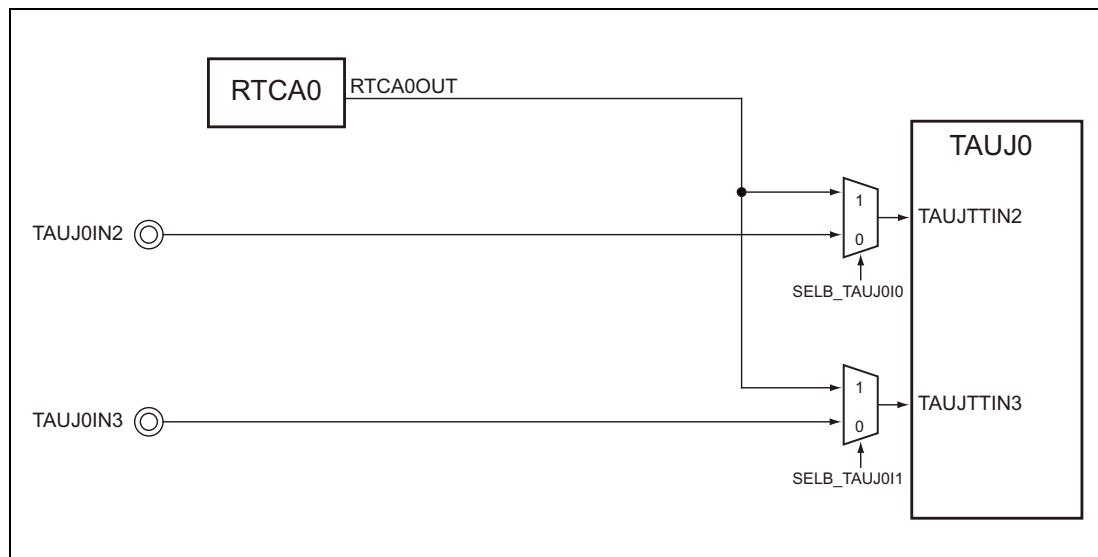


Figure 24.1 Selection of Signals Input to TAUJ0

The following table shows how to select signals input to the TAUJ.

Table 24.10 TAUJ0 Input Selections

Input Signal	Function	Settings
TAUJTTIN2	Port TAUJ0I2	SELB_TAUJ0I.SELB_TAUJ0I0 = 0
	RTCA0OUT (Real-Time Clock 1 Hz output)	SELB_TAUJ0I.SELB_TAUJ0I0 = 1
TAUJTTIN3	Port TAUJ0I3	SELB_TAUJ0I.SELB_TAUJ0I1 = 0
	RTCA0OUT (Real-Time Clock 1 Hz output)	SELB_TAUJ0I.SELB_TAUJ0I1 = 1

### 24.1.8.1 SELB\_TAUJ0I — TAUJTTINm Input Signal Selection Register

This register selects the TAUJ0 input signals.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFBC 0100<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SELB_TAUJ0I1	SELB_TAUJ0I0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 24.11 SELB\_TAUJ0I Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SELB_TAUJ0I1	Selection of TAUJTTIN3 input signal: 0: Port TAUJ013 1: RTCA0OUT
0	SELB_TAUJ0I0	Selection of TAUJTTIN2 input signal: 0: Port TAUJ012 1: RTCA0OUT

## 24.2 Overview

### 24.2.1 Functional Overview

The TAUJ has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUJ is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

#### Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels.

The synchronous operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

## 24.2.2 Terms

In this section, the following terms are used.

### **Independent channel operation function/synchronous operation channel operation function**

TAUJ has 4 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented by using a combination of channels.

- The independent channel operation function can be used any channel independently of all other channels.
- The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

### **Channel group**

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

### **Upper/lower channel**

Based on the channel number m, a channel with a smaller channel number or higher channel number can be referred to as “upper” or “lower” channel:

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel. Channel 0 is the highest channel and channel 3 is the lowest channel.

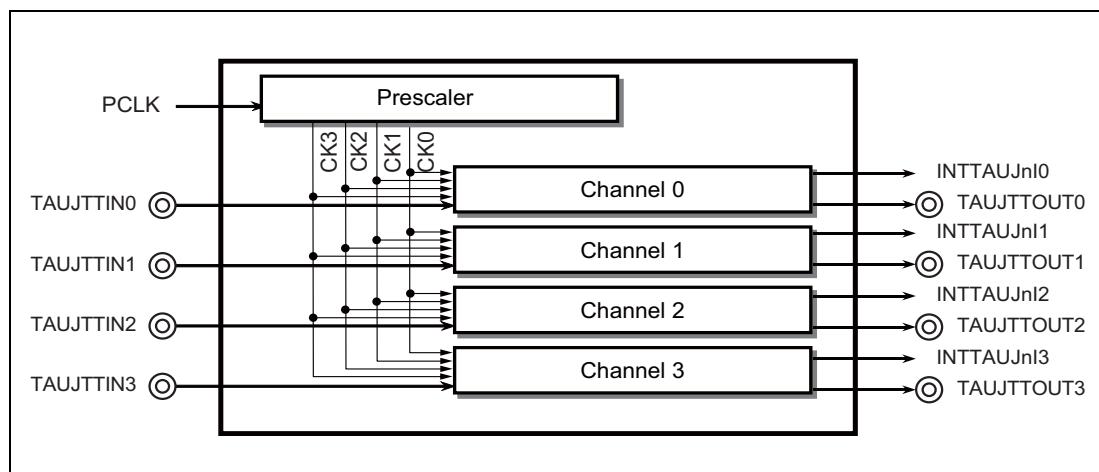
### 24.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

**Table 24.12 Functional List of TAUJ Operations**

Operation Function	Example
<b>Independent Channel Operation Functions</b>	<b>Section 24.12</b>
Interval Timer Function	<b>Section 24.12.1</b>
TAUJTTINm Input Interval Timer Function	<b>Section 24.12.2</b>
TAUJTTINm Input Pulse Interval Measurement Function	<b>Section 24.12.3</b>
TAUJTTINm Input Signal Width Measurement Function	<b>Section 24.12.4</b>
TAUJTTINm Input Position Detection Function	<b>Section 24.12.5</b>
TAUJTTINm Input Period Count Detection Function	<b>Section 24.12.6</b>
Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)	<b>Section 24.12.7</b>
Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)	<b>Section 24.12.8</b>
<b>Synchronous Channel Operation Functions</b>	<b>Section 24.13</b>
PWM Output Function	<b>Section 24.13.1</b>

### 24.2.4 TAUJ I/O and Interrupt Request Signals



**Figure 24.2 TAUJ I/O and Interrupt Request Signals**

### 24.2.5 Block Diagram

The following figure shows the main components of the TAUJ.

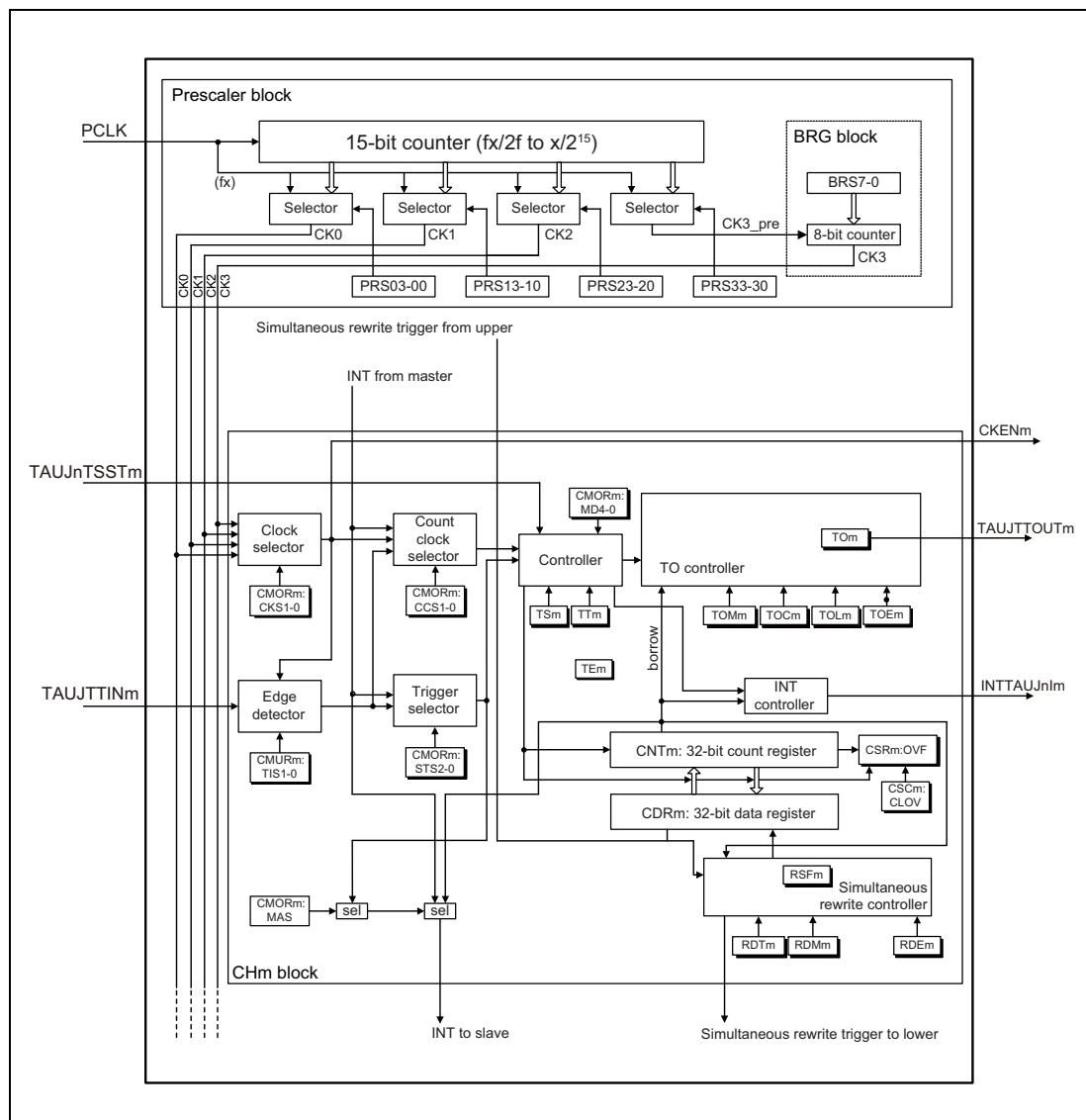


Figure 24.3 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

## 24.2.6 Description of Block Diagram

The following describes the functional blocks.

### Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived by dividing PCLK in the prescaler division factor of  $2^0$  to  $2^{15}$ . The fourth count clock, CK3, is derived by dividing PCLK by a division factor that is not a power of 2 by using the baud rate generator.

### Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of CK0 to CK3 clocks (selected by the clock selector)

### Controller

The controller controls the main operations of the counter.

- Operating mode (selected with the TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

### Trigger selector

The counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUJnTSSTm
- Valid edge of the TAUJTTINm input signal
- INTTAUJnIm from the master channel

### Simultaneous rewrite controller

Simultaneous rewrite control is enabled in synchronous operating modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

### TAUJnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals.

## 24.3 Registers

### 24.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn\_base>, see **Section 24.1.2, Register Base Address**.

Table 24.13 List of Registers

Module Name	Register Name	Symbol	Address
<b>TAUJn prescaler registers</b>			
TAUJn	TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base> + 90 <sub>H</sub>
TAUJn	TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base> + 94 <sub>H</sub>
<b>TAUJn control registers</b>			
TAUJn	TAUJn channel data register m	TAUJnCDRm	<TAUJn_base> + m × 4 <sub>H</sub>
TAUJn	TAUJn channel counter register m	TAUJnCNTm	<TAUJn_base> + 10 <sub>H</sub> + m × 4 <sub>H</sub>
TAUJn	TAUJn channel mode OS register m	TAUJnCMORM	<TAUJn_base> + 80 <sub>H</sub> + m × 4 <sub>H</sub>
TAUJn	TAUJn channel mode user register m	TAUJnCMURm	<TAUJn_base> + 20 <sub>H</sub> + m × 4 <sub>H</sub>
TAUJn	TAUJn channel status register m	TAUJnCSRm	<TAUJn_base> + 30 <sub>H</sub> + m × 4 <sub>H</sub>
TAUJn	TAUJn channel status clear trigger register m	TAUJnCSCm	<TAUJn_base> + 40 <sub>H</sub> + m × 4 <sub>H</sub>
TAUJn	TAUJn channel start trigger register	TAUJnTS	<TAUJn_base> + 54 <sub>H</sub>
TAUJn	TAUJn channel enable status register	TAUJnTE	<TAUJn_base> + 50 <sub>H</sub>
TAUJn	TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base> + 58 <sub>H</sub>
<b>TAUJn output registers</b>			
TAUJn	TAUJn channel output enable register	TAUJnTOE	<TAUJn_base> + 60 <sub>H</sub>
TAUJn	TAUJn channel output register	TAUJnTO	<TAUJn_base> + 5C <sub>H</sub>
TAUJn	TAUJn channel output mode register	TAUJnTOM	<TAUJn_base> + 98 <sub>H</sub>
TAUJn	TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base> + 9C <sub>H</sub>
TAUJn	TAUJn channel output active level register	TAUJnTOL	<TAUJn_base> + 64 <sub>H</sub>
<b>TAUJn reload data registers</b>			
TAUJn	TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base> + A0 <sub>H</sub>
TAUJn	TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base> + A4 <sub>H</sub>
TAUJn	TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base> + 68 <sub>H</sub>
TAUJn	TAUJn channel reload status register	TAUJnRSF	<TAUJn_base> + 6C <sub>H</sub>
<b>TAUJn emulation register</b>			
TAUJn	TAUJn emulation register	TAUJnEMU	<TAUJn_base> + A8 <sub>H</sub>

## 24.3.2 Details of TAUJn Prescaler Registers

### 24.3.2.1 TAUJnTPS — TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3\_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3\_PRE by the factor specified in TAUJnBRS.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAUJn\_base> + 90H

**Value after reset:** FFFFH

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnPRS3[3:0]				TAUJnPRS2[3:0]				TAUJnPRS1[3:0]				TAUJnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.14 TAUJnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUJnPRS3 [3:0]	Specifies a CK3_PRE clock. The CK3_PRE clock is an input clock of the BRG unit which supplies CK3 operation clocks to all channels.
<b>TAUJnPRS3[3:0]</b>		<b>CK3_PRE clock</b>
0000 <sub>B</sub>		PCLK/2 <sup>0</sup>
0001 <sub>B</sub>		PCLK/2 <sup>1</sup>
0010 <sub>B</sub>		PCLK/2 <sup>2</sup>
0011 <sub>B</sub>		PCLK/2 <sup>3</sup>
0100 <sub>B</sub>		PCLK/2 <sup>4</sup>
0101 <sub>B</sub>		PCLK/2 <sup>5</sup>
0110 <sub>B</sub>		PCLK/2 <sup>6</sup>
0111 <sub>B</sub>		PCLK/2 <sup>7</sup>
1000 <sub>B</sub>		PCLK/2 <sup>8</sup>
1001 <sub>B</sub>		PCLK/2 <sup>9</sup>
1010 <sub>B</sub>		PCLK/2 <sup>10</sup>
1011 <sub>B</sub>		PCLK/2 <sup>11</sup>
1100 <sub>B</sub>		PCLK/2 <sup>12</sup>
1101 <sub>B</sub>		PCLK/2 <sup>13</sup>
1110 <sub>B</sub>		PCLK/2 <sup>14</sup>
1111 <sub>B</sub>		PCLK/2 <sup>15</sup>

The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 24.14 TAUJnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUJnPRS2 [3:0]	Specifies a CK2 clock.
		<b>TAUJnPRS2[3:0]</b> <b>CK2 clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>
The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).		
7 to 4	TAUJnPRS1 [3:0]	Specifies a CK1 clock.
		<b>TAUJnPRS1[3:0]</b> <b>CK1 clock</b>
		0000 <sub>B</sub> PCLK/2 <sup>0</sup>
		0001 <sub>B</sub> PCLK/2 <sup>1</sup>
		0010 <sub>B</sub> PCLK/2 <sup>2</sup>
		0011 <sub>B</sub> PCLK/2 <sup>3</sup>
		0100 <sub>B</sub> PCLK/2 <sup>4</sup>
		0101 <sub>B</sub> PCLK/2 <sup>5</sup>
		0110 <sub>B</sub> PCLK/2 <sup>6</sup>
		0111 <sub>B</sub> PCLK/2 <sup>7</sup>
		1000 <sub>B</sub> PCLK/2 <sup>8</sup>
		1001 <sub>B</sub> PCLK/2 <sup>9</sup>
		1010 <sub>B</sub> PCLK/2 <sup>10</sup>
		1011 <sub>B</sub> PCLK/2 <sup>11</sup>
		1100 <sub>B</sub> PCLK/2 <sup>12</sup>
		1101 <sub>B</sub> PCLK/2 <sup>13</sup>
		1110 <sub>B</sub> PCLK/2 <sup>14</sup>
		1111 <sub>B</sub> PCLK/2 <sup>15</sup>
The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).		

Table 24.14 TAUJnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUJnPRS0 [3:0]	Specifies a CK0 clock.
		<b>TAUJnPRS0[3:0]</b>
		<b>CK0 clock</b>
	0000 <sub>B</sub>	PCLK/2 <sup>0</sup>
	0001 <sub>B</sub>	PCLK/2 <sup>1</sup>
	0010 <sub>B</sub>	PCLK/2 <sup>2</sup>
	0011 <sub>B</sub>	PCLK/2 <sup>3</sup>
	0100 <sub>B</sub>	PCLK/2 <sup>4</sup>
	0101 <sub>B</sub>	PCLK/2 <sup>5</sup>
	0110 <sub>B</sub>	PCLK/2 <sup>6</sup>
	0111 <sub>B</sub>	PCLK/2 <sup>7</sup>
	1000 <sub>B</sub>	PCLK/2 <sup>8</sup>
	1001 <sub>B</sub>	PCLK/2 <sup>9</sup>
	1010 <sub>B</sub>	PCLK/2 <sup>10</sup>
	1011 <sub>B</sub>	PCLK/2 <sup>11</sup>
	1100 <sub>B</sub>	PCLK/2 <sup>12</sup>
	1101 <sub>B</sub>	PCLK/2 <sup>13</sup>
	1110 <sub>B</sub>	PCLK/2 <sup>14</sup>
	1111 <sub>B</sub>	PCLK/2 <sup>15</sup>

The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).

#### NOTE

TAUJn clock input PCLK is defined in the first part of this section, **Section 24.1.3, Clock Supply**.

### 24.3.2.2 TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3\_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3\_PRE is specified in TAUJnTPS. TAUJnPRS3[3:0].

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUJn\_base> + 94H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
TAUJnBRS[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.15 TAUJnBRS Register Contents

Bit Position	Bit Name	Function
7 to 0	TAUJnBRS [7:0]	Specifies a CK3_PRE clock division factor for generating CK3.
	TAUJnBRS[7:0]	CK3 clock
	0000 0000 <sub>B</sub>	CK3_PRE / 1
	0000 0001 <sub>B</sub>	CK3_PRE / 2
	0000 0010 <sub>B</sub>	CK3_PRE / 3
	0000 0011 <sub>B</sub>	CK3_PRE / 4
	:	:
	1111 1110 <sub>B</sub>	CK3_PRE / 255
	1111 1111 <sub>B</sub>	CK3_PRE / 256

### 24.3.3 Details of TAUJn Control Registers

#### 24.3.3.1 TAUJnCDRm — TAUJn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

**Access:** This register can be read or written in 32-bit units..  
 • When this register functions as a capture register, only reading is possible. Write operation is ignored.  
 • When this register functions as a compare register, reading and writing is possible.

**Address:** <TAUJn\_base> + 0<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAUJnCDR[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TAUJnCDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.16 TAUJnCDRm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR [31:0]	Data register for capture/compare values

### 24.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register

This is a channel m counter register.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <TAUJn\_base> + 10<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** FFFF FFFF<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAUJnCNT[31:16]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
TAUJnCNT[15:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 24.17 TAUJnCNTm Register Contents**

Bit Position	Bit Name	Function
31 to 0	TAUJnCNT [31:0]	32-bit counter value

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEM = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 24.18 TAUJnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUJnCNTm		
		Start Value <sup>*1</sup>	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF FFFF <sub>H</sub>	Stop value	—
Capture mode	Count up	0000 0000 <sub>H</sub>	Stop value	—
One-count mode	Count down	FFFF FFFF <sub>H</sub>	Stop value	Stop value
Capture and one-count mode	Count up	0000 0000 <sub>H</sub>	Stop value	Capture value + 1 (TAUJnCDRm)
Count capture mode	Count up	0000 0000 <sub>H</sub>	Stop value	—
Gate count mode	Count down	FFFF FFFF <sub>H</sub>	Stop value	Stop value
Capture and gate count mode	Count up	0000 0000 <sub>H</sub>	Stop value	Stop value

Note 1. The value set for TAUJnCNTm when operating mode is changed after a reset is deasserted

### 24.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register

This register controls channel m operation.

**Access:** This register can be read or written in 16-bit units.  
Writable only when the counter is stopped (TAUJnTE.TAUJnTEM = 0).

**Address:** <TAUJn\_base> + 80H + m × 4H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]		TAUJnCOS[1:0]	—	TAUJnMD[4:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 24.19 TAUJnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Selects an operation clock, which is used with the TAUJTTINm input edge detection circuit. Setting of TAUJnCMORm.TAUJnCCS[1:0] bits also allows the operation clock to serve as the TAUJnCNTm count clock.
<b>TAUJn CKS1    TAUJn CKS0    Selection of Operation Clock</b>		
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3
13, 12	TAUJnCCS[1:0]	Selects a count clock for TAUJnCNTm counter.
<b>TAUJn CCS1    TAUJn CCS0    Selection of Operation Clock</b>		
0	0	Operation clock specified by TAUJnCMORm.TAUJnCKS[1:0].
0	1	Setting prohibited
1	0	
1	1	
11	TAUJnMAS	Specifies whether the channel is a master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm-odd) are fixed to 0.

Table 24.19 TAUJnCMORm Register Contents (2/3)

Bit Position	Bit Name	Function																																				
10 to 8	TAUJnSTS[2:0]	Selects an external start trigger.																																				
		<table border="1"> <thead> <tr> <th>TAUJn STS2</th> <th>TAUJn STS1</th> <th>TAUJn STS0</th> <th>Selection of Operation Clock</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>Valid edge of TAUJTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>Valid edge of TAUJTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>INT of master channel</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>Setting prohibited</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td></td></tr> <tr> <td>1</td><td>1</td><td>1</td><td></td></tr> </tbody> </table>	TAUJn STS2	TAUJn STS1	TAUJn STS0	Selection of Operation Clock	0	0	0	Software trigger	0	0	1	Valid edge of TAUJTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].	0	1	0	Valid edge of TAUJTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.	0	1	1	Setting prohibited	1	0	0	INT of master channel	1	0	1	Setting prohibited	1	1	0		1	1	1	
TAUJn STS2	TAUJn STS1	TAUJn STS0	Selection of Operation Clock																																			
0	0	0	Software trigger																																			
0	0	1	Valid edge of TAUJTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].																																			
0	1	0	Valid edge of TAUJTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.																																			
0	1	1	Setting prohibited																																			
1	0	0	INT of master channel																																			
1	0	1	Setting prohibited																																			
1	1	0																																				
1	1	1																																				
7, 6	TAUJnCOS[1:0]	<p>Specifies the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm.TAUJnOVF of channel m.</p> <p>These bits are only valid if channel m is for capture function (capture mode and capture &amp; one-count mode).</p>																																				
		<table border="1"> <thead> <tr> <th>TAUJn COS1</th> <th>TAUJn COS0</th> <th>TAUJnCDRm</th> <th>TAUJnCSRm.TAUJnOVF</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Updated when valid edge of TAUJTTINm input is detected.</td><td>           Updated (cleared or set) when valid edge of TAUJTTINm input is detected.           <ul style="list-style-type: none"> <li>Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.</li> <li>Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</li> </ul> </td></tr> <tr> <td>0</td><td>1</td><td></td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr> <tr> <td>1</td><td>0</td><td>Updated when valid edge of TAUJTTINm input is detected and when a counter overflow occurs.</td><td>No setting</td></tr> <tr> <td>1</td><td>1</td><td> <ul style="list-style-type: none"> <li>Detection of valid edge of TAUJTTINm input: The counter value is written into TAUJnCDRm.</li> <li>Occurrence of overflow: <math>FFFF\ FFFF_H</math> is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJTTINm is ignored.</li> </ul> </td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr> <tr> <td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr> </tbody> </table>	TAUJn COS1	TAUJn COS0	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	0	0	Updated when valid edge of TAUJTTINm input is detected.	Updated (cleared or set) when valid edge of TAUJTTINm input is detected. <ul style="list-style-type: none"> <li>Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.</li> <li>Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</li> </ul>	0	1		Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.	1	0	Updated when valid edge of TAUJTTINm input is detected and when a counter overflow occurs.	No setting	1	1	<ul style="list-style-type: none"> <li>Detection of valid edge of TAUJTTINm input: The counter value is written into TAUJnCDRm.</li> <li>Occurrence of overflow: <math>FFFF\ FFFF_H</math> is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJTTINm is ignored.</li> </ul>	Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.	5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.													
TAUJn COS1	TAUJn COS0	TAUJnCDRm	TAUJnCSRm.TAUJnOVF																																			
0	0	Updated when valid edge of TAUJTTINm input is detected.	Updated (cleared or set) when valid edge of TAUJTTINm input is detected. <ul style="list-style-type: none"> <li>Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.</li> <li>Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</li> </ul>																																			
0	1		Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.																																			
1	0	Updated when valid edge of TAUJTTINm input is detected and when a counter overflow occurs.	No setting																																			
1	1	<ul style="list-style-type: none"> <li>Detection of valid edge of TAUJTTINm input: The counter value is written into TAUJnCDRm.</li> <li>Occurrence of overflow: <math>FFFF\ FFFF_H</math> is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJTTINm is ignored.</li> </ul>	Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.																																			
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																																				

Table 24.19 TAUJnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function																																																																																										
4 to 0	TAUJnMD[4:0]	Specifies an operating mode.																																																																																										
		<table border="1"> <thead> <tr> <th>TAUJn MD4</th><th>TAUJn MD3</th><th>TAUJn MD2</th><th>TAUJn MD1</th><th>TAUJn MD0</th><th>Functional Description</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Setting prohibited</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Setting prohibited</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr> </tbody> </table>	TAUJn MD4	TAUJn MD3	TAUJn MD2	TAUJn MD1	TAUJn MD0	Functional Description	0	0	0	0	1/0	Interval timer mode	0	0	0	1	1/0	Setting prohibited	0	0	1	0	1/0	Capture mode	0	0	1	1	0	Setting prohibited	0	1	0	0	1/0	One-count mode	0	1	0	1	1/0	Setting prohibited	0	1	1	0	0	Capture and one-count mode	0	1	1	1	1/0	Setting prohibited	1	0	0	0	0	Setting prohibited	1	0	0	1	0	Setting prohibited	1	0	1	0	1/0	Setting prohibited	1	0	1	1	1/0	Count capture mode	1	1	0	0	0	Gate count mode	1	1	0	1	0	Capture and gate count mode
TAUJn MD4	TAUJn MD3	TAUJn MD2	TAUJn MD1	TAUJn MD0	Functional Description																																																																																							
0	0	0	0	1/0	Interval timer mode																																																																																							
0	0	0	1	1/0	Setting prohibited																																																																																							
0	0	1	0	1/0	Capture mode																																																																																							
0	0	1	1	0	Setting prohibited																																																																																							
0	1	0	0	1/0	One-count mode																																																																																							
0	1	0	1	1/0	Setting prohibited																																																																																							
0	1	1	0	0	Capture and one-count mode																																																																																							
0	1	1	1	1/0	Setting prohibited																																																																																							
1	0	0	0	0	Setting prohibited																																																																																							
1	0	0	1	0	Setting prohibited																																																																																							
1	0	1	0	1/0	Setting prohibited																																																																																							
1	0	1	1	1/0	Count capture mode																																																																																							
1	1	0	0	0	Gate count mode																																																																																							
1	1	0	1	0	Capture and gate count mode																																																																																							
		<table border="1"> <thead> <tr> <th>Mode</th><th>Role of TAUJnMD0 Bit</th></tr> </thead> <tbody> <tr> <td>Interval timer mode Capture mode Count capture mode</td><td>Specifies whether INTTAUJnlm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnlm is not generated. 1: INTTAUJnlm is generated.</td></tr> <tr> <td>One-count mode</td><td>Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. <b>CAUTION</b> In one-count mode, INTTAUJnlm signal is not output at the beginning of count operation.</td></tr> <tr> <td>Capture and one-count mode Gate count mode Capture and gate count mode</td><td>This bit should be set to 0. <b>CAUTION</b> INTTAUJnlm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.</td></tr> </tbody> </table>	Mode	Role of TAUJnMD0 Bit	Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUJnlm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnlm is not generated. 1: INTTAUJnlm is generated.	One-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. <b>CAUTION</b> In one-count mode, INTTAUJnlm signal is not output at the beginning of count operation.	Capture and one-count mode Gate count mode Capture and gate count mode	This bit should be set to 0. <b>CAUTION</b> INTTAUJnlm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.																																																																																		
Mode	Role of TAUJnMD0 Bit																																																																																											
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUJnlm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnlm is not generated. 1: INTTAUJnlm is generated.																																																																																											
One-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. <b>CAUTION</b> In one-count mode, INTTAUJnlm signal is not output at the beginning of count operation.																																																																																											
Capture and one-count mode Gate count mode Capture and gate count mode	This bit should be set to 0. <b>CAUTION</b> INTTAUJnlm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.																																																																																											

#### 24.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUJTTINm input.

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUJn\_base> + 20<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

**Table 24.20 TAUJnCMURm Register Contents**

Bit Position	Bit Name	Function															
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
1, 0	TAUJnTIS[1:0]	<p>Specifies a valid edge of TAUJTTINm input signal.</p> <table border="1"> <thead> <tr> <th>TAUJn TIS1</th> <th>TAUJn TIS0</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Falling edge</td> </tr> <tr> <td>0</td> <td>1</td> <td>Rising edge</td> </tr> <tr> <td>1</td> <td>0</td> <td>           Detection of falling and rising edges (selection of low width measurement)            Start trigger: Falling edge            Stop trigger (capture): Rising edge         </td> </tr> <tr> <td>1</td> <td>1</td> <td>           Detection of falling and rising edges (selection of high width measurement)            Start trigger: Rising edge            Stop trigger (capture): Falling edge         </td> </tr> </tbody> </table>	TAUJn TIS1	TAUJn TIS0	Functional Description	0	0	Falling edge	0	1	Rising edge	1	0	Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge	1	1	Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge
TAUJn TIS1	TAUJn TIS0	Functional Description															
0	0	Falling edge															
0	1	Rising edge															
1	0	Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge															
1	1	Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge															

Edge detection of TAUJTTINm input signal is based on the operation clock selected by TAUJnCMORm.TAUJnCKS[1:0].

### 24.3.3.5 TAUJnCSRm — TAUJn Channel Status Register

This register indicates the overflow status of channel m.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <TAUJn\_base> + 30<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 0x<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnOVF
Value after reset	0	0	0	0	0	0	—	0
R/W	R	R	R	R	R	R	R	R

Table 24.21 TAUJnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	Reserved	When read, unknown value is returned.
0	TAUJnOVF	<p>Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs</p> <p>This bit is used only in the following modes:</p> <ul style="list-style-type: none"> <li>• Capture mode</li> <li>• Capture and one-count mode</li> </ul> <p>The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].</p>

### 24.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <TAUJn\_base> + 40<sub>H</sub> + m × 4<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 24.22 TAUJnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

### 24.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <TAUJn\_base> + 54H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTS03	TAUJnTS02	TAUJnTS01	TAUJnTS00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 24.23 TAUJnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTSm	Enables the counter operation for channel m: 0: No function 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1.  Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1. Whether the counter is started or not depends on the selected operating mode.

### 24.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register indicates whether a counter operation is enabled.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <TAUJn\_base> + 50H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTE03	TAUJnTE02	TAUJnTE01	TAUJnTE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 24.24 TAUJnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnTEm	Indicates whether channel m's counter operation is enabled. 0: Counter operation is disabled 1: Counter operation is enabled  This bit is set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1. This bit is reset to 0 when TAUJnTT.TAUJnTTm is set to 1.

### 24.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <TAUJn\_base> + 58H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTT03	TAUJnTT02	TAUJnTT01	TAUJnTT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

**Table 24.25 TAUJnTT Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTTm	Stops channel m's counter operation. 0: No function 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm.  TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJTTOUTm retain the values provided before the counter is stopped.

### 24.3.4 Details of TAUJn Simultaneous Rewrite Register

#### 24.3.4.1 TAUJnRDE — TAUJn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm. It also enables and disables simultaneous rewrite of the data register TAUJnTOLm for the PWM output function.

**Access:** This register can be read or written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

**Address:** <TAUJn\_base> + A0H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDE03	TAUJnRDE02	TAUJnRDE01	TAUJnRDE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.26 TAUJnRDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

#### 24.3.4.2 TAUJnRDM — TAUJn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

**Access:** This register can be read or written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

**Address:** <TAUJn\_base> + A4H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDM03	TAUJnRDM02	TAUJnRDM01	TAUJnRDM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.27 TAUJnRDM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDMm	Specifies when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: No function
These bits only apply when TAUJnRDE.TAUJnRDEm = 1.		

### 24.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <TAUJn\_base> + 68H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDT03	TAUJnRDT02	TAUJnRDT01	TAUJnRDT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 24.28 TAUJnRDT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnRDTm	<p>Triggers the simultaneous rewrite enabling state.</p> <p>0: No function</p> <p>1: Simultaneous rewrite enabling state is triggered. The simultaneous rewrite enabling flag (TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger.</p> <p>These bits only apply when:</p> <ul style="list-style-type: none"> <li>• TAUJnRDE.TAUJnRDEm = 1</li> </ul>

### 24.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <TAUJn\_base> + 6CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRSF03	TAUJnRSF02	TAUJnRSF01	TAUJnRSF00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 24.29 TAUJnRSF Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnRSFm	<p>Indicates the simultaneous rewrite status.</p> <p>0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger.</p> <p>1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUJnRDFm = 1).</p>

### 24.3.5 Details of TAUJn Output Registers

#### 24.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUJn\_base> + 60H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOE03	TAUJnTOE02	TAUJnTOE01	TAUJnTOE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.30 TAUJnTOE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOEm	Enables/disables independent channel output function: 0: Disables independent timer output function (controlled by software) 1: Enables independent timer output function

#### 24.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJTTOUTm.

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUJn\_base> + 5CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTO03	TAUJnTO02	TAUJnTO01	TAUJnTO00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.31 TAUJnTO Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOM	Specifies and reads the level of TAUJTTOUTm: 0: Low 1: High
Only TAUJnTOM bits for which Independent Channel Output function is disabled (TAUJnTOEm = 0) can be written.		

### 24.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

**Access:** This register can be read or written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

**Address:** <TAUJn\_base> + 98H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOM03	TAUJnTOM02	TAUJnTOM01	TAUJnTOM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.32 TAUJnTOM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode  The output mode depends on the settings of channel output control (TAUJnTOE.TAUJnTOEm) bits.

### 24.3.5.4 TAUJnTOC — TAUJn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUJnTOMm.

**Access:** This register can be read or written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

**Address:** <TAUJn\_base> + 9CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

**Table 24.33 TAUJnTOC Register Contents**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>															
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
3 to 0	TAUJnTOCm	<p>Specifies the output mode:</p> <p>0: Operation mode 1 (= Toggle mode)      1: No function</p> <p>This bit must be set to 0 for all output modes except independent channel output mode controlled by software.      The output mode also depends on TAUJnTOM.TAUJnTOMm, as shown in the following table.</p> <table border="1"> <thead> <tr> <th>TAUJn TOMm</th> <th>TAUJn TOCm</th> <th>Functional Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Toggle mode: Toggling proceeds when INTTAUJnIm occurs.</td> </tr> <tr> <td>0</td> <td>1</td> <td>No function</td> </tr> <tr> <td>1</td> <td>0</td> <td>Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.</td> </tr> <tr> <td>1</td> <td>1</td> <td>No function</td> </tr> </tbody> </table>	TAUJn TOMm	TAUJn TOCm	Functional Description	0	0	Toggle mode: Toggling proceeds when INTTAUJnIm occurs.	0	1	No function	1	0	Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.	1	1	No function
TAUJn TOMm	TAUJn TOCm	Functional Description															
0	0	Toggle mode: Toggling proceeds when INTTAUJnIm occurs.															
0	1	No function															
1	0	Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.															
1	1	No function															

### 24.3.5.5 TAUJnTOL — TAUJn Channel Output Level Register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOM).

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAUJn\_base> + 64H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOL03	TAUJnTOL02	TAUJnTOL01	TAUJnTOL00
Value after reset	0	0	0	0	0	0	0	0

R/W	R	R	R	R	R/W	R/W	R/W	R/W
-----	---	---	---	---	-----	-----	-----	-----

**Table 24.34 TAUJnTOL Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOLm	Specifies the output logic of the channel m output bit (TAUJnTO.TAUJnTOM): 0: Positive logic (active high) 1: Negative logic (active low)
These bits apply in all channel output modes except independent channel output mode controlled by software.		

### 24.3.5.6 TAUJnEMU — TAUJn Emulation Register

This register controls operation by SVSTOP.

**Access:** This register can be read or written in 8-bit units.

A write should be performed when counters are stopped (TAUJnTE.TAUJnTEM = 0) and (EPC.SVSTOP = 0).

**Address:** <TAUJn\_base> + A8H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	TAUJn SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

R/W	R/W	R	R	R	R	R	R	R
-----	-----	---	---	---	---	---	---	---

**Table 24.35 TAUJnEMU Register Contents**

Bit Position	Bit Name	Function
7	TAUJnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 24.4 Operating Procedure

The following lists the general operation procedure for the TAUJn:

After a reset is deasserted, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUJTTOUTm is also initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.

2. Configure the desired TAUJn function:
  - Set the operation mode
  - Set any other control bits

3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.

The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.

4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUJnTT.TAUJnTTm bit to 1. The counter can be forcibly restarted by setting the TAUJnTS.TAUJnTSm bit to 1.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

### NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 24.12, Independent Channel Operation Functions** and **Section 24.13, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped ( $TAUJnTE.TAUJnTEm = 0$ ).

## 24.5 Concepts of Synchronous Channel Operation Function

The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in **Section 24.5.1, Rules of Synchronous Channel Operation Function**.

The synchronous channel operation function are detailed in the following section.

- **Section 24.13, Synchronous Channel Operation Functions**

### 24.5.1 Rules of Synchronous Channel Operation Function

#### Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel other than CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.  
Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.  
Example: If CH0 and CH2 are master channels, CH1 can be set as slave channel for CH0, but CH3 cannot.

#### Operation clock

- The same operation clock should be set for the master channel and the synchronized slave channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 24.4**.

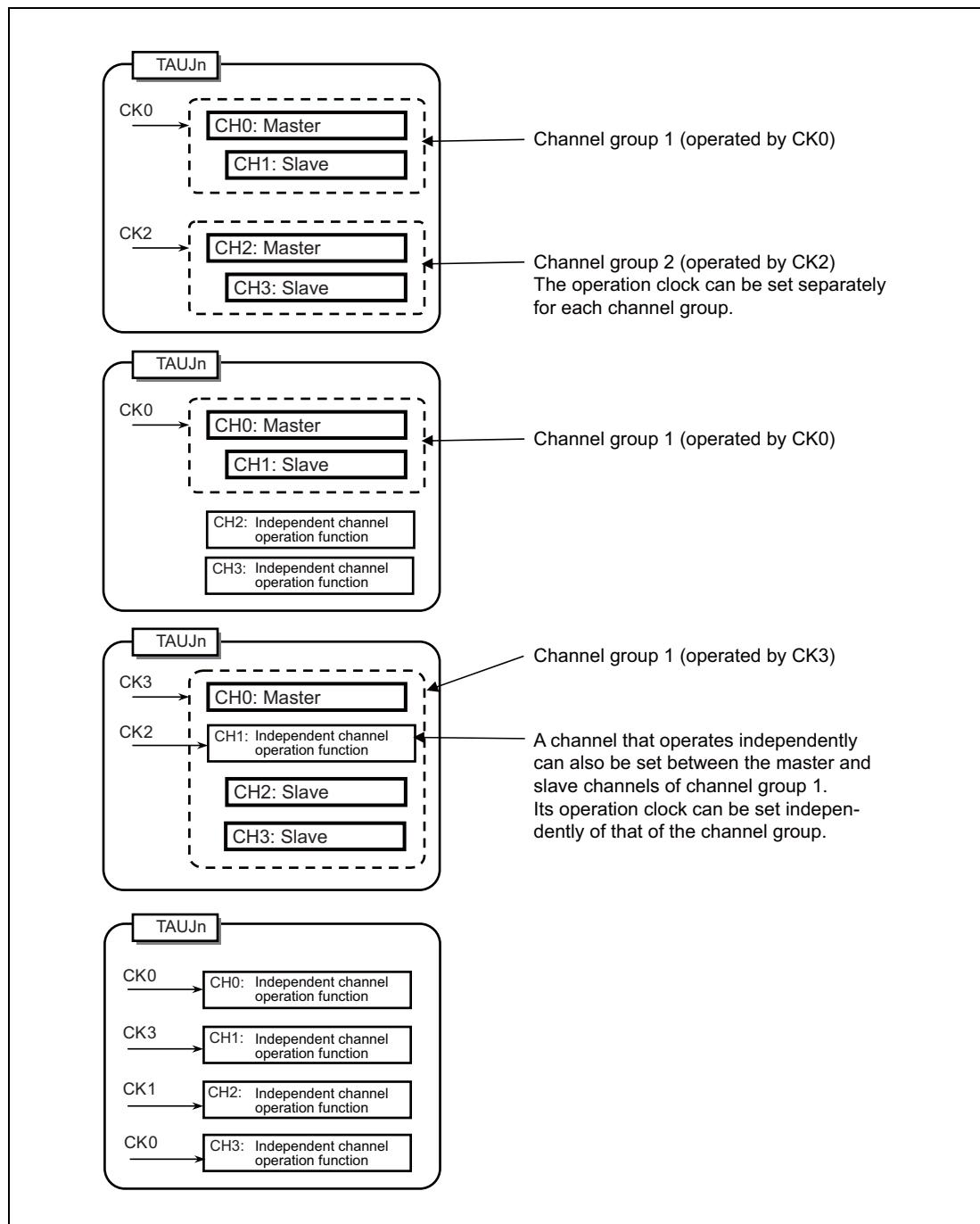


Figure 24.4 Grouping of Channels and Assignment of Operation Clocks

## 24.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

### 24.5.2.1 Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSm bits sets the corresponding TAUJnTE.TAUJnTEM bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

### 24.5.2.2 Simultaneous Start between TAUJ Units

Counters in different TAUJ units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

## 24.6 Simultaneous Rewrite

### 24.6.1 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

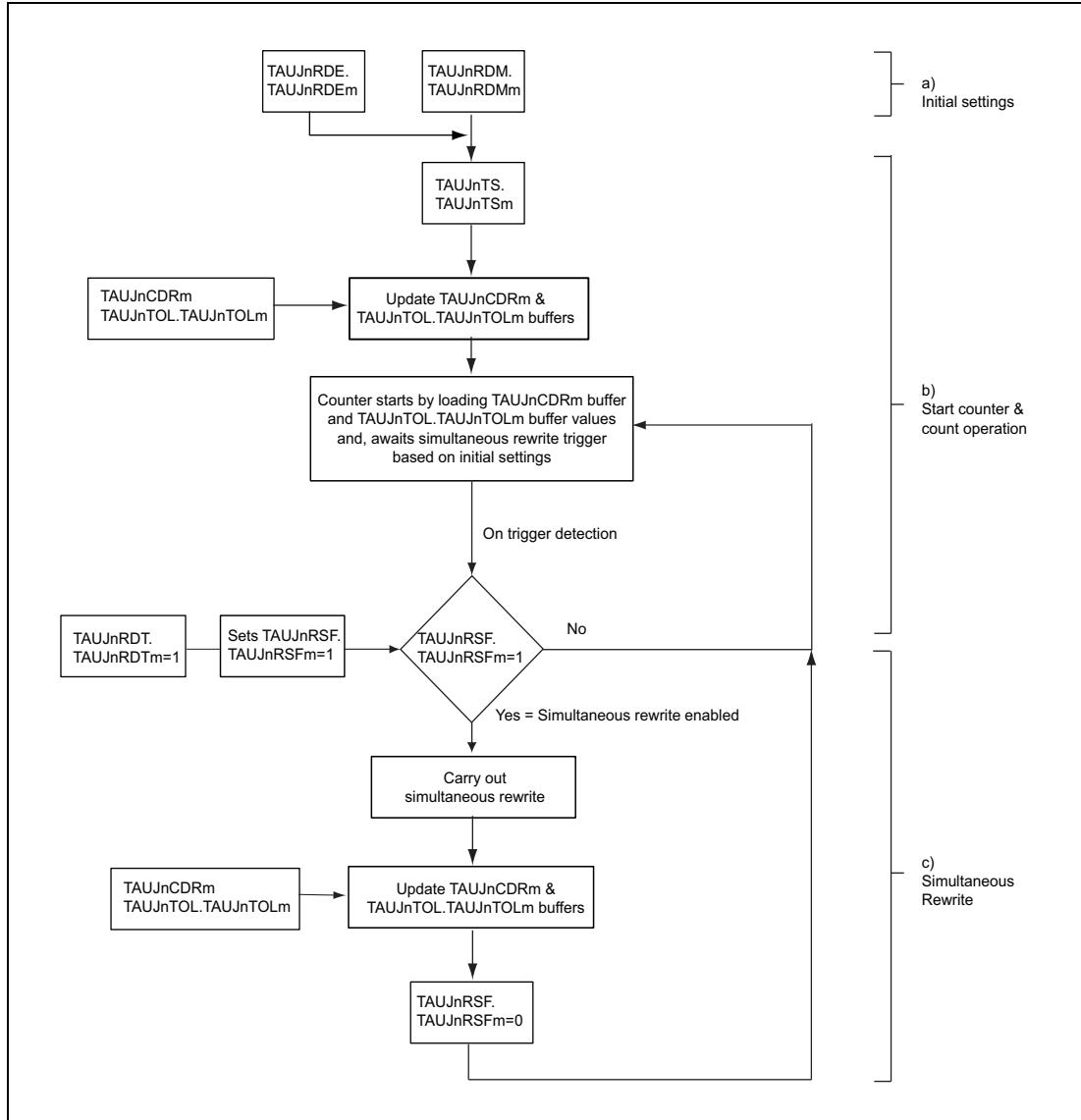


Figure 24.5 General Procedure for Simultaneous Rewrite

#### 24.6.1.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUJnRDE.TAUJnRDEm = 1.
- To select simultaneous rewrite when the master channel starts counting, set TAUJnRDM.TAUJnRDMm.

#### 24.6.1.2 Start Counter and Count Operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. The values of TAUJnTOL.TAUJnTOLm and the data registers (TAUJnCDRm) are loaded into the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the TAUJnRSFTAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

#### 24.6.1.3 Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is complete, the TAUJnRSF.TAUJnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

#### 24.6.2 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEM = 1).
- TAUJnTOL.TAUJnTOLm can be rewritten only during operation using the PWM output function. For all other functions, TAUJnTOL.TAUJnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUJTTOUTm outputs an invalid waveform.

### 24.6.3 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure with PWM output function is described in the following figure.

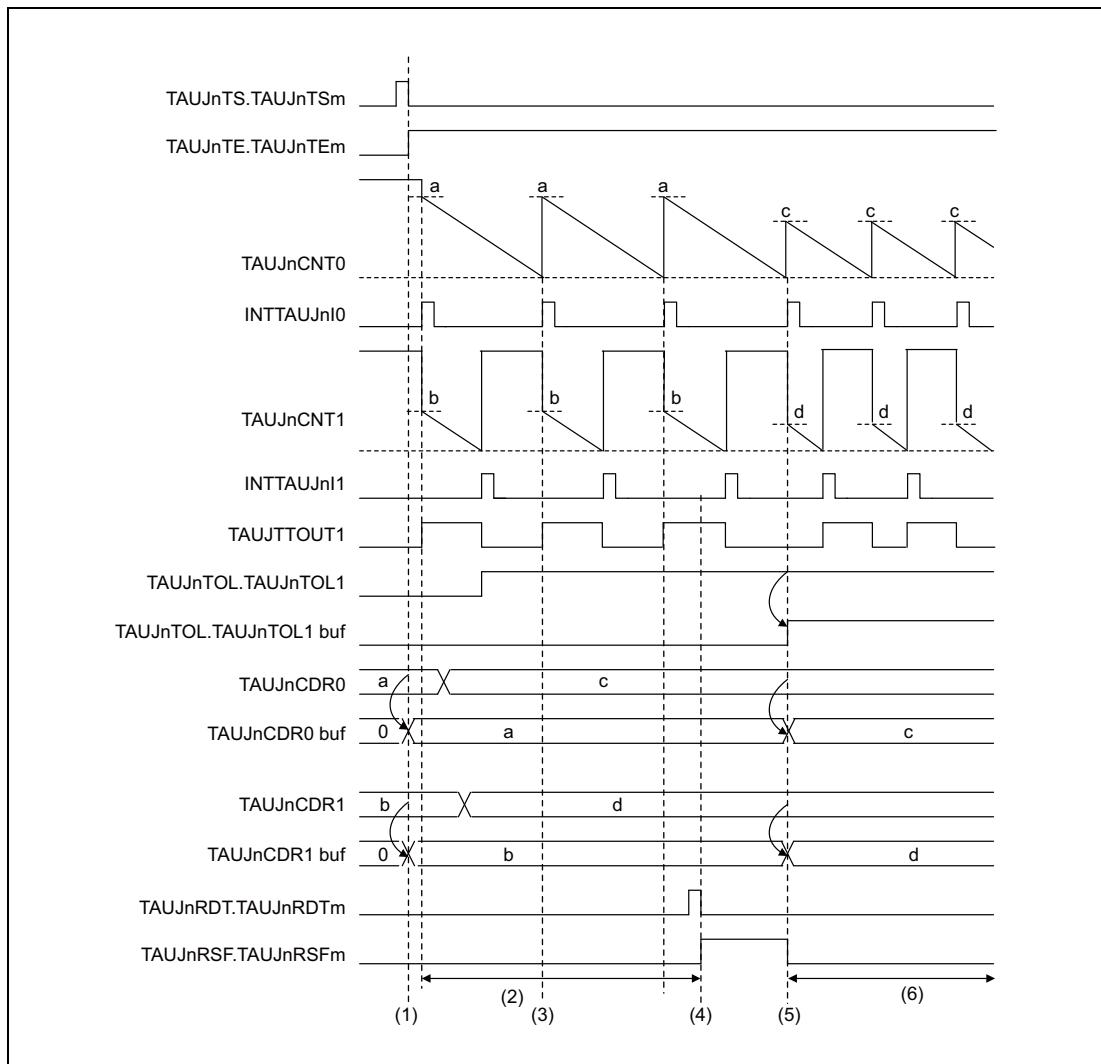


Figure 24.6    Simultaneous Rewrite with PWM Output Function

#### Setting:

CH0 is a master channel of PWM output function, and CH1 is a slave channel of PWM output function. Simultaneous rewrite is applied when the master channel starts counting.

**Description:**

- (1) When TAUJnTS.TAUJnTSm = 1 is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.
- (2) The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.
- (3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).
- (4) The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.
- (5) Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the TAUJnTOL.TAUJnTOLm value is loaded into the TAUJnTOL.TAUJnTOLm buffer.
- (6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

## 24.7 Channel Output Modes

The output of the TAUJTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUJnTOE.TAUJnTOEm = 0)

When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOM) is sent to the output pin (TAUJTTOUTm).

- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)

When controlled by TAUJ signals, the output level of TAUJTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOM is updated accordingly to reflect the value of TAUJTTOUTm.

- Independently (TAUJnTOM.TAUJnTOMm = 0)

In case of independent operation, the output of the TAUJTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).

- Synchronously (TAUJnTOM.TAUJnTOMm = 1)

In case of synchronous operation, the output of the TAUJTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOM bit can always be read to determine the current value of TAUJTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

### Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 24.36, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 24.7.2, Channel Output Modes Controlled Independently by TAUJn Signals**
- **Section 24.7.3, Channel Output Modes Controlled Synchronously by TAUJn Signals**

### Batch operation of TAUJnTOM bit

Whether a set value is reflected to the TAUJnTOM bit or not is controlled by the TAUJnTOE.TAUJnTOEm bit.

The TAUJnTOM setting is written only to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 0 when a write to the TAUJnTO register is attempted. No TAUJnTOM setting is reflected to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 1.

#### NOTE

---

TAUJnTO.TAUJnTOM bit is placed so that its bit number corresponds to a channel number.

---

### Output logic

Positive logic or negative logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function. Otherwise, changes to TAUJnTOL.TAUJnTOLm result in an undefined TAUJTTOUTm signal output.

See **Section 24.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 24.36**.

**Table 24.36 Channel Output Modes**

Channel Output Mode	TAUJnTOE.TAUJnTOEm	TAUJnTOM.TAUJnTOMm
<b>By software</b>		
Independent channel output mode controlled by software	0	x
<b>By TAUJ signals, independently</b>		
Independent channel output mode 1	1	0
<b>By TAUJ signals, synchronously</b>		
Synchronous channel output mode 1	1	1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

### NOTE

The following bits cannot be changed during count operation (TAUJnTE.TAUJnTEm = 1):

- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

### 24.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

- (1) Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJTTOUTm output.
- (2) Set channel output mode according to **Table 24.36, Channel Output Modes**, and the output logic using the TAUJnTOL.TAUJnTOLm bit.
- (3) Start the counter (TAUJnTS.TAUJnTSm = 1).

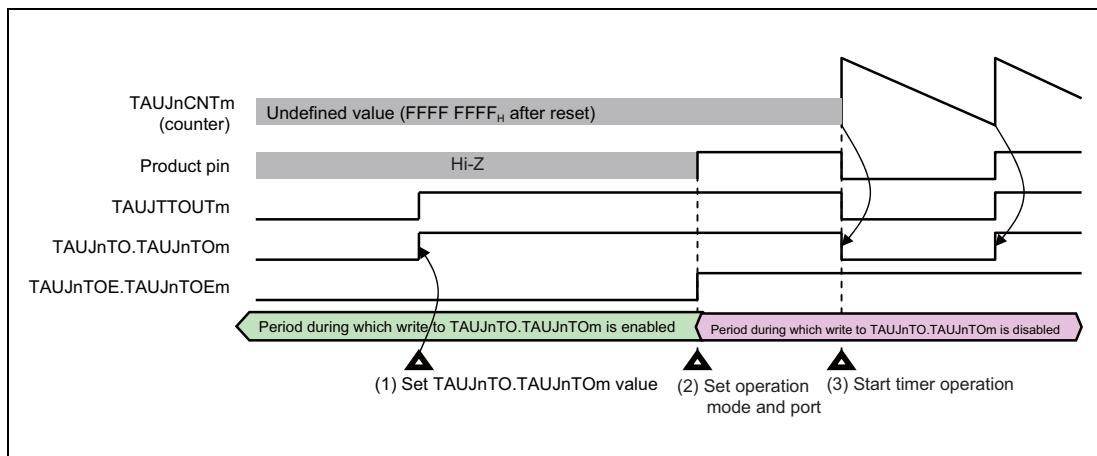


Figure 24.7 General Procedure for Specifying a TAUJTTOUTm Channel Output Mode

## 24.7.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in **Table 24.36, Channel Output Modes**.

### 24.7.2.1 Independent Channel Output Mode 1

#### Set/reset conditions

In this output mode, TAUJTTOUTm toggles when INTTAUJnIm is detected. The value of TAUJnTOL.TAUJnTOLm is ignored.

#### Prerequisites

There are no prerequisites other than those shown in **Table 24.36, Channel Output Modes**.

## 24.7.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in **Table 24.36, Channel Output Modes**.

### 24.7.3.1 Synchronous Channel Output Mode 1

#### Set/reset conditions

In this output mode, INTTAUJnIm of master channel serves as a set signal and INTTAUJnIm of the slave channel as a reset signal. If INTTAUJnIm of the master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel, i.e., the master channel is ignored.

#### Prerequisites

There are no prerequisites other than those shown in **Table 24.36, Channel Output Modes**.

## 24.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUJnTS.TAUJnTSm is set to 1 in each operating mode.

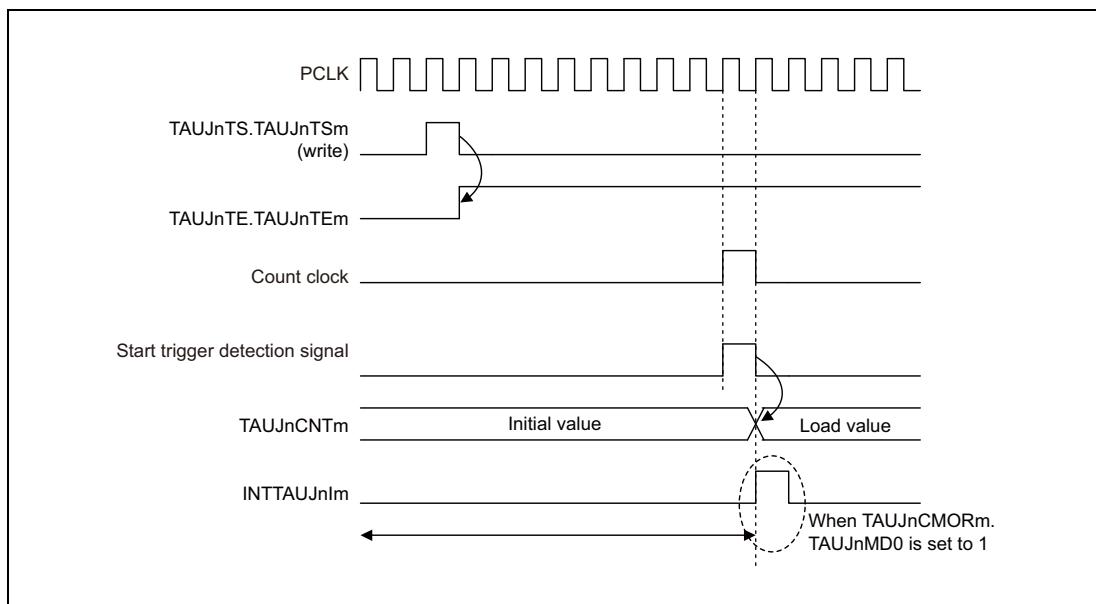
In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

### CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

### 24.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded when the counter starts.



**Figure 24.8 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode**

### 24.8.2 Other Operating Modes

In other operating modes, count clock cycle is irrelevant to start of counter operation. The counter operation start timing is triggered only upon detection of a valid edge of TAUJTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

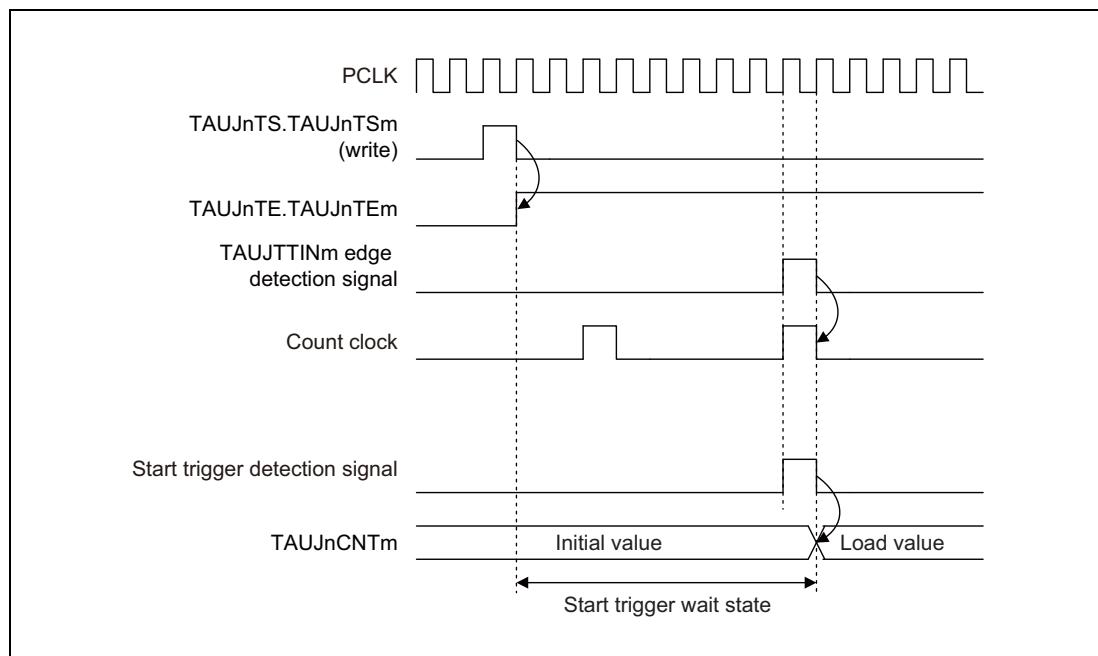


Figure 24.9 Count Start Timing in Other Operating Modes

## 24.9 TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The generation of INTTAUJnIm when the MD0 bit starts counting and the effect to TAUJTTOUTm depend on the selected function. For details, refer to the description of TAUJnCMORm.TAUJnMD0 of each function.

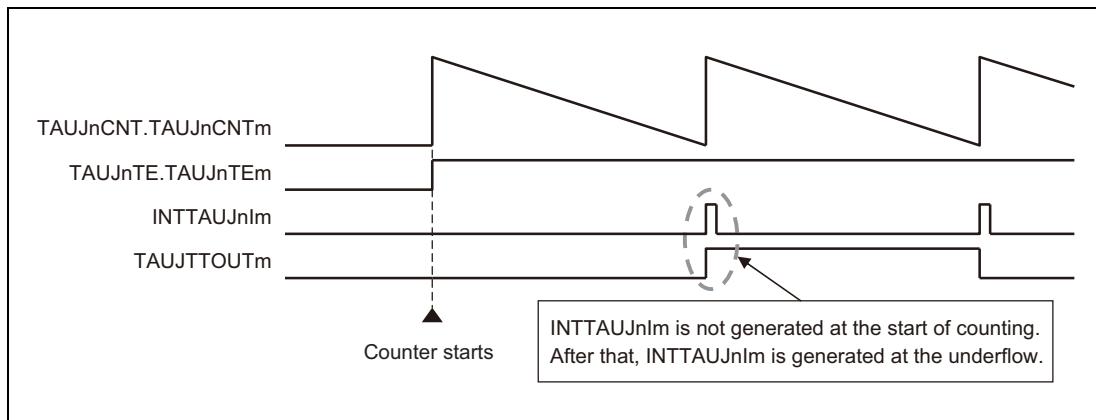


Figure 24.10 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 0)

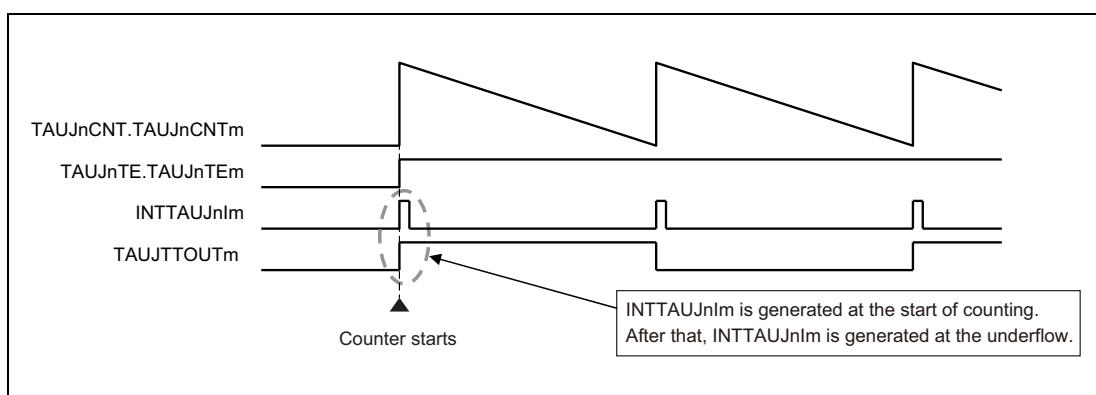


Figure 24.11 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 1)

## 24.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches  $FFFF\ FFFF_H$  and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operations in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches  $0000\ 0000_H$  at the same time as the first channel overflows ( $TAUJnCNTm = FFFF\ FFFF_H$ ).
- Set  $TAUJnCDRm$  of the second channel to  $FFFF\ FFFF_H$ .
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same  $TAUJTTINm$  input.
- The trigger detection settings ( $TAUJnCMORm.TAUJnSTS[2:0]$  and  $TAUJnCMURm.TAUJnTIS[1:0]$ ) must be identical for both channels.

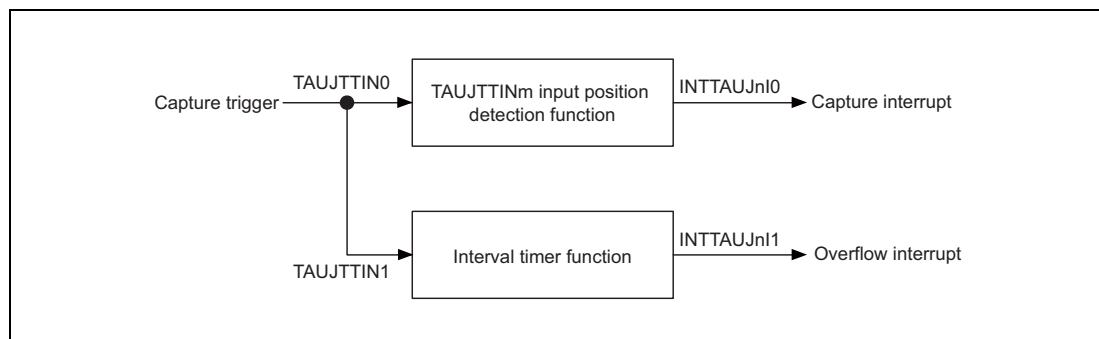
### Result:

The down-counter of the second channel reaches  $0000\ 0000_H$  at exactly the same time as the up-counter of the first channel overflows ( $TAUJnCNTm = FFFF\ FFFF_H$ ). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

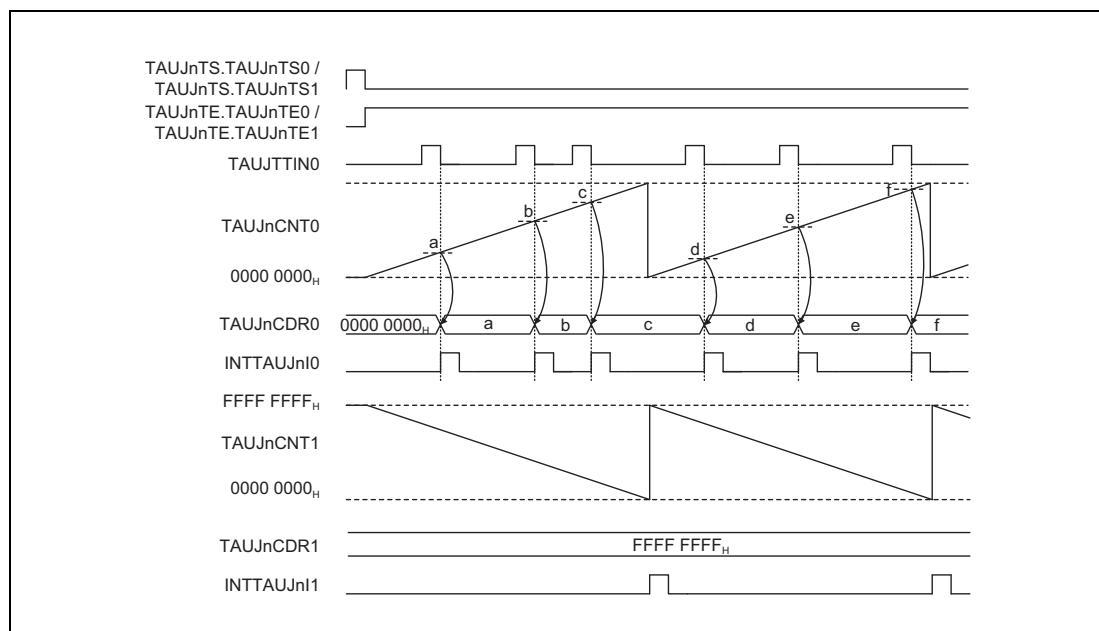
### 24.10.1 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

When the capture trigger is input simultaneously to TAUJTTINm of both channels, INTTAUJnIm of the interval timer function can detect the overflow when TAUJnCNTm of the TAUJTTINm input position detection function exceeds FFFF FFFF<sub>H</sub>.



**Figure 24.12 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function**

#### Timing diagram



**Figure 24.13 Interrupt Generation via Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function**

## 24.11 TAUJTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

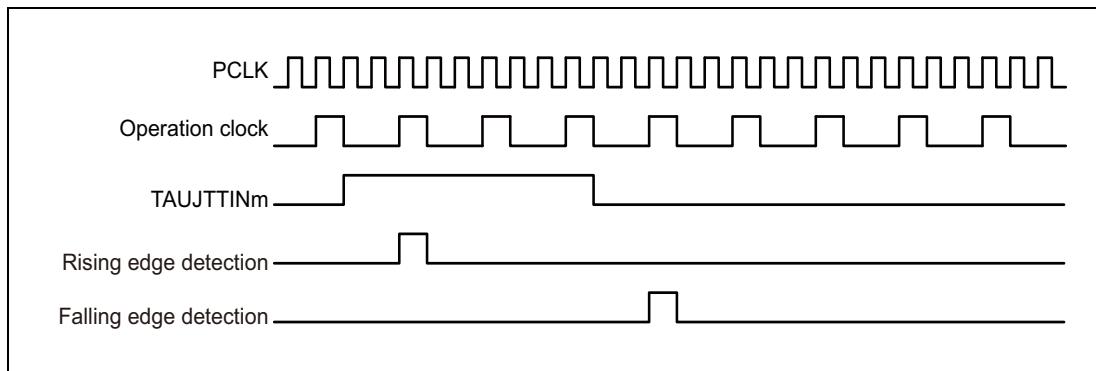


Figure 24.14 Basic Edge Detection Timing

**Figure 24.14** shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

## 24.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUJ. For a general overview of independent channel operation functions, see **Section 24.2, Overview**.

### 24.12.1 Interval Timer Function

#### 24.12.1.1 Overview

##### Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave.

##### Functional description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches 0000 0000<sub>H</sub>, INTTAUJnIm is generated and the TAUJTTOUTm signal toggles. TAUJnCNTm then loads the TAUJnCDRm value and subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEM to 0. TAUJnCNTm and TAUJTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.

##### Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJTTOUTm does not toggle. This results in a reverted TAUJTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1.

#### 24.12.1.2 Equations

$$\text{INTTAUJnIm cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1)$$

$$\text{TAUJTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm}+1) \times 2$$

### 24.12.1.3 Block Diagram and General Timing Diagram

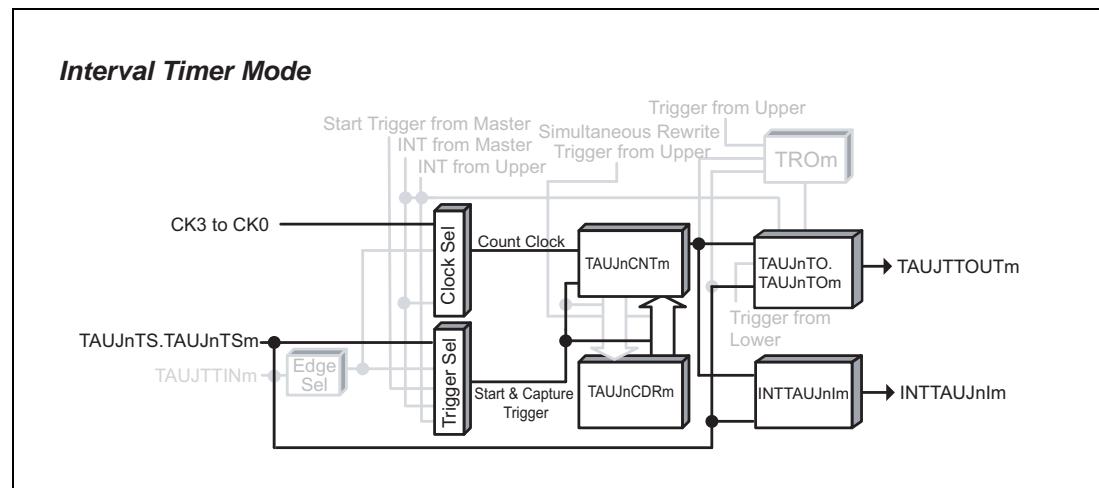


Figure 24.15 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).

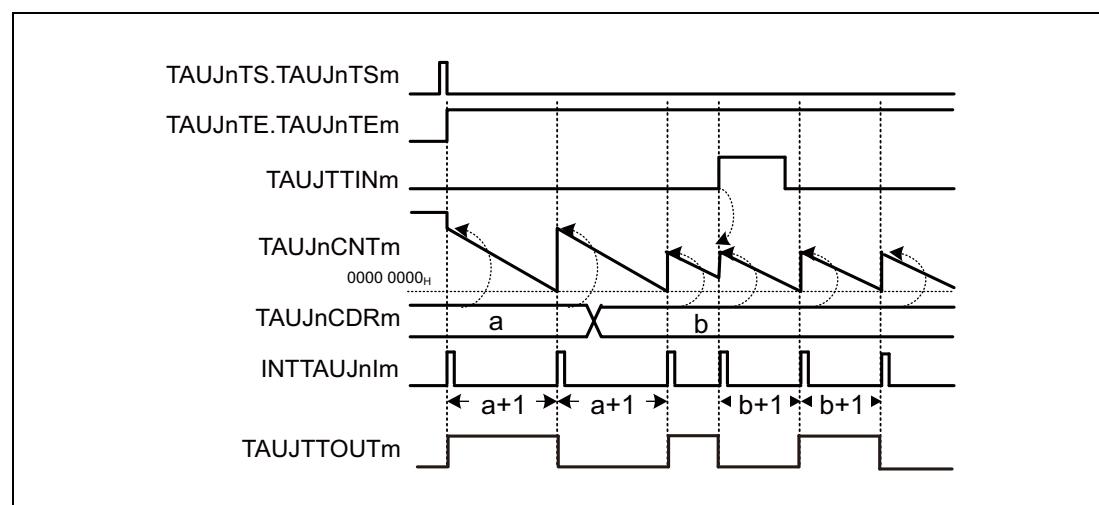


Figure 24.16 General Timing Diagram for Interval Timer Function

#### 24.12.1.4 Register Settings

##### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	—	TAUJnMD[4:1]	—	TAUJnMD0						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.37 Contents of the TAUJnCMORm register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUJnMD0	0: INTTAUJnlm is not generated and TAUJTTOUTm does not toggle when operation starts or restarts. 1: Generates INTTAUJnlm and toggles TAUJTTOUTm when operation starts or restarts.

##### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 24.38 Contents of the TAUJnCMURm register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

### (3) Channel output mode

**Table 24.39 Control Bit Settings in Independent Channel Output Mode 1**

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1 <sub>B</sub> .
TAUJnTOM.TAUJnTOMm	Write 0 <sub>B</sub> .
TAUJnTOC.TAUJnTOCm	Write 0 <sub>B</sub> .
TAUJnTOL.TAUJnTOLm	Write 0 <sub>B</sub> .

#### NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details refer to **Section 24.7, Channel Output Modes**.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the interval timer function. Therefore, these registers must be set to 0.

**Table 24.40 Simultaneous Rewrite Settings for Interval Timer Function**

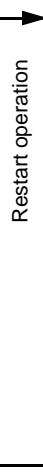
Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMM	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

### 24.12.1.5 Operating Procedure for Interval Timer Function

Table 24.41 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	<p>Set the TAUJnCMORm and TAUJnCMURm registers as described in <b>Table 24.37, Contents of the TAUJnCMORm register for Interval Timer Function</b> and <b>Table 24.38, Contents of the TAUJnCMURm register for Interval Timer Function</b>.</p> <p>Set the value of the TAUJnCDRm register.</p> <p>Set the channel output mode by setting the control bits as described in <b>Table 24.39, Control Bit Settings in Independent Channel Output Mode 1</b>.</p>	Channel operation is stopped.
Start operation	<p>Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUJnTE.TAUJnTEM is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnlm is generated and TAUJTTOUTm toggles.</p>
During operation	<p>The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.</p>	<p>TAUJnCNTm counts down. When the counter reaches 0000 0000H:</p> <ul style="list-style-type: none"> <li>• TAUJnCNTm reloads the TAUJnCDRm value and continues count operation.</li> <li>• INTTAUJnlm is generated and TAUJTTOUTm toggles.</li> </ul>
Stop operation	<p>Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUJnTE.TAUJnTEM is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.</p>

Restart operation



### 24.12.1.6 Specific Timing Diagrams

(1)  $\text{TAUJnCDRm} = 0000\ 0000_H$ , count clock = PCLK/2

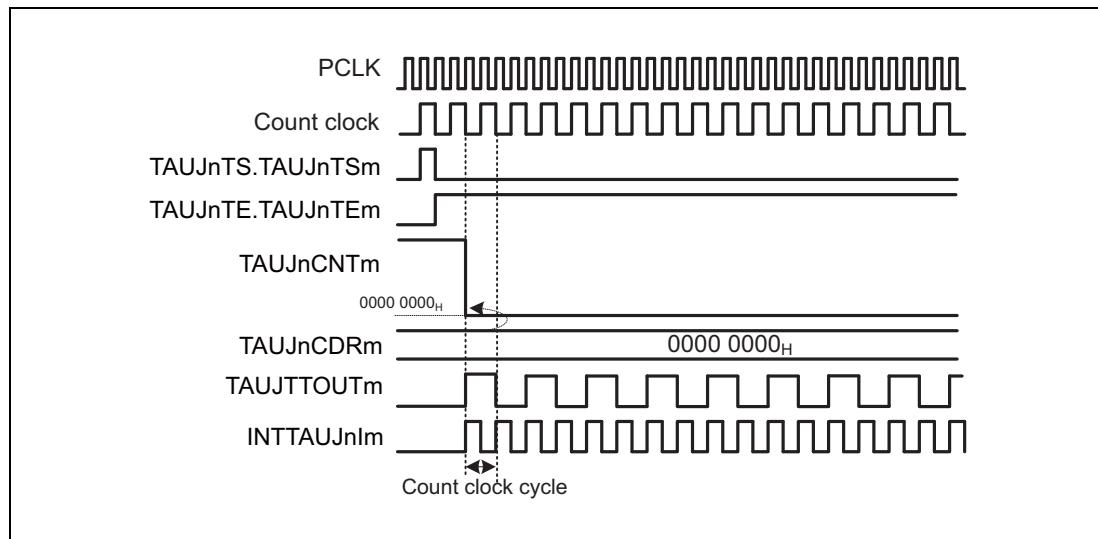
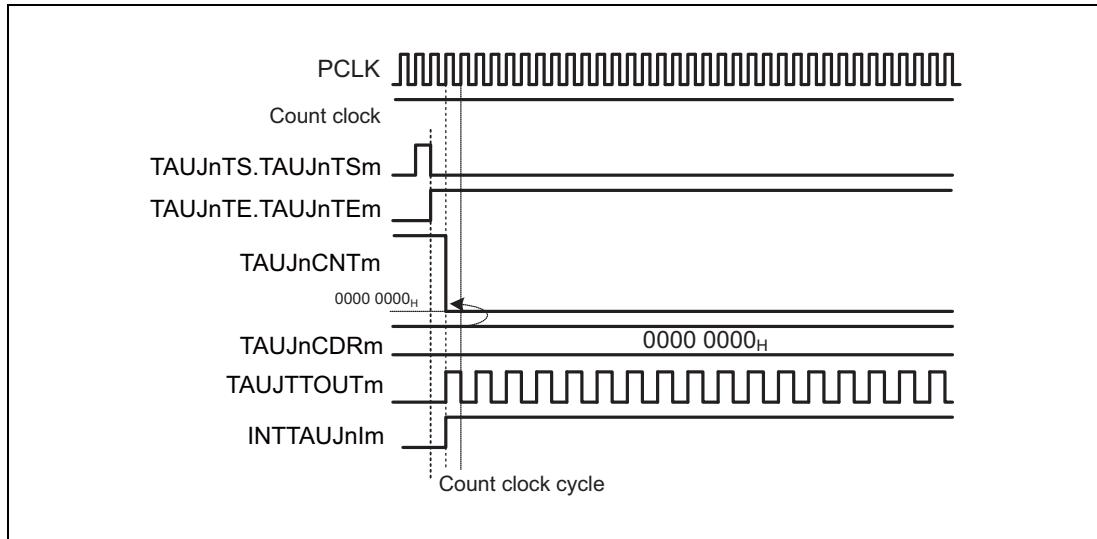


Figure 24.17  $\text{TAUJnCDRm} = 0000\ 0000_H$ , Count Clock = PCLK/2

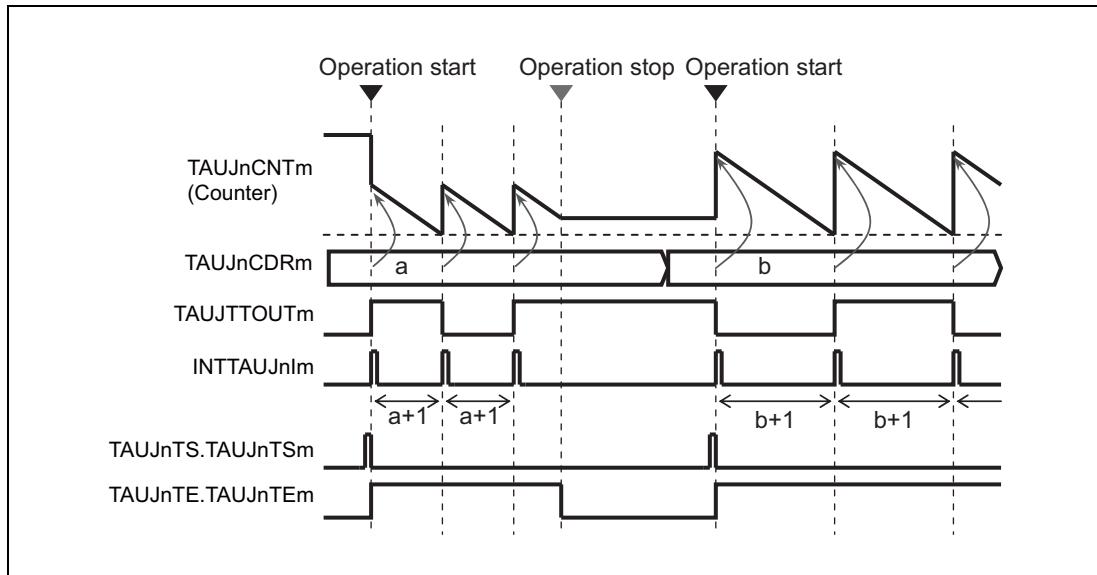
- If  $\text{TAUJnCDRm} = 0000\ 0000_H$  and the count clock = PCLK/2, the TAUJnCDRm value is loaded to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always  $0000\ 0000_H$ .
- INTTAUJnIm is generated every count clock, resulting in TAUJTTOUTm toggling every count clock.

(2)  $\text{TAUJnCDRm} = 0000\ 0000_H$ , count clock = PCLKFigure 24.18  $\text{TAUJnCDRm} = 0000\ 0000_H$ , Count Clock = PCLK

- If  $\text{TAUJnCDRm} = 0000\ 0000_H$  and the count clock = PCLK, the TAUJnCDRm value is loaded to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always  $0000\ 0000_H$ .
- INTTAUJnlm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.

TAUJTTOUTm is toggled every PCLK clock.

## (3) Operation stop and restart

Figure 24.19 Operation Stop and Restart ( $\text{TAUJnCMORm.TAUJnMD0} = 1$ )

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

#### (4) Forced restart

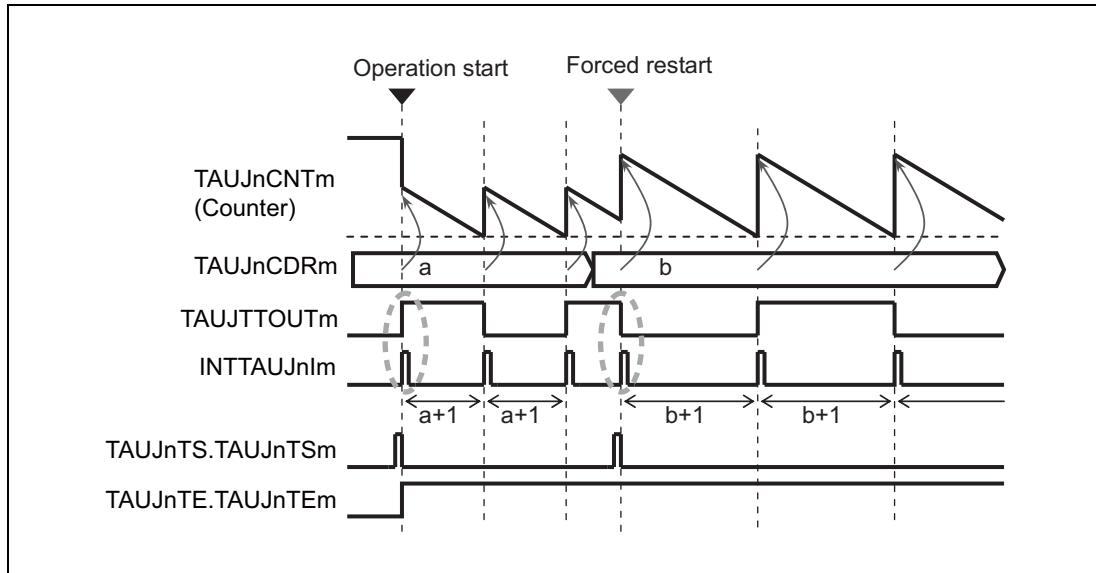


Figure 24.20 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUJnCDRm value is reflected to TAUJnCNTm and counting starts. Execute a forced restart to reflect the changed TAUJnCDRm value immediately.

### 24.12.2 TAUJTTINm Input Interval Timer Function

#### 24.12.2.1 Overview

##### Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJTTINm input edge is detected. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave. Output of square waves is only supported for TAUJ0.

##### Description

This function operates in an identical manner to the interval timer function (see **Section 24.12.1, Interval Timer Function**), except that this function is restarted by a valid TAUJTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

#### 24.12.2.2 Equations

$$\text{INTTAUJnIm cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1)$$

$$\text{TAUJTTOUTm square wave cycle} = \text{count clock cycle} \times (\text{TAUJnCDRm} + 1) \times 2$$

### 24.12.2.3 Block Diagram and General Timing Diagram

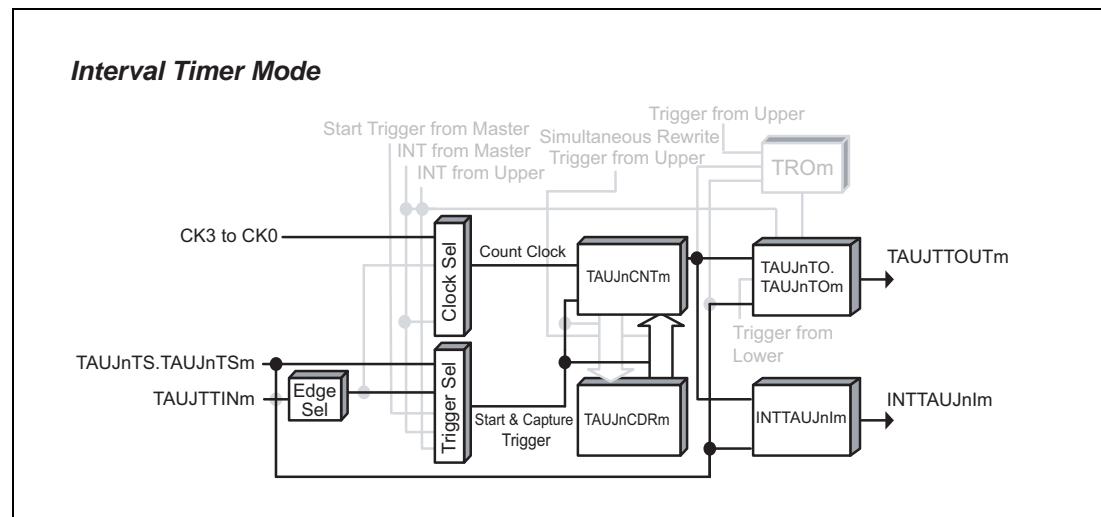


Figure 24.21 Block Diagram for TAUJTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnlm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01<sub>B</sub>)

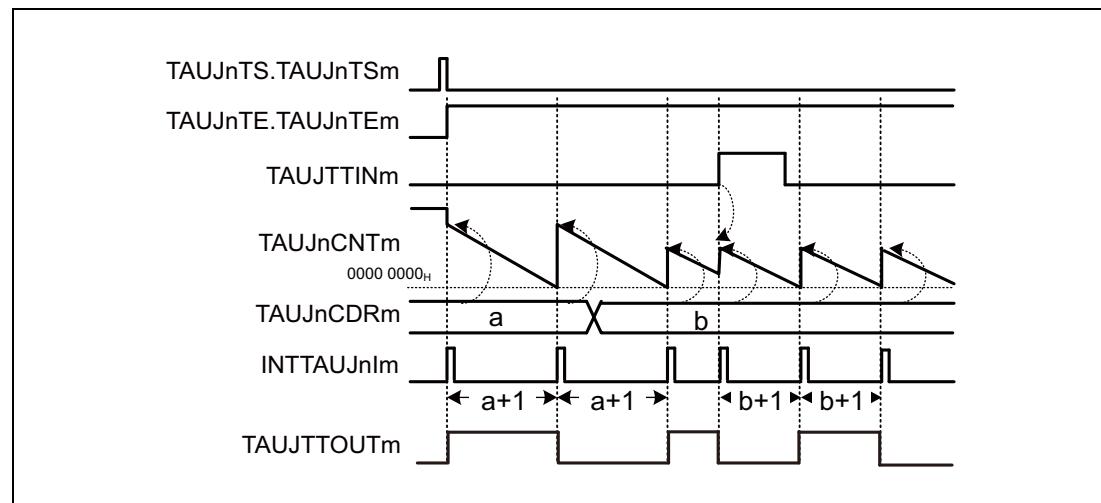


Figure 24.22 General Timing Diagram for TAUJTTINm Input Interval Timer Function

#### 24.12.2.4 Register Settings

##### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	TAUJnMD[4:1]	TAUJnMDO								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.42 Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUJnMDO	0: INTTAUJnlm is not generated and TAUJTTOUTm does not toggle when operation starts. 1: Generates INTTAUJnlm and toggles TAUJTTOUTm when operation starts.

##### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 24.43 Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

### (3) Channel output mode

**Table 24.44 Control Bit Settings for Independent Channel Output Mode 1**

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1 <sub>B</sub> .
TAUJnTOM.TAUJnTOMm	Write 0 <sub>B</sub> .
TAUJnTOC.TAUJnTOCm	Write 0 <sub>B</sub> .
TAUJnTOL.TAUJnTOLm	Write 0 <sub>B</sub> .

#### NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details refer to **Section 24.7, Channel Output Modes**.

### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input interval timer function. Therefore, these registers must be set to 0.

**Table 24.45 Simultaneous Rewrite Settings for TAUJTTINm Input Interval Timer Function**

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

### 24.12.2.5 Operating Procedure for TAUJTTINm Input Interval Timer Function

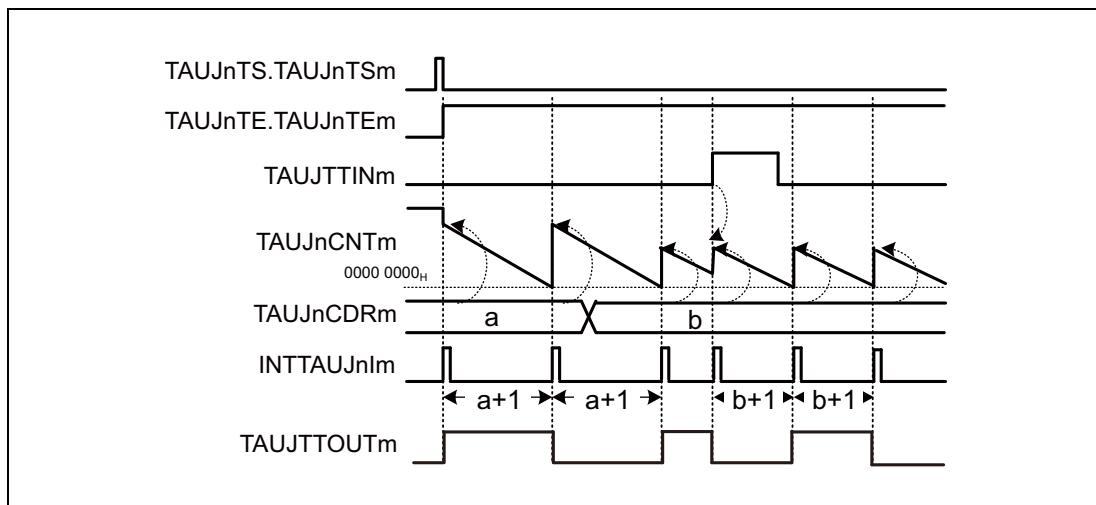
Table 24.46 Operating Procedure for TAUJTTINm Input Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	<p>Set the TAUJnCMORm and TAUJnCMURm registers as described in <b>Table 24.42, Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function</b> and <b>Table 24.43, Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function</b>.</p> <p>Set the value of the TAUJnCDRm register</p> <p>Set the channel output mode by setting the control bits as described in <b>Table 24.44, Control Bit Settings for Independent Channel Output Mode 1</b>.</p>	Channel operation is stopped.
Start operation	<p>Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUJnTE.TAUJnTEM is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnlm is generated and TAUJTTOUTm toggles.</p>
During operation	<p>The values of the TAUJnCMURm.TAUJnTIS[1:0] and TAUJnCDRm registers can be changed at any time. The TAUJnCNTm register can be read at all times.</p> <p>Detection of TAUJTTINm edge</p>	<p>TAUJnCNTm counts down. When the counter reaches 0000 0000H:</p> <ul style="list-style-type: none"> <li>• TAUJnCNTm reloads the TAUJnCDRm value and continues count operation.</li> <li>• INTTAUJnlm is generated and TAUJTTOUTm toggles.</li> </ul> <p>When a TAUJTTINm input valid edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.</p>
Stop operation	<p>Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUJnTE.TAUJnTEM is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.</p>



### 24.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 24.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUJTTINm input edge.



**Figure 24.23 Counter Triggered By Rising TAUJTTINm input edge  
( $TAUJnCMURm.TAUJnTIS[1:0] = 01_B$ ),  $TAUJnCMORm.TAUJnMD0 = 1$ )**

If a valid TAUJTTINm input edge is detected, an interrupt is generated which causes TAUJTTOUTm to toggle. In this example, the valid edge is a rising edge ( $TAUJnCMURm.TAUJnTIS[1:0] = 01_B$ ).

### 24.12.3 TAUJTTINm Input Pulse Interval Measurement Function

#### 24.12.3.1 Overview

##### Summary

This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJTTINm input signals.

##### Prerequisites

TAUJTTOUTm is not used for this function.

##### Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000<sub>H</sub>. When a valid TAUJTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000<sub>H</sub> and subsequently continues operation.

If the counter reaches FFFF FFFF<sub>H</sub> before a valid TAUJTTINm edge is detected, it overflows to 0000 0000<sub>H</sub>. The counter is reset to 0000 0000<sub>H</sub> and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

**Table 24.47 Effects of an Overflow**

TAUJnCMORm. COS[1:0]	When Overflow Occurs		When a Valid TAUJTTINm Input is then Detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF FFFF <sub>H</sub>	0	TAUJnCNTm set to 0, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEM = 0. TAUJnCNTm stops but retains its value. While the function is stopped, TAUJTTINm input valid edge detection and TAUJnCNTm capture are not performed.

##### Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, refer to **24.9, TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

**NOTE**

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

**24.12.3.2 Equations**

$$\begin{aligned} \text{TAUJTTINm input pulse interval} &= \text{count clock cycle} \times \\ &[(\text{TAUJnCSRm.TAUJnOVF} \times (\text{FFFF FFFF}_H + 1)) + \text{TAUJnCDRm capture value} + 1] \end{aligned}$$

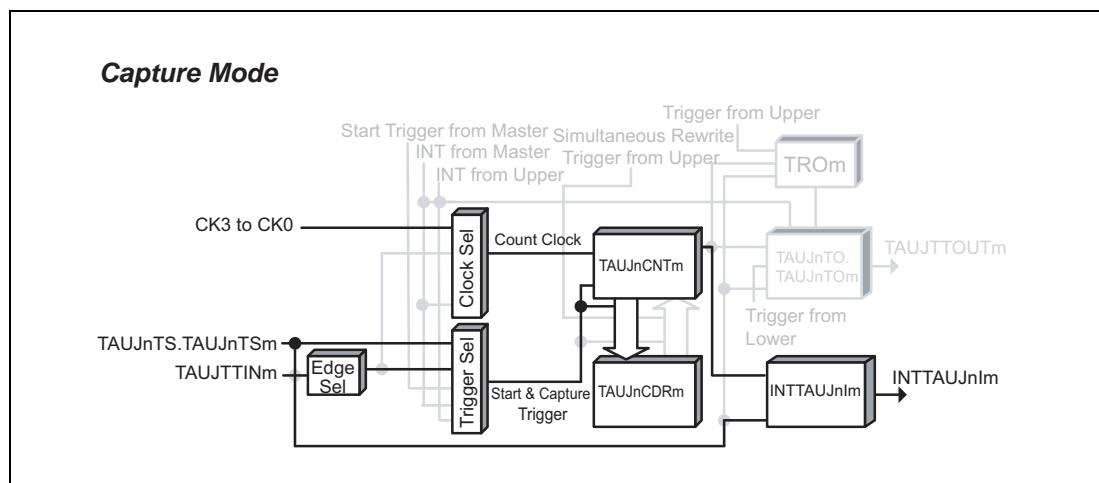
**24.12.3.3 Block Diagram and General Timing Diagram**

Figure 24.24 Block Diagram for TAUJTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUJnlm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00<sub>B</sub>)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00<sub>B</sub>).

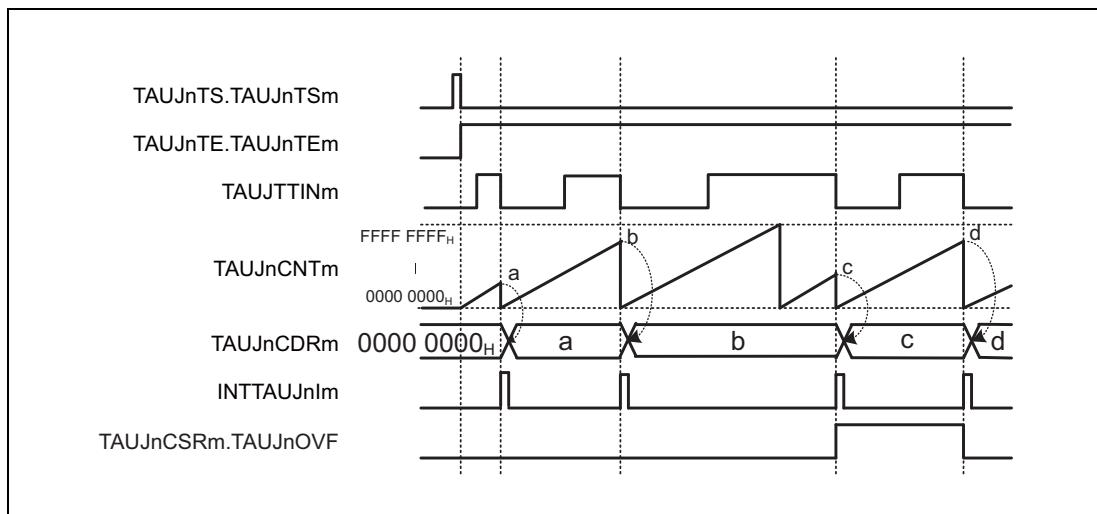


Figure 24.25 General Timing Diagram For TAUJTTINm Input Pulse Interval Measurement Function

### 24.12.3.4 Register Settings

#### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS		TAUJnSTS[2:0]	TAUJnCOS[1:0]	—		TAUJnMD[4:1]		TAUJnMD0					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.48 Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	See Table 24.47, Effects of an Overflow
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0010 <sub>B</sub> .
0	TAUJnMD0	0: INTTAUJnlm is not generated when operation starts. 1: Generates INTTAUJnlm when operation starts.

#### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 24.49 Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

#### (3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used by this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input pulse interval measurement function. Therefore, these registers must be set to 0.

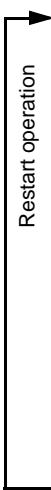
**Table 24.50 Simultaneous Rewrite Settings for TAUJTTINm Input Pulse Interval Measurement Function**

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

#### 24.12.3.5 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

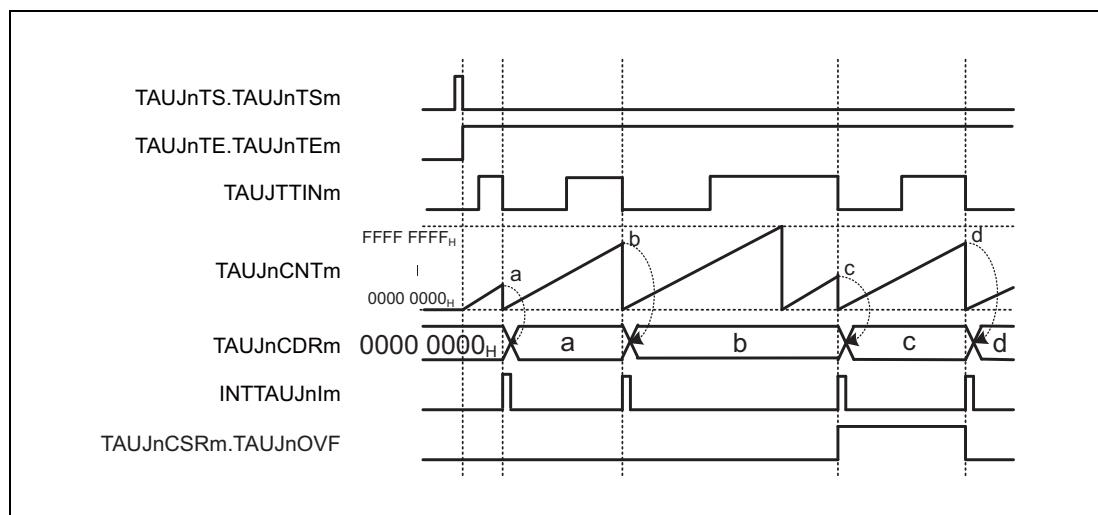
**Table 24.51 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function**

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in <b>Table 24.48, Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function</b> and <b>Table 24.49, Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function</b> .  The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000 <sub>H</sub> . INTTAUJnlm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	Detection of TAUJTTINm edges.  The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. TAUJnCSCm.TAUJnCLOV bit can be written to 1. (TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts to count up from 0000 0000 <sub>H</sub> . When a TAUJTTINm valid edge is detected: <ul style="list-style-type: none"> <li>• TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000<sub>H</sub>.</li> <li>• INTTAUJnlm is then generated.</li> </ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.



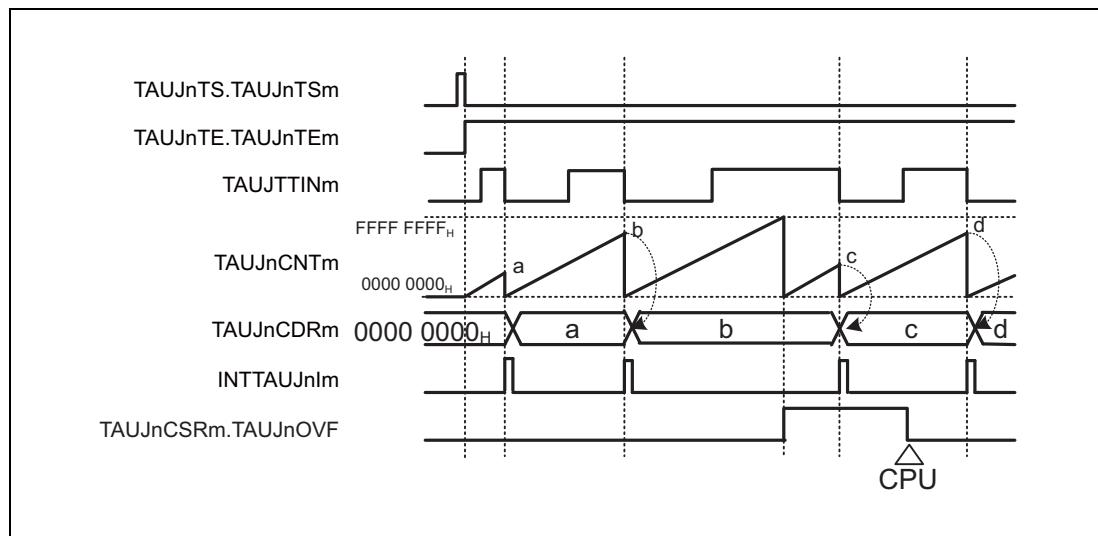
### 24.12.3.6 Specific Timing Diagrams: Overflow Behavior

#### (1) TAUJnCMORm.TAUJnCOS[1:0] = 00<sub>B</sub>



**Figure 24.26 TAUJnCMORm.TAUJnCOS[1:0] = 00<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 00<sub>B</sub>**

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- If the next valid TAUJTTINm input edge is detected when no overflow occurs, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01<sub>B</sub>

**Figure 24.27 TAUJnCMORm.TAUJnCOS[1:0] = 01<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 00<sub>B</sub>**

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

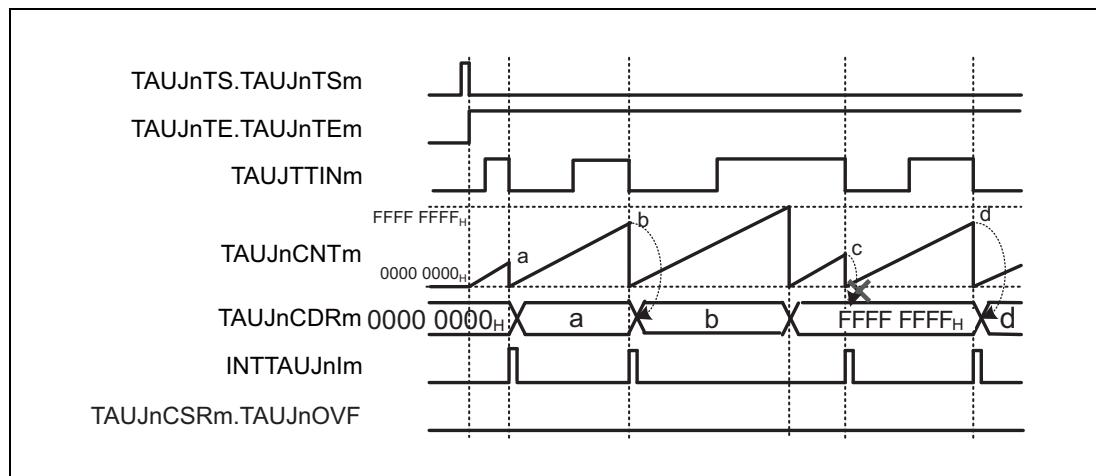
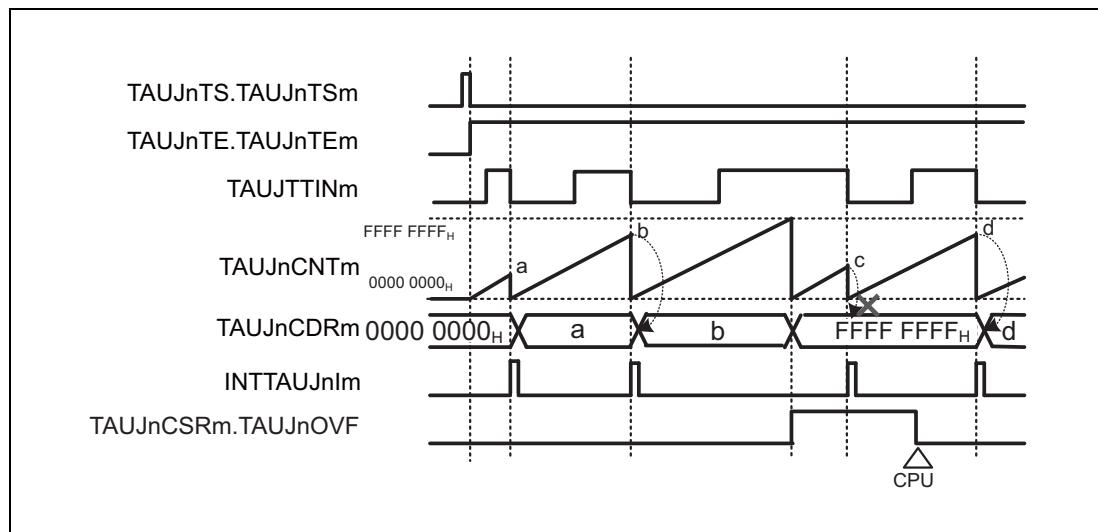
(3) TAUJnCMORm.TAUJnCOS[1:0] = 10<sub>B</sub>

Figure 24.28 TAUJnCMORm.TAUJnCOS[1:0] = 10<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF<sub>H</sub> and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11<sub>B</sub>

**Figure 24.29** TAUJnCMORm.TAUJnCOS[1:0] = 11<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 00<sub>B</sub>

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF<sub>H</sub>, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

## 24.12.4 TAUJTTINm Input Signal Width Measurement Function

### 24.12.4.1 Overview

#### Summary

This function measures the width of a TAUJTTINm signal by starting counting on one edge of the TAUJTTINm signal and capturing the counter value on the opposite edge.

#### Prerequisites

TAUJTTOUTm is not used for this function.

#### Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation. When a valid TAUJTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000<sub>H</sub>. When a valid TAUJTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJTTINm input start edge.

If the counter reaches FFFF FFFF<sub>H</sub> before a valid TAUJTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000<sub>H</sub> and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

**Table 24.52 Effects of an Overflow**

TAUJnCMORm. COS[1:0]	When Overflow Occurs		When a Valid TAUJTTINm Input Stop Edge is Detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded to TAUJnCDRm.	1
01		1		
10	Set to FFFF FFFF <sub>H</sub>	0	TAUJnCNTm stops counting, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

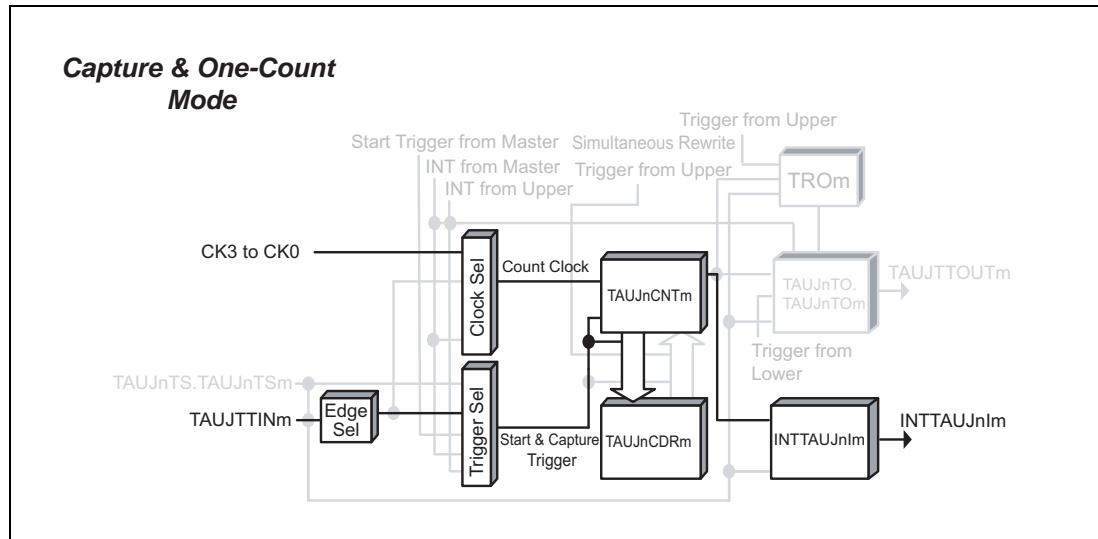
#### NOTE

When TAUJnCMORm.COS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

#### 24.12.4.2 Equations

$$\begin{aligned} \text{TAUJTTINm input signal width} &= \text{count clock cycle} \times \\ &[(\text{TAUJnCSRm.TAUJnOVF} \times (\text{FFFF FFFF}_H + 1)) + \text{TAUJnCDRm capture value} + 1] \end{aligned}$$

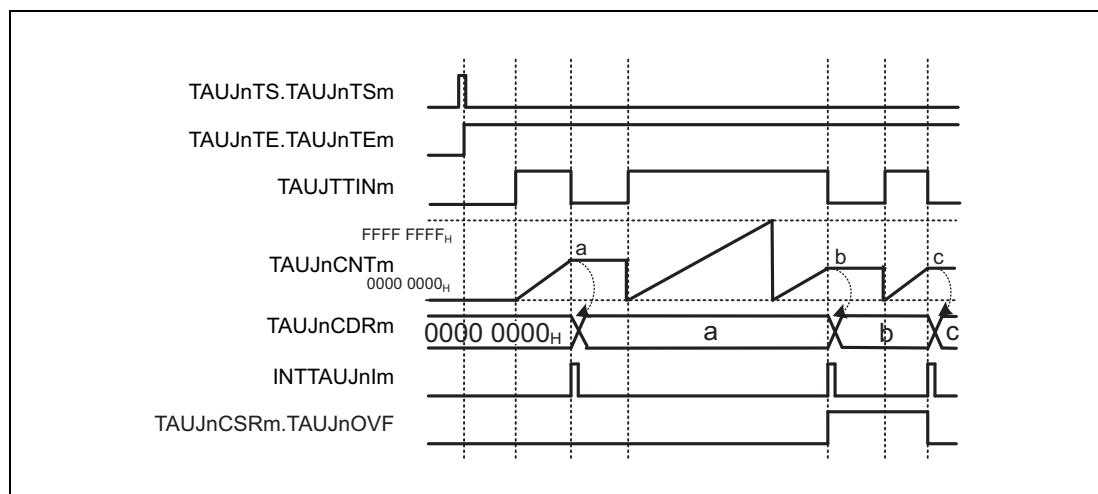
#### 24.12.4.3 Block Diagram and General Timing Diagram



**Figure 24.30 Block Diagram for TAUJTTINm Input Signal Width Measurement Function**

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
(TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00<sub>B</sub>).



**Figure 24.31 General Timing Diagram for TAUJTTINm Input Signal Width Measurement Function**

#### 24.12.4.4 Register Settings

##### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	TAUJnMD[4:1]	TAUJnMDO								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.53 Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	See Table 24.52, Effects of an Overflow.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0110 <sub>B</sub> .
0	TAUJnMDO	Write 0 <sub>B</sub> .

##### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W

Table 24.54 Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

##### (3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input signal width measurement function. Therefore, these registers must be set to 0.

**Table 24.55 Simultaneous Rewrite Settings for TAUJTTINm Input Signal Width Measurement Function**

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

#### 24.12.4.5 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

**Table 24.56 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function**

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in <b>Table 24.53, Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function</b> and <b>Table 24.54, Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function</b> .  The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When a TAUJTTINm start edge is detected, TAUJnCNTm starts to count up.
During operation	The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000H. When a TAUJTTINm valid edge is detected: <ul style="list-style-type: none"><li>• TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value.</li><li>• INTTAUJnlm is then generated.</li><li>• Counting stops at the “value that transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJTTINm start edge.</li></ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

#### 24.12.4.6 Specific Timing Diagrams: Overflow Behavior

##### (1) TAUJnCMORm.TAUJnCOS[1:0] = 00<sub>B</sub>

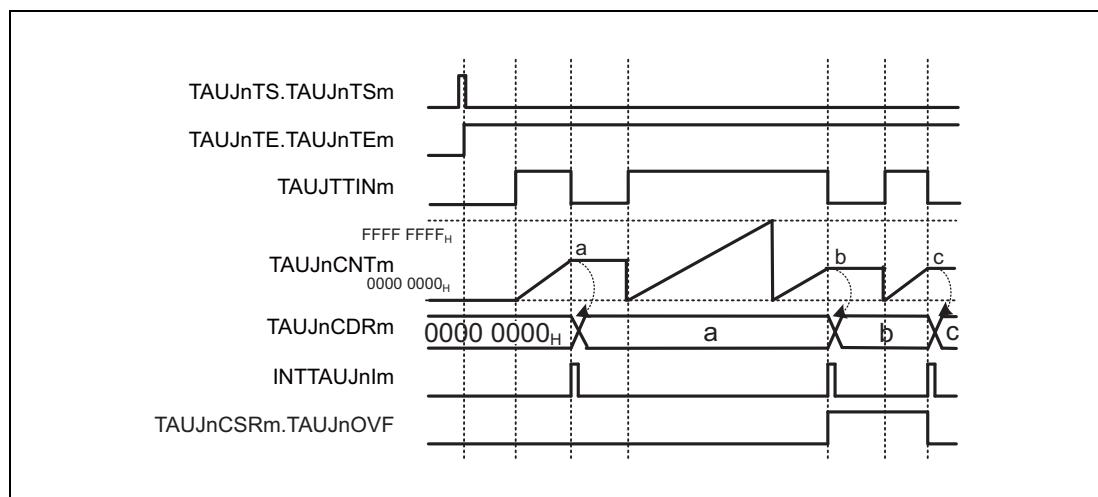


Figure 24.32 TAUJnCMORm.TAUJnCOS[1:0] = 00<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

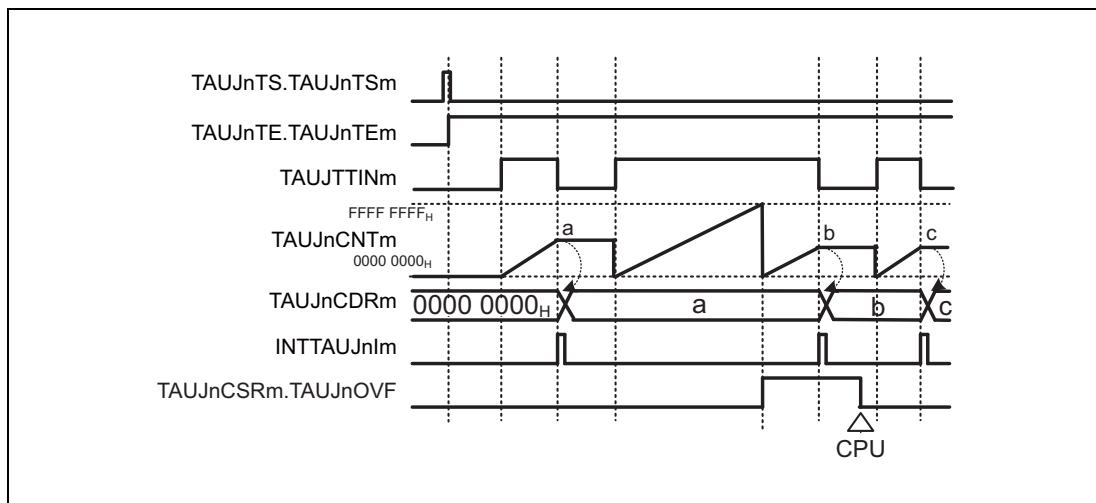
(2) TAUJnCMORm.TAUJnCOS[1:0] = 01<sub>B</sub>

Figure 24.33 TAUJnCMORm.TAUJnCOS[1:0] = 01<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

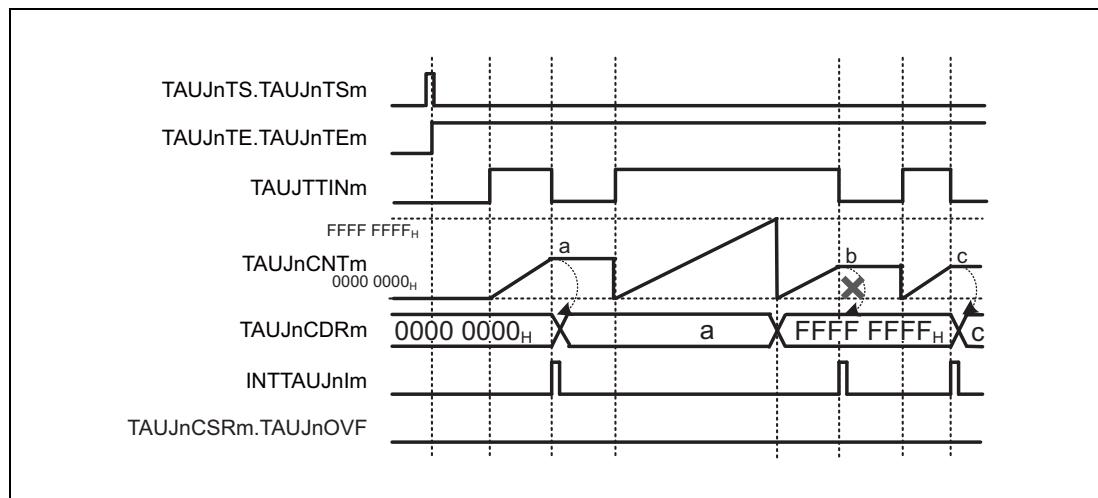
(3) TAUJnCMORm.TAUJnCOS[1:0] = 10<sub>B</sub>

Figure 24.34 TAUJnCMORm.TAUJnCOS[1:0] = 10<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF<sub>H</sub> and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

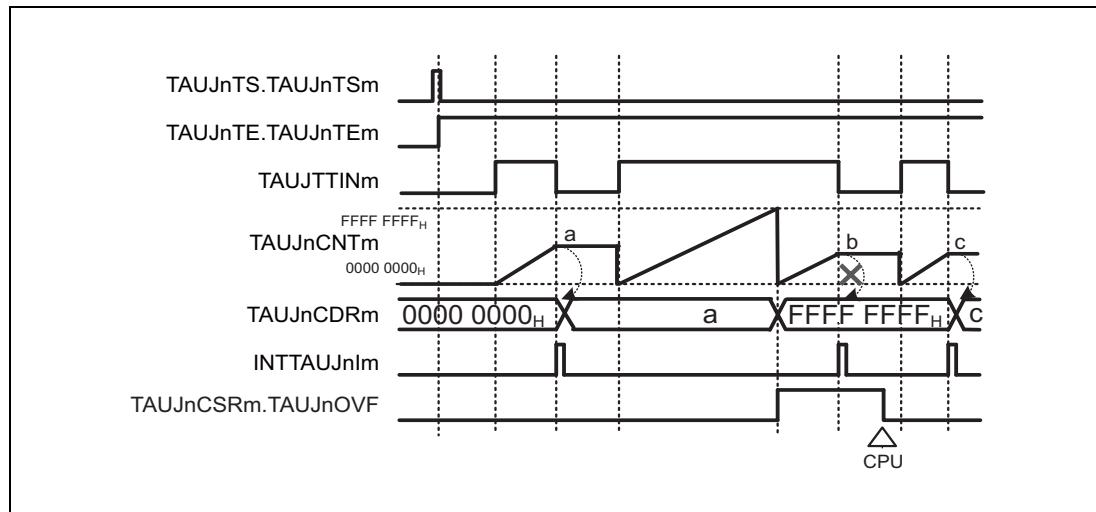
(4) TAUJnCMORm.TAUJnCOS[1:0] = 11<sub>B</sub>

Figure 24.35 TAUJnCMORm.TAUJnCOS[1:0] = 11<sub>B</sub>, TAUJnCMORm.TAUJnMD0 = 0,  
TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF<sub>H</sub>, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

## 24.12.5 TAUJTTINm Input Position Detection Function

### 24.12.5.1 Overview

#### Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUJTTINm signal.

#### Prerequisites

TAUJTTOUTm is not used for this function

#### Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation. The counter starts to count from 0000 0000<sub>H</sub>. When a valid TAUJTTINm input edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues to count.

When the counter reaches FFFF FFFF<sub>H</sub>, the counter restarts from 0000 0000<sub>H</sub>.

#### NOTE

The input TAUJTTINm is sampled at the frequency of the operation clock, specified by TAUJnCMORm.TAUJnCKS[1:0] bits. As a result, the output cycle of TAUJTTOUTm has an error of  $\pm 1$  operation clock cycle.

#### Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

### 24.12.5.2 Equations

Function duration at a TAUJTTINm input pulse =  
count clock cycle  $\times$  (TAUJnCDRm capture value + 1)

### 24.12.5.3 Block Diagram and General Timing Diagram

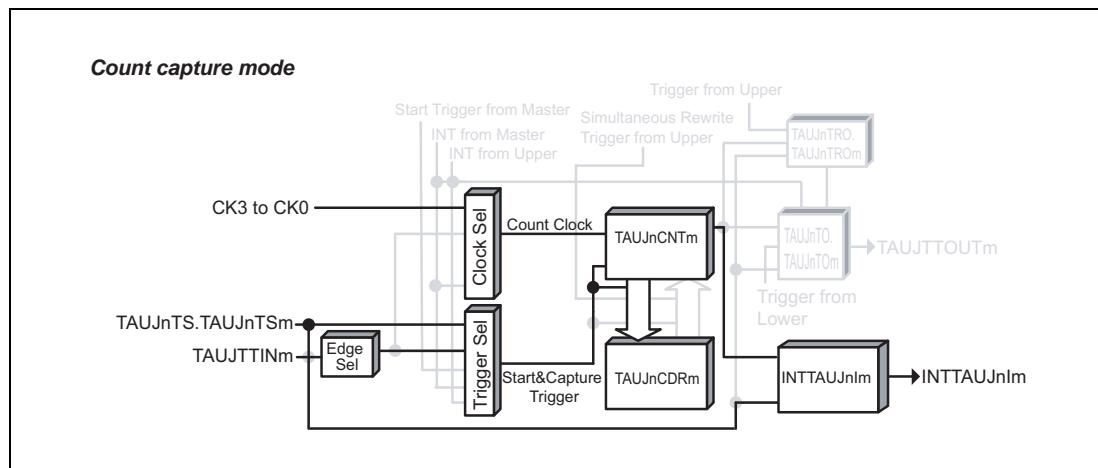


Figure 24.36 Block Diagram of TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- INTTAUJnlm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00<sub>B</sub>)

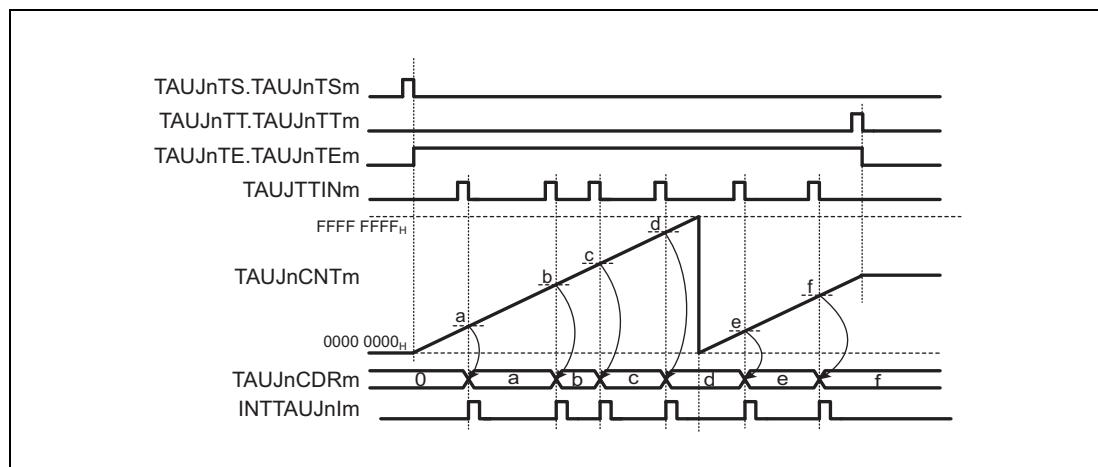


Figure 24.37 General Timing Diagram for TAUJTTINm Input Position Detection Function

#### 24.12.5.4 Register Settings

##### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS		TAUJnSTS[2:0]	TAUJnCOS[1:0]	—		TAUJnMD[4:1]		TAUJnMDO					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.57 Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 001 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 01 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1011 <sub>B</sub> .
0	TAUJnMDO	0: INTTAUJnlm is not generated when operation starts. 1: Generates INTTAUJnlm when operation starts.

##### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 24.58 Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

##### (3) Channel output mode

The channel output mode is not used by this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input position detection function. Therefore, these registers must be set to 0.

**Table 24.59 Simultaneous Rewrite Settings for TAUJTTINm Input Position Detection Function**

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

#### 24.12.5.5 Operating Procedure for TAUJTTINm Input Position Detection Function

**Table 24.60 Operating Procedure for TAUJTTINm Input Position Detection Function**

	Operation	Status of TAUJn
→ Restart operation	Initial channel setting  Set the TAUJnCMORm register and TAUJnCMURm registers as described in <b>Table 24.57, Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function</b> and <b>Table 24.58, Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function</b> .  The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
	Start operation  Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEM is set to 1 and the counter starts. INTTAUJnlm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
	During operation  The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time.	TAUJnCNTm starts to count up from 0000 0000 <sub>H</sub> . When a TAUJTTINm valid edge is detected: <ul style="list-style-type: none"><li>• TAUJnCNTm transfers (captures) its value to TAUJnCDRm.</li><li>• INTTAUJnlm is output.</li><li>• The counter value is not cleared to 0000 0000<sub>H</sub> and TAUJnCNTm continues count operation.</li></ul> Afterwards, this procedure is repeated. When TAUJnCNTm reaches FFFF FFFF <sub>H</sub> , the counter restarts from 0000 0000 <sub>H</sub> .
	Stop operation  Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEM is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

### 24.12.5.6 Specific Timing Diagrams

#### (1) Operation stop and restart

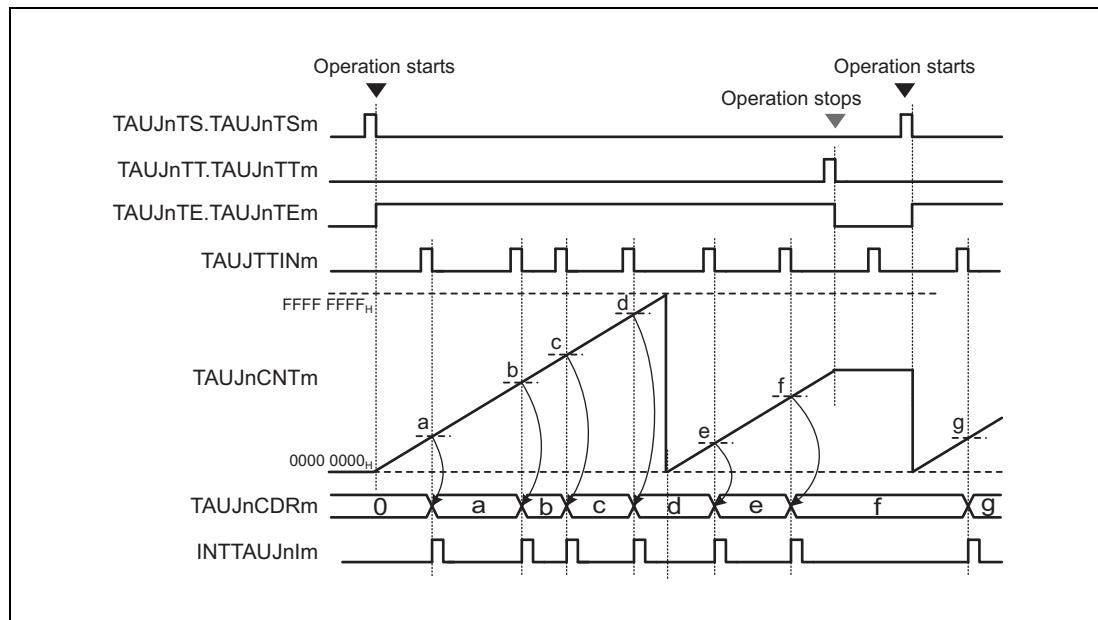


Figure 24.38 Operation Stop and Restart ( $\text{TAUJnCMORm}.\text{TAUJnMD0} = 0$ ,  $\text{TAUJnCMURm}.\text{TAUJnTIS}[1:0] = 00_B$ )

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000<sub>H</sub>.

## 24.12.6 TAUJTTINm Input Period Count Detection Function

### 24.12.6.1 Overview

#### Summary

This function measures the cumulative width of a TAUJTTINm input signal.

#### Prerequisites

TAUJTTOUTm is not used for this function.

#### Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJTTINm input edge.

When a valid TAUJTTINm input start edge is detected, the counter starts to count from 0000 0000<sub>H</sub>.

When a valid TAUJTTINm input stop edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJTTINm input start edge is detected.

When the next valid TAUJTTINm input start edge is detected, the counter restarts counting from the stop value.

When the counter reaches FFFF FFFF<sub>H</sub>, the counter restarts from 0000 0000<sub>H</sub>.

This function cannot be forcibly restarted.

#### NOTE

The input TAUJTTINm signal is sampled at the frequency of the operation clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

#### Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10<sub>B</sub>, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

### 24.12.6.2 Equations

Cumulative TAUJTTINm input width =  
count clock cycle × (TAUJnCDRm capture value + 1)

### 24.12.6.3 Block Diagram and General Timing Diagram

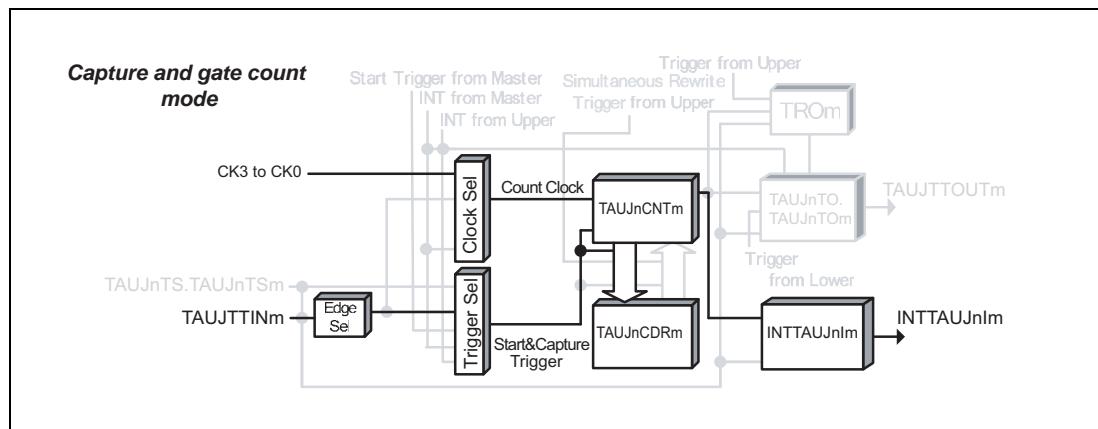


Figure 24.39 Block Diagram for TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
(TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>)

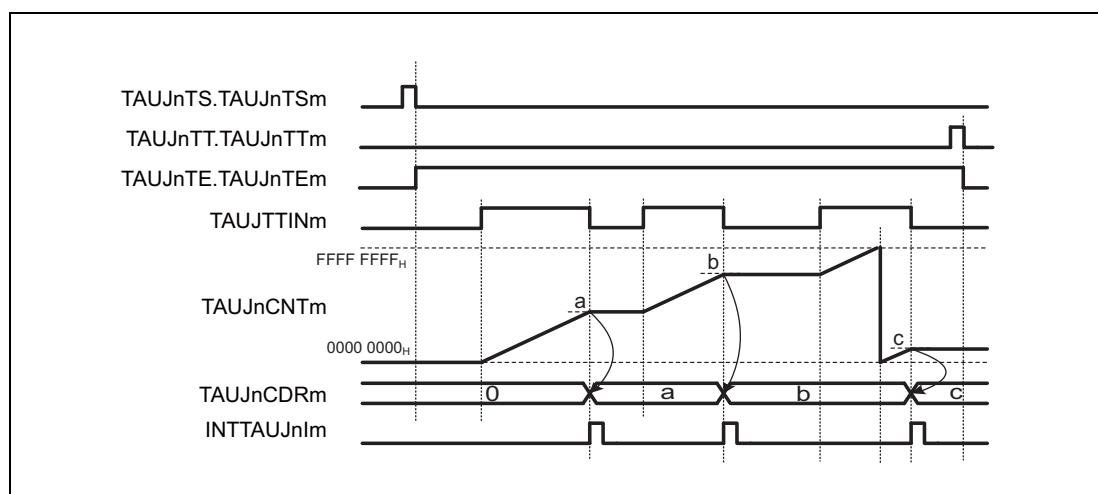


Figure 24.40 General Timing Diagram for TAUJTTINm Input Period Count Detection Function

#### 24.12.6.4 Register Settings

##### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.61 Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 01 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1101 <sub>B</sub> .
0	TAUJnMD0	Write 0 <sub>B</sub> .

##### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 24.62 Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

##### (3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input period count detection function. Therefore, these registers must be set to 0.

**Table 24.63 Simultaneous Rewrite Settings for TAUJTTINm Input Period Count Detection Function**

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

#### 24.12.6.5 Operating Procedure for TAUJTTINm Input Period Count Detection Function

**Table 24.64 Operating Procedure for TAUJTTINm Input Period Count Detection Function**

Operation	Status of TAUJn	
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in <b>Table 24.61, Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function</b> and <b>Table 24.62, Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function</b> . The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEM is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge.
During operation	TAUJTTINm edge detection The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time.	When a TAUJTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts to count up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnlm is generated. Counting stops at the “value transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When TAUJnCNTm reaches FFFF FFFF <sub>H</sub> , the counter restarts from 0000 0000 <sub>H</sub> . Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEM is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

### 24.12.6.6 Specific Timing Diagrams

#### (1) Operation Stop and Restart

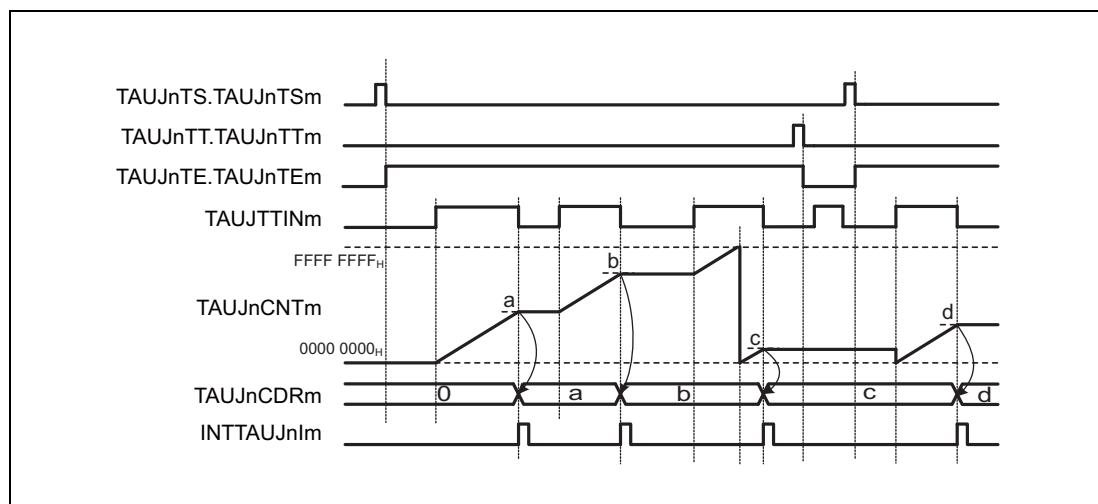


Figure 24.41 Operation Stop and Restart ( $\text{TAUJnCMURm}.\text{TAUJnTIS}[1:0] = 11_B$ )

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000<sub>H</sub>.

## 24.12.7 Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

### 24.12.7.1 Overview

#### Summary

This function measures the width of an individual TAUJTTINm input signal. An interrupt is generated if the TAUJTTINm input width is longer than FFFF FFFF<sub>H</sub> + 1.

#### Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF FFFF<sub>H</sub>.

#### Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF FFFF<sub>H</sub> is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUJTTINm input start edge is detected, TAUJnCNTm loads FFFF FFFF<sub>H</sub> and starts to count down.

If the counter reaches 0000 0000<sub>H</sub> before a stop edge is detected, an interrupt is generated.

#### Conditions

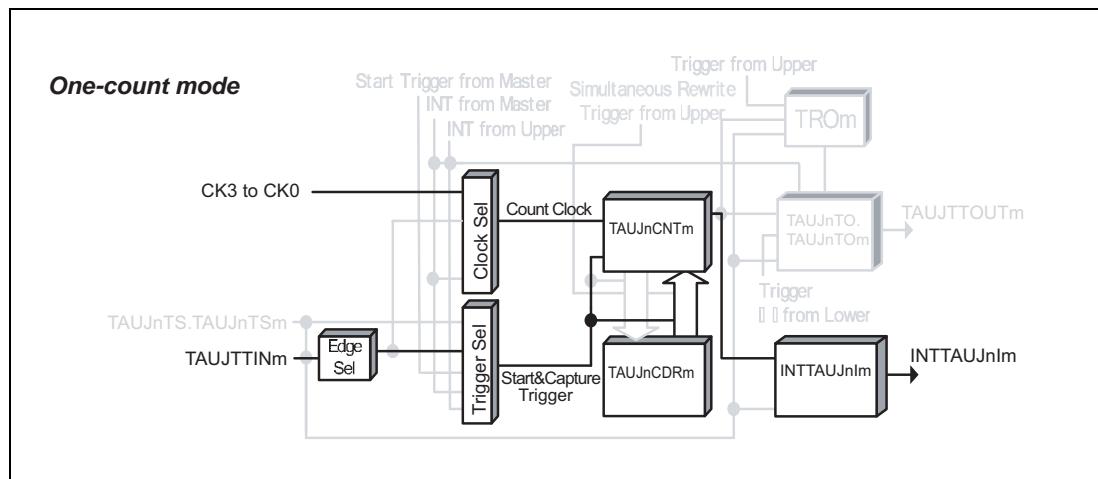
The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10<sub>B</sub>, the TAUJTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>, the TAUJTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

#### NOTE

The counter cannot be restarted during operation.

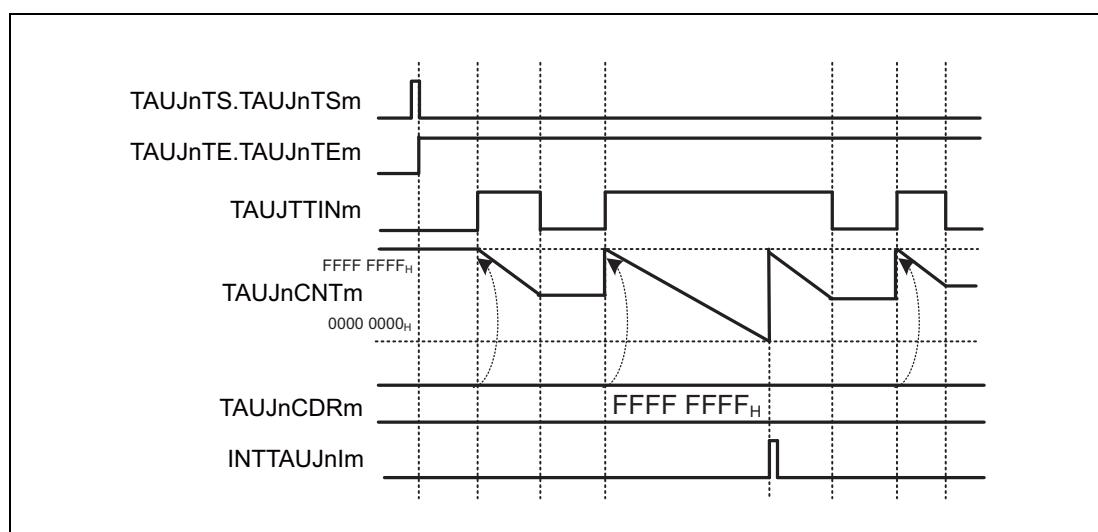
### 24.12.7.2 Block Diagram and General Timing Diagram



**Figure 24.42** Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
( $\text{TAUJnCMURm.TAUJnTIS}[1:0] = 11_B$ )



**Figure 24.43** General Timing Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

### 24.12.7.3 Register Settings

#### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	TAUJnMD[4:1]	TAUJnMDO								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.65 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUJnMDO	Write 0 <sub>B</sub> .

#### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W

Table 24.66 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

#### (3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the overflow interrupt output function (during TAUJTTINm width measurement). Therefore, these registers must be set to 0.

**Table 24.67 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)**

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

#### 24.12.7.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

**Table 24.68 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)**

Operation	Status of TAUJn	
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in <b>Table 24.65, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)</b> and <b>Table 24.66, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)</b> . Set the value of the TAUJnCDRm register to FFFF FFFF <sub>H</sub> .	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.  Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge.  When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF <sub>H</sub> ).
During operation	The TAUJnCNTm register can be read at any time.	TAUJnCNTm counts down. When the counter reaches 0000 0000 <sub>H</sub> : <ul style="list-style-type: none"><li>• INTTAUJnlm is generated.</li></ul> When a reverse edge of TAUJTTINm is detected during count operation: <ul style="list-style-type: none"><li>• TAUJnCNTm stops counting and waits for a trigger.</li></ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

## 24.12.8 Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

### 24.12.8.1 Overview

#### Summary

This function measures the cumulative width of a TAUJTTINm input signal. An interrupt is generated if the cumulative TAUJTTINm input width is longer than  $\text{FFFF FFFF}_H$ , and an overflow interrupt can be output.

#### Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to  $\text{FFFF FFFF}_H$ .

#### Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected.  $\text{FFFF FFFF}_H$  is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUJTTINm input start edge and then continues to count down from the current value.

When the counter reaches  $0000\ 0000_H$  an interrupt is generated.  $\text{FFFF FFFF}_H$  is loaded to TAUJnCNTm and the counter continues to count down until a TAUJTTINm input stop edge is detected.

#### Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] =  $10_B$ , the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] =  $11_B$ , the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

#### NOTE

The counter cannot be restarted during operation.

### 24.12.8.2 Block Diagram and General Timing Diagram

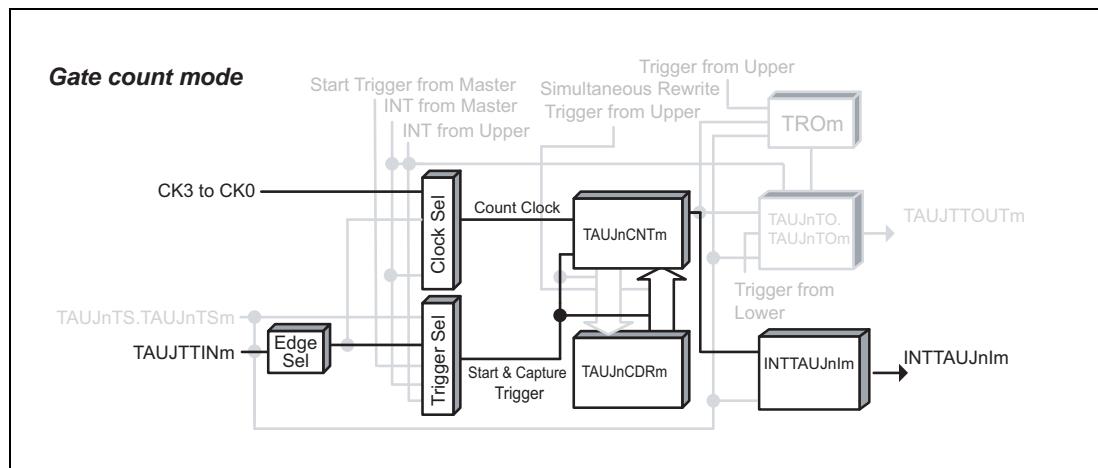


Figure 24.44 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement  
(TAUJnCMURm.TAUJnTIS[1:0] = 11<sub>B</sub>)

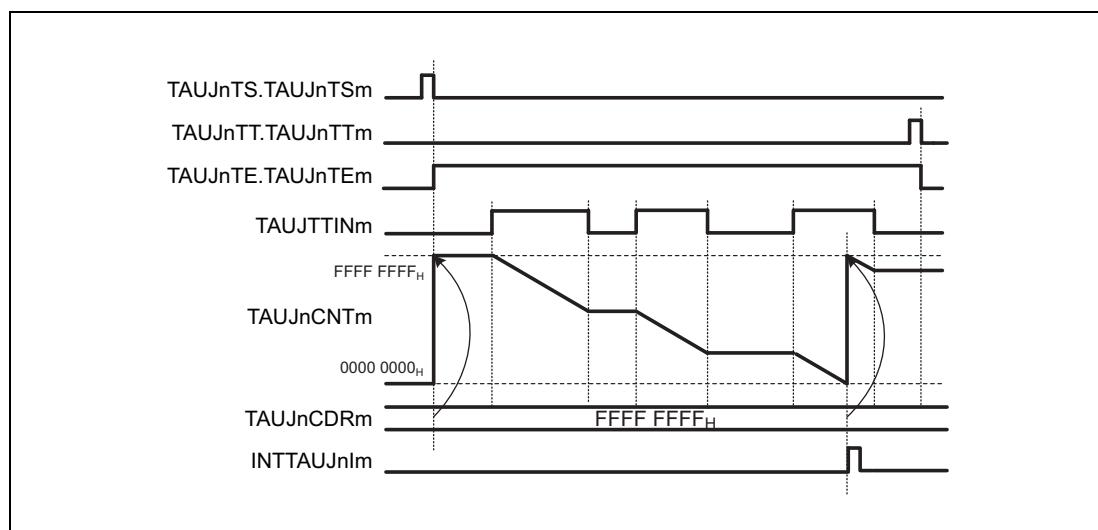


Figure 24.45 General Timing Diagram For Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

### 24.12.8.3 Register Settings

#### (1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	TAUJnMD[4:1]	TAUJnMDO								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.69 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 010 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1100 <sub>B</sub> .
0	TAUJnMDO	Write 0 <sub>B</sub> .

#### (2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R/W	R	R	R	R	R	R/W	R/W

Table 24.70 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

#### (3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

#### (4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

**Table 24.71 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)**

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

#### 24.12.8.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

**Table 24.72 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)**

Operation	Status of TAUJn	
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in <b>Table 24.69, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)</b> and <b>Table 24.70, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)</b> .  Set the value of the TAUJnCDRm register to FFFF FFFF <sub>H</sub> .	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.  Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEM is set to 1 and TAUJnCNTm waits for detection of the start edge.  When a start edge is detected, the value of TAUJnCDRm (FFFF FFFF <sub>H</sub> ) is loaded to TAUJnCNTm.
During operation	The TAUJnCNTm register can be read at all times	TAUJnCNTm counts down. When the counter reaches 0000 0000 <sub>H</sub> : <ul style="list-style-type: none"><li>• INTTAUJnlm is generated.</li><li>• TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF<sub>H</sub>) and continues to count down.</li></ul> When a reverse edge of TAUJTTINm is detected during count operation: <ul style="list-style-type: none"><li>• TAUJnCNTm stops and retains the stop value.</li></ul> When a TAUJTTINm valid edge is detected while the counter is stopped: <ul style="list-style-type: none"><li>• TAUJnCNTm counts down from the stop value.</li></ul> Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEM is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

## 24.13 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the TAUJ. For a general overview of synchronous channel operation, see **Section 24.2, Overview**.

### 24.13.1 PWM Output Function

#### 24.13.1.1 Overview

##### Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

##### Prerequisites

- Two channels
- The operation mode for the master channel should be set to the interval timer mode. (See **Table 24.73, Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operation mode for the slave channel should be set to the one-count mode. (See **Table 24.76, Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUJTTOUTm is not used for the master channel of this function.
- The channel output mode for the slave channels should be set to synchronous channel output mode 1. (See **24.7, Channel Output Modes**.)

##### Description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEM = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counters start to count down from these values. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set or reset to realize the PWM output.

- Master channel:

When the counter of the master channel reaches 0000 0000<sub>H</sub> and pulse cycle time has elapsed, INTTAUJnIm is generated. The TAUJnCDRm value is loaded to TAUJnCNTm, and the counter counts down.

- Slave channel(s):

INTTAUJnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUJnCDRm (slave) is loaded to TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJTTOUTm signal is set to the active level.

When the counter reaches 0000 0000<sub>H</sub>, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJTTOUTm signal is set to the inactive level. The counter returns to FFFF FFFF<sub>H</sub> and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEM to 0. TAUJnCNTm and TAUJTTOUTm of master and slave

channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

### Conditions

Simultaneous rewrite can be used with this function. Please refer to **Section 24.6, Simultaneous Rewrite**.

### 24.13.1.2 Equations

$$\text{Pulse cycle} = (\text{TAUJnCDRm (master)} + 1) \times \text{count clock cycle}$$

$$\text{Duty cycle [%]} = (\text{TAUJnCDRm (slave)}) / (\text{TAUJnCDRm (master)} + 1) \times 100$$

- Duty cycle = 0%

$\text{TAUJnCDRm (slave)} = 0000\ 0000_H$

- Duty cycle = 100%

$\text{TAUJnCDRm (slave)} \geq \text{TAUJnCDRm (master)} + 1$

### 24.13.1.3 Block Diagram and General Timing Diagram

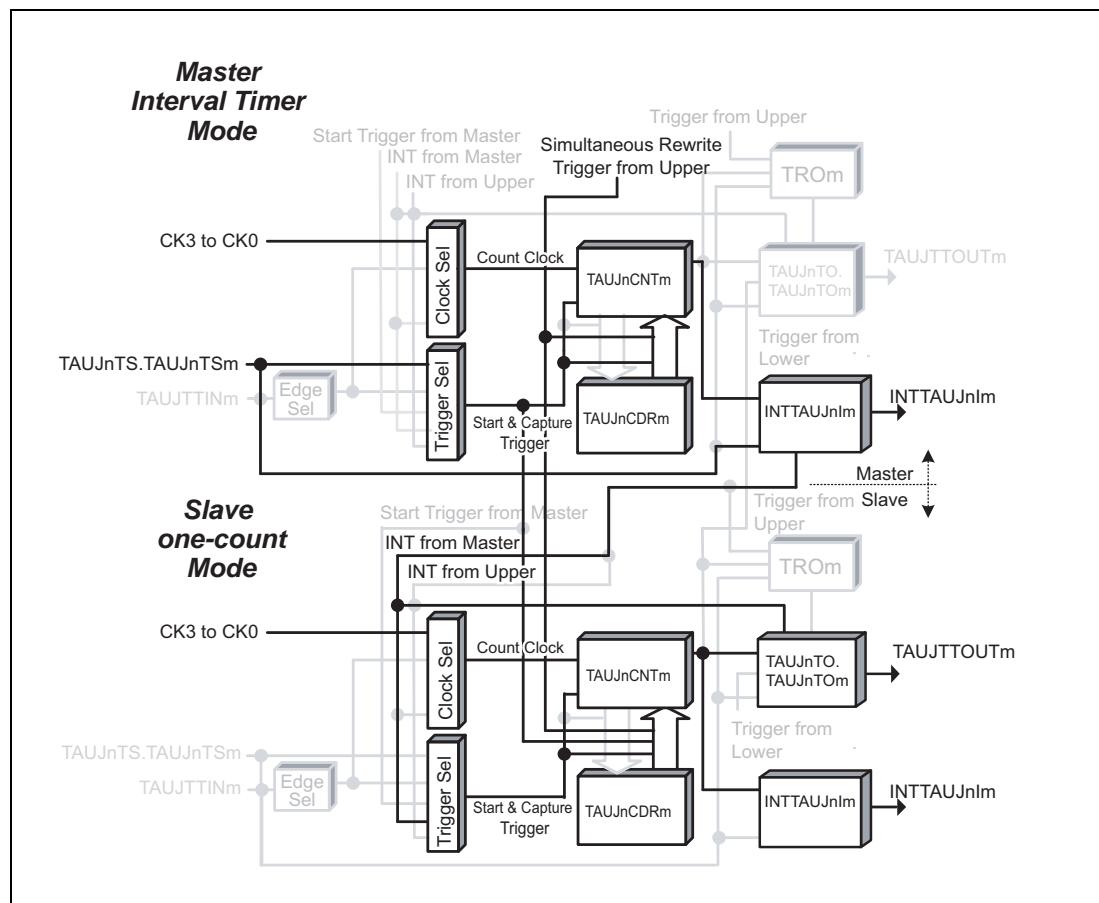


Figure 24.46 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

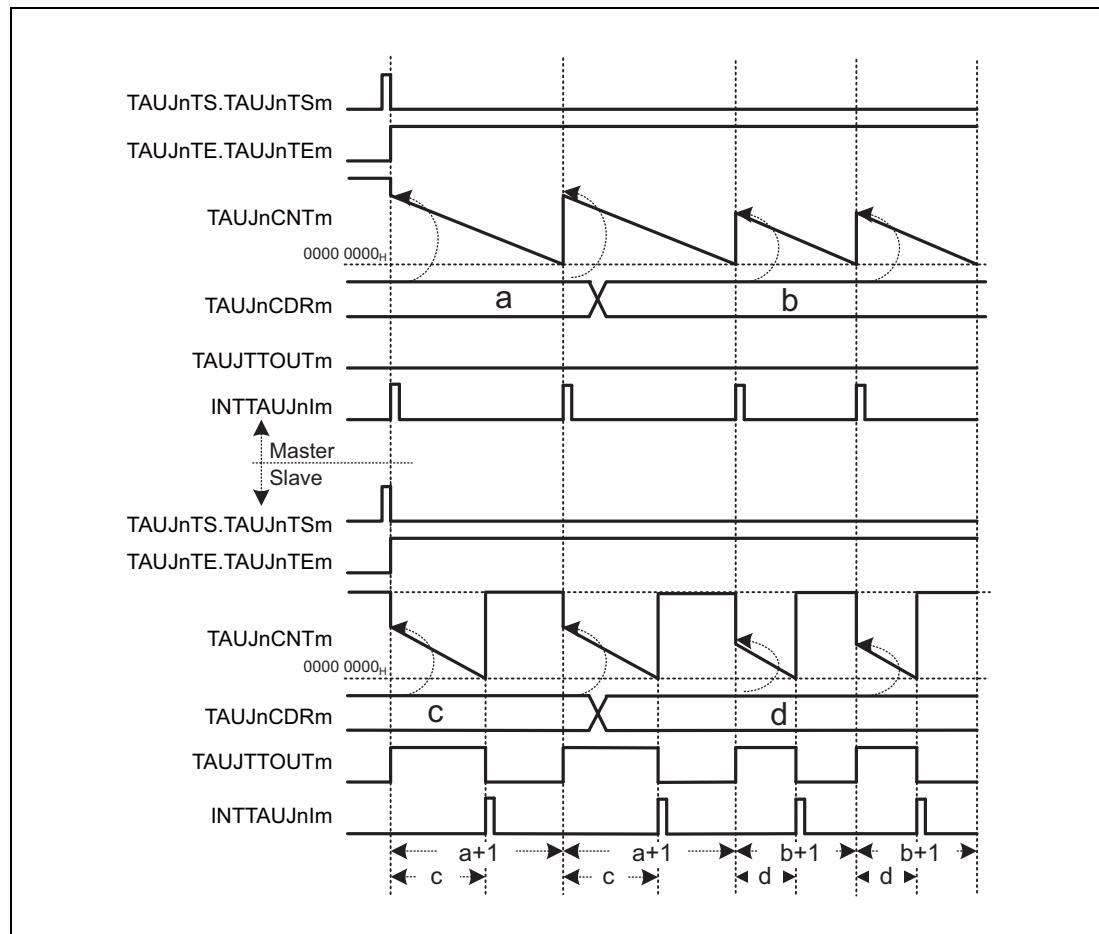


Figure 24.47 General Timing Diagram for PWM Output Function

#### NOTE

- The interval between the starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm + 1.
- TAUJTTOUTm of the slave channel will rise with a delay of one count clock after the rising of INTTAUJnlm of the master channel.

#### 24.13.1.4 Register Settings for the Master Channel

##### (1) TAUJnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	TAUJnMD[4:1]	TAUJnMD0								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.73 Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 1 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 000 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000 <sub>B</sub> .
0	TAUJnMD0	Write 1 <sub>B</sub> .

##### (2) TAUJnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 24.74 Contents of the TAUJnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

##### (3) Channel output mode for the master channel

The channel output mode is not used by this function.

**(4) Simultaneous rewrite for the master channel**

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 24.75 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function**

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMM	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

### 24.13.1.5 Register Settings for the Slave Channel(s)

#### (1) TAUJnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]	TAUJnCCS[1:0]	TAUJnMAS	TAUJnSTS[2:0]	TAUJnCOS[1:0]	—	TAUJnMD[4:1]	TAUJnMD0								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	R	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.76 Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00 <sub>B</sub> .
11	TAUJnMAS	Write 0 <sub>B</sub> .
10 to 8	TAUJnSTS[2:0]	Write 100 <sub>B</sub> .
7, 6	TAUJnCOS[1:0]	Write 00 <sub>B</sub> .
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100 <sub>B</sub> .
0	TAUJnMD0	Write 1 <sub>B</sub> .

#### (2) TAUJnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnTIS[1:0]
Value after reset	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R/W	R/W

Table 24.77 Contents of the TAUJnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

**(3) Channel output mode for the slave channel(s)**

**Table 24.78 Control Bit Settings for Independent Channel Output Mode 1**

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	Write 1 <sub>B</sub> .
TAUJnTOM.TAUJnTOMm	Write 1 <sub>B</sub> .
TAUJnTOC.TAUJnTOCm	Write 0 <sub>B</sub> .
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Negative logic

**(4) Simultaneous rewrite for the slave channel(s)**

The simultaneous rewrite settings of the master and slave channels must be identical.

**Table 24.79 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function**

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

### 24.13.1.6 Operating Procedure for PWM Output Function

Table 24.80 Operating Procedure for PWM Output Function

	Operation	Status of TAUJn
Initial channel setting	<p>Master channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in <b>Section 24.13.1.4, Register Settings for the Master Channel</b>.</p> <p>Slave channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in <b>Section 24.13.1.5, Register Settings for the Slave Channel(s)</b>.</p> <p>Set the values of the TAUJnCDRm registers of all channels.</p>	Channel operation is stopped.
Start operation	<p>Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnlm is generated on the master channel and TAUJTTOUTm (slave) is set.</p>
During operation	<p>TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time.</p> <p>TAUJnRDT.TAUJnRDTm can be changed during operation.</p>	<p>TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches 0000 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUJnlm (master) is generated.</li> <li>TAUJnCNTm (master) loads the TAUJnCDRm value and continues count operation.</li> <li>TAUJnCNTm (slave) loads the TAUJnCDRm value and counts down.</li> <li>TAUJTTOUTm (slave) is set to the active level.</li> </ul> <p>When TAUJnCNTm (slave) reaches 0000 0000H:</p> <ul style="list-style-type: none"> <li>INTTAUJnlm (slave) is generated.</li> <li>TAUJTTOUTm (slave) is set to the inactive level.</li> </ul>
Stop operation	<p>Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.</p>	<p>TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.</p>



### 24.13.1.7 Specific Timing Diagrams

#### (1) Duty cycle = 0%

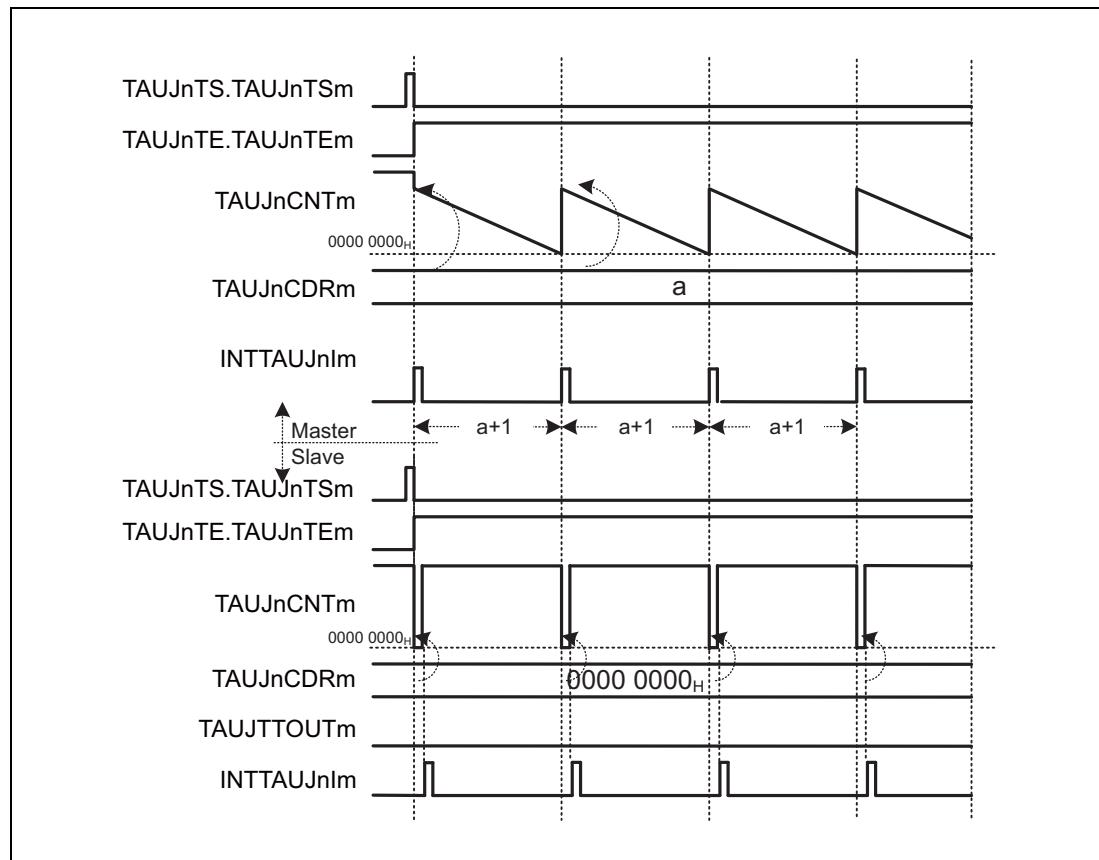
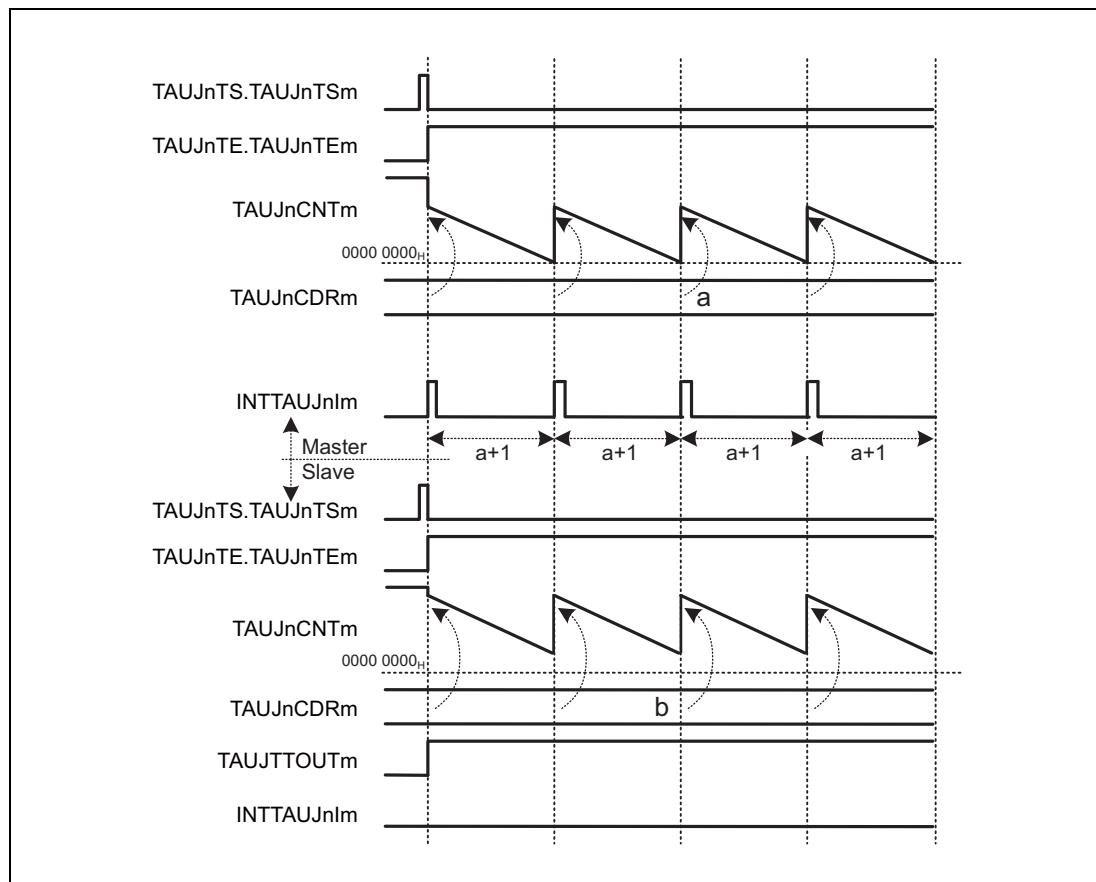


Figure 24.48 TAUJnCDRm (slave) =  $0000\ 0000_H$ , Positive Logic  
 $(TAUJnTOL.TAUJnTOLm (slave) = 0)$

- Every time the master channel generates an interrupt (INTTAUJnIm),  $0000\ 0000_H$  is loaded to TAUJnCNTm (slave). As a result, a slave channel interrupt (INTTAUJnIm) is generated at the same time and TAUJTTOUTm remains inactive.
- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

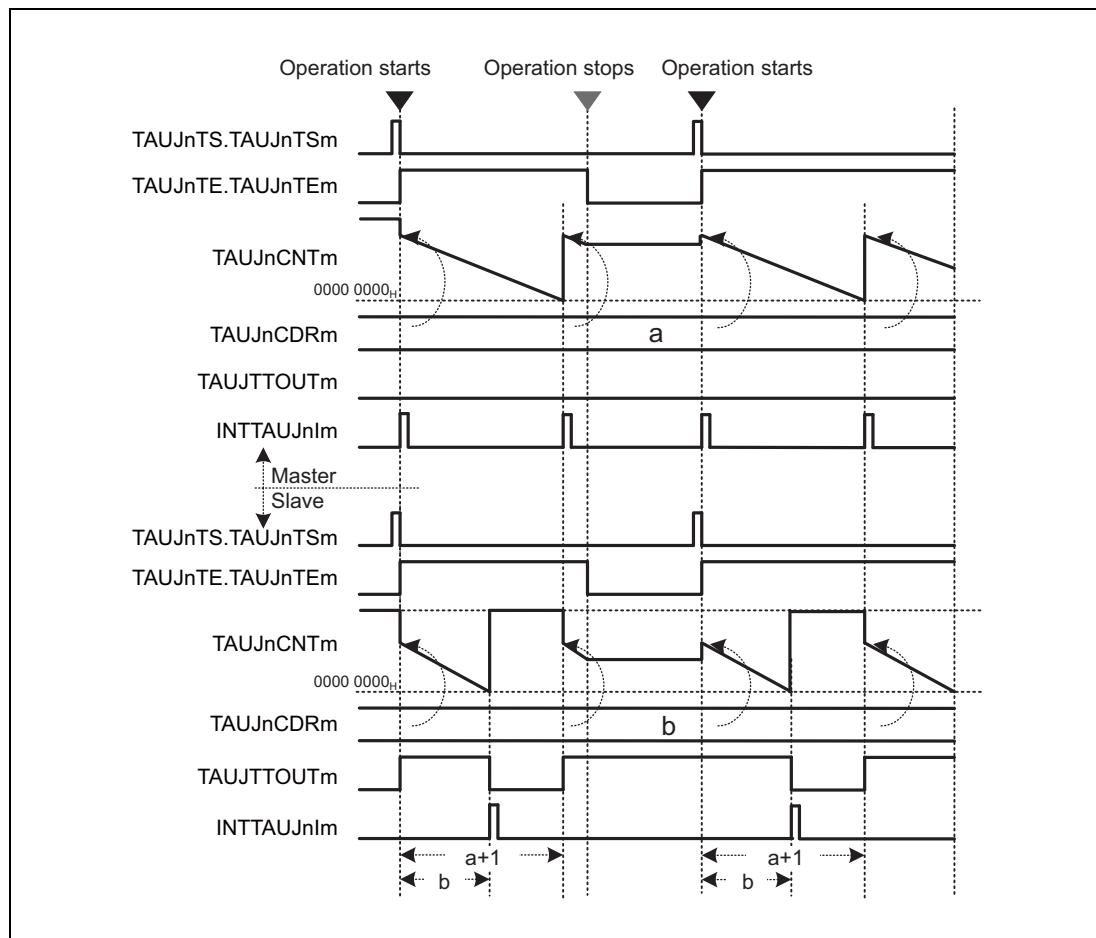
## (2) Duty cycle = 100%



**Figure 24.49 TAUJnCDRm (slave)  $\geq$  TAUJnCDRm (master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)**

If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach 0000 0000<sub>H</sub>. TAUJTTOUT<sub>m</sub> remains active.

### (3) Operation stop and restart



**Figure 24.50 Operation Stop and Restart, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)**

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEM to 0.
- TAUJnCNTm and TAUJTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channels to 1. The TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

## Section 25 Real-Time Clock (RTCA)

This section contains a generic description of the Real-Time Clock (RTCA).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the RTCA.

### 25.1 Features of RH850/F1L RTCA

#### 25.1.1 Number of Units and Channels

This microcontroller has the following number of RTCA units.

Each RTCA unit has one channel RTCA. “Number of channels” is used with the same meaning as “number of units” in this section.

**Table 25.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	—	—	—	—	1	1
Name	—	—	—	—	RTCAn (n = 0)	RTCAn (n = 0)

**Table 25.2 Index**

Index	Description
n	Throughout this section, the individual RTCA units are identified by the index “n” (n = 0); for example, RTCAnCTL0 is the RTCAn control register 0.

#### 25.1.2 Register Base Address

RTCAn base address is listed in the following table.

RTCAn register addresses are given as offsets from the base address.

**Table 25.3 Register Base Address**

Base Address Name	Base Address
<RTCA0_base>	FFE7 8000 <sub>H</sub>

### 25.1.3 Clock Supply

The RTCAn clock supply is shown in the following table.

**Table 25.4 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
RTCA0	RTCATCKI	CKSCLK_ARTCA
	PCLK	CPUCLK2
	Register access clock	CPUCLK2

### 25.1.4 Interrupt Requests

RTCAn interrupt requests are listed in the following table.

**Table 25.5 Interrupt Requests**

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
<b>RTCA0</b>			
RTCATINT1S	1-second interval interrupt	201	—
RTCATINTAL	Alarm interrupt	202	—
RTCATINTR	Fixed interval interrupt	203	—

### 25.1.5 Reset Sources

RTCAn reset sources are listed in the following table. RTCAn is initialized by these reset sources.

**Table 25.6 Reset Sources**

Unit Name	Reset Source
RTCA0	Power-up reset (PURES)

### 25.1.6 External Input/Output Signals

External input/output signals of RTCAn are listed below.

**Table 25.7 External Input/Output Signals**

Unit Signal Name	Description	Alternative Port Pin Signal Name
<b>RTCA0</b>		
RTCAT1HZ	1-Hz pulse output	RTCA0OUT <sup>*1</sup>

Note 1. RTCA0OUT is connected to TAUJ0. For details, see **Section 24, Timer Array Unit J (TAUJ)**.

## 25.2 Overview

### 25.2.1 Functional Overview

The Real-Time Clock (RTCA) has the following features:

- Count clock selection from 32 kHz to 4.194304 MHz
- Counters for years, months, day of the month, day of the week, hours, minutes, seconds, and a sub-counter. The calendar covers 99 years. Leap years are handled by hardware automatically.
- One Hz pulse output function
- Fixed interval interrupt function
- Alarm interrupt function
- Clock error correction function if a 32.768-kHz count clock is used

### 25.2.2 Block Diagram

The block diagram shows the main components of the RTCA.

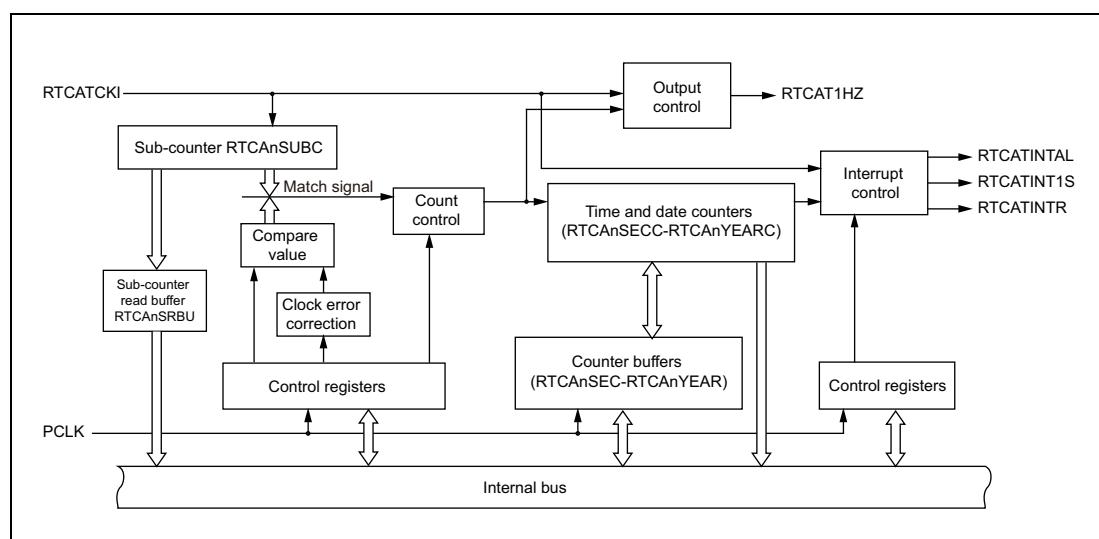


Figure 25.1 Block Diagram of the RTCA

### 25.2.3 Description of Blocks

The Real-Time Clock RTCA provides information about the present time and date and can generate wake-up signals (interrupts, alarms). This information is derived from the count clock RTCATCKI.

#### Sub-counter

RTCATCKI is the input to the sub-counter RTCAnSUBC. The sub-counter counts up from 0 until it reaches the compare value. The compare value is always defined as the frequency of RTCATCKI – 1 (in Hz). Thus, the sub-counter overflows after one second. It is then reset to 0 and triggers the seconds counter RTCAnSECC (and, if specified, the interrupt RTCATINT1S).

The sub-counter can generate a fixed interval interrupt every 0.25 seconds, 0.5 seconds, or 1 second, and a 1-Hz output pulse.

#### Time and date counters

The counters for minutes, hours, day of the week, day of the month, months, and years also count up. They have their own overflow limits. If all the lower counters overflow, the upper counter counts up.

The overflow limit of the counter for the day of the month (RTCAnDAYC) depends on the present month (28, 30, or 31 days) and (in February) on the year counter RTCAnYEARC (years 0, 4, 8, 12, etc. are considered leap years).

The hours counter RTCAnHOURC can be switched between 12- and 24-hour formats.

The counters for seconds, minutes, hours, day of the month, and months can generate a fixed interval interrupt upon overflow (RTCATINTR).

The counters for minutes, hours, and day of the week can also generate an alarm interrupt (RTCATINTAL), e.g. every Tuesday and Thursday at 10:32.

#### Counter buffers

All counters can be read directly at any time. The clock signal used to access the read/write registers and the count clock are usually asynchronous. An overflow of the sub-counter during the read operation can make all read values obsolete. Therefore, reading the counters must be performed using a special procedure. For details, see **Section 25.5.3, Reading Clock Counters**.

For reasons of synchronization, the counters cannot be written directly.

For reading and writing, all counters are accompanied by buffer registers. The buffer registers provide a synchronized way for reading the counters and for setting time and date. When they are used, the operation of the sub-counter must first be suspended and then re-activated (see also **Section 25.5.3, Reading Clock Counters** and **Section 25.5.2, Updating Clock Counters**).

The RTCAnTIMEC and RTCAnCALC registers and their corresponding buffer registers can be used to check and set the time (hours, minutes and seconds) or the date (day of the week, day of the month, month, and year) with one read/write operation.

## 25.3 Registers

### 25.3.1 List of Registers

RTCA registers are listed in the following table.

<RTCAn\_base> is defined in **Section 25.1.2, Register Base Address**.

Table 25.8 List of Registers

Module Name	Register Name	Symbol	Address
<b>Control registers</b>			
RTCAn	Control register 0	RTCAnCTL0	<RTCAn_base> + 00 <sub>H</sub>
RTCAn	Control register 1	RTCAnCTL1	<RTCAn_base> + 04 <sub>H</sub>
RTCAn	Control register 2	RTCAnCTL2	<RTCAn_base> + 08 <sub>H</sub>
<b>Sub-counter registers</b>			
RTCAn	Sub-count register	RTCAnSUBC	<RTCAn_base> + 0C <sub>H</sub>
RTCAn	Sub-count register read buffer	RTCAnSRBU	<RTCAn_base> + 10 <sub>H</sub>
RTCAn	Clock error correction register	RTCAnSUBU	<RTCAn_base> + 38 <sub>H</sub>
RTCAn	Sub-counter compare register	RTCAnSCMP	<RTCAn_base> + 3C <sub>H</sub>
<b>Clock counter and buffer registers</b>			
RTCAn	Seconds count register	RTCAnSECC	<RTCAn_base> + 4C <sub>H</sub>
RTCAn	Seconds count buffer register	RTCAnSEC	<RTCAn_base> + 14 <sub>H</sub>
RTCAn	Minute count register	RTCAnMINC	<RTCAn_base> + 50 <sub>H</sub>
RTCAn	Minute count buffer register	RTCAnMIN	<RTCAn_base> + 18 <sub>H</sub>
RTCAn	Hour count register	RTCAnHOURC	<RTCAn_base> + 54 <sub>H</sub>
RTCAn	Hour count buffer register	RTCAnHOUR	<RTCAn_base> + 1C <sub>H</sub>
RTCAn	Day of the week count register	RTCAnWEEKC	<RTCAn_base> + 58 <sub>H</sub>
RTCAn	Day of the week count buffer register	RTCAnWEEK	<RTCAn_base> + 20 <sub>H</sub>
RTCAn	Day count register	RTCAnDAYC	<RTCAn_base> + 5C <sub>H</sub>
RTCAn	Day count buffer register	RTCAnDAY	<RTCAn_base> + 24 <sub>H</sub>
RTCAn	Month count register	RTCAnMONC	<RTCAn_base> + 60 <sub>H</sub>
RTCAn	Month count buffer register	RTCAnMONTH	<RTCAn_base> + 28 <sub>H</sub>
RTCAn	Year count register	RTCAnYEARC	<RTCAn_base> + 64 <sub>H</sub>
RTCAn	Year count buffer register	RTCAnYEAR	<RTCAn_base> + 2C <sub>H</sub>
<b>Special counter and buffer registers</b>			
RTCAn	Time count register	RTCAnTIMEC	<RTCAn_base> + 68 <sub>H</sub>
RTCAn	Time count buffer register	RTCAnTIME	<RTCAn_base> + 30 <sub>H</sub>
RTCAn	Calendar count register	RTCAnCALC	<RTCAn_base> + 6C <sub>H</sub>
RTCAn	Calendar count buffer register	RTCAnCAL	<RTCAn_base> + 34 <sub>H</sub>
<b>Alarm time setting registers</b>			
RTCAn	Alarm minute setting register	RTCAnALM	<RTCAn_base> + 40 <sub>H</sub>
RTCAn	Alarm hour setting register	RTCAnALH	<RTCAn_base> + 44 <sub>H</sub>
RTCAn	Alarm day of the week setting register	RTCAnALW	<RTCAn_base> + 48 <sub>H</sub>
<b>Emulation register</b>			
RTCAn	Emulation register	RTCAnEMU	<RTCAn_base> + 74 <sub>H</sub>

## 25.3.2 Details of RTCA Control Registers

### 25.3.2.1 RTCAnCTL0 — RTCA Control Register 0

This register controls the count operation of the sub-counter RTCAnSUBC, the format (12-hour/24-hour) of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH, and the operation mode.

**Access:** This register can be read or written in 8-bit or 1-bit units.

**Address:** <RTCAn\_base> + 00H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	RTCAnCE	RTCAnCEST	RTCAnAMPM	RTCAnSLSB	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R	R

**Table 25.9 RTCAnCTL0 Register Contents**

Bit Position	Bit Name	Function
7	RTCAnCE	Starts/stops the sub-counter RTCAnSUBC operation. 0: Stops the sub-counter operation. All output pins and all status flags in control register RTCAnCTL2 are cleared. 1: Starts the sub-counter operation. The sub-counter counts up.
6	RTCAnCEST	Indicates the operation enabled/stopped status of the sub-counter: 0: Operation stopped status 1: Operation enabled status For details on how to use this status flag, see <b>Section 25.5.1, Initial Setting of the RTCA</b> .
5	RTCAnAMPM	Selects the format of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH: 0: 12-hour format (1 to 12, am/pm) 1: 24-hour format (0 to 23, military time) For details on the format, see <b>Table 25.21, 12- and 24-Hour Format</b> .
4	RTCAnSLSB	Selects the operation mode: 0: 32.768 kHz mode 1: Frequency selection mode For details on the operation modes, see <b>Section 25.4, Operation</b> . The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1). For details on the initialization of RTCAn, see <b>Section 25.5.1, Initial Setting of the RTCA</b> .
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

### 25.3.2.2 RTCAnCTL1 — RTCA Control Register 1

This register controls the interrupt request generation and the 1-Hz pulse output.

**Access:** This register can be read or written in 8-bit or 1-bit units.

**Address:** <RTCAn\_base> + 04H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnEN1HZ	RTCAnENALM	RTCAnEN1S	RTCAnCT2	RTCAnCT1	RTCAnCT0
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.10 RTCAnCTL1 Register Contents

Bit Position	Bit Name	Function																														
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																														
5	RTCAnEN1HZ	Enables/stops 1-Hz pulse output (RTCAT1HZ): 0: RTCAT1HZ disabled (RTCAT1HZ is fixed to 0) 1: RTCAT1HZ enabled																														
4	RTCAnENALM	Enables/disables alarm interrupt request generation (RTCATINTAL): 0: RTCATINTAL disabled 1: RTCATINTAL enabled																														
3	RTCAnEN1S	Enables/disables 1-second interrupt request generation (RTCATINT1S): 0: RTCATINT1S disabled 1: RTCATINT1S enabled																														
2 to 0	RTCAnCT[2:0]	Specifies the fixed interval interrupt request (RTCATINTR) setting:																														
		<table border="1"> <thead> <tr> <th colspan="3">RTCATINTR Interrupt Request Generation</th> </tr> <tr> <th>RTCAnCT[2:0]</th> <th>Interval</th> <th>Timing</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>No interrupt request generation</td> <td></td> </tr> <tr> <td>001</td> <td>Every 0.25 seconds</td> <td>Every 0.25, 0.5, 0.75 and 1 second</td> </tr> <tr> <td>010</td> <td>Every 0.5 seconds</td> <td>Every 0.5 and 1 second</td> </tr> <tr> <td>011</td> <td>Every second</td> <td>Every 1 second</td> </tr> <tr> <td>100</td> <td>Every minute</td> <td>Every 1 minute 00 seconds</td> </tr> <tr> <td>101</td> <td>Every hour</td> <td>Every 1 hour 00 minutes 00 seconds</td> </tr> <tr> <td>110</td> <td>Every day</td> <td>Every 1 day 00 hours 00 minutes 00 seconds (i.e., every midnight)</td> </tr> <tr> <td>111</td> <td>Every month</td> <td>Every 1 month first day 00 hours 00 minutes 00 seconds (i.e., every first midnight of a month)</td> </tr> </tbody> </table>	RTCATINTR Interrupt Request Generation			RTCAnCT[2:0]	Interval	Timing	000	No interrupt request generation		001	Every 0.25 seconds	Every 0.25, 0.5, 0.75 and 1 second	010	Every 0.5 seconds	Every 0.5 and 1 second	011	Every second	Every 1 second	100	Every minute	Every 1 minute 00 seconds	101	Every hour	Every 1 hour 00 minutes 00 seconds	110	Every day	Every 1 day 00 hours 00 minutes 00 seconds (i.e., every midnight)	111	Every month	Every 1 month first day 00 hours 00 minutes 00 seconds (i.e., every first midnight of a month)
RTCATINTR Interrupt Request Generation																																
RTCAnCT[2:0]	Interval	Timing																														
000	No interrupt request generation																															
001	Every 0.25 seconds	Every 0.25, 0.5, 0.75 and 1 second																														
010	Every 0.5 seconds	Every 0.5 and 1 second																														
011	Every second	Every 1 second																														
100	Every minute	Every 1 minute 00 seconds																														
101	Every hour	Every 1 hour 00 minutes 00 seconds																														
110	Every day	Every 1 day 00 hours 00 minutes 00 seconds (i.e., every midnight)																														
111	Every month	Every 1 month first day 00 hours 00 minutes 00 seconds (i.e., every first midnight of a month)																														

If the settings of RTCAnCT[2:0] are changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to RTCATINTR. Implement appropriate interrupt mask processing procedures.

### 25.3.2.3 RTCAnCTL2 — RTCA Control Register 2

This register contains status information and controls the data transfer from the sub-counter RTCAnSUBC to the dedicated sub-counter read buffer RTCAnSRBU and the operation setting of the clock counters (RTCAnSECC to RTCAnYEARC).

<b>Access:</b>	This register can be read or written in 8-bit or 1-bit units.							
<b>Address:</b>	<RTCAn_base> + 08H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnWUST	RTCAnWSST	RTCAnRSST	RTCAnRSUB	RTCAnWST	RTCAnWAIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W

**Table 25.11 RTCAnCTL2 Register Contents (1/2)**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	RTCAnWUST	<p>Indicates whether RTCAnSUBU write operation has been completed: 0: RTCAnSUBU write completed 1: RTCAnSUBU write in progress</p> <p>The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSUBU is completed, this bit is set to 1.</p> <p>See <a href="#">Section 25.5.5, Writing to RTCAnSUBU</a>, for details.</p>
4	RTCAnWSST	<p>Indicates whether RTCAnSCMP write operation has been completed: 0: RTCAnSCMP write completed 1: RTCAnSCMP write in progress</p> <p>The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSCMP is completed, this bit is set to 1.</p> <p>See <a href="#">Section 25.5.6, Writing to RTCAnSCMP</a>, for details.</p>
3	RTCAnRSST	<p>Indicates whether the value of the sub-counter (RTCAnSUBC) has been transferred to the sub-count register read buffer (RTCAnSRBU): 0: Transfer in progress, or waiting for a transfer trigger 1: Transfer completed</p> <p>This bit is cleared (transfer is triggered) by RTCAnRSUB=1. This bit is automatically set when the transfer is completed.</p> <p>See <a href="#">Section 25.5.4, Reading RTCAnSRBU</a>, for details.</p>
2	RTCAnRSUB	<p>Triggers transfer of the value of the sub-counter (RTCAnSUBC) to the dedicated read buffer (RTCAnSRBU) or clears the transfer state of the sub-counter: 0: Transfer status (RTCAnRSST) is cleared. 1: Transfer is triggered.</p> <p>This bit is used to read the value of RTCAnSRBU when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1). The value of RTCAnSUBC is synchronized with RTCATCKI and loaded to RTCAnSRBU.</p> <p>For details, see <a href="#">Section 25.5.4, Reading RTCAnSRBU</a>.</p>

**Table 25.11 RTCAnCTL2 Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
1	RTCAnWST	<p>Indicates the status of all clock counters (RTCAnSECC to RTCAnYEARC):</p> <p>0: All clock counters are running. 1: All clock counters are stopped The sub-counter is still running.</p> <p>The clock counters must be stopped before reading or writing clock counter values during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). To stop the clock counters, set RTCAnWAIT = 1.</p>
0	RTCAnWAIT	<p>Restarts/stops all clock counters (RTCAnSECC to RTCAnYEARC):</p> <p>0: Restarts all clock counters either immediately or immediately after the clock counter write operation finishes. 1: Stops all clock counters temporarily. The sub-counter is still running.</p> <p>The clock counters must be stopped before reading or writing counter buffers during sub-counter operation (RTCAnCTL0.RTCAnCE = 1).</p> <p><b>CAUTION</b></p> <p>Only one overflow can be held internally. When two overflows occur, the seconds counter is incremented only by one when it is restarted. Thus, the procedure must be completed within one second.</p>

### 25.3.3 Details of RTCA Sub-Counter Registers

#### 25.3.3.1 RTCAnSUBC — RTCA Sub-Count Register

This counter counts the 1-second reference time. It operates using the count clock RTCATCKI.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <RTCAn\_base> + 0C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

This register is initialized:

- When write operation is performed to the seconds count buffer register (RTCAnSEC) or to the time count buffer register (RTCAnTIME) and the value is reflected to the seconds count register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA nSUBC[21:16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCA nSUBC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.12 RTCA nSUBC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCA nSUBC [21:0]	Sub-counter value The sub-counter only operates while RTCAnCTL0.RTCAnCEST = 1.

#### NOTES

- This sub-counter operates with RTCATCKI while the read operation is clocked by PCLK. Reading this sub-counter during operation (RTCAnCTL0.RTCAnCEST = 1) is asynchronous to RTCATCKI and can lead to wrong results.  
Use the sub-count register read buffer (RTCAnSRBU) to read the sub-counter value during operation.  
For details, see **Section 25.5.4, Reading RTCAnSRBU**.
- The count-operation of this sub-counter depends on the selected operation mode. See **Section 25.4, Operation**, for details.

### 25.3.3.2 RTCAnSRBU — RTCA Sub-Count Register Read Buffer

This register is the read buffer for the sub-counter RTCAnSUBC.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <RTCAn\_base> + 10H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCAAnSRBU[21:16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAAnSRBU[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 25.13 RTCAAnSRBU Register Contents**

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAAnSRBU [21:0]	Sub-counter value at the time of the last RTCAAnSUBC read. When RTCACTL2.RTCAAnRSUB is set to 1, the value of the RTCAAnSUBC is loaded to the read buffer in synchronization with RTCATCKI.

#### NOTE

Perform RTCAAnSRBU read according to the flow described in **Section 25.5.4, Reading RTCAAnSRBU**.

### 25.3.3.3 RTCAnSUBU — RTCA Clock Error Correction Register

This register enables and specifies clock error correction. This register only applies in 32.768-kHz mode (RTCAnCTL0.RTCAnSLSB = 0).

For details on clock error correction, see **Section 25.4.4, Clock Error Correction**.

**Access:** This register can be read or written in 8-bit units.

Note the following when writing this register during sub-counter operation:

- Previous RTCAnSUBU write must be completed (RTCAnCTL2.RTCAnWUST = 0).
- The write operation ends with the next sub-counter overflow.

**Address:** <RTCAn\_base> + 38<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	RTCAnDEV	RTCAnF6			RTCAnF[5:0]			
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Value after reset	0	0	0	0	0	0	0	0

**Table 25.14 RTCAnSUBU Register Contents**

Bit Position	Bit Name	Function
7	RTCAnDEV	Specifies how often clock error correction is performed per minute: 0: Three times every minute (when RTCAnSECC equals 00, 20, and 40) 1: Once every minute (when RTCAnSECC equals 00)
6	RTCAnF6	Specifies whether the sub-counter value is incremented or decremented: 0: Incremented (+ correction) Incrementation value = (RTCAnF[5:0] value – 1) × 2 1: Decrementated (– correction) Decrementation value = (inverted data of RTCAnF[5:0] value + 1) × 2
5 to 0	RTCAnF[5:0]	Error correction value

#### NOTES

1. When RTCAnF[5:1] = 00000<sub>B</sub>, clock error correction is not performed.
2. Perform RTCAnSUBU write as described in
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.5, Writing to RTCAnSUBU**.

### 25.3.3.4 RTCAnSCMP — RTCA Sub-Counter Compare Register

This register sets the compare value of the sub-counter RTCAnSUBC in frequency selection mode (RTCAnCTL0.RTCAnSLSB = 1).

When the sub-counter values matches the value of this register, an overflow signal is output to the seconds counter RTCAnSECC and the sub-counter is cleared.

Set the value for this register according to the frequency of the input clock RTCATCKI.

**Access:** This register can be read or written in 32-bit units.

Note the following when writing this register during sub-counter operation:

- Previous RTCAnSCMP write must be completed (RTCAnCTL2.RTCAnWSST = 0).
- The write operation ends with the next sub-counter overflow.

**Address:** <RTCAn\_base> + 3CH

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA nSCMP[21:16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCA nSCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.15 RTCA nSCMP Register Contents**

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 0	RTCA nSCMP [21:0]	Sub-counter compare value in frequency selection mode.

#### Example

The following example illustrates the setting of RTCA nSCMP:

- RTCATCKI = 4 MHz = 4,000,000 Hz
- RTCA nSCMP = 4,000,000 – 1 = 3,999,999 (decimal code) = 3D08FFH
- The seconds counter RTCA nSECC is triggered when the sub-counter value changes from 3D08FFH to 0H.

#### NOTES

1. The operation of the RTCA cannot be guaranteed if a value of 3198 (decimal code) or lower is set in this register.
2. Perform RTCA nSCMP write as described in **Section 25.5.1, Initial Setting of the RTCA** and **Section 25.5.6, Writing to RTCA nSCMP**.

## 25.3.4 Details of RTCA Clock Counter and Buffer Registers

### 25.3.4.1 RTCAnSECC — RTCA Seconds Count Register

This register is the seconds counter. It counts seconds from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the sub-counter RTCAnSUBC.

If the sub-counter overflows while the seconds counter is stopped

(RTCAnCTL2.RTCAnWST = 1), the seconds counter behaves as follows:

- If one sub-counter overflow occurs while the seconds counter is stopped, the overflow is held internally.

The seconds counter is incremented by one when it is restarted.

- If two or more overflows occur while the seconds counter is stopped, the overflow count cannot be held internally.

The seconds counter is incremented by one when it is restarted.

- If the seconds counter was updated while the seconds counter is stopped, the sub-counter overflow(s) are ignored.

- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the minutes counter (RTCAnMINC).

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 4CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0	
	—	RTCAnSECC[6:0]							
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	

Table 25.16 RTCAnSECC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

#### NOTES

- Perform RTCAnSECC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
- A start value can be assigned to this register by writing to the seconds count buffer register RTCAnSEC or to the clock time setting register RTCAnTIME. See
  - Section 25.5.1, Initial Setting of the RTCA**, and
  - Section 25.5.2, Updating Clock Counters**

### 25.3.4.2 RTCAnSEC — RTCA Seconds Count Buffer Register

This register is a buffer register to read/write the seconds counter RTCAnSECC.

**Access:** This register can be read or written in 8-bit units.

**Address:** <RTCAn\_base> + 14<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0	
	—	RTCAnSEC[6:0]							
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

**Table 25.17 RTCAnSEC Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

#### NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnSEC read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters, and**
  - **Section 25.5.3, Reading Clock Counters.**

### 25.3.4.3 RTCAnMINC — RTCA Minutes Count Register

This register is the minutes counter. It counts minutes from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the seconds counter RTCAnSECC.
- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the hours counter (RTCAnHOURC).

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 50<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0	
	—	RTCAnMINC[6:0]							
Value after reset	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	

**Table 25.18 RTCAnMINC Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnMINC	Minutes in BCD [6:0]

#### NOTES

1. Perform RTCAnMINC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the minutes count buffer register RTCAnMIN or to the time count buffer register RTCAnTIME. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

#### 25.3.4.4 RTCAnMIN — RTCA Minutes Count Buffer Register

This register is a buffer register to read/write the minutes counter RTCAnMINC.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RTCAn_base> + 18H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.19 RTCAnMIN Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnMIN [6:0]	Minutes in BCD

#### NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnMIN read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters, and**
  - **Section 25.5.3, Reading Clock Counters.**

### 25.3.4.5 RTCAnHOURC — RTCA Hours Count Register

This register is the hours counter. It counts the hours in BCD. The count range depends on the selected hour format. See **Table 25.21, 12- and 24-Hour Format**.

This register counts as follows.

- It is triggered by every overflow of the minutes counter RTCAnMINC.
- It outputs an overflow signal when the value changes from 23 to 00 (in 24-hour format) or from 31 to 12 (in 12-hour format). The overflow signal triggers two counters:
  - Day of the week counter (RTCAnWEEKC)
  - Day of the month counter (RTCAnDAYC)

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 54<sub>H</sub>

**Value after reset:** 12<sub>H</sub>

Bit	7	6	5	4	3	2	1	0		
	—	—	RTCAnHOURC[5:0]							
Value after reset	0	0	0	1	0	0	1	0		
R/W	R	R	R	R	R	R	R	R		

**Table 25.20 RTCAnHOURC Register Contents**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnHOURC [5:0]	Hours in BCD. See <b>Table 25.21, 12- and 24-Hour Format</b> , for details.

#### NOTES

1. Perform RTCAnHOURC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the hours count buffer register RTCAnHOUR or to the time count buffer register RTCAnTIME. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 12- or 24-hour format

The count values of RTCAnHOURC depend on the selected hour format.

If 12-hour format is selected (RTCAnCTL0.RTCAnAMPM = 0), bit 5 in the RTCAnHOURC register is the am/pm indicator:

- RTCAnHOURC[5] = 0: am
- RTCAnHOURC[5] = 1: pm

The following table shows the count range of RTCAnHOURC in both 12- and 24-hour format.

**Table 25.21 12- and 24-Hour Format**

12-Hour Format (RTCAnAMPM = 0)		24-Hour Format (RTCAnAMPM = 1)	
Time	RTCAnHOURC	Time	RTCAnHOURC
0 am	12 <sub>H</sub>	0	00 <sub>H</sub>
1 am	01 <sub>H</sub>	1	01 <sub>H</sub>
2 am	02 <sub>H</sub>	2	02 <sub>H</sub>
3 am	03 <sub>H</sub>	3	03 <sub>H</sub>
4 am	04 <sub>H</sub>	4	04 <sub>H</sub>
5 am	05 <sub>H</sub>	5	05 <sub>H</sub>
6 am	06 <sub>H</sub>	6	06 <sub>H</sub>
7 am	07 <sub>H</sub>	7	07 <sub>H</sub>
8 am	08 <sub>H</sub>	8	08 <sub>H</sub>
9 am	09 <sub>H</sub>	9	09 <sub>H</sub>
10 am	10 <sub>H</sub>	10	10 <sub>H</sub>
11 am	11 <sub>H</sub>	11	11 <sub>H</sub>
0 pm	32 <sub>H</sub>	12	12 <sub>H</sub>
1 pm	21 <sub>H</sub>	13	13 <sub>H</sub>
2 pm	22 <sub>H</sub>	14	14 <sub>H</sub>
3 pm	23 <sub>H</sub>	15	15 <sub>H</sub>
4 pm	24 <sub>H</sub>	16	16 <sub>H</sub>
5 pm	25 <sub>H</sub>	17	17 <sub>H</sub>
6 pm	26 <sub>H</sub>	18	18 <sub>H</sub>
7 pm	27 <sub>H</sub>	19	19 <sub>H</sub>
8 pm	28 <sub>H</sub>	20	20 <sub>H</sub>
9 pm	29 <sub>H</sub>	21	21 <sub>H</sub>
10 pm	30 <sub>H</sub>	22	22 <sub>H</sub>
11 pm	31 <sub>H</sub>	23	23 <sub>H</sub>

↓  
pm indicator in 12-hour format:  
RTCAnHOUR.RTCAnHOUR[5] = 1

### 25.3.4.6 RTCAnHOUR — RTCA Hours Count Buffer Register

This register is a buffer register to read/write the hours counter RTCAnHOURC.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RTCAn_base> + 1C <sub>H</sub>							
<b>Value after reset:</b>	12 <sub>H</sub>							
Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.22 RTCAnHOUR Register Contents**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnHOUR [5:0]	Hours in BCD See <b>Table 25.21, 12- and 24-Hour Format</b> , for details.

#### NOTES

1. When writing this register, only the following decimal values in BCD are allowed:
  - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0):  
01 to 12 or 21 to 32
  - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1):  
00 to 23
2. Perform RTCAnHOUR read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters,** and
  - **Section 25.5.3, Reading Clock Counters.**

### 25.3.4.7 RTCAnWEEKC — RTCA Day of the Week Count Register

This register is the day of the week counter. It counts from 0 to 6.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 58<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RTCAWEEKC[2:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 25.23 RTCAWEEKC Register Contents**

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAWEEKC [2:0]	Day of the week

#### NOTES

1. Perform RTCAWEEKC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day of the week count buffer register RTCAWEEK or to the calendar count buffer register RTCACAL. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 25.3.4.8 RTCAnWEEK — RTCA Day of the Week Count Buffer Register

This register is a buffer register to read/write the day of the week counter RTCAnWEEKC.

There is no particular correspondence between the value of RTCAnWEEK and the day of the week. Set the correspondence according to the application to be used.

Example: 0 = Sunday, 1 = Monday, ..., 6 = Saturday

**Access:** This register can be read or written in 8-bit units.

**Address:** <RTCAn\_base> + 20H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTCAWEEK[2:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

**Table 25.24 RTCAWEEK Register Contents**

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	RTCAWEEK [2:0]	Day of the week

#### NOTES

1. When writing this register, only decimal values between 0 and 6 in BCD are allowed.
2. Perform RTCAWEEK read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters, and**
  - **Section 25.5.3, Reading Clock Counters.**

### 25.3.4.9 RTCAnDAYC — RTCA Day of the Month Count Register

This register is the day of the month counter. It counts from 01 to a maximum of 31 in BCD, depending on the value of the month counter (RTCAnMONC) and the year counter (RTCAnYEARC):

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, non-leap year)

Years 0, 4, 8, 12, etc., are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.
- It outputs an overflow signal when the value changes from 28, 29, 30, or 31 to 01, depending on the current month and year. The overflow signal triggers the month counter (RTCAnMONC).

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 5CH

**Value after reset:** 01H

Bit	7	6	5	4	3	2	1	0		
	—	—	RTCAnDAYC[5:0]							
Value after reset	0	0	0	0	0	0	0	1		
R/W	R	R	R	R	R	R	R	R		

**Table 25.25 RTCAnDAYC Register Contents**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnDAYC [5:0]	Day of the month in BCD

#### NOTES

1. Perform RTCAnDAYC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day of the month count buffer register RTCAnDAY or to the calendar count buffer register RTCAnCAL. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 25.3.4.10 RTCAnDAY — RTCA Day of the Month Count Buffer Register

This register is a buffer register to read/write the day of the month counter RTCAnDAYC.

**Access:** This register can be read or written in 8-bit units.

**Address:** <RTCAn\_base> + 24H

**Value after reset:** 01H

Bit	7	6	5	4	3	2	1	0
	—	—						RTCAAnDAY[5:0]
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.26 RTCAAnDAY Register Contents**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAAnDAY [5:0]	Day of the month in BCD

#### NOTES

1. When writing this register, only decimal values between 01 and 31 in BCD are allowed:
  - 01 to 31 (January, March, May, July, August, October, December)
  - 01 to 30 (April, June, September, November)
  - 01 to 29 (February, leap year)
  - 01 to 28 (February, non-leap year)
2. Perform RTCAAnDAY read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters, and**
  - **Section 25.5.3, Reading Clock Counters.**

### 25.3.4.11 RTCAnMONC — RTCA Month Count Register

This register is the month counter. It counts the month of the year, starting from 01 to 12 in BCD.

This register counts as follows.

- It is triggered by every overflow of the counter for the day of the month RTCAnDAYC.
- It outputs an overflow signal when the value changes from 12 to 01. The overflow signal triggers the year counter (RTCAnYEARC).

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 60<sub>H</sub>

**Value after reset:** 01<sub>H</sub>

Bit	7	6	5	4	3	2	1	0			
	—	—	—	RTCAnMONC[4:0]							
Value after reset	0	0	0	0	0	0	0	1			
R/W	R	R	R	R	R	R	R	R			

**Table 25.27 RTCAnMONC Register Contents**

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	RTCAnMONC	Month of the year in BCD [4:0]

#### NOTES

1. Perform RTCAnMONC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the month count buffer register RTCAnMONTH or to the calendar count buffer register RTCAnCAL. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 25.3.4.12 RTCAnMONTH — RTCA Month Count Buffer Register

This register is a buffer register to read/write the month counter RTCAnMONC.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RTCAn_base> + 28H							
<b>Value after reset:</b>	01H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

**Table 25.28 RTCAnMONTH Register Contents**

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	RTCAnMONTH	Month of the year in BCD [4:0]

#### NOTES

1. When writing this register, only decimal values between 01 and 12 in BCD are allowed.
2. Perform RTCAnMONTH read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters, and**
  - **Section 25.5.3, Reading Clock Counters.**

### 25.3.4.13 RTCAnYEARC — RTCA Year Count Register

This register is the year counter. It counts years from 00 to a maximum of 99 in BCD.

Years 00, 04, 08, ..., 92, and 96 (every four years) are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the month counter RTCAnMONC.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <RTCAn\_base> + 64<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
RTCAnYEARC[7:0]								
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 25.29 RTCAnYEARC Register Contents**

Bit Position	Bit Name	Function
7 to 0	RTCAnYEARC [7:0]	Year in BCD

#### NOTES

1. Perform RTCAnYEARC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the year count buffer register RTCAnYEAR or to the calendar count buffer register RTCAnCAL. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 25.3.4.14 RTCAnYEAR — RTCA Year Count Buffer Register

This register is a buffer register to read/write the year counter RTCAnYEARC.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RTCAn_base> + 2CH							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	RTCAAnYEAR[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.30 RTCAAnYEAR Register Contents**

Bit Position	Bit Name	Function
7 to 0	RTCAAnYEAR [7:0]	Year in BCD

#### NOTES

1. When writing this register, only decimal values between 00 and 99 in BCD are allowed.
2. Perform RTCAAnYEAR read/write as described in
  - **Section 25.5.1, Initial Setting of the RTCA,**
  - **Section 25.5.2, Updating Clock Counters,** and
  - **Section 25.5.3, Reading Clock Counters.**

## 25.3.5 Details of RTCA Special Counter and Buffer Registers

### 25.3.5.1 RTCAnTIMEC — RTCA Time Count Register

This register enables the RTCAnHOURC, RTCAnMINC, and RTCAnSECC counters to be read simultaneously.

Access:	This register is a read-only register that can be read in 32-bit units.															
Address:	<RTCAn_base> + 68H															
Value after reset:	0012 0000H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]						—	RTCAnSECC[6:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 25.31 RTCAnTIMEC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	RTCAnHOURC [5:0]	Hours in BCD. See <b>Table 25.21, 12- and 24-Hour Format</b> , for details.
15	Reserved	When read, the value after reset is returned.
14 to 8	RTCAnMINC [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

#### NOTES

1. Perform RTCAnTIMEC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the time count buffer register RTCAnTIME. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 25.3.5.2 RTCAnTIME — RTCA Time Count Buffer Register

This register enables the RTCAnHOUR, RTCAnMIN, and RTCAnSEC buffer registers to be read/written simultaneously.

**Access:** This register can be read or written in 32-bit units.

**Address:** <RTCAn\_base> + 30<sub>H</sub>

**Value after reset:** 0012 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—		RTCAnHOUR[5:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]						—		RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.32 RTCAnTIME Register Contents**

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	RTCAnHOUR [5:0]	Hours in BCD See <b>Table 25.21, 12- and 24-Hour Format</b> , for details.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	RTCAnMIN [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

#### NOTE

Perform RTCAnTIME read/write as described in

- **Section 25.5.1, Initial Setting of the RTCA,**
- **Section 25.5.2, Updating Clock Counters, and**
- **Section 25.5.3, Reading Clock Counters.**

### 25.3.5.3 RTCAnCALC — RTCA Calendar Count Register

This register enables the RTCAnYEARC, RTCAnMONC, RTCAnDAYC, and RTCAnWEEKC counters to be read simultaneously.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <RTCAn\_base> + 6C<sub>H</sub>

**Value after reset:** 0001 0100<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTCAAnYEARC[7:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
RTCAAnMONC[4:0]																
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCAAnDAYC[5:0]																
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.33 RTCAAnCALC Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAAnYEARC [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned.
20 to 16	RTCAAnMONC	Month of the year in BCD [4:0]
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RTCAAnDAYC [5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAAnWEEKC [2:0]	Day of the week in BCD

#### NOTES

1. Perform RTCAAnCALC read according to the flow described in **Section 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the clock time setting register RTCAAnCAL. See
  - **Section 25.5.1, Initial Setting of the RTCA**, and
  - **Section 25.5.2, Updating Clock Counters**.

### 25.3.5.4 RTCAnCAL — RTCA Calendar Count Buffer Register

This register enables the RTCAnYEAR, RTCAnMONTH, RTCAnDAY, and RTCAnWEEK buffer registers to be read/written simultaneously.

**Access:** This register can be read or written in 32-bit units.

**Address:** <RTCAn\_base> + 34H

**Value after reset:** 0001 0100H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEAR[7:0]										—	—	—	RTCAnMONTH[4:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]						—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 25.34 RTCAnCAL Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEAR [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	RTCAnMONTH [4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	RTCAnDAY [5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of the week in BCD

#### NOTE

Perform RTCAnCAL read/write as described in

- **Section 25.5.1, Initial Setting of the RTCA.**
- **Section 25.5.2, Updating Clock Counters, and**
- **Section 25.5.3, Reading Clock Counters.**

## 25.3.6 Details of RTCA Alarm Setting Registers

### 25.3.6.1 RTCAnALM — RTCA Alarm Minute Setting Register

This register specifies the minute of the alarm interrupt.

For details and example settings, see **Section 25.4.3, Alarm Interrupt Function**.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RTCAn_base> + 40H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	RTCAnALM[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.35 RTCAnALM Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnALM [6:0]	Minute of the alarm interrupt in BCD

#### NOTES

1. If decimal values outside the range of 00 to 59 in BCD are set, no alarm interrupt request will be generated.
2. When the setting of RTCAnALM is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

### 25.3.6.2 RTCAnALH — RTCA Alarm Hour Setting Register

This register specifies the hour of the alarm interrupt.

For details and example settings, see **Section 25.4.3, Alarm Interrupt Function**.

**Access:** This register can be read or written in 8-bit units.

**Address:** <RTCAn\_base> + 44<sub>H</sub>

**Value after reset:** 12<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—						RTCAAnALH[5:0]
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.36 RTCAAnALH Register Contents**

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAAnALH [5:0]	Hour of the alarm interrupt in BCD

#### NOTES

- If decimal values outside the following range are set, no alarm interrupt request will be generated:
  - 12-hour format (RTCACTL0.RTCAAnAMPM = 0): 01 to 12 or 21 to 32
  - 24-hour format (RTCACTL0.RTCAAnAMPM = 1): 00 to 23
- When the setting of RTCAAnALH is changed during sub-counter operation (RTCACTL0.RTCAAnCEST = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

### 25.3.6.3 RTCAnALW — RTCA Alarm Day of the Week Setting Register

This register specifies the day(s) of the week of the alarm interrupt.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<RTCAn_base> + 48H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	RTCAnALW[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 25.37 RTCAnALW Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnALW [6:0]	Specifies day of the week m (m = 0 to 6) as a day, when an alarm interrupt request is generated: 0: No alarm interrupt request is generated on day m. 1: Alarm interrupt request is generated on day m at the time set using RTCAnALM and RTCAnALH. The bits of this register correspond to the count value of the day of the week counter (RTCAnWEEKC).

#### NOTE

When the setting of RTCAnALW is changed during sub-counter operation (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

#### Example

If Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, ..., Saturday is RTCAnWEEK = 6:

- To set the alarm for Sunday, set RTCAnALW = 0000 0001<sub>B</sub>.
- To set the alarm for Monday and Wednesday, set RTCAnALW = 0000 1010<sub>B</sub>.
- To set the alarm for Tuesday, Thursday, and Saturday, set RTCAnALW = 0101 0100<sub>B</sub>.

For more examples, see **Section 25.4.3, Alarm Interrupt Function**.

## 25.3.7 RTCA Emulation Register

### 25.3.7.1 RTCAnEMU — RTCA Emulation Register

This register controls operation by SVSTOP.

**Access:** This register can be read or written in 8-bit or 1-bit units.  
A write should be performed when EPC.SVSTOP = 0.

**Address:** <RTCAn\_base> + 74<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	RTCAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

**Table 25.38 RTCAnEMU Register Contents**

Bit Position	Bit Name	Function
7	RTCAnSVSDIS	When the EPC.SVSTOP bit is set to 0: Count clock is supplied when the debugger gains microcontroller control (at a breakpoint, etc.) regardless of the value of this bit.  When the EPC.SVSTOP bit is set to 1: 0: Count clock is stopped when the debugger gains microcontroller control (as at a breakpoint). 1: Count clock continues to be supplied when the debugger gains microcontroller control (at a breakpoint, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 25.4 Operation

The RTCA provides two operation modes:

- Frequency selection mode
- 32.768-kHz mode

The operation mode that can be used depends on the available input clock RTCATCKI. The operation mode specifies the sub-counter compare value that is used to trigger the seconds counter and thus all subsequent counters. Clock error correction is only possible in 32.768-kHz mode.

The following table provides an overview of the properties of the two operation modes.

**Table 25.39 RTCA Operation Mode Overview**

32.768-kHz Mode			
	Frequency Selection Mode	Clock Correction Disabled	Clock Correction Enabled
Allowed input clock RTCATCKI	Any frequency from 32 kHz to 4.194304 MHz	32.768 kHz	Any frequency from 32.76180000 kHz to 32.77420000 kHz
Sub-counter RTCAnSUBC operation	<ul style="list-style-type: none"> <li>• Counter overflow at value of RTCAnSCMP</li> <li>• RTCAnSCMP must be set to RTCATCKI-1 (in Hz)</li> </ul>	Counter overflow at $7FFF_H$	Counter overflow at $7FFF_H$ or Every 20 or 60 seconds: $7FFF_H$ $\pm RTCAnSUBU.RTCAnF[5:0]$

The operation mode is selected by control bit RTCAnCTL0.RTCAnSLSB. For details on how to set the operation mode during RTCA initialization, see **Section 25.5.1, Initial Setting of the RTCA**.

### CAUTIONS

1. The input clock RTCATCKI must not be outside the allowed frequency range.
2. The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1).

### 25.4.1 Clock Counter Format

The clock counters (RTCAnSECC to RTCAnYEARC) operate on binary coded decimals (BCD): Each digit is represented by its own binary sequence.

Depending on the valid data range, the number of bits for a digit differs. For example, the tens digit of the month of the year counter has only one bit (for 0 and 1) whereas the tens digit of the minutes counter has 3 bits (for 0 to 5).

The following table lists the decimals 0 to 59 in binary and BCD.

**Table 25.40 Example of BCD Code – Seconds or Minutes Counter (0 to 59)**

Decimal	Binary	BCD
0	000000	000 0000
1	000001	000 0001
2	000010	000 0010
3	000011	000 0011
4	000100	000 0100
5	000101	000 0101
6	000110	000 0110
7	000111	000 0111
8	001000	000 1000
9	001001	000 1001
10	001010	001 0000
11	001011	001 0001
12	001100	001 0010
:	:	:
58	111010	101 1000
59	111011	101 1001

### 25.4.2 Fixed Interval Interrupt Function

Interrupt RTCATINTR can be specified to occur after every 0.25 seconds, 0.5 seconds, 1 (full) second, 1 (full) minute, 1 (full) hour, 1 (full) day, or 1 (full) month.

The fixed interval interrupt function is controlled by bits RTCAnCTL1.RTCAnCT[2:0].

### 25.4.3 Alarm Interrupt Function

Interrupt RTCATINTAL can be specified to occur at a certain time on one or several days of the week. This interrupt can be used as a wake-up signal.

The alarm interrupt function is enabled and disabled by bit RTCAnCTL1.RTCAnENALM.

The alarm setting is specified by the following control registers:

- RTCAnALW selects the weekday(s).

The allocation of bits to weekdays is defined by the day of the week count buffer register RTCAnWEEK.

- RTCAnALH and RTCAnALM specify the hour and minute in BCD.

### Examples

The following tables show some exemplary settings of the alarm control registers for both 12-hour and 24-hour format.

In this example, Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, ..., Saturday is RTCAnWEEK = 6:

**Table 25.41 Alarm Setting in 12-Hour Format (RTCAnCTL0.RTCAnAMPM = 0)**

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00 am	01 <sub>H</sub>	07 <sub>H</sub>	00 <sub>H</sub>
Sunday, Monday 12:15 pm	03 <sub>H</sub>	32 <sub>H</sub>	15 <sub>H</sub>
Monday, Wednesday, Friday 5:30 pm	2A <sub>H</sub>	25 <sub>H</sub>	30 <sub>H</sub>
Daily, 10:45 pm	7F <sub>H</sub>	30 <sub>H</sub>	45 <sub>H</sub>

**Table 25.42 Alarm Setting in 24-Hour Format (RTCAnCTL0.RTCAnAMPM = 1)**

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00	01 <sub>H</sub>	07 <sub>H</sub>	00 <sub>H</sub>
Sunday, Monday 12:15	03 <sub>H</sub>	12 <sub>H</sub>	15 <sub>H</sub>
Monday, Wednesday, Friday 17:30	2A <sub>H</sub>	17 <sub>H</sub>	30 <sub>H</sub>
Daily, 22:45	7F <sub>H</sub>	22 <sub>H</sub>	45 <sub>H</sub>

### 25.4.4 Clock Error Correction

Clock error correction compensates for deviations of the oscillator from the nominal clock rate. With clock error correction input clock rates from 32.76180 kHz to 32.77420 kHz are possible.

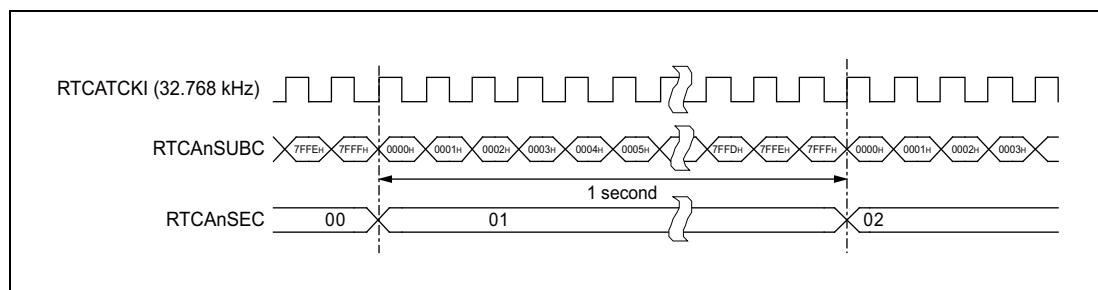
The clock error correction function is only available in 32.768-kHz operation mode. In this operation mode, a nominal clock rate of 32.768 kHz is expected and the sub-counters overflow value is fixed to 7FFF<sub>H</sub>.

The following figures illustrate the clock error when the input clock rate deviates from the nominal clock.

#### RTCATCKI = 32.768 kHz

**Figure 25.2, RTCATCKI = 32.768 kHz, No Clock Error Correction Required** shows the timing diagram if RTCATCKI matches the nominal clock rate of 32.768 kHz. No clock error correction is required.

Counting from 0 to 32767 (0 to 7FFF<sub>H</sub>) with a 32.768-kHz clock is exactly equal to one second.



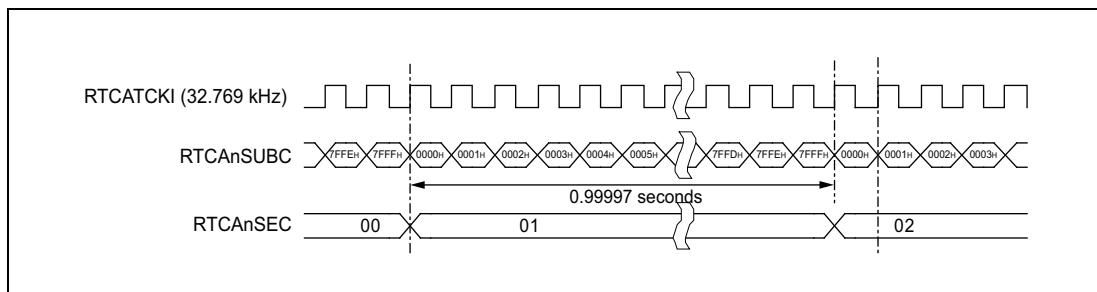
**Figure 25.2 RTCATCKI = 32.768 kHz, No Clock Error Correction Required**

**RTCATCKI = 32.769 kHz**

**Figure 25.3, RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled** shows the timing diagram if RTCATCKI deviates from the nominal clock rate of 32.768 kHz. In this example, RTCATCKI is connected to a 32.769-kHz oscillator. Clock error correction is not enabled.

Counting from 0 to 32767 (0 to  $7FFF_H$ ) with a 32.769-kHz clock is equal to approximately 0.99997 seconds ( $32768/32769$ ). A “+ error” (faster than 32.768-kHz) occurs. In one month, RTCA deviates approximately –79 seconds from the real time.

$$\text{Error} = (32768/32769 - 1) \times 60 \text{ (s)} \times 60 \text{ (min)} \times 24 \text{ (h)} \times 30 \text{ (d)}$$



**Figure 25.3 RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled**

Clock error correction is performed by stretching/reducing the 1-second period of the sub-counter at regular intervals. The sub-counter’s upper limit of  $7FFF_H$  is increased or decreased by setting the following parameters in register RTCAnSUBU:

- A correction value greater than one
- An operator (add/subtract)
- An interval (20 or 60 seconds)

The corrected overflow value becomes effective every 20 or 60 seconds, so that on the average RTCAnSECC is triggered exactly every second.

#### 25.4.4.1 Setting the Correction Value and the Operator

The correction value and operator are specified by the RTCAnF6, RTCAnF[5:0] bits of the RTCAnSUBU register:

- RTCAnF6 specifies whether the overflow value is incremented or decremented.
- RTCAnF[5:0] specifies the correction value.

The correction values are calculated as follows:

**Table 25.43 Correction Value Settings**

RTCAnF6	Increment/Decrement	Correction Value
0	Increment	(Value of RTCAnF[5:0] – 1) × 2
1	Decrement	(Inverted value of RTCAnF[5:0] + 1) × 2

Some examples are given in the following table:

**Table 25.44 Correction Value Examples**

RTCAnF6	RTCAnF[5:0]	Correction Value	Count Limit of RTCAnSUBC
0	15 <sub>H</sub>	(15 <sub>H</sub> – 1) × 2 = 40	32768 + 40 = 32808
1	15 <sub>H</sub>	(15 <sub>H</sub> + 1) × 2 = (2A <sub>H</sub> + 1) × 2 = 86	32768 – 86 = 32682

#### 25.4.4.2 Impact of the Repetition Interval

The correction value set by RTCAnF6, RTCAnF[5:0] does not change the count limit of RTCAnSUBC every second. The repetition interval at which the correction value becomes effective is specified by bit RTCAnDEV.

This bit also influences the size of the correctable frequency range and the correction accuracy.

The following table summarizes the RTCAnDEV settings.

**Table 25.45 Setting of Bit RTCAnSUBU.RTCAnDEV**

RTCAnDEV	Count Limit of RTCAnSUBC is Changed	Frequency Range that can be Corrected	Correction Accuracy
0	Every 20 seconds when RTCAnSECC = 00, 20, or 40	32.76180000 to 32.77420000 kHz	
1	Every 60 seconds when RTCAnSECC = 00	32.76593333 to 32.77006667 kHz	Three times higher than for RTCAnDEV = 0

### 25.4.4.3 Sample Settings

The frequencies that can be corrected, as well as the setting values of bits RTCAnDEV, RTCAnF6, and RTCAnF[5:0], are listed in the following table.

**Table 25.46 Correctable Frequency Range when RTCAnDEV = 0**

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76810000 kHz	0	000010	Once every 20 s, RTCAnSUBC count value + 2
32.76820000 kHz	0	000011	Once every 20 s, RTCAnSUBC count value + 4
32.76830000 kHz	0	000100	Once every 20 s, RTCAnSUBC count value + 6
:	:	:	:
32.77400000 kHz	0	111101	Once every 20 s, RTCAnSUBC count value + 120
32.77410000 kHz	0	111110	Once every 20 s, RTCAnSUBC count value + 122
32.77420000 kHz (upper limit)	0	111111	Once every 20 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76180000 kHz (lower limit)	1	000010	Once every 20 s, RTCAnSUBC count value – 124
32.76190000 kHz	1	000011	Once every 20 s, RTCAnSUBC count value – 122
32.76200000 kHz	1	000100	Once every 20 s, RTCAnSUBC count value – 120
:	:	:	:
32.76770000 kHz	1	111101	Once every 20 s, RTCAnSUBC count value – 6
32.76780000 kHz	1	111110	Once every 20 s, RTCAnSUBC count value – 4
32.76790000 kHz	1	111111	Once every 20 s, RTCAnSUBC count value – 2

**Table 25.47 Correctable Frequency Range when RTCAnDEV = 1**

<b>Input Clock Frequency</b>	<b>RTCAnF6</b>	<b>RTCAnF[5:0]</b>	<b>Correction Value of RTCAnSUBC</b>
—	0	000000	No correction
—	0	000001	No correction
32.76803333 kHz	0	000010	Once every 60 s, RTCAnSUBC count value + 2
32.76806667 kHz	0	000011	Once every 60 s, RTCAnSUBC count value + 4
32.76810000 kHz	0	000100	Once every 60 s, RTCAnSUBC count value + 6
:	:	:	:
32.77000000 kHz	0	111101	Once every 60 s, RTCAnSUBC count value + 120
32.77003333 kHz	0	111110	Once every 60 s, RTCAnSUBC count value + 122
32.77006667 kHz (upper limit)	0	111111	Once every 60 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76593333 kHz (lower limit)	1	000010	Once every 60 s, RTCAnSUBC count value - 124
32.76596667 kHz	1	000011	Once every 60 s, RTCAnSUBC count value - 122
32.76600000 kHz	1	000100	Once every 60 s, RTCAnSUBC count value - 120
:	:	:	:
32.76790000 kHz	1	111101	Once every 60 s, RTCAnSUBC count value - 6
32.76793333 kHz	1	111110	Once every 60 s, RTCAnSUBC count value - 4
32.76796667 kHz	1	111111	Once every 60 s, RTCAnSUBC count value - 2

## 25.5 Procedures for Setup, Writing and Reading

The following subsections provide flow charts that illustrate the procedures for RTCA setup and for reading and writing the RTCA clock counters.

### 25.5.1 Initial Setting of the RTCA

The RTCA must be stopped before setting the initial setting value of each counter.

#### 25.5.1.1 RTCA Stop Procedure

Stop the RTCA according to the following flow.

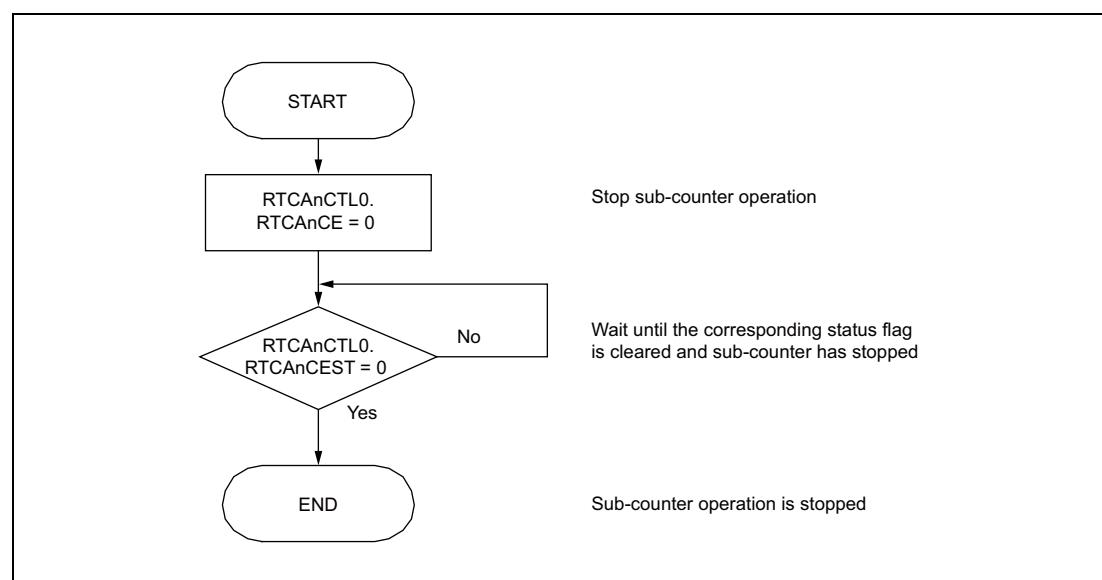


Figure 25.4 RTCA Stop Procedure

### 25.5.1.2 RTCA Initialization Procedure

Perform the initial setting of the RTCA according to the following flow:

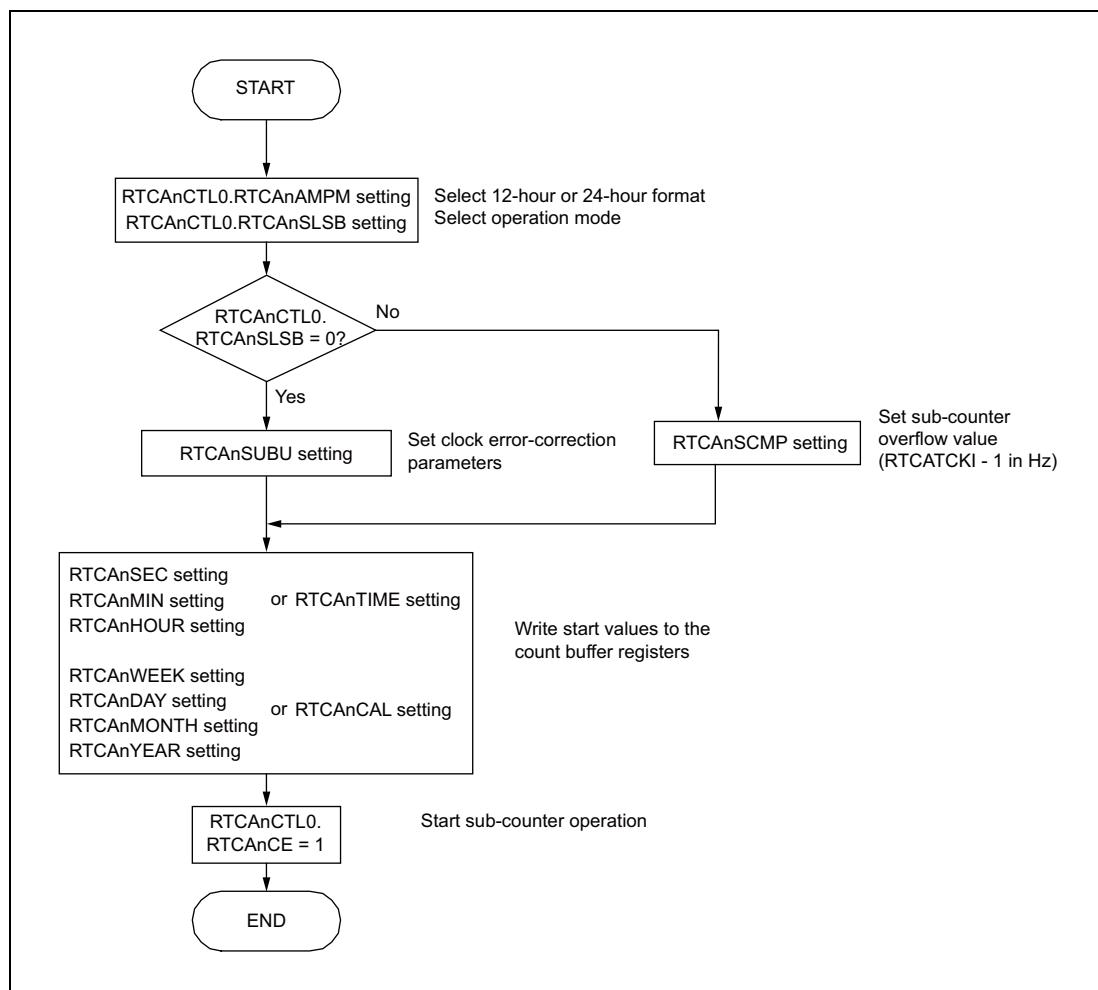


Figure 25.5 RTCA Initial Setup Procedure

#### CAUTION

The internal clock counter is synchronized with RTCATCKI.

In addition, two RTCATCKI periods are required before the clock counter starting behind END of the above flow.

Therefore, PCLK must be continuously supplied until the completion of the initial setting.

Check that RTCAnCTL0.RTCAnCEST = 1, when the supply of PCLK is stopped after setting the initial setting value of RTCA.

### 25.5.2 Updating Clock Counters

The clock counters RTCAnSECC to RTCAnYEARC can be stopped and updated while the sub-counter is running.

To update the clock counter when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

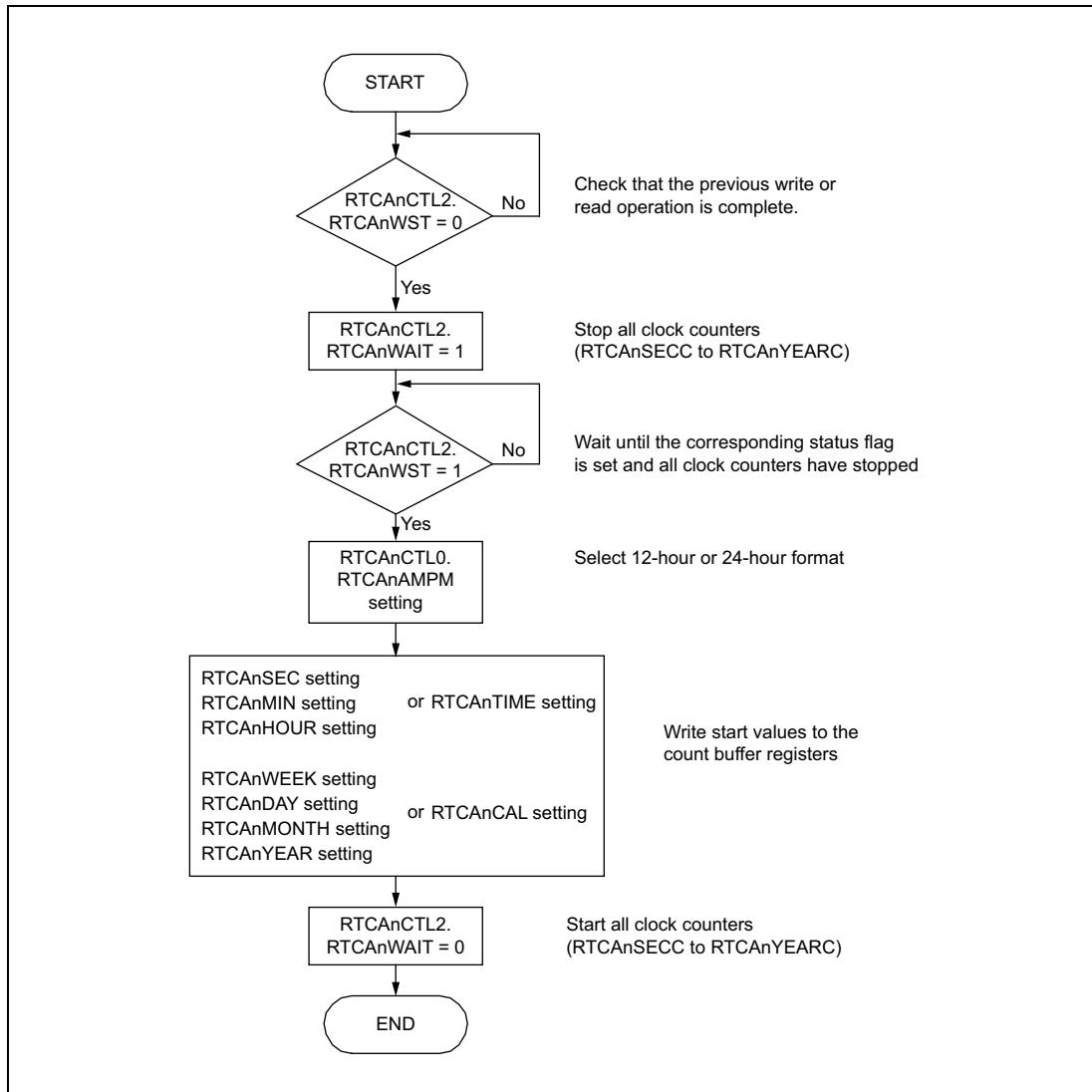


Figure 25.6 Updating Clock Counter Values

#### CAUTIONS

- The internal clock counter is synchronized with RTCATCKI.**  
In addition, two RTCATCKI periods are required before the clock counter updating behind **END** of the above flow.  
Therefore, PCLK must be continuously supplied until the completion of the clock counter updating.  
Check that RTCAnCTL2.RTCAnWST = 0 before stopping the supply of PCLK after the completion of clock counter updating.
- The update procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more:**

3. Only one sub-counter overflow can be held internally and increment the seconds counter after restarting the clock counters if the value is held.
4. If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.

### 25.5.3 Reading Clock Counters

There are two methods to read the clock counters while sub-counter operation is enabled:

- Reading count buffer registers
- Reading counter registers

The advantages and disadvantages of the two methods are summarized in the following table.

**Table 25.48 Comparison of the Two Read Methods**

	<b>Advantage</b>	<b>Disadvantage</b>
Reading count buffer registers	It is unnecessary to read clock counters several times because the clock counters are read synchronously.	A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.
Reading count registers	Program wait state does not occur.	If the sub-counter increments, the clock counters must be read several times because they are read asynchronously to RTCATCKI.

#### 25.5.3.1 Procedure for Reading Count Buffer Registers

The following operations are necessary:

1. Stop all clock counters (RTCAnCTL2.RTCAnWAIT = 1). The value of the clock counters is transferred to the corresponding count buffer registers.
2. Read the count buffer registers.

A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.

The maximum delay is three PCLK periods plus two RTCATCKI periods. For example, if the RTCA operates with PCLK = 40 MHz and RTCATCKI = 32.768 kHz, the delay is about 61  $\mu$ s.

To read the count buffer register when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1), follow the flowchart shown below.

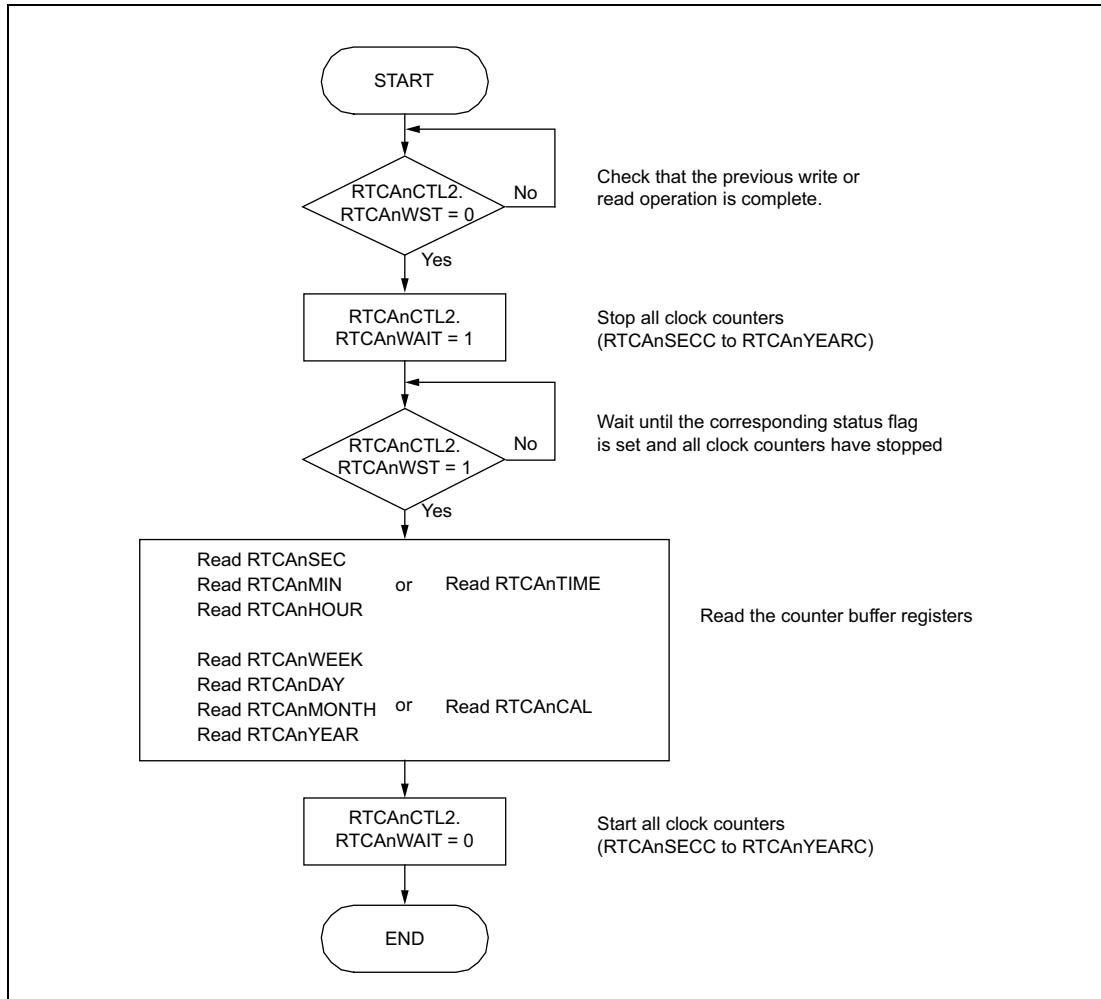


Figure 25.7 Reading Clock Count Buffer Registers

#### CAUTIONS

1. The internal clock counter is synchronized with RTCATCKI. In addition, two RTCATCKI periods are required before resuming counter behind END of the above flow. Therefore, PCLK must be continuously supplied until the counter resuming. Check that RTCAnCTL0.RTCAnCEST = 1 first to stop the supply of PCLK after count buffer register reading.
2. The reading procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more.
3. Only one sub-counter overflow can be held internally. If there is a value held internally when the clock counter restarts, the seconds counter will be incremented by 1.
4. If the sub-counter overflows more than once during clock counter stop the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.

### 25.5.3.2 Procedure for Reading Counter Registers Directly

To ensure that the sub-counter did not overflow while reading the counters, the seconds counter RTCAnSECC must be read twice in the beginning and at the end of the procedure. The first read value is compared with the second read value.

- First read value = second read value:  
No overflow of sub-counter occurred during counter read operation.
- First read value  $\neq$  second read value:  
Overflow of the sub-counter occurred during counter read operation. The counters must be read again to get the current counter values.

To read the counter register directly when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

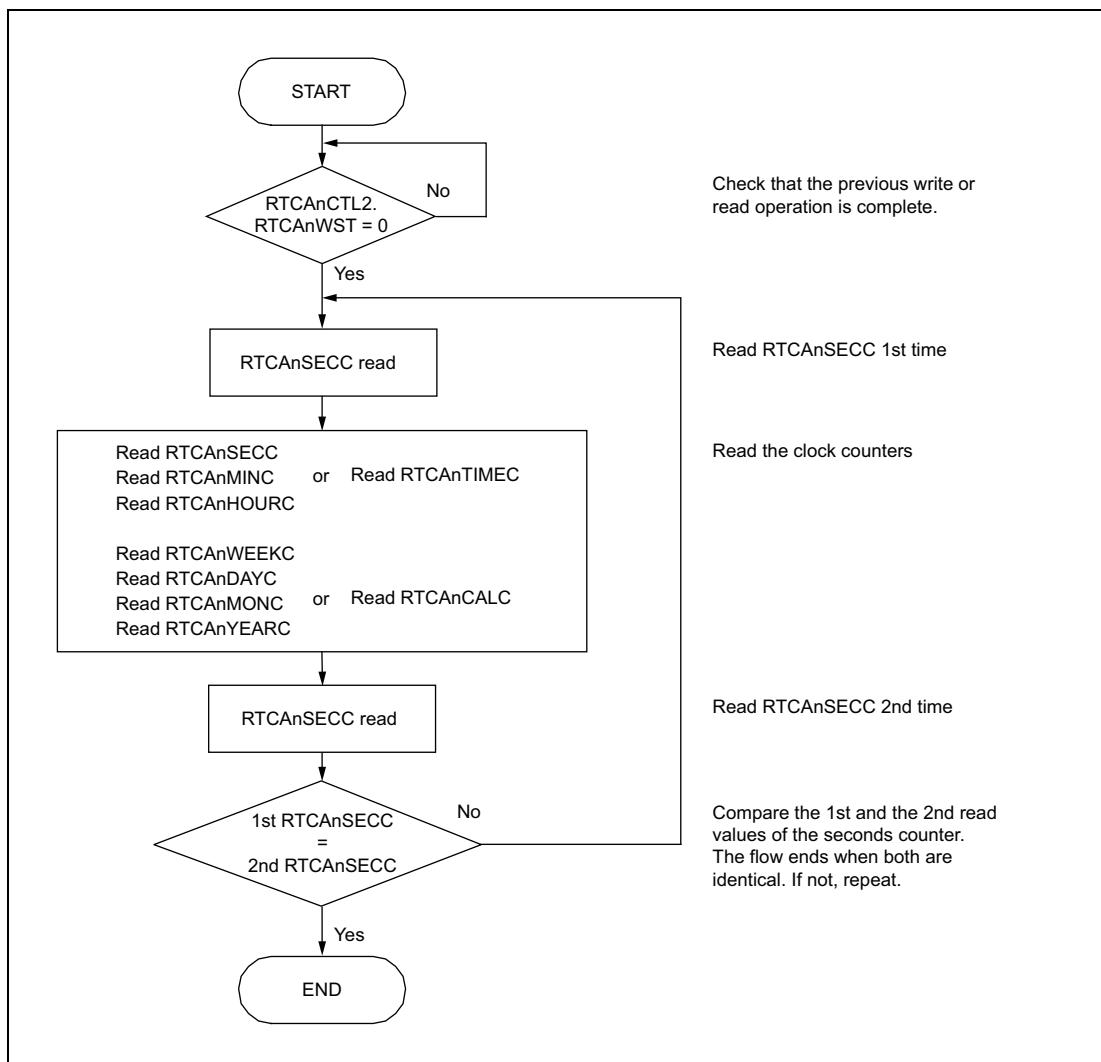


Figure 25.8 Reading Clock Counter Registers

#### NOTE

The procedure must be completed within one second.

### 25.5.4 Reading RTCAnSRBU

RTCAnSRBU is the read buffer register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), read RTCAnSRBU according to the following flow.

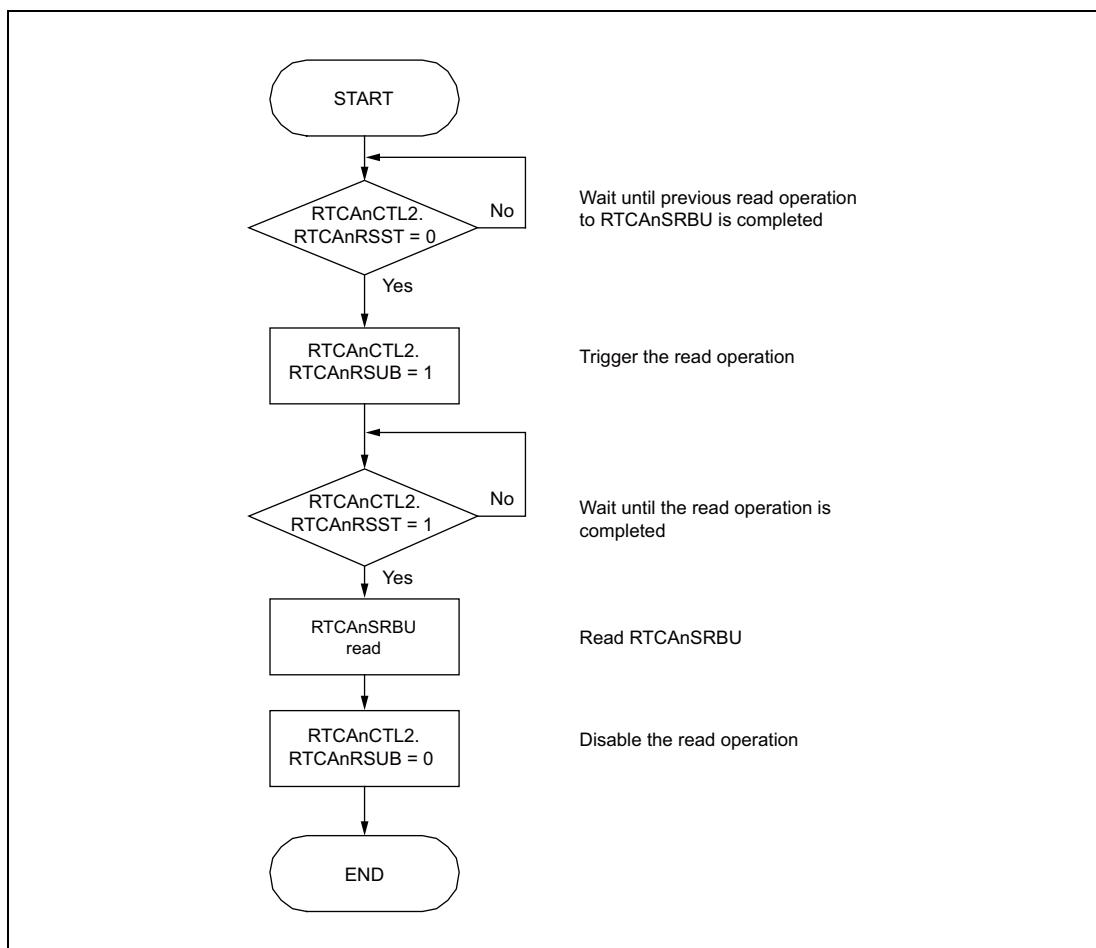


Figure 25.9 Reading the RTCAnSRBU Register

### 25.5.5 Writing to RTCAnSUBU

RTCAnSUBU is the clock error correction register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSUBU according to the flow described below.

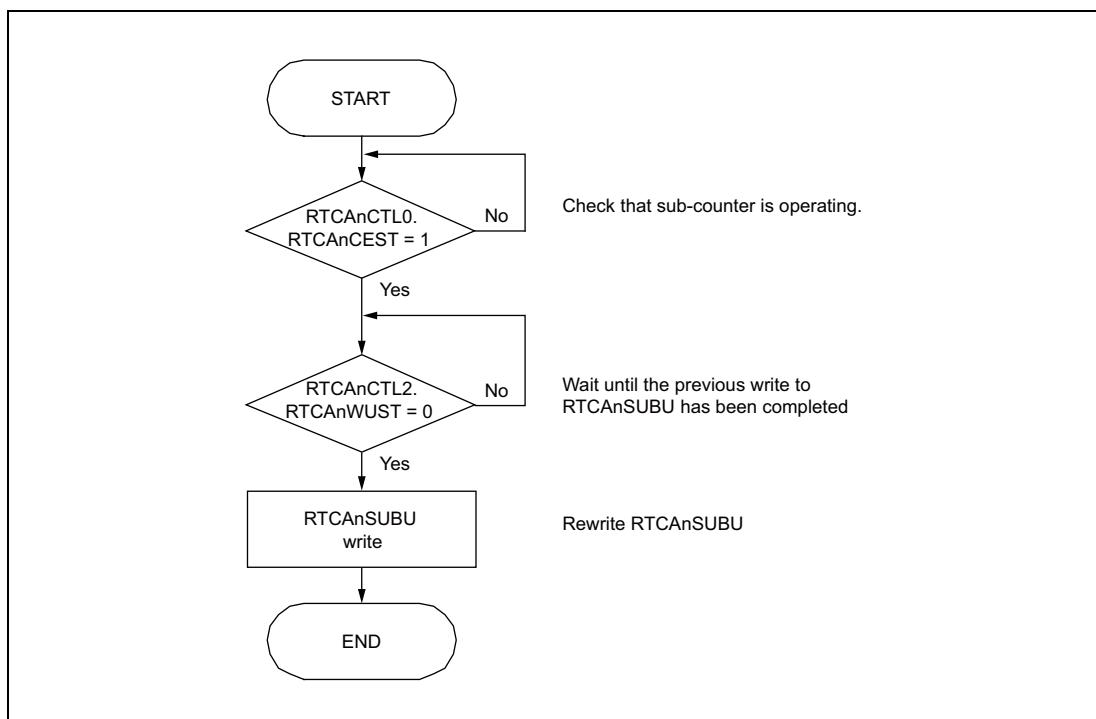


Figure 25.10 Writing to the RTCAnSUBU Register

#### NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWUST is set when RTCAnSUBU is written to. It is cleared when the write operation to RTCAnSUBU is completed. This is synchronous with the next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWUST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWUST = 1 at the beginning of this flow).

### 25.5.6 Writing to RTCAnSCMP

RTCAnSCMP is the sub-counter compare register.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSCMP according to the flow described below.

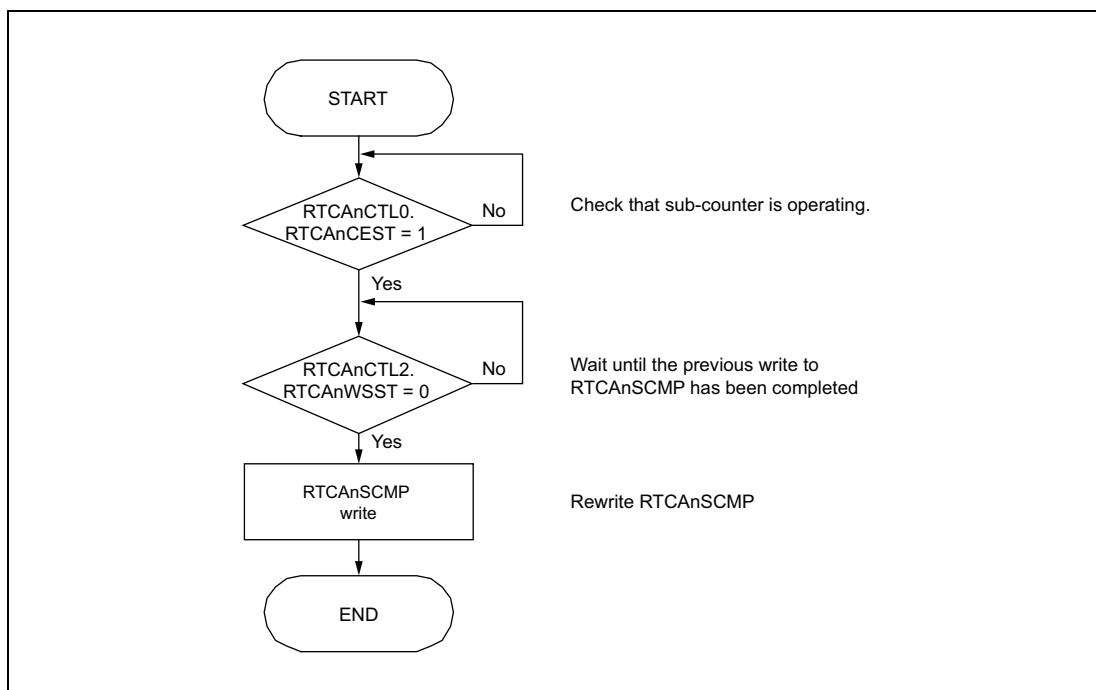


Figure 25.11 Writing the RTCAnSCMP Register

#### NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWSST is set when RTCAnSCMP is written to. It is cleared when the write operation to RTCAnSCMP is completed. This is synchronous with the next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWSST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWSST = 1 at the beginning of this flow).

## 25.6 Timing Diagrams

### 25.6.1 Timing of Counter Start

The following diagram illustrates the counter start after setting the time in the buffer registers.

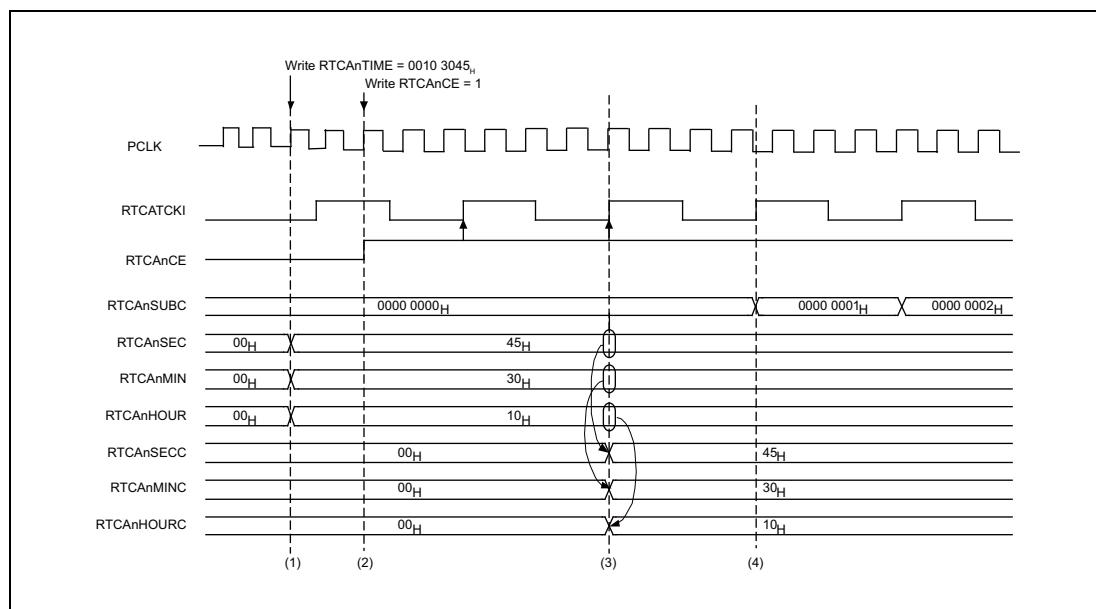


Figure 25.12 Counter Start Timing

The timing diagram above shows the following:

- (1) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME =  $0010\ 3045_H$ . Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.
- (2) Sub-counter operation is started by setting RTCAnCTL0.RTCAnCE = 1.
- (3) When the second rising edge of RTCATCKI occurs, the buffer register values are loaded to the corresponding count registers.
- (4) When the next rising edge of RTCATCKI occurs, count up of the sub-counter starts.

## 25.6.2 Timing of Clock Counter Update while Counter Is Enabled

The following diagram illustrates the counter restart after setting the time in the buffer registers.

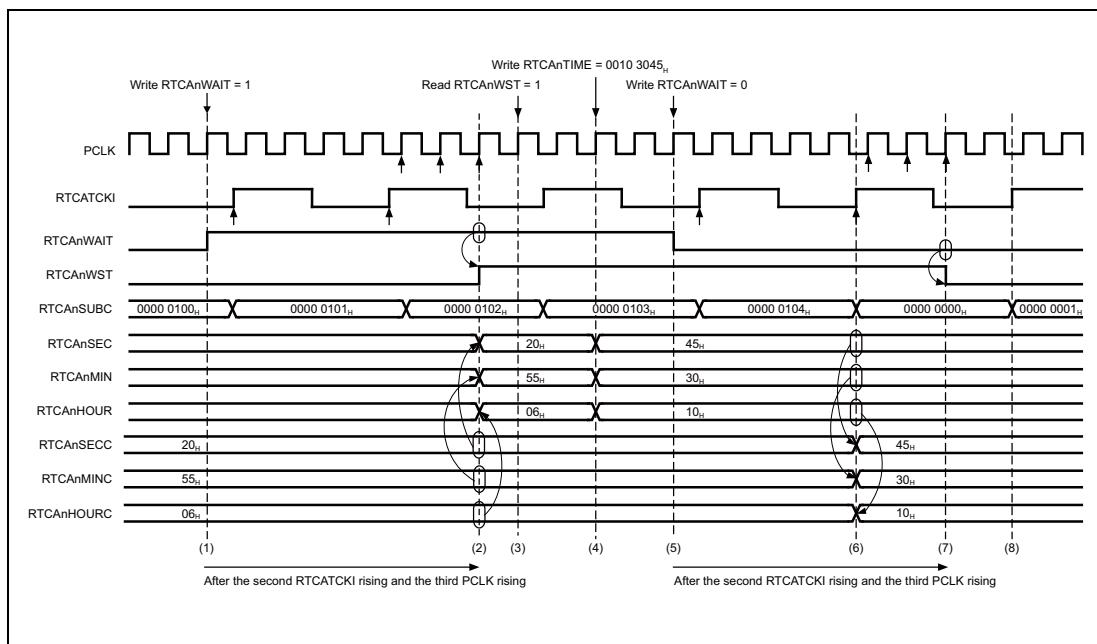


Figure 25.13 Clock Counter Update Timing

The timing diagram above shows the following:

- (1) Trigger the clock counters stop (RTCAnCTL2.RTCAnWAIT = 1).
- (2) RTCAnCTL2.RTCAnWST is set to 1 after the second rising edge of RTCATCKI and the third rising edge of PCLK, and the counter clock stops. The sub-counter continues counting.
- (3) RTCAnCTL2.RTCAnWST = 1 can be readable.
- (4) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME to 0010 3045<sub>H</sub>. Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.
- (5) Trigger the clock counters restart (RTCAnCTL2.RTCAnWAIT = 0).
- (6) When the second rising edge of RTCATCKI occurs, the values of the buffer registers are loaded to the corresponding count registers. Write operation to RTCAnSECC is performed and RTCAnSUBC is cleared.
- (7) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnWST is set to 0.
- (8) Clock counter operation is resumed.

### 25.6.3 Timing of Sub-Counter Read Buffer Reading while Counter is Enabled

The following diagram illustrates the timing when reading the sub-counter read buffer RTCAnSRBU.

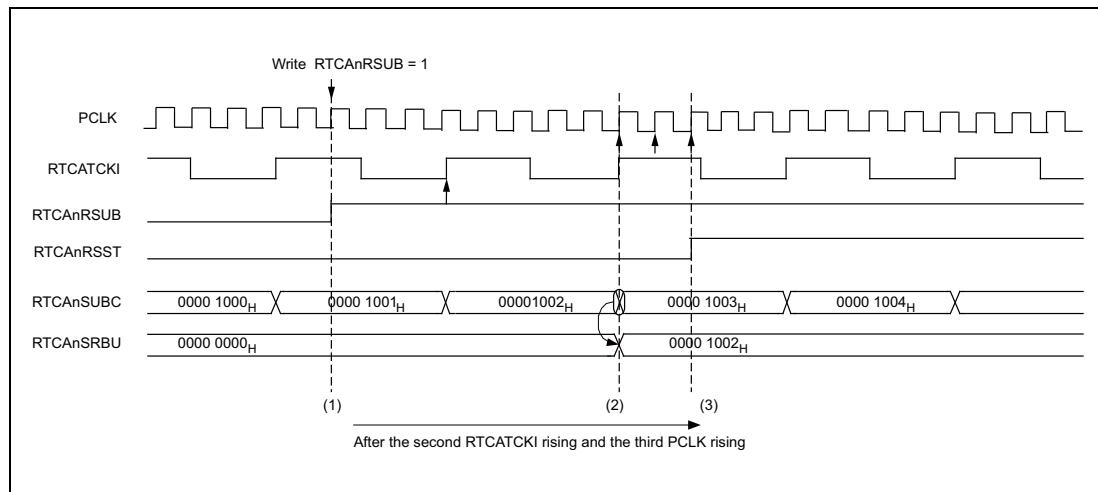


Figure 25.14 Timing when Reading the Sub-Counter Read Buffer Register Value

The timing diagram above shows the following:

- (1) Setting RTCAnRSUB = 1 triggers loading of the sub-counter value to RTCAnSRBU.
- (2) When the second rising edge of RTCATCKI occurs, the value of RTCAnSUBC is loaded to RTCAnSRBU.
- (3) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnRSST is set to 1 and RTCAnSRBU can be read.

## Section 26 Encoder Timer (ENCA)

This section contains a generic description of the Encoder Timer (ENCA).

The first part in this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the ENCA.

### 26.1 Features of RH850/F1L ENCA

#### 26.1.1 Number of Units and Channels

This microcontroller has the following number of ENCA units.

Each ENCA unit has one channel ENCA. “Number of channels” is used with the same meaning as “number of units” in this section.

**Table 26.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units				1		
Name				ENCA <sub>n</sub> (n = 0)		

**Table 26.2 Index**

Index	Description
n	Throughout this section, the individual ENCA units are identified by the index “n” (n = 0); for example, ENCA <sub>n</sub> CTL is the ENCA <sub>n</sub> control register.

#### 26.1.2 Register Base Address

ENCA<sub>n</sub> base address is listed in the following table.

ENCA<sub>n</sub> register addresses are given as offsets from the base address.

**Table 26.3 Register Base Address**

Base Address Name	Base Address
<ENCA0_base>	FFE8 0000 <sub>H</sub>

### 26.1.3 Clock Supply

The ENCA0 clock supply is shown in the following table.

**Table 26.4 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
ENCA0	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

### 26.1.4 Interrupt Requests

ENCA0 interrupt requests are listed in the following table.

**Table 26.5 Interrupt Requests**

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
<b>ENCA0</b>			
ENCATIOV	Overflow interrupt	77	—
ENCATIUD	Underflow interrupt	78	—
ENCATINT0	Capture/compare match interrupt 0	79	—
ENCATINT1	Capture/compare match interrupt 1	80	—
ENCATIEC	Encoder clear interrupt	81	—

### 26.1.5 Reset Sources

ENCA0 reset sources are listed in the following table. ENCA0 is initialized by these reset sources.

**Table 26.6 Reset Sources**

Unit Name	Reset Source
ENCA0	All reset sources (ISORES)

### 26.1.6 External Input/Output Signals

External input/output signals of ENCA0 are listed below.

**Table 26.7 External Input/Output Signals**

Unit Signal Name	Description	Alternative Port Pin Signal Name
<b>ENCA0</b>		
ENCATTIN0	ENCA0 capture trigger input 0* <sup>1</sup>	ENCA0TIN0
ENCATTIN1	ENCA0 capture trigger input 1* <sup>1</sup>	ENCA0TIN1
ENCA0E0	ENCA0 encoder input 0* <sup>1</sup>	ENCA0E0
ENCA0E1	ENCA0 encoder input 1* <sup>1</sup>	ENCA0E1
ENCA0EC	ENCA0 encoder clear input* <sup>1</sup>	ENCA0EC

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**.

### 26.1.7 Internal Input/Output Signals

Input/output signals to be connected between ENCA and PIC are listed below.

**Table 26.8 Internal Input/Output Signals**

Unit Signal Name	Description	Connected to
ENCATSST	Simultaneous start trigger	PIC
ENCATTIN1	ENCAn capture trigger input 1	PIC

## 26.2 Overview

### 26.2.1 Functional Overview

- Generation of the counter control signal from the encoder input signal, and count operation in synchronization with PCLK
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture/compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt request signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture/compare register to the counter upon underflow occurrence
- Encoder input signal can be used as the timer counter clear condition
- Edge or level can be selected for determining the presence of the encoder input signal that is used as the timer counter clear condition
- Detection of counter overflow and underflow and output of error flags and error occurrence interrupts
- Five interrupts: two capture/compare interrupts, one counter clear interrupt, one overflow interrupt, and one underflow interrupt

## 26.2.2 Block Diagram

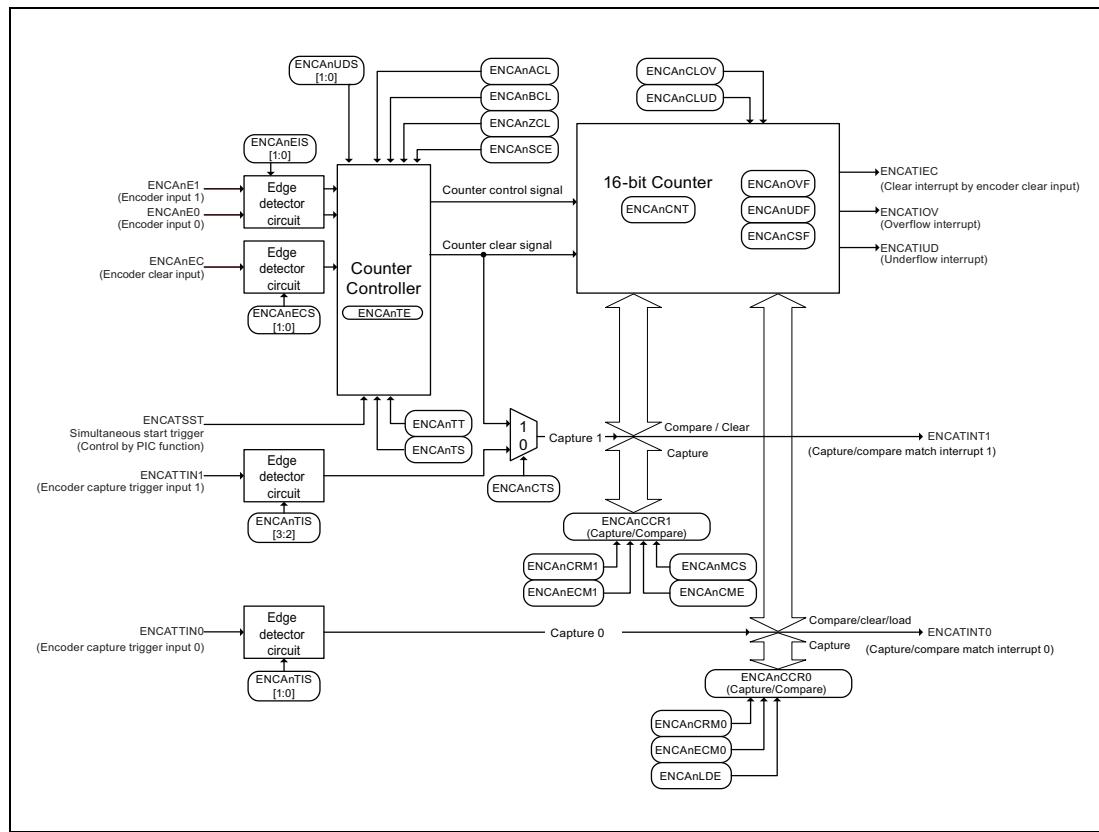


Figure 26.1 ENCA Block Diagram

## 26.3 Registers

### 26.3.1 List of Registers

ENCA registers are listed in the following table.

<ENCA<sub>n</sub>\_base> is defined in **Section 26.1.2, Register Base Address**.

Table 26.9 List of Registers

Module Name	Register Name	Symbol	Address
ENCA <sub>n</sub>	ENCA <sub>n</sub> capture/compare register 0	ENCA <sub>n</sub> CCR0	<ENCA <sub>n</sub> _base>
ENCA <sub>n</sub>	ENCA <sub>n</sub> capture/compare register 1	ENCA <sub>n</sub> CCR1	<ENCA <sub>n</sub> _base> + 04 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> counter register	ENCA <sub>n</sub> CNT	<ENCA <sub>n</sub> _base> + 08 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> status flag register	ENCA <sub>n</sub> FLG	<ENCA <sub>n</sub> _base> + 0C <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> status flag clear register	ENCA <sub>n</sub> FGC	<ENCA <sub>n</sub> _base> + 10 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> timer enable status register	ENCA <sub>n</sub> TE	<ENCA <sub>n</sub> _base> + 14 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> timer start trigger register	ENCA <sub>n</sub> TS	<ENCA <sub>n</sub> _base> + 18 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> timer stop trigger register	ENCA <sub>n</sub> TT	<ENCA <sub>n</sub> _base> + 1C <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> I/O control register 0	ENCA <sub>n</sub> IOC0	<ENCA <sub>n</sub> _base> + 20 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> control register	ENCA <sub>n</sub> CTL	<ENCA <sub>n</sub> _base> + 40 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> I/O control register 1	ENCA <sub>n</sub> IOC1	<ENCA <sub>n</sub> _base> + 44 <sub>H</sub>
ENCA <sub>n</sub>	ENCA <sub>n</sub> emulation register	ENCA <sub>n</sub> EMU	<ENCA <sub>n</sub> _base> + 48 <sub>H</sub>

### 26.3.2 ENCACTL — ENCA Control Register

This register is used to configure various operation settings of the Encoder Timer.

**Access:** This register can be read or written in 16-bit units.  
Writing to this register during operation is prohibited.

**Address:** <ENCA\_base> + 40H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 26.10 ENCACTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	ENCACTL CME	<p>Encoder Clear Mask Enable This bit is used to enable/disable masking of compare-match interrupt detection when the compare function is used. 0: Disables the compare-match interrupt (ENCATINT1) mask function for the ENCAAnCCR1 register 1: Enables the compare-match interrupt (ENCATINT1) mask function for the ENCAAnCCR1 register. This bit is valid only when ENCAAnCRM1 = 0. When this bit is set to "1", setting ENCAAnECM1 to "1" is prohibited.</p>
14	ENCACTL MCS	<p>Encoder Mask Clear Select This bit is used to select the trigger for cancelling masking of compare-match interrupt detection ENCATTINT1 when the compare function is used. This bit is valid only when ENCAAnCRM1 = 0. 0: Masking of compare-match interrupt detection is canceled when the ENCAAnCCR1 register is written. 1: Masking of compare match interrupt detection is canceled when one of the following three operations is performed. – Timer counter clear operation accompanying encoder clear input – Timer counter clear operation upon compare-match between ENCAAnCNT and ENCAAnCCR0 when ENCAAnECM0 = 1 – Loading from ENCAAnCCR0 to the timer counter upon underflow detection when ENCAAnLDE = 1</p>
13 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	ENCACTL CRM1	ENCAAnCCR1 Register Mode 0: ENCAAnCCR1 used as compare register. 1: ENCAAnCCR1 used as capture register.
8	ENCACTL CRM0	ENCAAnCCR0 Register Mode 0: ENCAAnCCR0 used as compare register. 1: ENCAAnCCR0 used as capture register.
7	ENCACTL CTS	ENCAAnCCR1 Capture Trigger Select This is a trigger selection bit for the capture operation to the ENCAAnCCR1 register. This bit is valid only when ENCAAnCRM1 = 1. 0: Uses ENCATTIN1 of capture trigger 1 signal as the capture trigger for the ENCAAnCCR1 register. 1: The counter clear signal selected with ENCAAnSCE is used as the capture trigger for the ENCAAnCCR1 register.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

**Table 26.10 ENCAAnCTL Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
4	ENCAAnLDE	<p>ENCAAn Counter Load Enable This bit is used to enable/disable setting value loading to the counter upon underflow occurrence. This bit is valid only when ENCAAnCRM0 = 0. When ENCAAnCRM0 = 1, loading of the ENCAAnCCR0 register setting value to the counter upon occurrence of an underflow is not performed, regardless of the value of this bit.</p> <p>0: Disables loading of ENCAAnCCR0 register setting value to counter upon occurrence of a counter underflow. 1: Enables loading of ENCAAnCCR0 register setting value to counter upon occurrence of a counter underflow.</p>
3	ENCAAnECM1	<p>Encoder Clear Mode 1 This bit is used to set the counter clear operation upon match between the counter value and ENCAAnCCR1 setting value. This bit is valid only when ENCAAnCRM1 = 0.</p> <p>0: Does not clear the counter to <math>0000_H</math> upon match of timer counter value and ENCAAnCCR1 setting value. 1: Clears the counter to <math>0000_H</math> upon match of timer counter value and ENCAAnCCR1 setting value if the next count is a down-count.</p>
2	ENCAAnECM0	<p>Encoder Clear Mode 0 This bit is used to set the counter clear operation upon match between the counter value and ENCAAnCCR0 setting value. This bit is valid only when ENCAAnCRM0 = 0.</p> <p>0: Does not clear the counter to <math>0000_H</math> upon match of timer counter value and ENCAAnCCR0 setting value. 1: Clears the counter to <math>0000_H</math> upon match of timer counter value and ENCAAnCCR0 setting value if the next count is a up-count.</p>
1, 0	ENCAAnUDS[1:0]	<p>Up/down Count Selection 1 and 0 These are the counter up/down control bits using ENCAAnE0 and ENCAAnE1.</p> <p>00: Upon detection of valid edge of ENCAAnE0, - down-count when ENCAAnE1 = H, - up-count when ENCAAnE1 = L</p> <p>01: Upon detection of valid edge of ENCAAnE0, up-count, Upon detection of valid edge of ENCAAnE1, down-count</p> <p>10: At rising edge of ENCAAnE0, down-count At falling edge of ENCAAnE0, up-count However, count operation is performed only when ENCAAnE1 = L.</p> <p>11: Detection of both edges of ENCAAnE0, ENCAAnE1. The count operation is determined based on the combination of the detected edge and level.</p>

### 26.3.3 ENCAAnIOC0 — ENCAAn I/O Control Register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCATTIN0, ENCATTIN1).

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<ENCAAn_base> + 20H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	ENCAAnTIS[3:2]	ENCAAnTIS[1:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

**Table 26.11 ENCAAnIOC0 Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	ENCAAnTIS[3:2]	Input Edge Selection for Capture Trigger 1 These bits are valid only when ENCAAnCTL.ENCAAnCRM1 = 1 and ENCAAnCTL.ENCAAnCTS = 0. All other settings of ENCAAnCRM1 and ENCAAnCTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAAnTIS[1:0]	Input Edge Selection for Capture Trigger 0 These bits are valid only when ENCAAnCTL.ENCAAnCRM0 = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

### 26.3.4 ENCAAnIOC1 — ENCAAn I/O Control Register 1

This register is used to perform the clear condition setting and edge selection for input from the encoder.

**Access:** This register can be read or written in 8-bit units.  
Writing to this register during operation is prohibited.

**Address:** <ENCAAn\_base> + 44H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	ENCAAnSCE	ENCAAnZCL	ENCAAnBCL	ENCAAnACL	ENCAAnECS[1:0]		ENCAAnEIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 26.12 ENCAAnIOC1 Register Contents (1/2)**

Bit Position	Bit Name	Function
7	ENCAAnSCE	Encoder Special-Clear Enable This is an encoder special clear enable bit. When setting this bit to 1, set ENCAAnUDS1 and ENCAAnUDS0 to 10 <sub>B</sub> or 11 <sub>B</sub> . The operation is not guaranteed if this bit is set to 1 with ENCAAnUDS1 and ENCAAnUDS0 set to 00 <sub>B</sub> or 01 <sub>B</sub> . 0: Clears the counter upon detection of ENCAAnEC valid edge (set with ENCAAnECS1 and ENCAAnECS0). 1: Clears the counter upon detection of input level condition of ENCAAnE0, ENCAAnE1 and ENCAAnEC (set with ENCAAnZCL bit, ENCAAnBCL bit, and ENCAAnACL bit).
6	ENCAAnZCL	Input-Z Clear Condition Selection This bit is used to set the condition for clearing the encoder clear input (ENCAAnEC) when using the encoder special clear function. This bit is valid only when ENCAAnSCE = 1; it is invalid when ENCAAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
5	ENCAAnBCL	Input-B Clear Condition Selection This bit is used to set the condition for clearing the encoder input 1 (ENCAAnE1) when using the encoder special clear function. This bit is valid only when ENCAAnSCE = 1; it is invalid when ENCAAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
4	ENCAAnACL	Input-A Clear Condition Selection This bit is used to set the condition for clearing the encoder input 0 (ENCAAnE0) when using the encoder special clear function. This bit is valid only when ENCAAnSCE = 1; it is invalid when ENCAAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
3, 2	ENCAAnECS[1:0]	Encoder Clear Input Edge Selection 1 and 0 These are the encoder clear input edge selection bits. These bits are valid only when ENCAAnSCE = 0; they are invalid when ENCAAnSCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

**Table 26.12 ENCAAnIOC1 Register Contents (2/2)**

Bit Position	Bit Name	Function
1, 0	ENCAAnEIS[1:0]	<p>Encoder Edge Input Selection 1 and 0 These are the encoder input edge selection bits. These bits are valid when ENCAAnUDS1 and ENCAAnUDS0 = 00<sub>B</sub> or 01<sub>B</sub>, and are invalid when ENCAAnUDS1 and ENCAAnUDS0 = 10<sub>B</sub> or 11<sub>B</sub>.</p> <p>00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection</p>

### 26.3.5 ENCAAnFLG — ENCAAn Status Flag Register

This register holds the status flags of the timer counter of ENCAAn.

**Access:** This register can only be read in 8-bit units.

**Address:** <ENCAAn\_base> + 0C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENCAAnCSF	ENCAAnUDF	ENCAAnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 26.13 ENCAAnFLG Register Contents**

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	ENCAAnCSF	Counter Status Flag This bit reflects the current timer counter operation. 0: Timer counter in up-count status 1: Timer counter in down-count status
1	ENCAAnUDF	Underflow Flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared to 0 at the start of count operation. 0: This flag is cleared upon any of the following events: <ul style="list-style-type: none"><li>– “1” is written to ENCAAnFGC.ENCAAnCLUD</li><li>– The flag is cleared to 0 by setting ENCAAnTS bit to “1” when ENCAAnTE = 0 or by setting the simultaneous start trigger input (ENCATSSST signal) to “High”.</li></ul> 1: This flag is set to “1” upon occurrence of an underflow during the encoder timer count operation.
0	ENCAAnOVF	Overflow Flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared to 0 at the start of count operation. 0: This flag is cleared upon any of the following events: <ul style="list-style-type: none"><li>– “1” is written to ENCAAnFGC.ENCAAnCLOV</li><li>– The flag is cleared to 0 by setting ENCAAnTS bit to “1” when ENCAAnTE = 0 or by setting the simultaneous start trigger input (ENCATSSST signal) to “High”.</li></ul> 1: This flag is set to “1” upon occurrence of an overflow during the encoder timer count operation.

### 26.3.6 ENCAAnFGC — ENCAAn Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCAAnFLG.

**Access:** This register is a write-only register that can be written in 8-bit units.  
This register always returns 0 when read.

**Address:** <ENCAAn\_base> + 10H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ENCAAnCLUD	ENCAAnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 26.14 ENCAAnFGC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	ENCAAnCLUD	<p>Underflow Flag Clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCAAnUDF of the ENCAAnFLG register (clears underflow detection).</p>
0	ENCAAnCLOV	<p>Overflow Flag Clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCAAnOVF of the ENCAAnFLG register (clears overflow detection).</p>

### 26.3.7 ENCAAnCCR0 — ENCAAn Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

**Access:** This register can be read or written in 16-bit units.

When this register functions as a capture register, only reading is possible. Write operation is ignored.

When this register functions as a compare register, reading and writing is possible.

**Address:** <ENCAAn\_base> + 00H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAAnCCR0[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 26.15 ENCAAnCCR0 Register Contents**

Bit Position	Bit Name	Function
15 to 0	ENCAAnCCR0 [15:0]	<p>Capture/Compare Register 0</p> <p>Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCAAnCTL.ENCAAnLDE setting. See the description of the ENCAAnLDE bit in ENCA control register ENCAAnCTL for details.</p> <ul style="list-style-type: none"> <li>• If ENCAAnCTL.ENCAAnCRM0 = 0: ENCAAnCCR0 is a compare register. Set the value to be compared with the timer counter value.</li> <li>• If ENCAAnCTL.ENCAAnCRM0 = 1: ENCAAnCCR0 is a capture register. The captured timer counter value is stored.</li> </ul>

### 26.3.8 ENCAAnCCR1 — ENCAAn Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

**Access:** This register can be read or written in 16-bit units.

When this register functions as a capture register, only reading is possible. Write operation is ignored.

When this register functions as a compare register, reading and writing is possible.

**Address:** <ENCAAn\_base> + 04H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAAnCCR1[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 26.16 ENCAAnCCR1 Register Contents**

Bit Position	Bit Name	Function
15 to 0	ENCAAnCCR1 [15:0]	<p>Capture/Compare Register 1</p> <p>During capture operation, the capture trigger to this register differs according to the ENCAAnCTL.ENCAAnCTS setting. See the description of the ENCAAnCTS bit in ENCA control register ENCAAnCTL for details.</p> <ul style="list-style-type: none"> <li>• If ENCAAnCTL.ENCAAnCRM1 = 0: ENCAAnCCR1 is a compare register. Set the value to be compared with the timer counter value.</li> <li>• If ENCAAnCTL.ENCAAnCRM1 = 1: ENCAAnCCR1 is a capture register. The captured timer counter value is stored.</li> </ul>

### 26.3.9 ENCA<sub>n</sub>CNT — ENCA<sub>n</sub> Counter Register

This register is the 16-bit timer counter register.

**Access:** This register can be read or written in 16-bit units.  
This register can be written only when the operation is stopped.

**Address:** <ENCA<sub>n</sub>\_base> + 08<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCA <sub>n</sub> CNT[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.17 ENCA<sub>n</sub>CNT Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCA <sub>n</sub> CNT [15:0]	<p>Counter Register</p> <ul style="list-style-type: none"> <li>ENCA<sub>n</sub>TE.ENCA<sub>n</sub>TE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter.</li> <li>ENCA<sub>n</sub>TE.ENCA<sub>n</sub>TE status: 0 → 1 (operation start): Count operation start Up/down count operation is started with the set arbitrary value.</li> <li>ENCA<sub>n</sub>TE.ENCA<sub>n</sub>TE status: 1 (operating): Counting Up/down count operation is performed.</li> <li>ENCA<sub>n</sub>TE.ENCA<sub>n</sub>TE status: 1 → 0 (stopped): Count stop The counter value immediately before the operation was stopped is held, and the count operation is stopped.</li> </ul>

### 26.3.10 ENCA<sub>n</sub>TE — ENCA<sub>n</sub> Timer Enable Status Register

This register indicates the operating status of ENCA<sub>n</sub>.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <ENCA<sub>n</sub>\_base> + 14<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCA <sub>n</sub> TE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 26.18 ENCA<sub>n</sub>TE Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ENCA <sub>n</sub> TE	<p>Timer Status Enable</p> <p>This is a status bit that indicates the operation enabled/stopped status of ENCA<sub>n</sub>.</p> <p>This bit is cleared to 0 when “1” is written to ENCA<sub>n</sub>TT.ENCA<sub>n</sub>TT.</p> <p>This bit is set to “1” when “1” is written to ENCA<sub>n</sub>TS.ENCA<sub>n</sub>TS, or when the input signal of ENCATSST is set to High level.</p> <p>0: Operation stopped status 1: Operation enabled status</p>

### 26.3.11 ENCAnts — ENCA Timer Start Trigger Register

This register provides the trigger bit for setting the ENCA to the operation enabled state.

**Access:** This register is a write-only register that can be written in 8-bit units.  
It is always read as 00H. This register can be written only when ENCAnts.ENCAnts is 0.

**Address:** <ENCA\_base> + 18H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnts
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 26.19 ENCAnts Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnts	<p>Timer Start Trigger This is the trigger bit that sets the ENCA to the operation enabled state. 0: Writing is ignored. 1: The ENCA is set to the operation enabled state by setting ENCAnts.ENCAnts = 1.</p>

### 26.3.12 ENCA<sub>n</sub>TT — ENCA<sub>n</sub> Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCA<sub>n</sub> to the operation stopped state.

**Access:** This register is a write-only register that can be written in 8-bit units.  
It is always read as 00<sub>H</sub>.

**Address:** <ENCA<sub>n</sub>\_base> + 1C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCA <sub>n</sub> TT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

**Table 26.20 ENCA<sub>n</sub>TT Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCA <sub>n</sub> TT	<p>Timer Stop Trigger This is the trigger bit that sets the ENCA<sub>n</sub> to the operation stopped state. 0: Writing is ignored. 1: Clears ENCA<sub>n</sub>TE.ENCA<sub>n</sub>TE to “0”, to set the ENCA<sub>n</sub> to the count operation stopped state.</p>

### 26.3.13 ENCAAnEMU — ENCAAn Emulation Register

This register controls operations by SVSTOP.

**Access:** This register can be read or written in 8-bit units.  
Writing to this register should be performed in the counter operation stopped status (ENCAAnTE.ENCAAnTE = 0 and EPC.SVSTOP = 0).

**Address:** <ENCAAn\_base> + 48H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	ENCAAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

**Table 26.21 ENCAAnEMU Register Contents**

Bit Position	Bit Name	Function
7	ENCAAnSVSDIS	<ul style="list-style-type: none"> <li>When EPC.SVSTOP bit = 0: The count clock continues to be provided when the debugger assumes control of the microcontroller (at a break point, etc.), regardless of the value of this bit (1 or 0).</li> <li>When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger assumes control of the microcontroller (at a break point, etc.). 1: The count clock continues to be provided when the debugger assumes control of the microcontroller (at a break point, etc.).</li> </ul>
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 26.4 Operation

The ENCAAn operates the timer counter with counter up/down control and clear control by encoder inputs. The ENCAAnCCR0 and ENCAAnCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

### 26.4.1 Timer Counter Operation

The timer counter operations of the ENCAAn are described below.

The figure below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

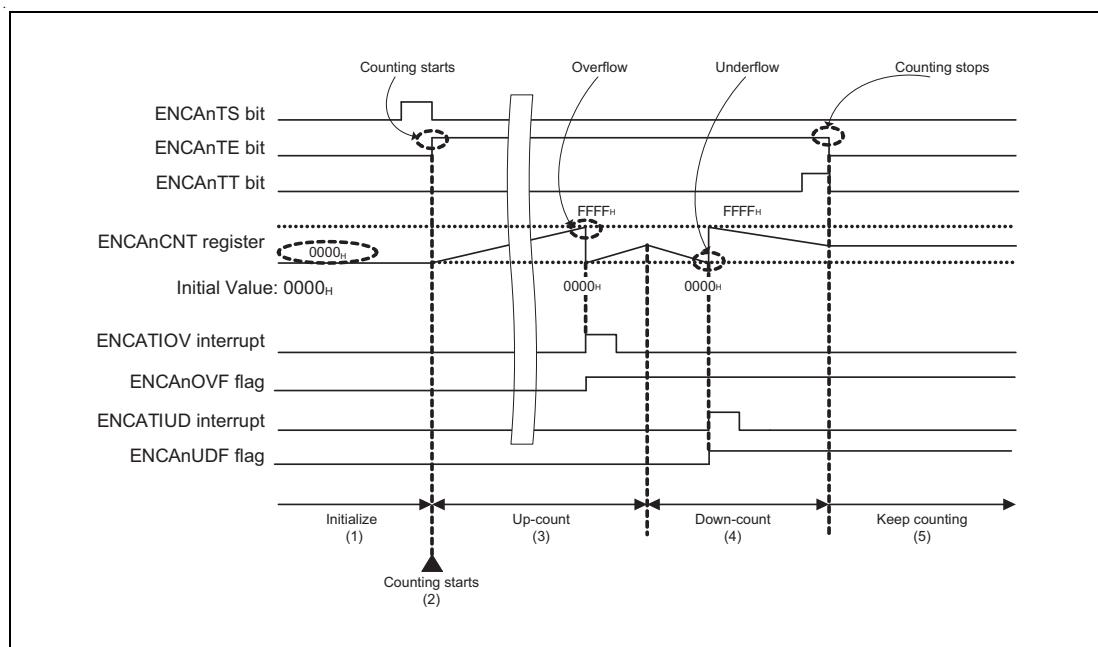


Figure 26.2 Timer Counter Initial Value Setting/Start/Stop

#### (1) Timer Counter Initial Value Setting

The initial value of the ENCAAn counter register (ENCAAnCNT) can be set in the counter operation stopped status (ENCAAnTE = 0).

#### (2) Timer Counter Startup

By writing “1” to the timer start trigger bit (ENCAAnTS), the timer status enable bit (ENCAAnTE) is set to “1”, the count operation is enabled, and counting operation is performed upon detection of the valid edge of the encoder input.

#### (3) Overflow Operation

An overflow occurs when up-counting is performed when the counter value is  $FFFF_{\text{H}}$ . If the counter value changes from  $FFFF_{\text{H}}$  to  $0000_{\text{H}}$ , an overflow interrupt (ENCATIOV) is generated, and the overflow flag (ENCAAnOVF) is set to “1”. The overflow flag (ENCAAnOVF) is cleared to “0” when “1” is set to the overflow flag clear bit (ENCAAnCLOV). For details about the operation, see **Section 26.6.1, Overflow Occurrence and Overflow Flag Clear Operation**.

#### (4) Underflow Operation

An underflow occurs when down-counting is performed when the counter value is  $0000_H$ . If the counter value changes from  $0000_H$  to  $FFFF_H$ , an underflow interrupt (ENCATIUD) is generated, and the underflow flag (ENCAAnUDF) is set to “1”. The underflow flag (ENCAAnUDF) is cleared to “0” when “1” is set to the underflow flag clear bit (ENCAAnCLUD). For details about the operation, see **Section 26.6.2, Underflow Occurrence and Underflow Flag Clear Operation.**

#### (5) Timer Counter Stop

By writing “1” to the timer stop trigger bit (ENCAAnTT), the timer status enable bit (ENCAAnTE) is cleared to “0”, and the count operation is stopped. At this time, the timer counter is not reset to  $0000_H$  and holds the value before count operation stop.

### 26.4.2 Up/Down Control of Timer Counter

Up/down control is performed by judging the phase of the encoder inputs (ENCAnE0, ENCAnE1) according to the settings of ENCAAnUDS1 and ENCAAnUDS0.

#### 26.4.2.1 When the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 00<sub>B</sub>

Table 26.22 When ENCAAnUDS1/ENCAAnUDS0 Bits = 00<sub>B</sub>

ENCAAnUDS1	ENCAAnUDS0	Operation Description		
		ENCAAnE0 Pin	ENCAAnE1 Pin	Count Operation
0	0	Rising edge	High level	Down
		Falling edge		
		Rising and falling edges		
		Rising edge	Low level	Up
		Falling edge		
		Rising and falling edges		

The valid edge for ENCAAnE0 is specified by setting ENCAAnEIS1 and ENCAAnEIS0.

Up/down count operation is performed when the valid edges and levels of ENCAAnE0 and ENCAAnE1 match.

The following timing chart shows the count operation when ENCAAnUDS1 and ENCAAnUDS0 bits = 00<sub>B</sub>.

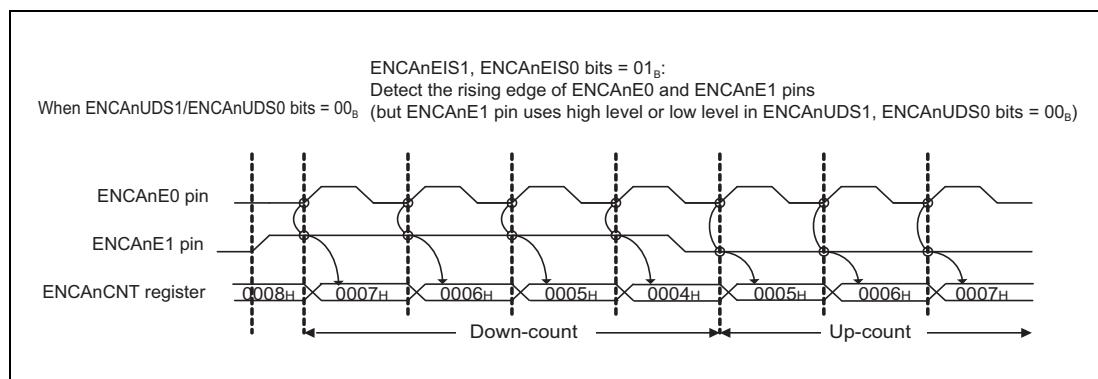


Figure 26.3 Count Operation when the ENCAAnUDS1/ENCAAnUDS0 Bits in the ENCAAnCTL Register = 00<sub>B</sub>

### 26.4.2.2 When the ENCA<sub>n</sub>UDS1/ENCA<sub>n</sub>UDS0 Bits in the ENCA<sub>n</sub>CTL Register = 01<sub>B</sub>

Table 26.23 When the ENCA<sub>n</sub>UDS1/ENCA<sub>n</sub>UDS0 Bits = 01<sub>B</sub>

ENCA <sub>n</sub> UDS1	ENCA <sub>n</sub> UDS0	Operation Description		
		ENCA <sub>n</sub> E0 Pin	ENCA <sub>n</sub> E1 Pin	Count Operation
0	1	Low level	Rising edge	Down
			Falling edge	
			Rising and falling edges	
		High level	Rising edge	Up
			Falling edge	
			Rising and falling edges	
		Rising edge	Low level	
			Falling edge	
			Rising and falling edges	
		Simultaneous input		Hold

The valid edges for ENCA<sub>n</sub>E0 and ENCA<sub>n</sub>E1 are specified by setting ENCA<sub>n</sub>EIS1 and ENCA<sub>n</sub>EIS0.

Up/down count operation is performed when the valid edges and levels of the ENCA<sub>n</sub>E0/ENCA<sub>n</sub>E1 pins match, and the count is held when the valid edges overlap.

The following timing chart shows the count operation when ENCA<sub>n</sub>UDS1 and ENCA<sub>n</sub>UDS0 bits = 01<sub>B</sub>.

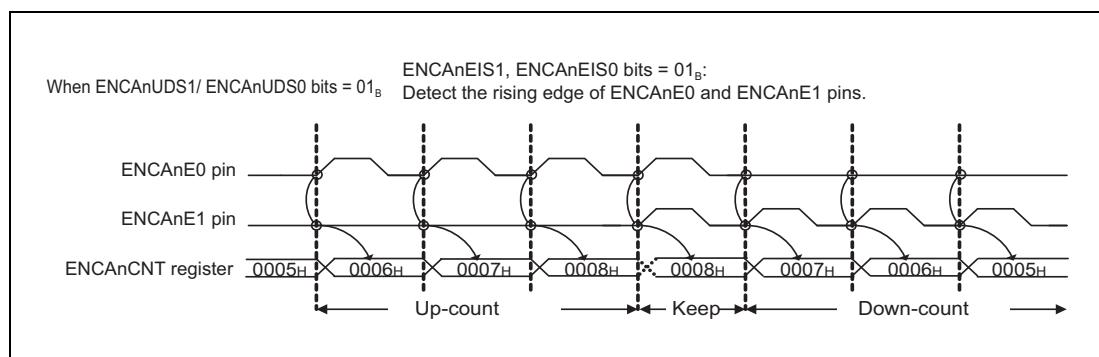


Figure 26.4 Count Operation when the ENCA<sub>n</sub>UDS1/ENCA<sub>n</sub>UDS0 Bits in the ENCA<sub>n</sub>CTL Register = 01<sub>B</sub>

### 26.4.2.3 When the ENCA<sub>n</sub>UDS1 and ENCA<sub>n</sub>UDS0 Bits in the ENCA<sub>n</sub>CTL Register = 10<sub>B</sub>

Table 26.24 When the ENCA<sub>n</sub>UDS1, ENCA<sub>n</sub>UDS0 Bits = 10<sub>B</sub>

ENCA <sub>n</sub> UDS1	ENCA <sub>n</sub> UDS0	Operation Description		
		ENCA <sub>n</sub> E0 Pin	ENCA <sub>n</sub> E1 Pin	Count Operation
1	0	Rising edge	Low level	Down
		Rising edge	Falling edge	
		Falling edge	Low level	Up
		Falling edge	Falling edge	
		Low level	Rising edge	Hold
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	

The valid edge specification for ENCA<sub>n</sub>E0 and ENCA<sub>n</sub>E1 (settings of ENCA<sub>n</sub>EIS1 and ENCA<sub>n</sub>EIS0) is invalid.

The following timing chart shows the count operation when the ENCA<sub>n</sub>UDS1/ENCA<sub>n</sub>UDS0 bits = 10<sub>B</sub>.

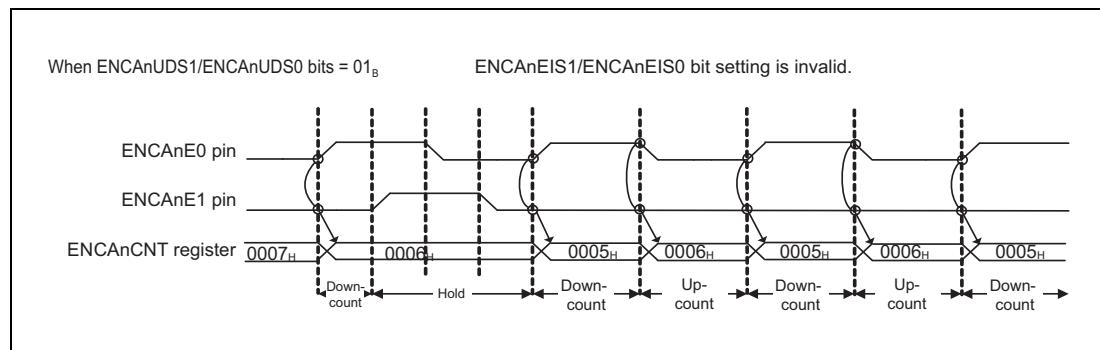


Figure 26.5 Count Operation when ENCA<sub>n</sub>UDS1/ENCA<sub>n</sub>UDS0 Bits in ENCA<sub>n</sub>CTL Register = 10<sub>B</sub>

#### 26.4.2.4 When ENCANUDS1/ENCANUDS0 Bits in the ENCANCTL Register = 11<sub>B</sub>

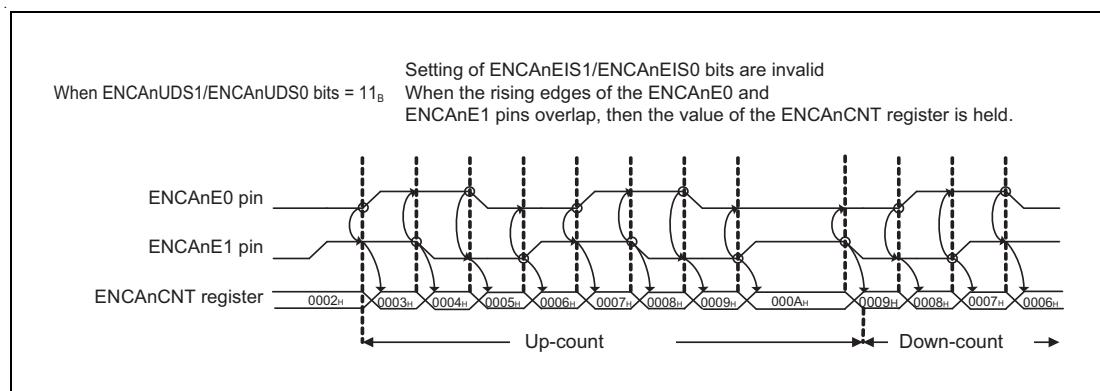
**Table 26.25** When ENCANUDS1/ENCAnUDS0 Bits = 11<sub>B</sub>

ENCAAnUDS1	ENCAAnUDS0	Operation Description		
		ENCAAnE0 Pin	ENCAAnE1 Pin	Count Operation
1	1	Low level	Falling edge	Down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		Hold

Valid edge specification for ENCAne0 and ENCAne1 (settings of ENCAneis1 and ENCAneis0) is invalid.

The counter value is held when the valid edges of ENCA<sub>n</sub>E0 and ENCA<sub>n</sub>E1 overlap.

The following timing chart shows the count operation when ENCAAnUDS1/ENCAAnUDS0 bits = 11<sub>B</sub>.



**Figure 26.6** Count Operation when ENCAnUDS1 and ENCAnUDS0 Bits in the ENCAnCTL Register =  $11_B$

### 26.4.3 Timer Counter Clear Control by Encoder Input

The timer counter is cleared to  $0000_H$  by encoder clear input (ENCAAnEC).

Two types of clearing methods can be selected by controlling the ENCAAnSCE, ENCAAnZCL, ENCAAnBCL, ENCAAnACL, ENCAAnECS1, and ENCAAnECS0 bits of the ENCAAnIOC1 register.

**Table 26.26 Timer Counter Clear Control by Encoder Input**

Clearing method	ENCAAnSCE	ENCAAnZCL	ENCAAnBCL	ENCAAnACL	ENCAAnECS1, ENCAAnECS0
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

#### 26.4.3.1 Clearing Method when ENCAAnSCE = 0

- Upon detection of the valid edge of ENCAAnEC, the timer counter is cleared to  $0000_H$  in synchronization with the operation clock.
- The valid edge of ENCAAnEC is specified by the setting of the ENCAAnECS1 and ENCAAnECS0 bits.
- The settings of the ENCAAnZCL, ENCAAnBCL, and ENCAAnACL bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

For details about clear operation when ENCAAnSCE = 0, see the timing chart in **Section 26.6.19, Capture Operation Performed upon Clearing by ENCAAnEC when ENCAAnSCE = 0**.

#### 26.4.3.2 Clearing Method when ENCAAnSCE = 1

- When the clear levels of the ENCAAnEC, ENCAAnE1, ENCAAnE0 inputs are detected, the timer counter is cleared to  $0000_H$  in synchronization with the operating clock.
- Specify the clear levels of the ENCAAnEC, ENCAAnE1, ENCAAnE0 inputs by the settings of the ENCAAnZCL, ENCAAnBCL, and ENCAAnACL bits.
- The settings of the ENCAAnECS1 and ENCAAnECS0 bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCAAnZCL, ENCAAnBCL, and ENCAAnACL settings are listed in the table below.

**Table 26.27 Clearing Conditions of the Timer Counter**

Counter Clear Condition Setting			Encoder Pin Input Level		
ENCAAnZCL	ENCAAnBCL	ENCAAnACL	ENCAAnEC	ENCAAnE1	ENCAAnE0
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

## 26.4.4 Functions of ENCAAnCCR0

### 26.4.4.1 Compare Function

- When ENCAAnCRM0 = 0, the ENCAAnCCR0 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAAnCCR0 setting value, a compare 0 match interrupt (ENCATINT0) is output.
- When ENCAAnECM0 = 1, the timer counter is cleared to 0000<sub>H</sub> in synchronization with the operating clock upon compare match if the next count operation is up-count.

Table 26.28 Compare Function of ENCAAnCCR0

ENCAAnCCR0 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCAAnCCR0
ENCAAnCRM0	ENCAAnECM0		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Clears timer counter to 0000 <sub>H</sub> .
		Down-count	Does not clear (continues count operation).

#### When ENCAAnLDE = 1

- Upon occurrence of an underflow, the setting value of the ENCAAnCCR0 register is loaded to the timer counter.
- An underflow interrupt (ENCATIUD) is output.

#### NOTE

For the timing chart when ENCAAnLDE = 1, see **Section 26.6.8, Using the ENCAAnLDE Function Immediately after Startup** to **Section 26.6.12, Up-count after Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input.**

### 26.4.4.2 Capture Function

- When ENCAAnCRM0 = 1, the ENCAAnCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCATTIN0), the value of the timer counter is stored into ENCAAnCCR0.
- A capture 0 interrupt (ENCATINT0) is output during capture operation.

#### NOTE

For details about capture operation for ENCAAnCCR0, see the timing charts in **Section 26.6.14, Capture Operation between Count Clocks (ENCAAnCCR0)** and **Section 26.6.17, Encoder Operation when Compare Match Clear Control is Disabled**.

## 26.4.5 Functions of ENCAAnCCR1

### 26.4.5.1 Compare Function

- When ENCAAnCRM1 = 0, the ENCAAnCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAAnCCR1 setting value, a compare 1 match interrupt (ENCATINT1) is output.
- When ENCAAnECM1 = 1, the timer counter is cleared to 0000<sub>H</sub> in synchronization with the operating clock upon compare match if the next count operation is down-count.

Table 26.29 Compare Function of ENCAAnCCR1

ENCAAnCCR1 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCAAnCCR1
ENCAAnCRM1	ENCAAnECM1		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Does not clear (continues count operation).
		Down-count	Clears timer counter to 0000 <sub>H</sub> .

### Compare match interrupt mask function

- When ENCAAnCME = 1, the compare 1 match interrupt mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCAAnCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCAAnCME = 1 and ENCAAnMCS = 0, a compare 1 match interrupt is output once upon the first compare match by writing to the ENCAAnCCR1 register (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCAAnCME = 1 and ENCAAnMCS = 1, a compare 1 match interrupt is output once upon the first compare match by a timer counter clear operation accompanying encoder clear input or by a timer counter clear operation upon match between the ENCAAnCCR0 register value and the timer counter value (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCAAnCME = 1, ENCAAnMCS = 1 and ENCAAnLDE = 1, a compare 1 match interrupt is output once upon the first compare match by a loading operation of the ENCAAnCCR0 register to the timer counter upon underflow detection (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- Setting ENCAAnECM1 to “1” is prohibited when enabling the compare 1 match interrupt mask function.

Table 26.30 Compare Match Interrupt Mask Function of ENCAAnCCR1

ENCAAnCCR1 Function	Compare 1 Match Interrupt Mask	Interrupt Mask Cancel Trigger	Compare 1 Match Interrupt Output upon Compare Match with ENCAAnCCR1
ENCAAnCRM1	ENCAAnCME	ENCAAnMCS	
0 (Compare)	0 (Mask function disabled)	— (Setting invalid)	Outputs compare 1 match interrupt upon each compare match.
	1 (Mask function enabled)	0 (Write operation to ENCAAnCCR1)  1 (Timer counter clear operation) (Loading from ENCAAnCCR0 to the timer counter upon underflow occurrence when ENCAAnLDE = 1)	Outputs compare 1 match interrupt once upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)

#### 26.4.5.2 Capture Function

When ENCAAnCRM1 = 1, the ENCAAnCCR1 register functions as a dedicated capture register.

##### NOTE

For details about capture operation to ENCAAnCCR1, see the timing chart in **Section 26.6.13, Capture Operation between Count Clocks (ENCAAnCCR1)**.

The operations for each of the ENCAAnCTS settings are shown in the table below.

Table 26.31 Operations for Each of the ENCAAnCTS Settings

ENCAAnCCR1 Function	Capture Trigger Selection	Capture Trigger Signal	Timer Counter Clearing	Interrupt Occurrence
ENCAAnCRM1	ENCAAnCTS			
1 (Capture)	0	Capture trigger 1 input (ENCATTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (ENCATINT1)
	1	Encoder clear input (set with ENCAAnSCE)	Clears timer counter.	(1) Capture 1 interrupt (ENCATINT1) (2) Encoder clear interrupt (ENCATIEC)

##### NOTE

For details about the timing chart when ENCAAnCTS = 0 or ENCAAnCTS = 1, see the following:

**Section 26.6.3, Count Clearing and Capture Operation by Encoder Clear Input (ENCAAnEC Pin), Section 26.6.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin), Section 26.6.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin), Section 26.6.11, Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAAnEC Pin) and Section 26.6.12, Up-count after Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input.**

### 26.4.5.3 Timer Counter Clearing upon Compare Register Match

Timer counter clearing upon compare match between the value of the timer counter and the ENCAAnCCR0/1 setting value, according to the settings of the ENCAAnECM1 and ENCAAnECM0 bits in the ENCAAnCTL register, is detailed in the following table.

**Table 26.32 Timer Counter Clearing Operation upon Compare Register Match**

ENCAAnECM1 and ENCAAnECM0	Next Count Operation	Timer Counter Clearing upon Compare Match with ENCAAnCCR1	Timer Counter Clearing upon Compare Match with ENCAAnCCR0
00	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
01	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000 <sub>H</sub> .
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
10	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Clears timer counter to 0000 <sub>H</sub> .	Does not clear (continues count operation).
11	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000 <sub>H</sub> .
	Down-count	Clears timer counter to 0000 <sub>H</sub> .	Does not clear (continues count operation).

## 26.4.6 Startup/Stop of Timer Counter

### 26.4.6.1 Startup of Timer

The timer operation can be started by setting the ENCAAnTS bit to “1”.

PIC setting enables simultaneous start with other timers. For details, see **Section 27.8, Simultaneous Start Trigger Function**.

### 26.4.6.2 Stop of Timer

When the ENCAAnTT bit is set to “1”, the ENCAAnTE bit becomes “0” and the timer stops.

## 26.5 ENCA Setting Sequences

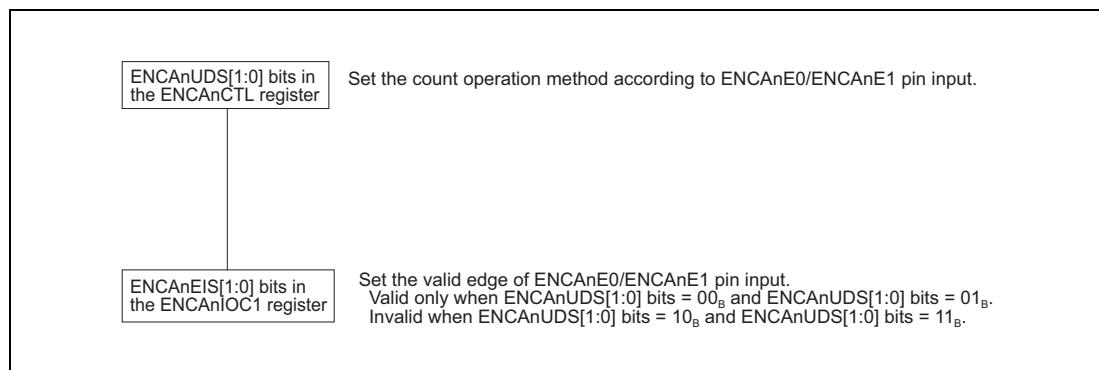
### 26.5.1 ENCA Setting Procedure

The ENCA setting procedure is described below.

**Table 26.33 ENCA Setting Procedure**

Initial Setting	Action	Setting status
Initial setting	Reset deassertion	Power-on status, operation stopped status. (Writing to each register is enabled)
ENCA initial setting	Perform the following initial settings. <ul style="list-style-type: none"> <li>• Setting for counter</li> <li>• Setting for counter clear</li> <li>• Setting for ENCAAnCCR0 register</li> <li>• Setting for ENCAAnCCR1 register</li> </ul>	This is the count operation stopped status. The value of the ENCAAnTE bit indicating the operating status is 0.
	Perform the counter initial value settings. <ul style="list-style-type: none"> <li>• Set any 16-bit value to ENCAAnCNT register. (When, after setting this register, the ENCAAnTS bit is set to "1", the counter operation starts from the set count value.)</li> </ul>	The set value is set as the initial value of the counter register.
Operation start	Perform the counter operation start setting. <ul style="list-style-type: none"> <li>• Set the ENCAAnTS bit to "1".</li> </ul>	This is the counter operation starts status. The value of the ENCAAnTE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. <ul style="list-style-type: none"> <li>• ENCAAnCCR0 register setting.</li> <li>• ENCAAnCCR1 register setting.</li> <li>• ENCAAnIOC0 register setting.</li> </ul>	The count operation set with the initial setting is performed, and up/down counting is performed according to ENCAAnE0 and ENCAAnE1 pins.
Operation stop	Perform the counter operation stop setting during operation. <ul style="list-style-type: none"> <li>• Set the ENCAAnTT bit to "1".</li> </ul>	This is the counter operation stopped status. The value of the ENCAAnTE bit indicating the operating status is 0.
ENCA stop	Reset	The setting registers are initialized.

#### 26.5.1.1 Initial Setting Procedure for the Counter



**Figure 26.7 Initial Setting Procedure for the Counter**

### 26.5.1.2 Initial Setting Procedure for Counter Clear

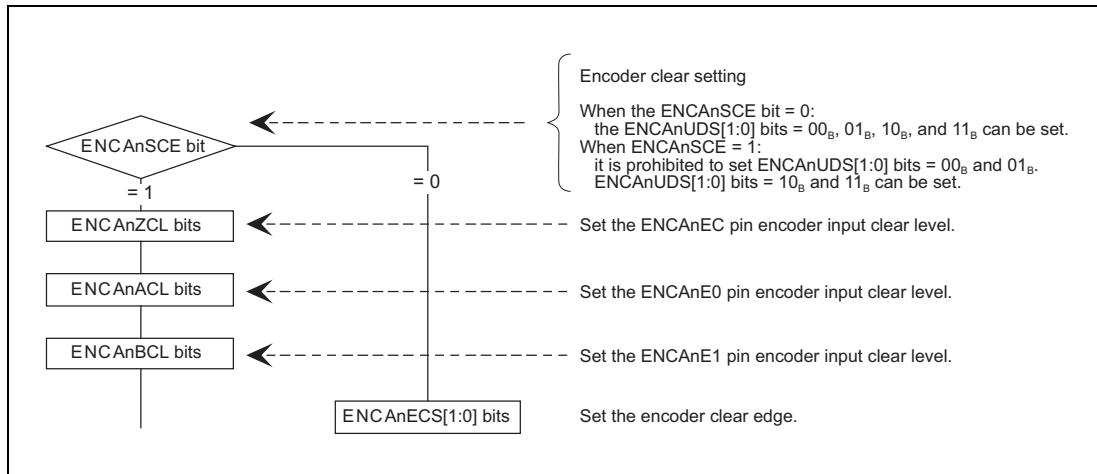


Figure 26.8 Initial Setting Procedure for Counter Clear

### 26.5.1.3 Setting Procedure for ENCAAnCCR0 Register

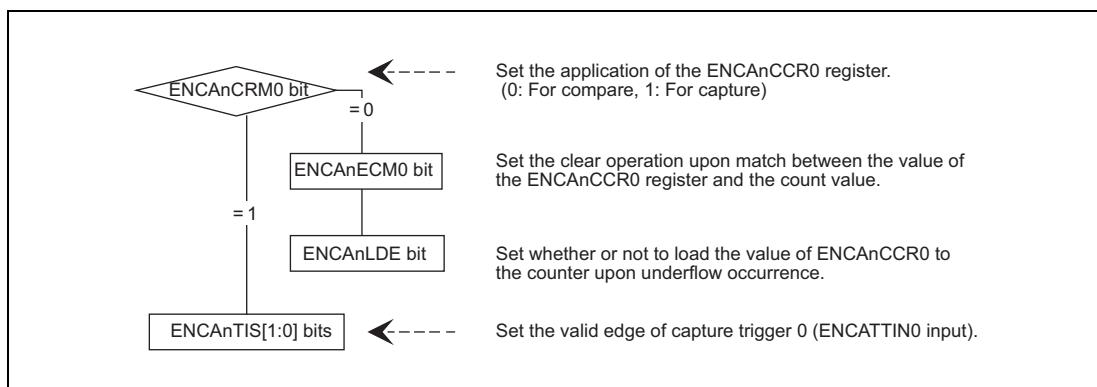


Figure 26.9 Setting Procedure for ENCAAnCCR0 Register

### 26.5.1.4 Setting Procedure for ENCAAnCCR1 Register

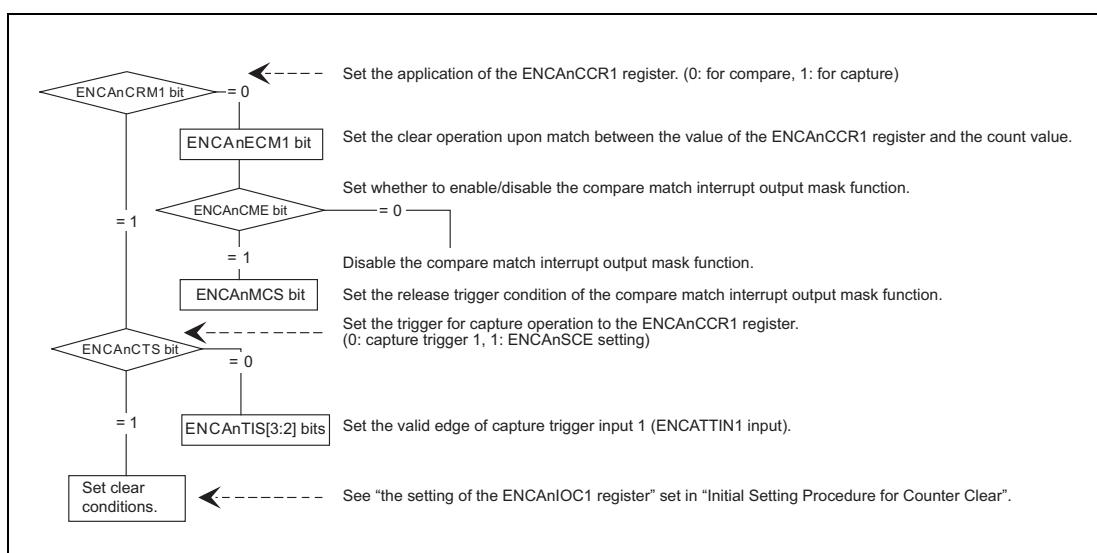


Figure 26.10 Setting Procedure for ENCAAnCCR1 Register

## 26.6 Timing Chart

### 26.6.1 Overflow Occurrence and Overflow Flag Clear Operation

An overflow occurs when up-counting is performed when the counter value is  $FFFF_{H}$ . Once an overflow occurs, an overflow interrupt (ENCATIOV) is output and the overflow flag (ENCAAnOVF) is set to 1. When the overflow clear bit (ENCAAnCLOV) is set to 1, the overflow flag (ENCAAnOVF) is cleared to 0.

The operations of overflow occurrence and overflow flag clearing are described below.

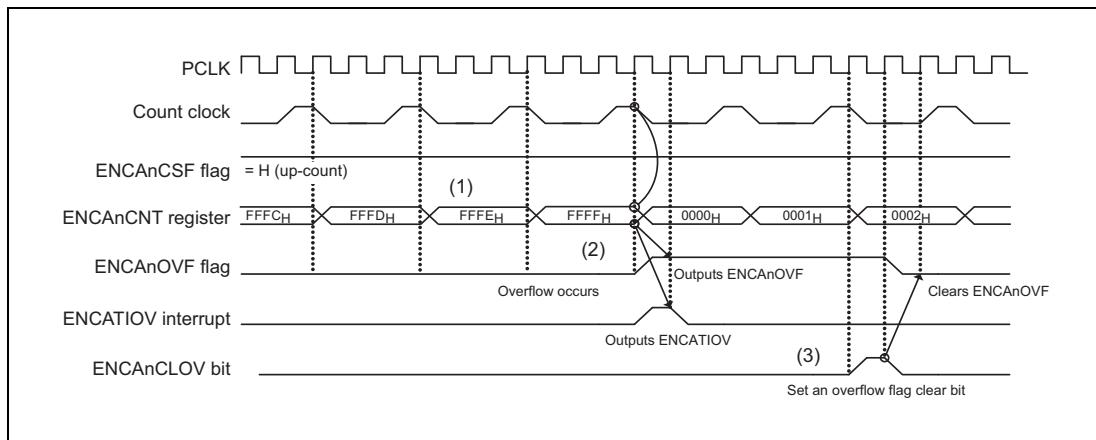


Figure 26.11 Settings of Overflow Occurrence and Overflow Flag Clear

- (1) The count value is counted up from  $FFFF_{H}$  to  $FFFF_{H}$ .
- (2) When the count value changes from  $FFFF_{H}$  to  $0000_{H}$ , an overflow occurs. At the same time, an overflow interrupt is output and the overflow flag is set to 1.
- (3) By setting the ENCAAnCLOV bit in the ENCAAnFGC register to 1 by the overflow flag clearing method, the overflow flag is cleared to 0. The overflow flag is also cleared by setting the ENCAAnTS bit in the ENCAAnTS register to 1 when the ENCAAnTE bit in the ENCAAnTE register is 0, or setting the input signal of ENCATSST (simultaneous start trigger input) to "High".

## 26.6.2 Underflow Occurrence and Underflow Flag Clear Operation

An underflow occurs when down-counting is performed when the counter value is  $0000_{\text{H}}$ .

Once an underflow occurs, an underflow interrupt (ENCATIUD) is output and the underflow flag (ENCAAnUDF) is set to 1. When the underflow clear bit (ENCAAnCLUD) is set to 1, the underflow flag (ENCAAnUDF) is cleared to 0.

The operations of underflow occurrence and underflow flag clearing are described below.

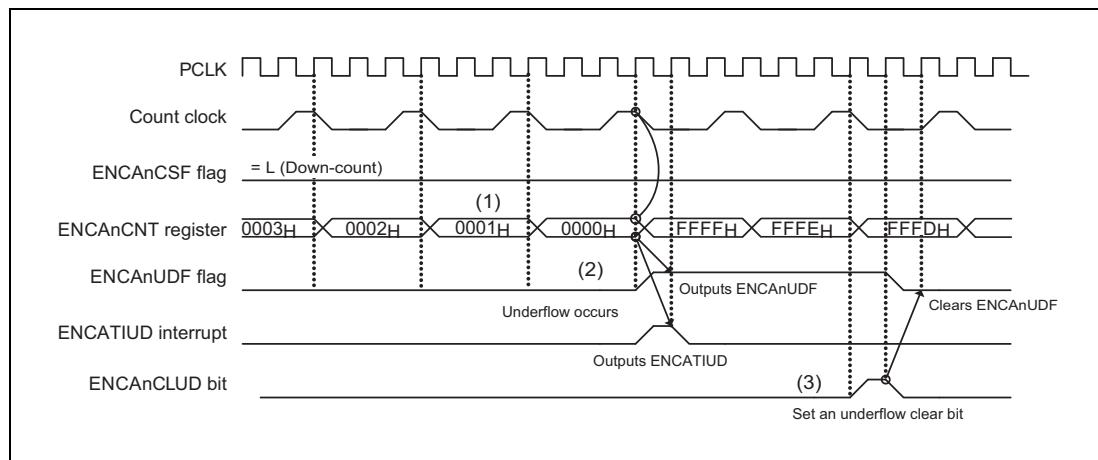


Figure 26.12 Settings of Underflow Occurrence and Underflow Flag Clear

- (1) The count value is counted down from  $0001_{\text{H}}$  to  $0000_{\text{H}}$ .
- (2) When the count value changes from  $0000_{\text{H}}$  to  $FFFF_{\text{H}}$ , an underflow occurs. At the same time, an underflow interrupt is output and the underflow flag is set to 1.
- (3) By setting the ENCAAnCLUD bit in the ENCAAnFGC register to 1 by the underflow flag clearing method, the underflow flag is cleared to 0. The underflow flag is also cleared by setting the ENCAAnTS bit in the ENCAAnTS register to 1 when the ENCAAnTE bit in the ENCAAnTE register is 0, or by setting the input signal of ENCAtSST (simultaneous start trigger) to "High".

### 26.6.3 Count Clearing and Capture Operation by Encoder Clear Input (ENCAAnEC Pin)

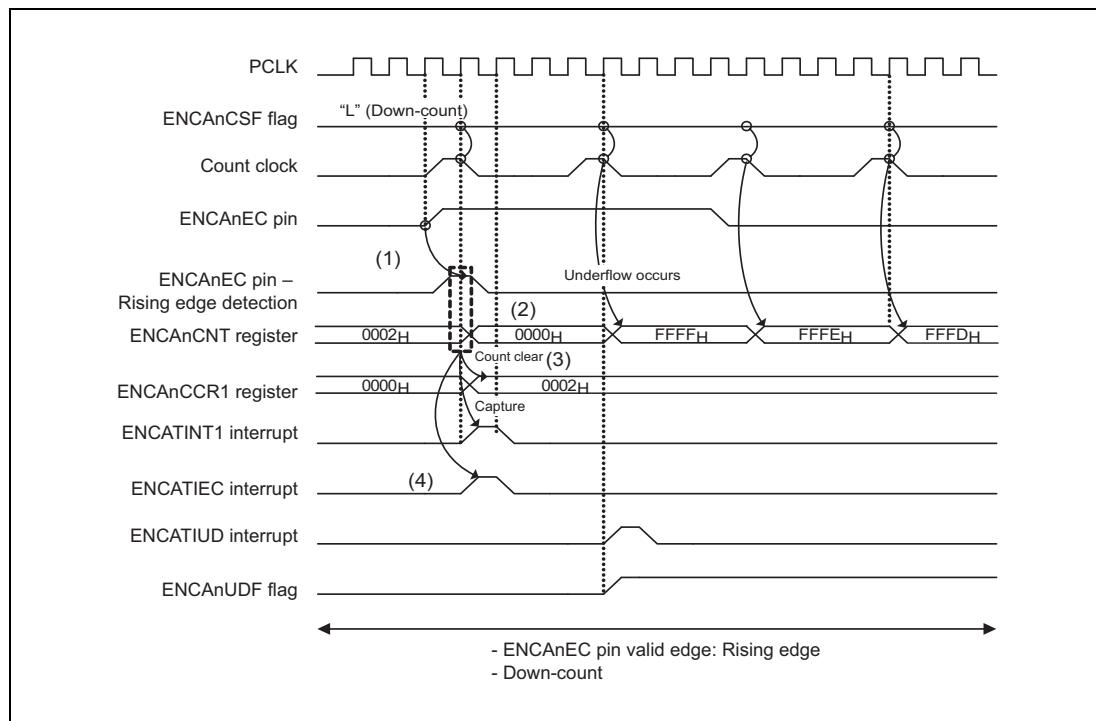
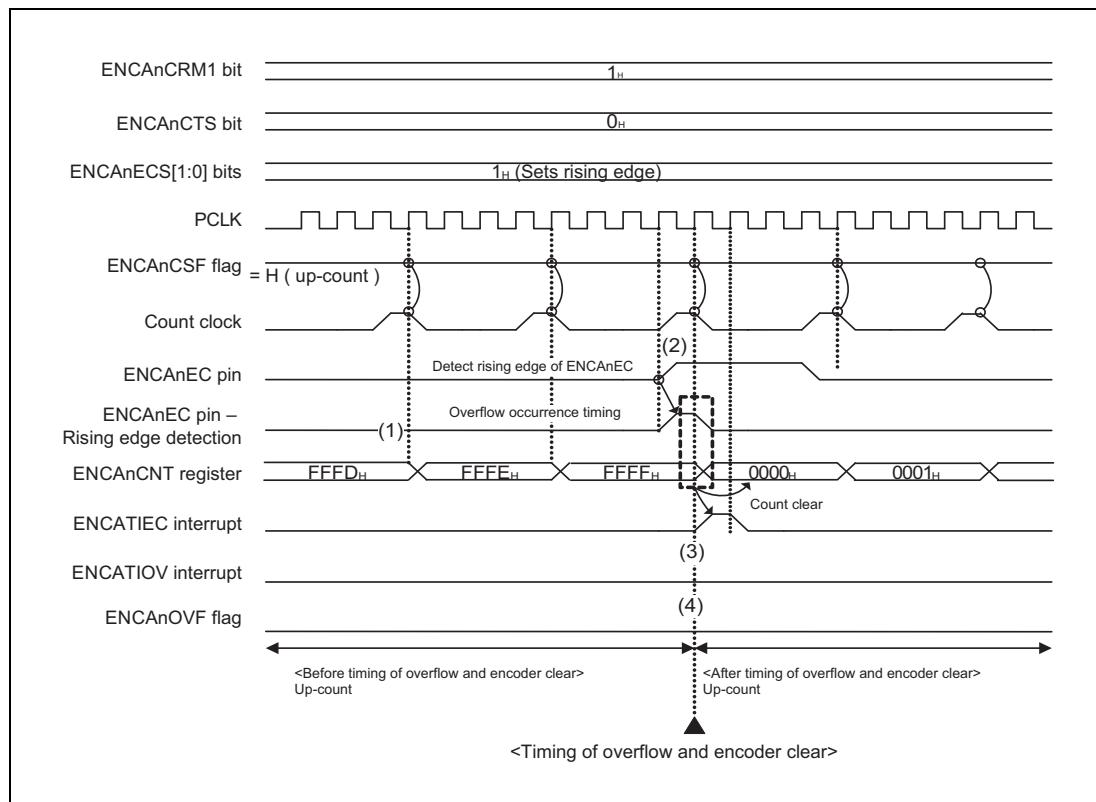


Figure 26.13 Timing Chart of Count Clearing and Capture Operation by Encoder Clear Input (ENCAAnEC Pin)

#### Setting conditions

- ENCAAnCRM1 bit in the ENCAAnCTL register = 1  
(Select the ENCAAnCCR1 register as capture.)
  - ENCAAnCTS bit in the ENCAAnCTL register = 1  
(Select the ENCAAnEC pin input as capture trigger input.)
  - ENCAAnECS1 and ENCAAnECS0 bits in the ENCAAnIOC1 register = 01<sub>B</sub>  
(Select the ENCAAnEC pin input as rising edge detection.)
- (1) Capture operation is performed by the rising edge of the ENCAAnEC pin input trigger.
  - (2) Clearing is performed by the ENCAAnEC pin input and the count value is set to 0000<sub>H</sub>.
  - (3) The counter value (0002<sub>H</sub>) is captured in the ENCAAnCCR1 register by the rising edge of the ENCAAnEC pin input.
  - (4) At the same time, a clear interrupt (ENCATIEC) and capture interrupt (ENCATINT1) due to the ENCAAnEC pin input are output.

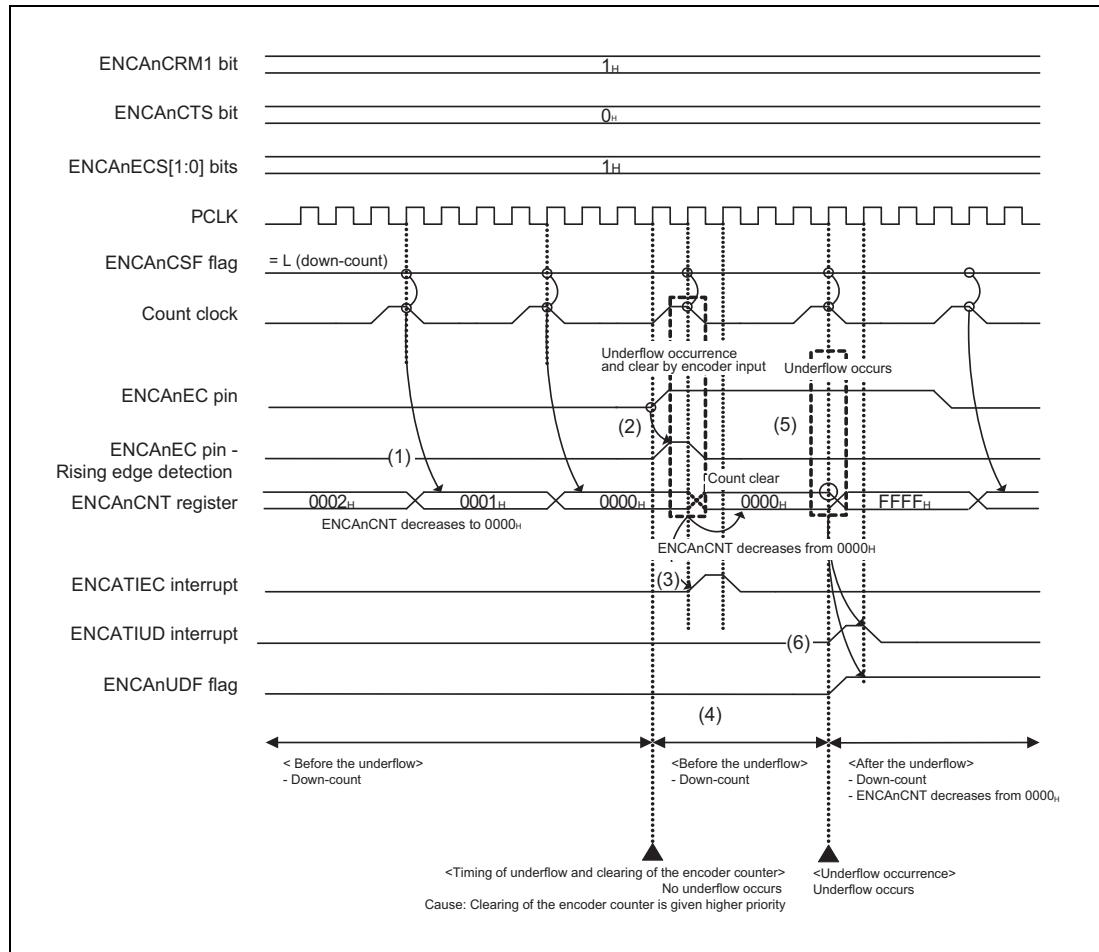
#### 26.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)



**Figure 26.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC pin)**

- (1) An up-count from  $FFFFD_H$  is continuously performed.
- (2) When an overflow occurs if the count value is  $FFFF_H$ , and the rising edge of ENCAAnEC is detected simultaneously, clear operation by the encoder clear input is performed. The counter value is cleared to  $0000_H$ .
- (3) When the counter value is cleared by the encoder clear input, a clear interrupt (ENCATIEC) by encoder clear input is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output (An overflow does not occur. Clear operation is performed by the encoder clear input).
- (4) Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

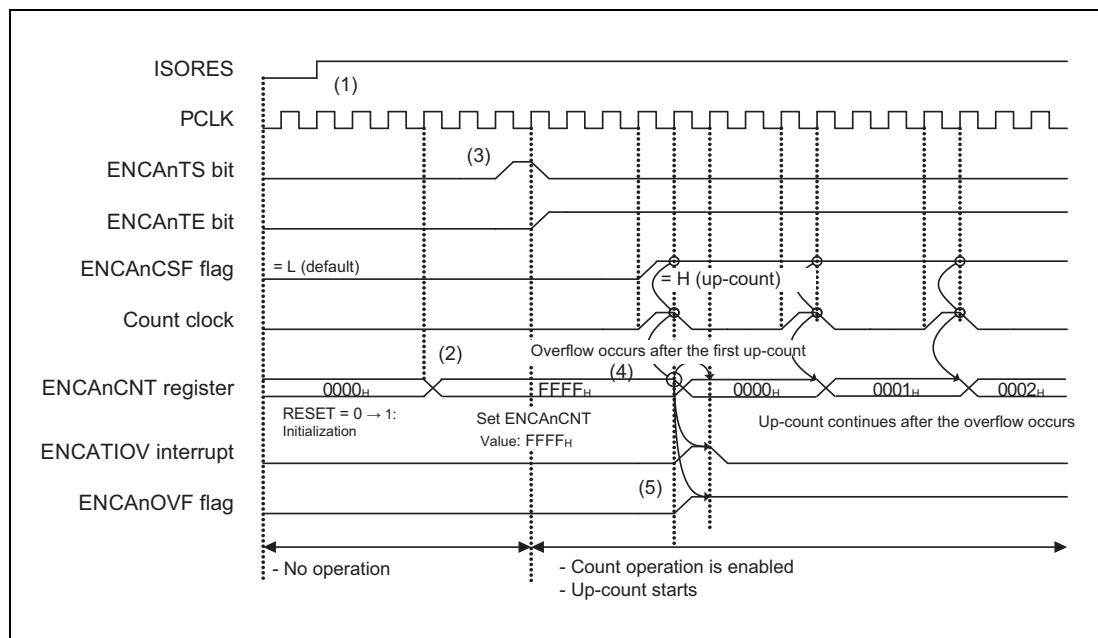
### 26.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin)



**Figure 26.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAAnEC Pin)**

- (1) A down-count from 0002H is continuously performed.
- (2) When an underflow occurs if the count value is 0000H, and the rising edge of ENCAAnEC is detected simultaneously, clear operation by the encoder clear input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000H.
- (3) When the counter value is cleared by the encoder clear input, an encoder clear interrupt (ENCATIEC) is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output (An underflow does not occur. Clear operation is performed by the encoder clear input).
- (4) Because an underflow does not occur as is the case with step 3, the underflow flag is not set.
- (5) When a further down-count is performed after the counter value changes to 0000H by clear operation by the encoder clear input, the counter value changes from 0000H to FFFFH, and an underflow occurs.
- (6) When an underflow occurs, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAAnUDF) is set.

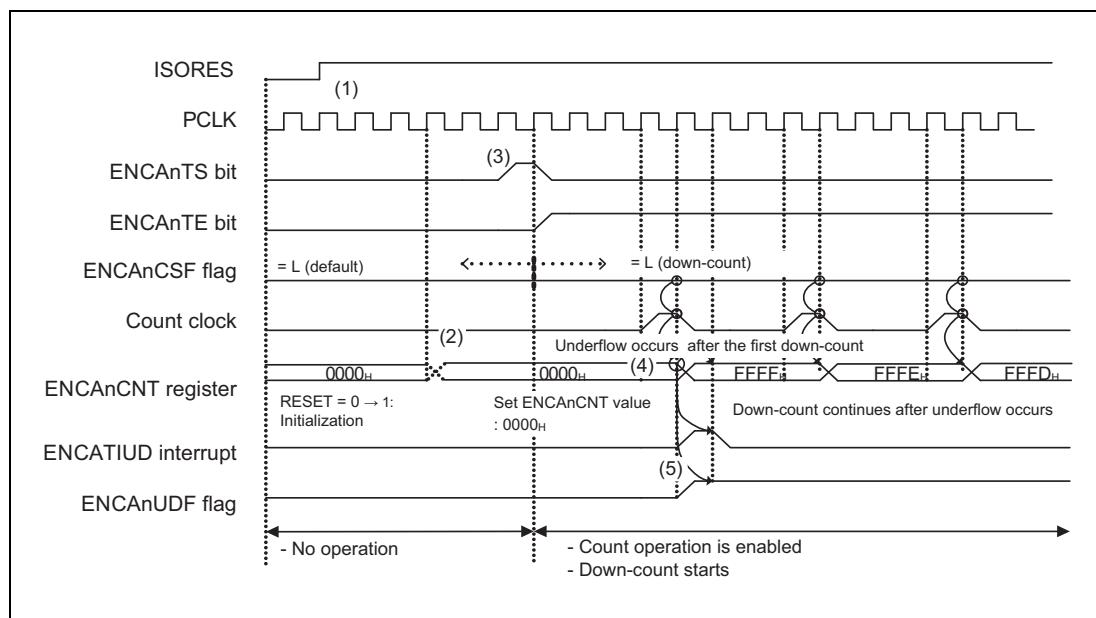
### 26.6.6 Overflow Operation Immediately after Startup



**Figure 26.16 Overflow Operation Immediately after Startup**

- (1) When the ISORES value changes from “0” to “1”, the status changes from “reset asserted” to “reset deasserted”.
- (2) The timer counter is set to  $\text{FFFF}_H$  as the initial value.
- (3) ENCAAnTS is set to “1”, and operation starts. ENCAAnTE changes to “1”, which indicates that operation is enabled.
- (4) When an up-count is performed from  $\text{FFFF}_H$  which is the initially set count value, the counter value changes from  $\text{FFFF}_H$  to  $0000_H$ , and an overflow occurs immediately after operation starts.
- (5) At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (ENCATIOV) is output, and the overflow flag (ENCAAnOVF) is set.

### 26.6.7 Underflow Operation Immediately after Startup



**Figure 26.17 Underflow Operation Immediately after Startup**

- (1) When the ISORES value changes from “0” to “1”, the status changes from “reset asserted” to “reset deasserted”.
- (2) The timer counter is set to 0000<sub>H</sub> as the initial value.
- (3) ENCAAnTS is set to “1”, and operation starts. ENCAAnTE changes to “1”, which indicates that operation is enabled.
- (4) When a down-count is performed from 0000<sub>H</sub> which is the initially set count value, the counter value changes from 0000<sub>H</sub> to FFFF<sub>H</sub>, and an underflow occurs immediately after operation starts.
- (5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATTIUD) is output, and the underflow flag (ENCAAnUDF) is set.

### 26.6.8 Using the ENCAAnLDE Function Immediately after Startup

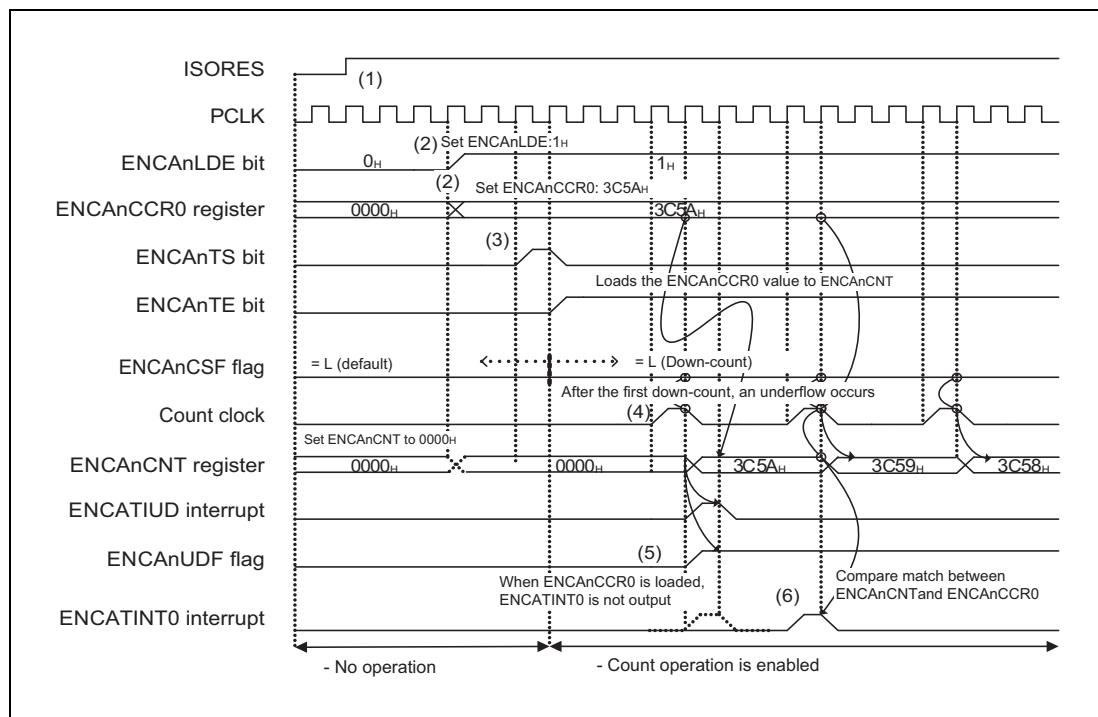


Figure 26.18 Using the ENCAAnLDE Function Immediately after Startup

- (1) When the ISORES value changes from “0” to “1”, the status changes from “reset asserted” to “reset deasserted”.
- (2) The load enable bit (ENCAAnLDE) is set to “1”, capture/compare register 0 (ENCAAnCCR0) is set to  $3C5A_{H}$ , and the timer counter is set to the initial value  $0000_{H}$ .
- (3) ENCAAnTS is set to “1”, and operation starts. ENCAAnTE changes to “1”, which indicates that operation is enabled.
- (4) When a down-count is performed from  $0000_{H}$  which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCAAnLDE is set to “1”, the ENCAAnCCR0 value,  $3C5A_{H}$ , is loaded to the timer counter (ENCAtINT0 is not output during loading).
- (5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCAtIUD) is output, and the underflow flag (ENCAAnUDF) is set (after an underflow occurs, down-count operation from the loaded value ( $3C5A_{H}$ ) continues).
- (6) After the ENCAAnCCR0 value is loaded to ENCAAnCNT, a match with ENCAAnCCR0 is detected, and ENCAtINT0 is output.

### 26.6.9 ENCA<sub>n</sub>LDE Function (Loading Count Value)

#### (1) <When ENCA<sub>n</sub>LDE = 0>

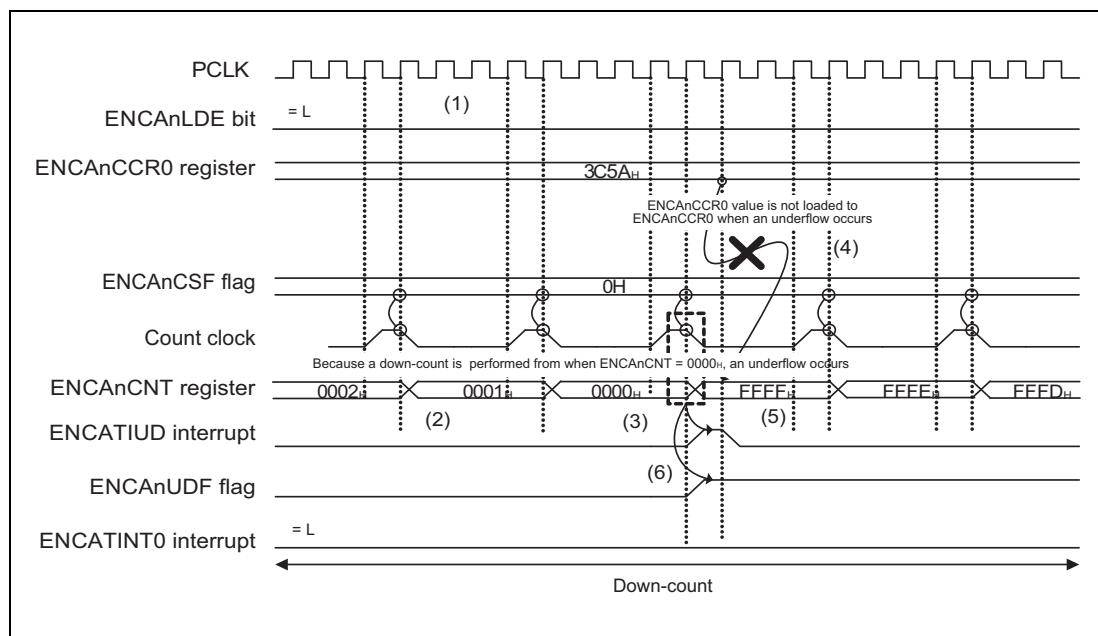
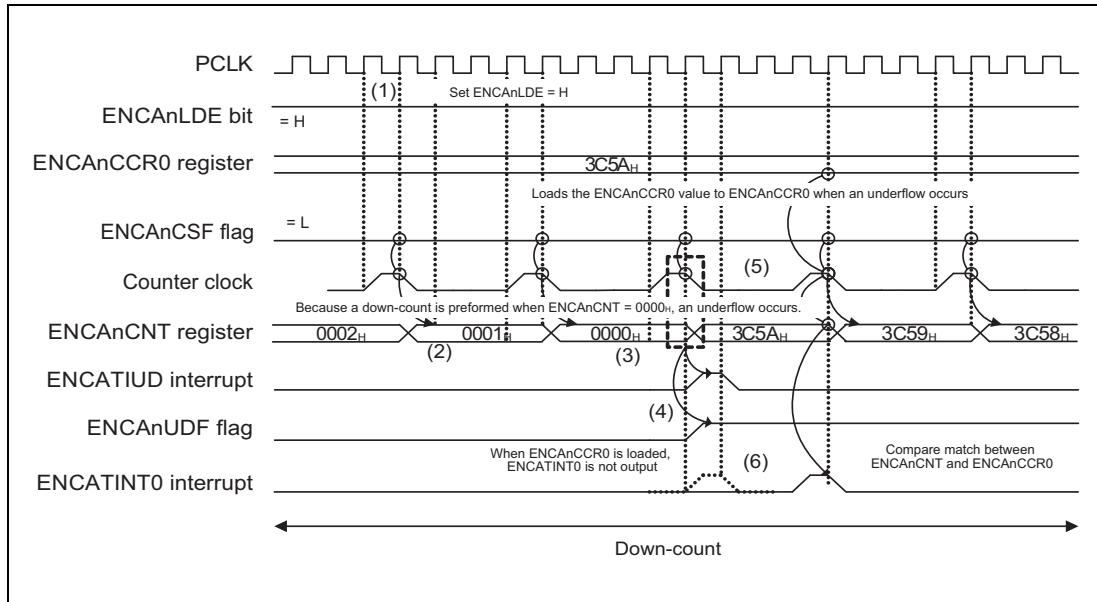


Figure 26.19 ENCA<sub>n</sub>LDE Function (when ENCA<sub>n</sub>LDE = 0)

- (1) ENCA<sub>n</sub>LDE is set to “0” (even if an underflow occurs, the ENCA<sub>n</sub>CCR0 value is not loaded).
- (2) A down-count is performed:  $0002_H \rightarrow 0001_H \rightarrow 0000_H$
- (3) When a further down-count is performed after the counter value changes to  $0000_H$ , an underflow occurs.
- (4) Because ENCA<sub>n</sub>LDE is set to “0”, the setting value of the ENCA<sub>n</sub>CCR0 register is not loaded to the counter even if an underflow occurs.
- (5) Operation changes to underflow operation (counter value:  $0000_H \rightarrow FFFF_H$ ).
- (6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCA<sub>n</sub>UDF) is set.

## (2) &lt;When ENCAAnLDE = 1&gt;

**Figure 26.20 ENCAAnLDE Function (when ENCAAnLDE = 1)**

- (1) ENCAAnLDE is set to “1” (if an underflow occurs, the ENCAAnCCR0 value is loaded to the counter).
- (2) A down-count is performed: 0002<sub>H</sub> → 0001<sub>H</sub> → 0000<sub>H</sub>
- (3) When a further down-count is performed after the counter value changes to 0000<sub>H</sub>, an underflow occurs.
- (4) An underflow interrupt is output, and the underflow flag is set.
- (5) Because ENCAAnLDE is set to “1”, the setting value of the ENCAAnCCR0 register is loaded to the counter if an underflow occurs. ENCAAnCNT is set to 3C5A<sub>H</sub>.
- (6) After the ENCAAnCCR0 value is set to ENCAAnCNT, if the ENCAAnCNT value matches the ENCAAnCCR0 value on a count clock, a compare match interrupt (ENCAtINT0) is output.

### 26.6.10 Conflict between ENCAAnLDE Function (Loading Counter Value) and Rewrite of ENCAAnCCR0 Register

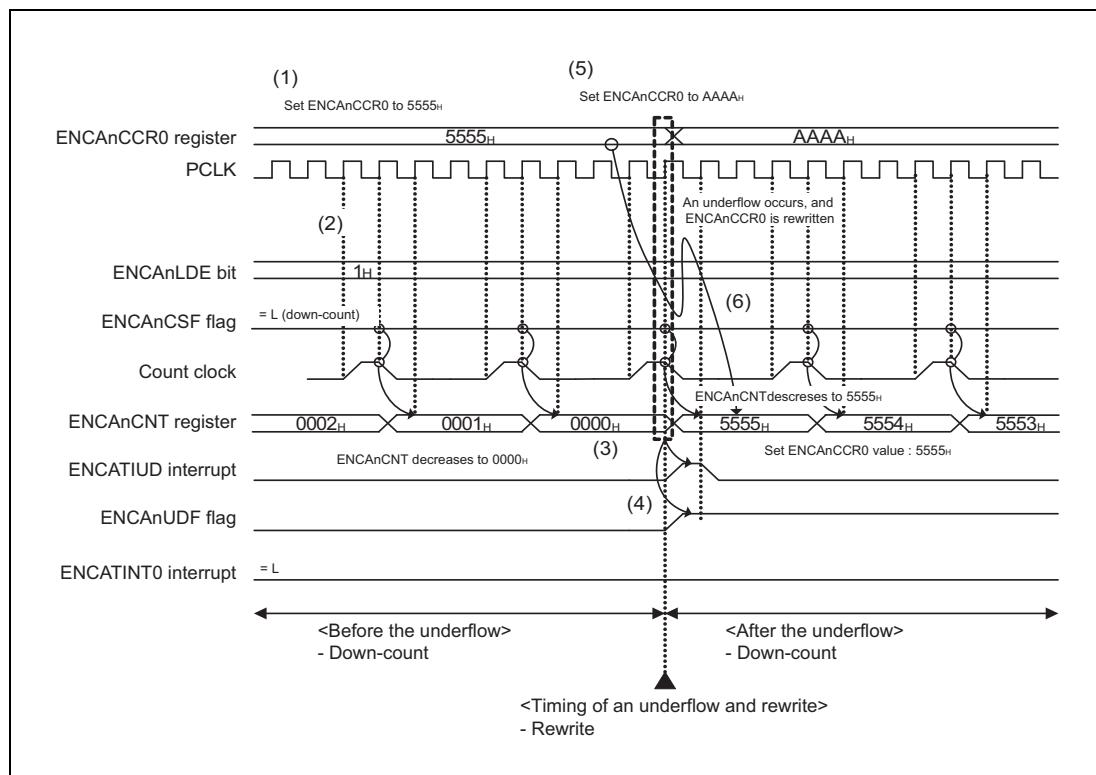
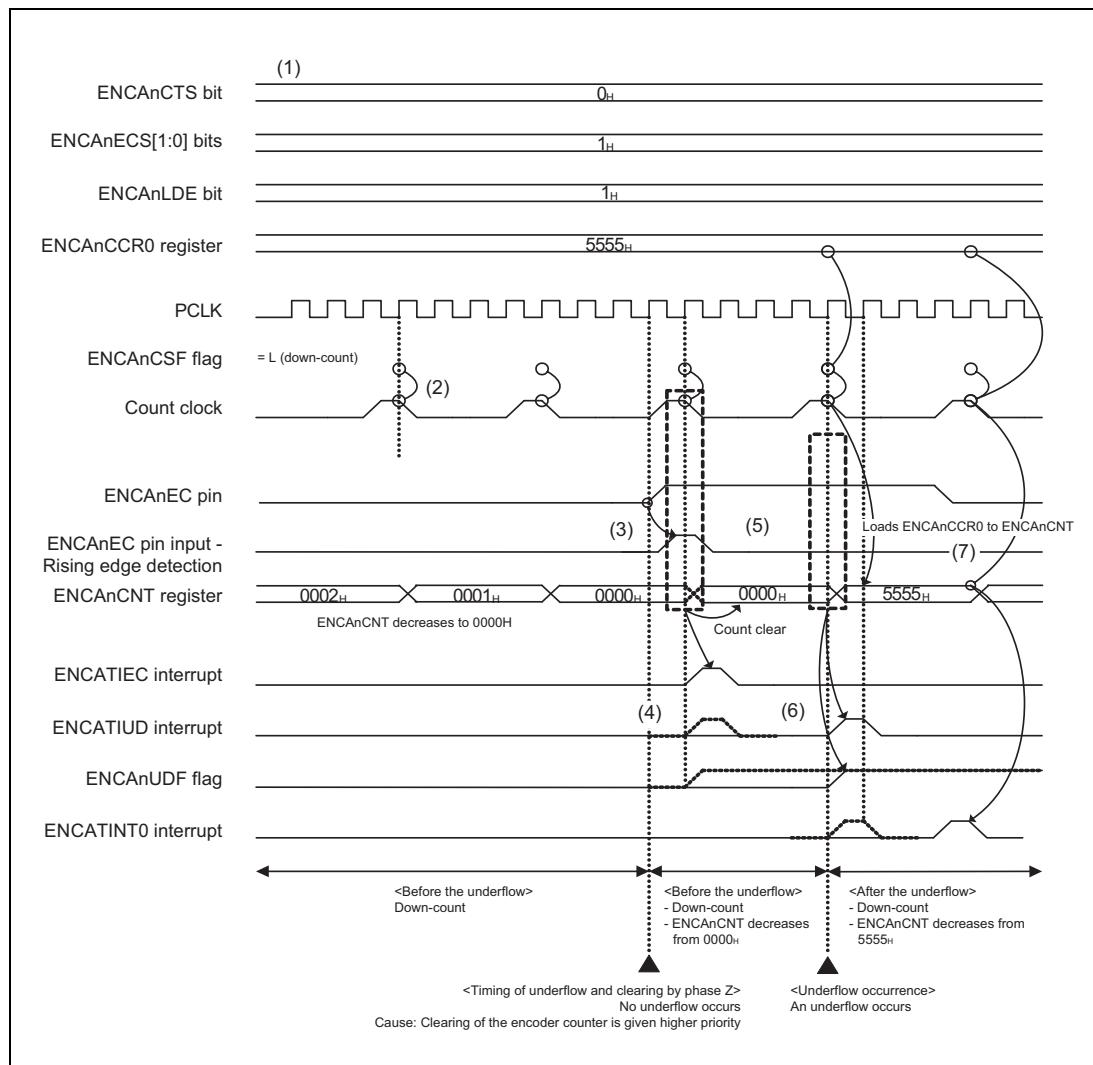


Figure 26.21 Conflict between ENCAAnLDE Function and Rewrite of ENCAAnCCR0 Register

- (1) The ENCAAnCCR0 register is currently set to 5555<sub>H</sub>.
- (2) ENCAAnLDE is currently set to “1”.
- (3) A down-count is performed (0002<sub>H</sub> → 0001<sub>H</sub> → 0000<sub>H</sub>), and an underflow occurs.
- (4) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAAnUDF) is set.
- (5) When an underflow occurs, the ENCAAnCCR0 register value is changed from 5555<sub>H</sub> to AAAA<sub>H</sub>.
- (6) Additionally, when an underflow occurs, the ENCAAnCCR0 value before the rewrite was performed (5555<sub>H</sub>) is set in ENCAAnCNT.

### 26.6.11 Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAAnEC Pin)



**Figure 26.22 Conflict between ENCAAnLDE Function and Clear Operation by Encoder Clear Input**

- (1) The values are set as follows: ENCAAnCTS = 0, ENCAAnECS[1:0] = 01<sub>B</sub>, ENCAAnLDE = 1, and ENCAAnCCR0 = 5555<sub>H</sub>.
- (2) A down-count is performed: 0002<sub>H</sub> → 0001<sub>H</sub> → 0000<sub>H</sub>
- (3) When the count value becomes 0000<sub>H</sub>, the rising edge of ENCAAnEC pin is detected, and clear operation by the encoder clear input is performed.
- (4) Because a count clear is performed when the count value reaches 0000<sub>H</sub>, a counter clear interrupt (ENCATIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000<sub>H</sub>. Therefore, an underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAAnUDF) is not set.
- (5) After the count value is cleared to 0000<sub>H</sub> by clear operation by the encoder clear input, a down-count is performed and an underflow occurs.
- (6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAAnUDF) is set.

- (7) Because ENCAAnLDE = “1”, if an underflow occurs, the ENCAAnCCR0 value is loaded to ENCAAnCNT.
- (8) After the ENCAAnCCR0 value is set to ENCAAnCNT, a compare match is detected according to the count clock. If the ENCAAnCNT value matches the ENCAAnCCR0 value, a compare match interrupt (ENCATINT0) is output.

### 26.6.12 Up-count after Conflict between ENCAAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

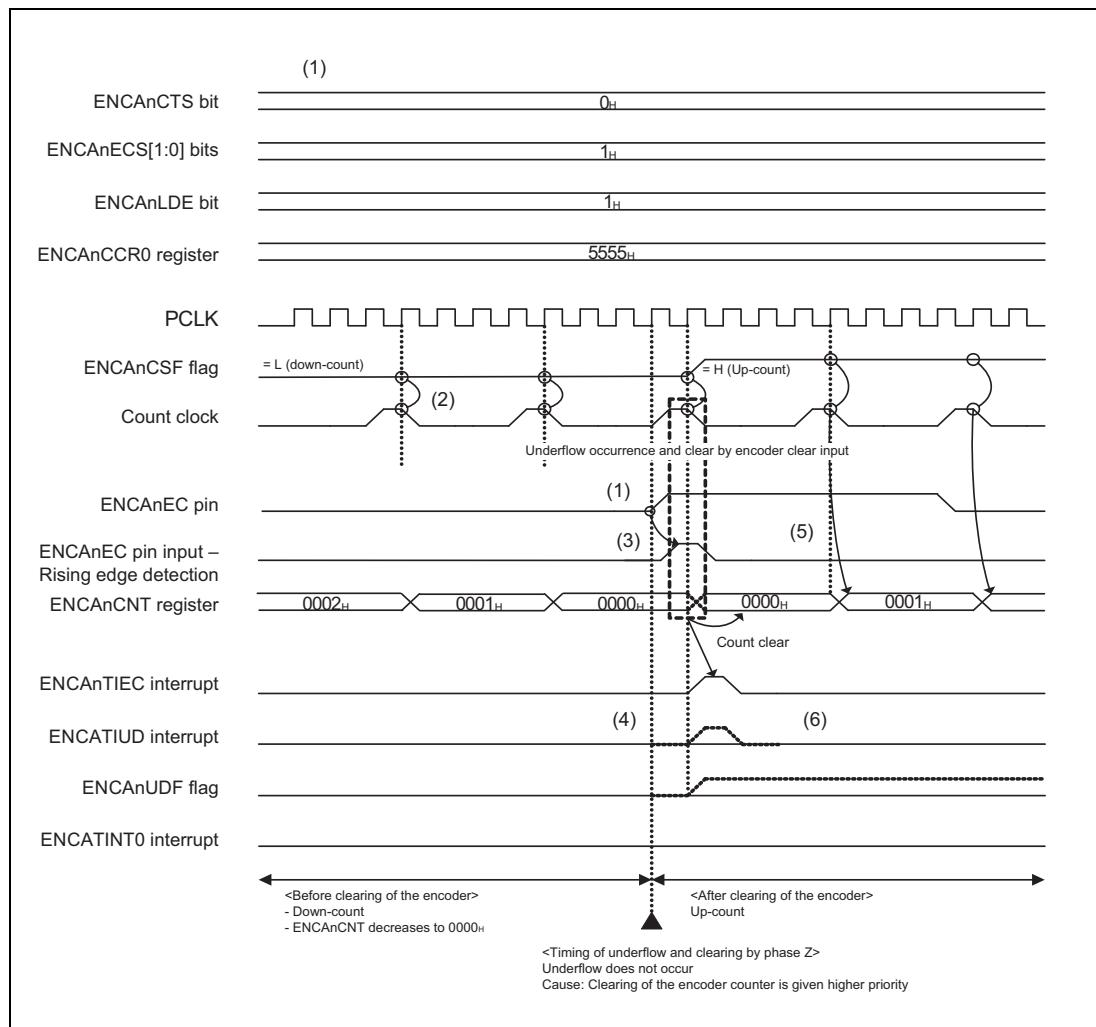
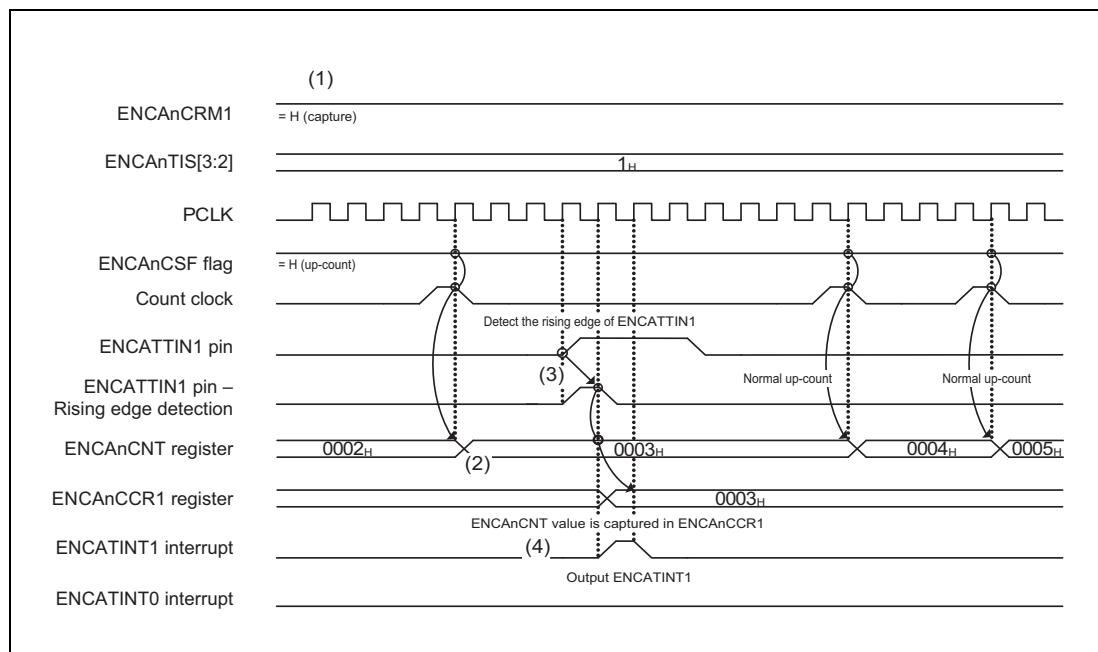


Figure 26.23 Up-count after Conflict between ENCAAnLDE Function and Encoder Clear

- (1) The values are set as follows: ENCAAnCTS = 0, ENCAAnECS[1:0] = 01<sub>B</sub>, ENCAAnLDE = 1, and ENCAAnCCR0 = 5555<sub>H</sub>.
- (2) A down-count is performed: 0002<sub>H</sub> → 0001<sub>H</sub> → 0000<sub>H</sub>
- (3) When the count value becomes 0000<sub>H</sub>, the rising edge of ENCAAnEC pin is detected, and clear operation by the encoder clear input is performed.
- (4) Because a count clear is performed when the count value reaches 0000<sub>H</sub>, a counter clear interrupt (ENCATIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000<sub>H</sub>. Therefore, an underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAAnUDF) is not set.
- (5) After the count value is cleared to 0000<sub>H</sub> by clear operation by the encoder clear input, an up-count is performed.
- (6) An underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAAnUDF) is not set.

### 26.6.13 Capture Operation between Count Clocks (ENCA<sub>n</sub>CCR1)



**Figure 26.24 Capture Operation between Count Clocks (ENCA<sub>n</sub>CCR1)**

- (1) The values are set as follows: ENCA<sub>n</sub>CRM1 = 1, and ENCA<sub>n</sub>TIS[3:2] = 01<sub>B</sub>.
- (2) An up-count is performed.
- (3) The rising edge of the ENCATTIN1 input is detected, and the count value is captured in ENCA<sub>n</sub>CCR1.
- (4) An interrupt (ENCATINT1) corresponding to the capture to the ENCA<sub>n</sub>CCR1 register is output.

### 26.6.14 Capture Operation between Count Clocks (ENCA<sub>n</sub>CCR0)

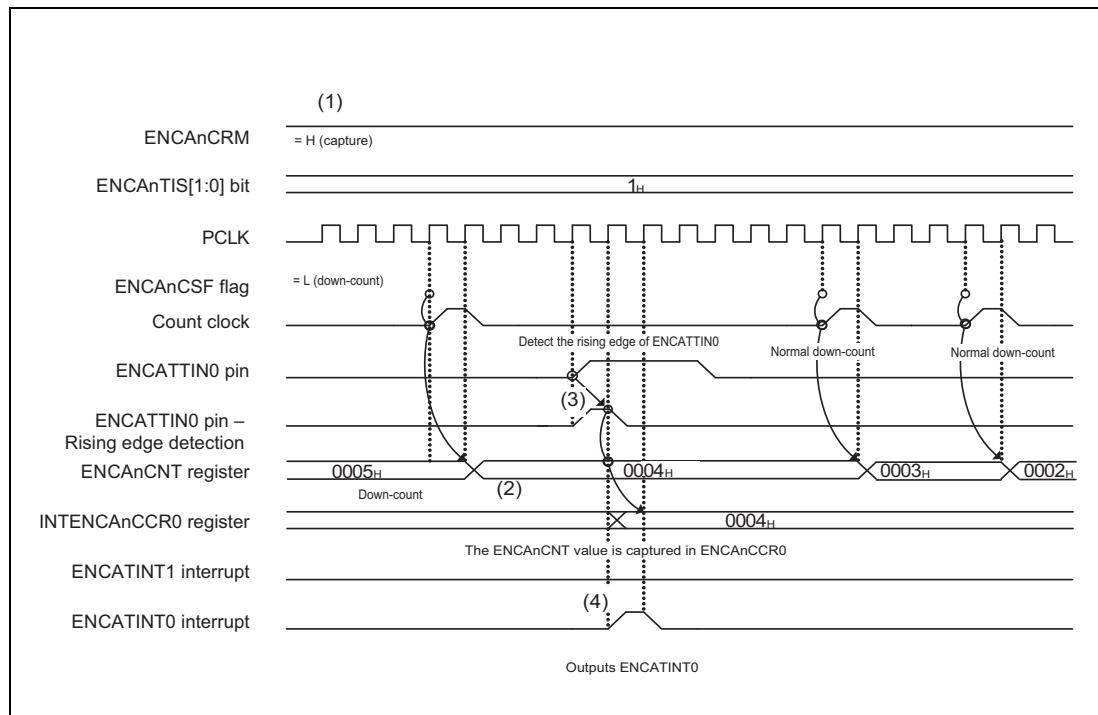
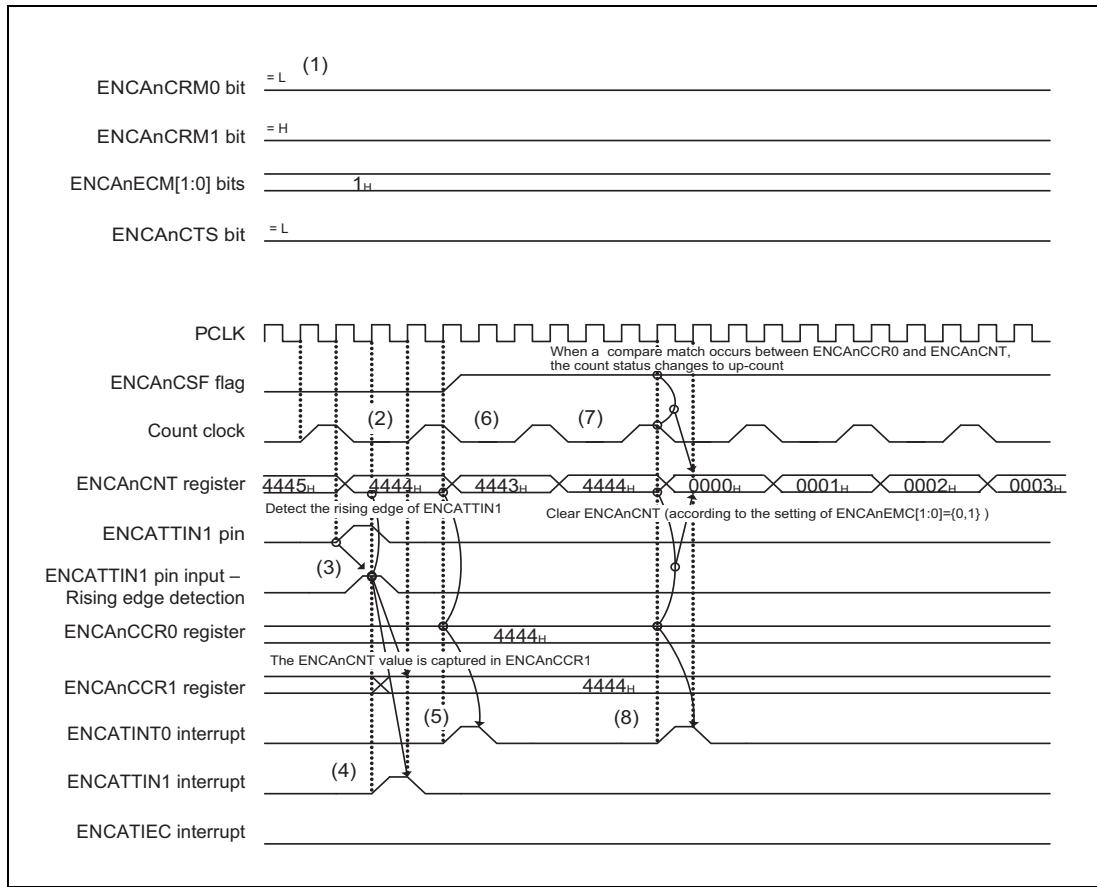


Figure 26.25 Capture Operation between Count Clocks (ENCA<sub>n</sub>CCR0)

- (1) The values are set as follows: ENCA<sub>n</sub>CRM0 = 1, and ENCA<sub>n</sub>TIS[1:0] = 01<sub>B</sub>.
- (2) A down-count is performed.
- (3) The rising edge of the ENCATTIN0 input is detected, and the count value is captured in ENCA<sub>n</sub>CCR0.
- (4) An interrupt (ENCATINT0) corresponding to the capture to the ENCA<sub>n</sub>CCR0 register is output.

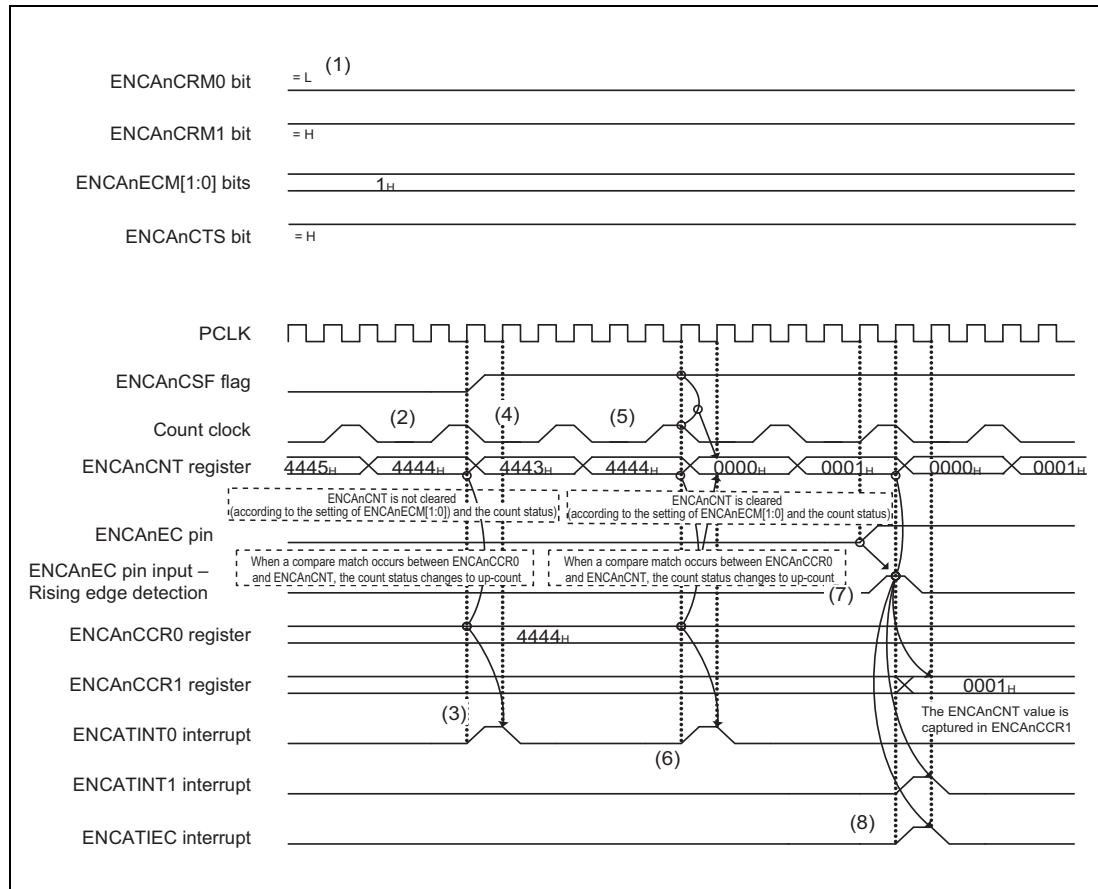
### 26.6.15 Encoder Operation when Compare Match Clear Control is Enabled and ENCAAnCTS = 0



**Figure 26.26 Encoder Operation when Compare Match Clear Control is Enabled and ENCAAnCTS = 0**

- (1) The values are set as follows: ENCAAnCCR0 = 4444<sub>H</sub>, ENCAAnCRM0 = 0, ENCAAnCRM1 = 1, ENCAAnECM[1:0] = 01<sub>B</sub>, and ENCAAnCTS = 0.
- (2) A down-count is performed.
- (3) The rising edge of the ENCATTIN1 input is detected, and the ENCAAnCNT value (4444<sub>H</sub>) is captured in the ENCAAnCCR1 register.
- (4) An interrupt signal (ENCATINT1) corresponding to the capture to the ENCAAnCCR1 register is output.
- (5) When a compare match occurs between ENCAAnCNT (counted down from 4445<sub>H</sub> to 4444<sub>H</sub>) and ENCAAnCCR0 (4444<sub>H</sub>), a compare match interrupt (ENCATINT0) with ENCAAnCCR0 is output.
- (6) The count operation changes to up-count.
- (7) When ENCAAnCNT is counted up from 4443<sub>H</sub> to 4444<sub>H</sub>, a compare match with ENCAAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAAnECM1 and ENCAAnECM0 (01<sub>B</sub>), and the ENCAAnCNT value changes to 0000<sub>H</sub>.
- (8) When ENCAAnCNT changes to 4444<sub>H</sub>, a compare match interrupt (ENCATINT0) with ENCAAnCCR0 is output.

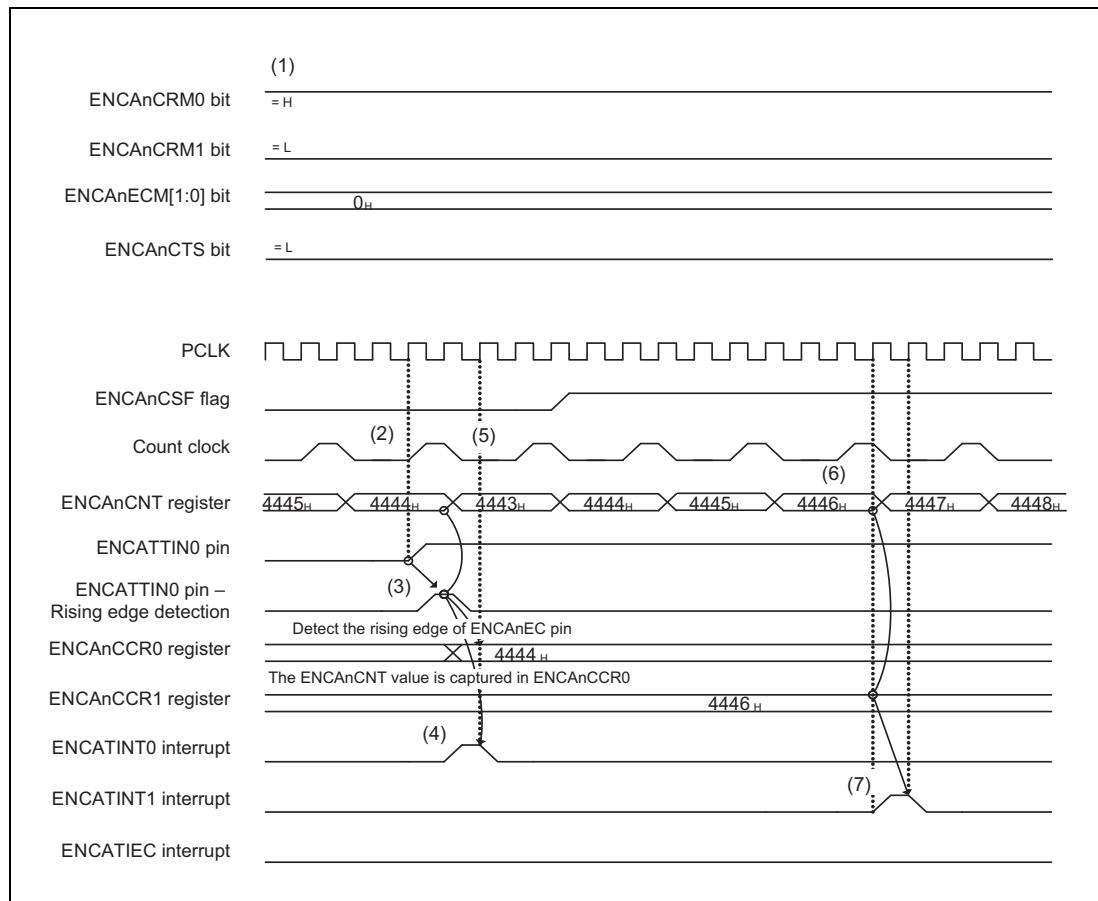
### 26.6.16 Encoder Operation when Compare Match Clear Control is Enabled and ENCAAnCTS = 1



**Figure 26.27 Encoder Operation when Compare Match Clear Control is Enabled and ENCAAnCTS = 1**

- (1) The values are set as follows: ENCAAnCCR0 = 4444<sub>H</sub>, ENCAAnCRM0 = 0, ENCAAnCRM1 = 1, ENCAAnECM[1:0] = 01<sub>B</sub>, and ENCAAnCTS = 1.
- (2) A down-count is performed.
- (3) When a compare match occurs between ENCAAnCNT (counted down from 4445<sub>H</sub> to 4444<sub>H</sub>) and ENCAAnCCR0 (4444<sub>H</sub>), a compare/capture interrupt (ENCATINT0) is output.
- (4) The count operation changes to up-count.
- (5) When ENCAAnCNT is counted up from 4443<sub>H</sub> to 4444<sub>H</sub>, a compare match with ENCAAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAAnECM1 and ENCAAnECM0 (01<sub>B</sub>), and the ENCAAnCNT value changes to 0000<sub>H</sub>.
- (6) When ENCAAnCNT changes to 4444<sub>H</sub>, a compare match interrupt (ENCATINT0) with ENCAAnCCR0 is output.
- (7) After the count value is cleared, an up-count is performed, and the count value changes to 0001<sub>H</sub>. At this point, the ENCAAnCNT value (0001<sub>H</sub>) is captured in ENCAAnCCR1 by detecting the rising edge of the ENCAAnEC signal, and the counter is cleared to 0000<sub>H</sub>.
- (8) An interrupt (ENCATINT1) corresponding to the capture to the ENCAAnCCR1 register and a clear interrupt (ENCATICC) by ENCAAnEC are output.

### 26.6.17 Encoder Operation when Compare Match Clear Control is Disabled



**Figure 26.28 Encoder Operation when Compare Match Clear Control is Disabled**

- (1) The values are set as follows: ENCAncCCR1 = 4446H, ENCAncCRM0 = 1, ENCAncCRM1 = 0, ENCAncECM[1:0] = 00B, and ENCAncCTS = 0.
- (2) A down-count is performed.
- (3) When the rising edge of ENCATTIN0 is detected, the ENCAncNT value (4444H) is captured in ENCAncCR0.
- (4) An interrupt signal (ENCATINT0) corresponding to the capture to the ENCAncCCR0 register is output.
- (5) The count operation changes to up-count.
- (6) When ENCAncNT changes to 4446H, a compare match with ENCAncCCR1 is detected.
- (7) A compare match interrupt (ENCATINT1) with ENCAncCCR1 is output.

## 26.6.18 Capture Operation Performed upon Clearing by ENCA<sub>n</sub>EC, ENCA<sub>n</sub>E0, or ENCA<sub>n</sub>E1 when ENCA<sub>n</sub>SCE = 1

### 26.6.18.1 Accompanying capture operation

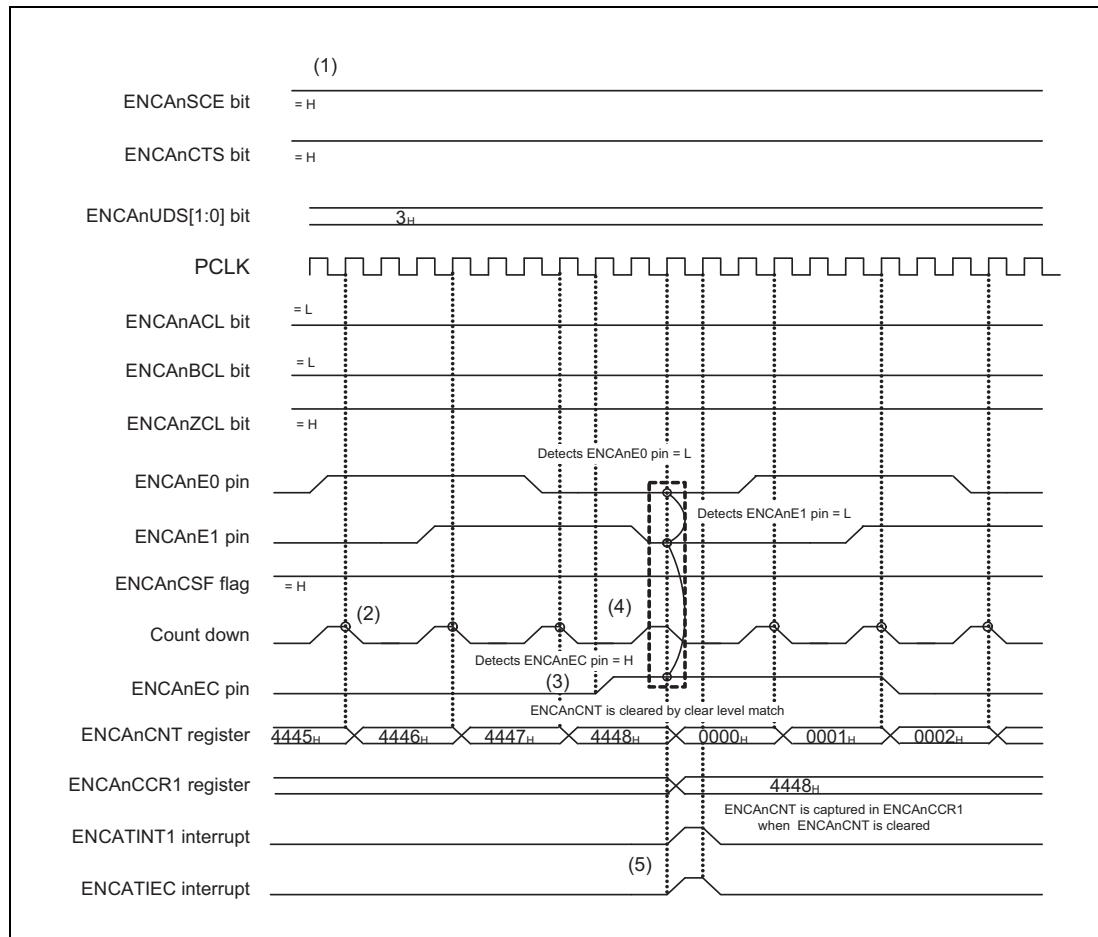
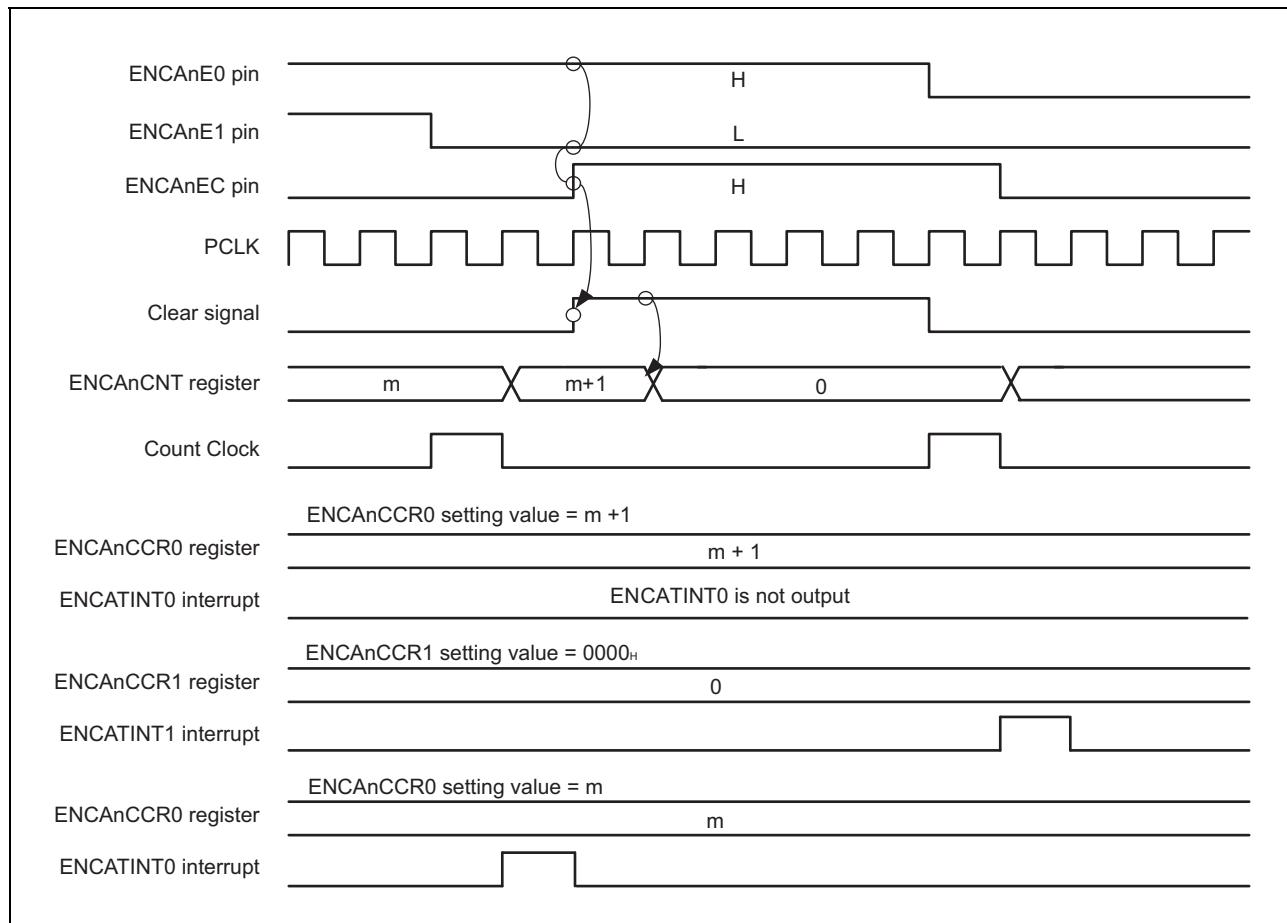


Figure 26.29 Capture Operation Performed upon Clearing by ENCA<sub>n</sub>EC, ENCA<sub>n</sub>E0, or ENCA<sub>n</sub>E1 when ENCA<sub>n</sub>SCE = 1

- (1) The values are set as follows: ENCA<sub>n</sub>SCE = 1, ENCA<sub>n</sub>CTS = 1, ENCA<sub>n</sub>UDS[1:0] = 11<sub>B</sub>, ENCA<sub>n</sub>ACL = 0, ENCA<sub>n</sub>BCL = 0, and ENCA<sub>n</sub>ZCL = 1.
- (2) An up-count is performed.
- (3) The count value is not cleared upon the rising edge of ENCA<sub>n</sub>EC.
- (4) When ENCA<sub>n</sub>E0, ENCA<sub>n</sub>E1 and ENCA<sub>n</sub>EC reach the set clear level, the count value is cleared. The count value is captured in ENCA<sub>n</sub>CCR1 at the time of the clearing.
- (5) At the time of the clearing, an interrupt (ENCATINT1) corresponding to the capture to the ENCA<sub>n</sub>CCR1 register and a clear interrupt (ENCATICC) by ENCA<sub>n</sub>EC are output.

**26.6.18.2 When the Timing of the ENCA<sub>n</sub>EC Input is Later than that of the ENCA<sub>n</sub>E1 Input during Up-count (When ENCA<sub>n</sub>ACL = 1, ENCA<sub>n</sub>BCL = 0, ENCA<sub>n</sub>ZCL = 1, and ENCA<sub>n</sub>UDS[1:0] = 11<sub>B</sub>)**



**Figure 26.30 Clearing Timing for when the Timing of the ENCA<sub>n</sub>EC Input is Later than that of the ENCA<sub>n</sub>E1 Input during Up-count**

**26.6.18.3 When the Timing of the ENCA<sub>n</sub>EC Input is the Same as that of the ENCA<sub>n</sub>E1 Input during Up-count (When ENCA<sub>n</sub>ACL = 1, ENCA<sub>n</sub>BCL = 0, ENCA<sub>n</sub>ZCL = 1, and ENCA<sub>n</sub>UDS[1:0] = 11<sub>B</sub>)**

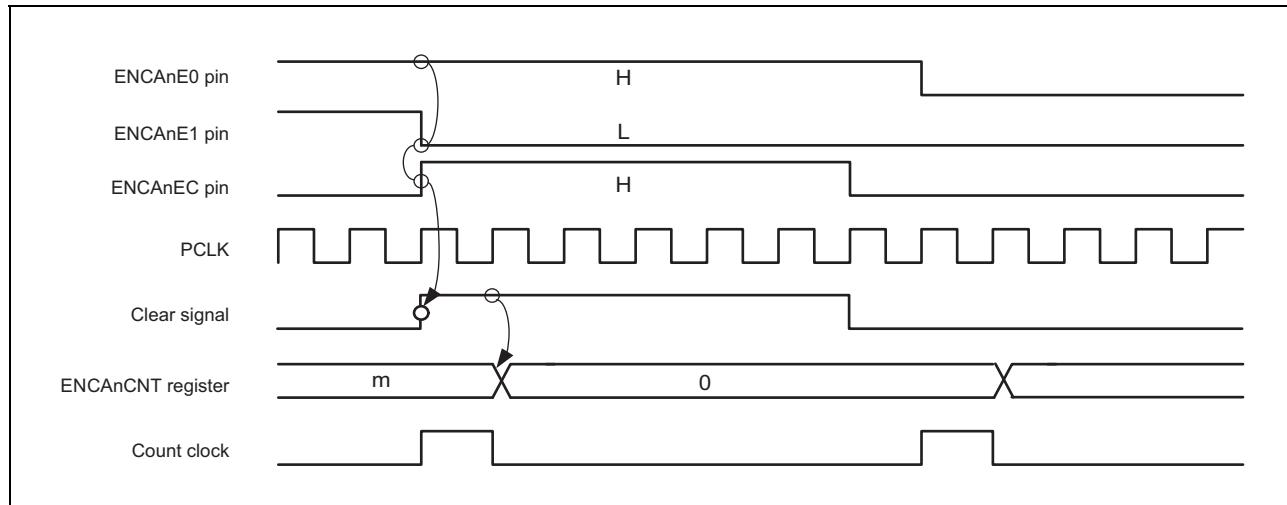


Figure 26.31 Clearing Timing for when the Timing of the ENCA<sub>n</sub>EC Input is the Same as that of the ENCA<sub>n</sub>E1 Input during Up-count

**26.6.18.4 When the Timing of the ENCA<sub>n</sub>EC Input is Earlier than that of the ENCA<sub>n</sub>E1 Input during Up-count (When ENCA<sub>n</sub>ACL = 1, ENCA<sub>n</sub>BCL = 0, ENCA<sub>n</sub>ZCL = 1, and ENCA<sub>n</sub>UDS[1:0] = 11<sub>B</sub>)**

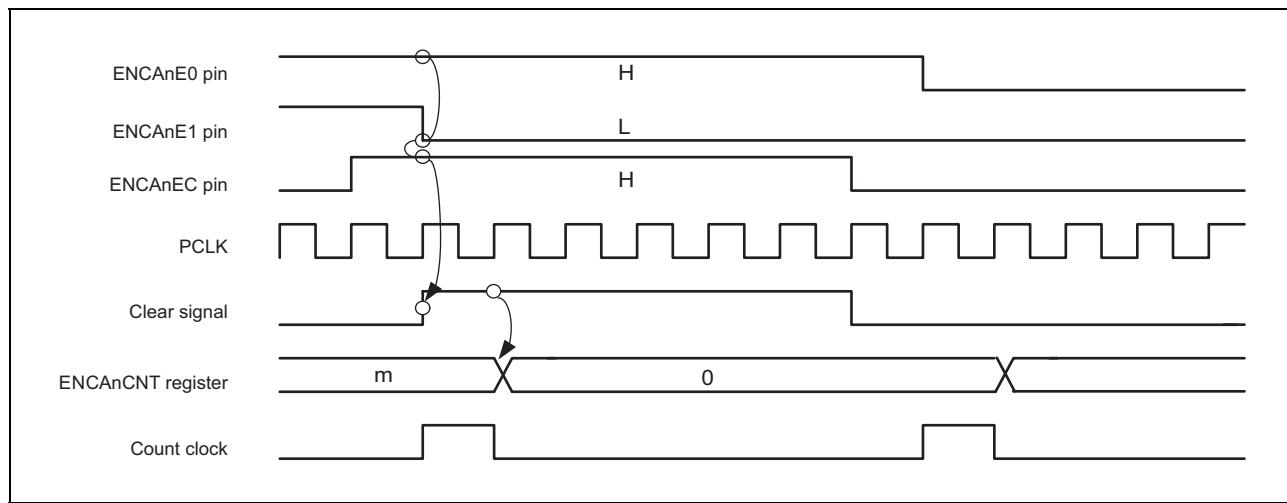
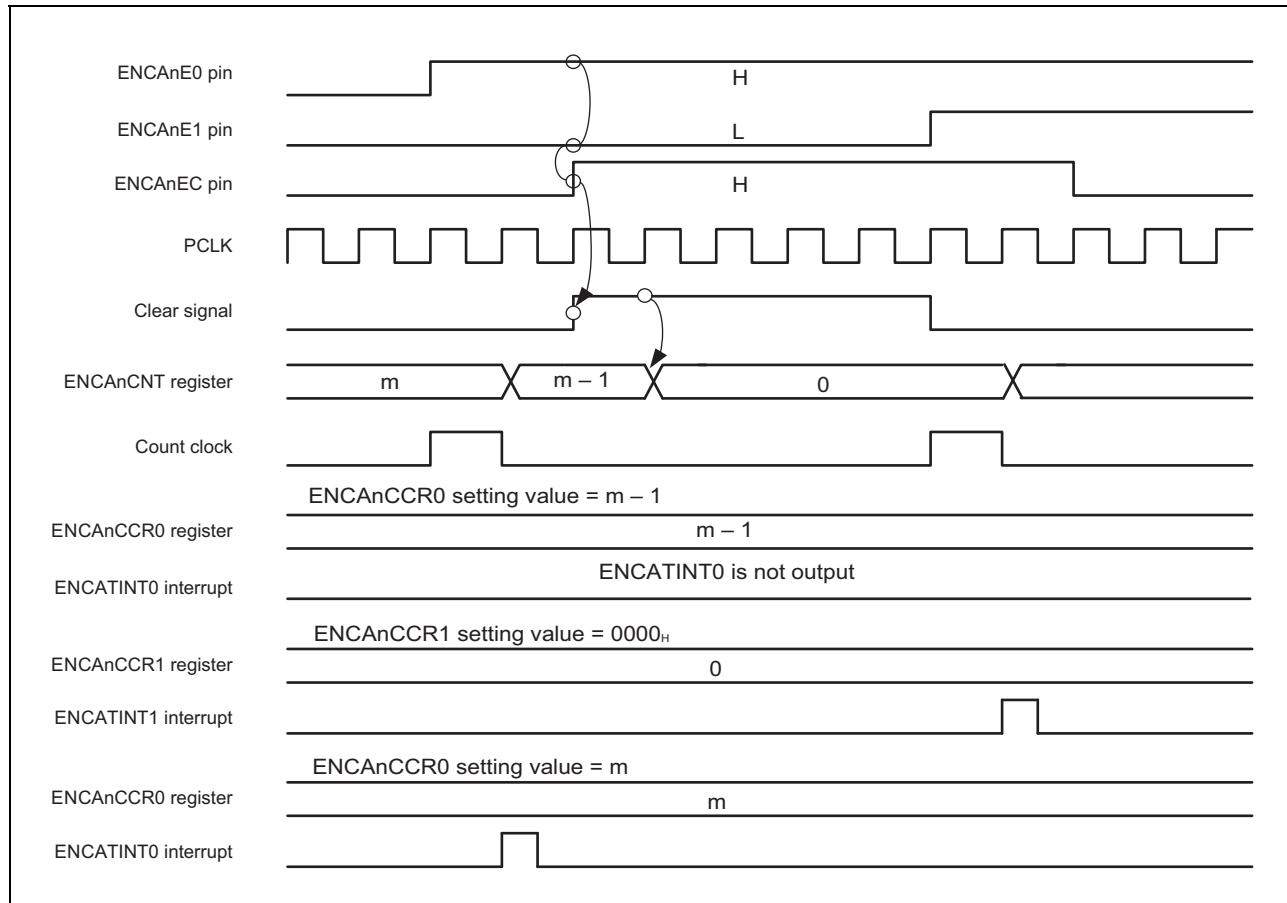


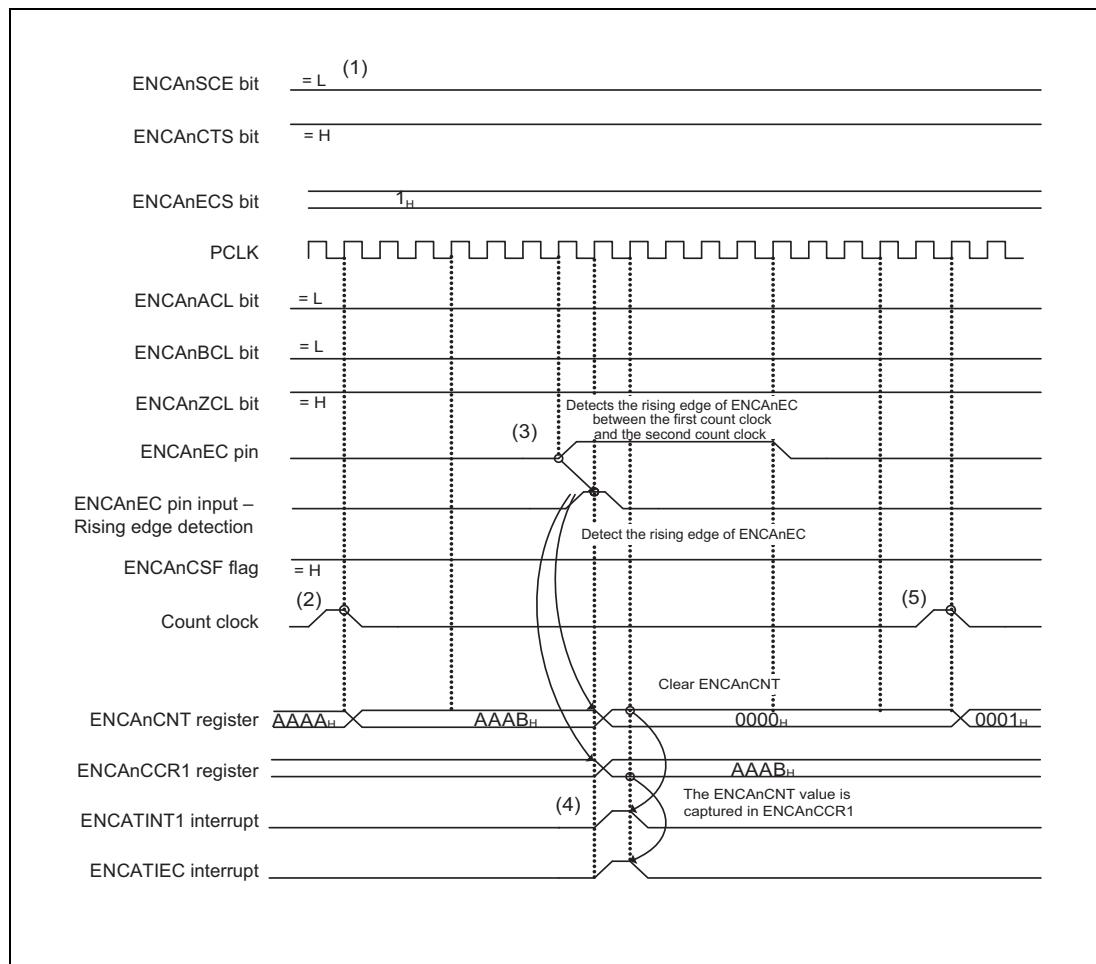
Figure 26.32 Clearing Timing for when the Timing of the ENCA<sub>n</sub>EC Input is Earlier than that of the ENCA<sub>n</sub>E1 Input during Up-count

**26.6.18.5 When the Timing of the ENCA<sub>n</sub>EC Input is Later than that of the ENCA<sub>n</sub>E1 Input during Down-count (When ENCA<sub>n</sub>ACL = 1, ENCA<sub>n</sub>BCL = 0, ENCA<sub>n</sub>ZCL = 1, and ENCA<sub>n</sub>UDS[1:0] = 11<sub>B</sub>)**



**Figure 26.33 Clearing Timing for when the Timing of the ENCA<sub>n</sub>EC Input is Later than that of the ENCA<sub>n</sub>E1 Input during Down-count**

### 26.6.19 Capture Operation Performed upon Clearing by ENCAAnEC when ENCAAnSCE = 0



**Figure 26.34 Capture Operation Performed upon Clearing by ENCAAnEC when ENCAAnSCE = 0**

- (1) The values are set as follows: ENCAAnSCE = 0, ENCAAnCTS = 1, and ENCAAnECS[1:0] = 01<sub>B</sub>.
- (2) An up-count is performed.
- (3) The rising edge of the ENCAAnEC input is detected, and the ENCAAnCNT value (AAAB<sub>H</sub>) is captured in the ENCAAnCCR1 register. Concurrently, clear operation by ENCAAnEC is performed, and ENCAAnCNT is cleared to 0000<sub>H</sub>.
- (4) A capture interrupt 1 (ENCAtINT1) to the ENCAAnCCR1 register and an encoder clear interrupt (ENCAtIEC) by ENCAAnEC are output.
- (5) After the count value is cleared, an up-count is performed, and the count value changes to 0001<sub>H</sub>.

## Section 27 Motor Control

This section contains a generic description of the Motor Control.

The first part in this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the Motor Control.

### 27.1 Features of RH850/F1L Motor Control

#### 27.1.1 Number of Units and Channels

The motor control function comprises a timer motor control unit (TAPA) and a peripheral interconnection unit (PIC) to connect the TAPA unit to peripheral timers, and generates motor control waveforms by using a combination of peripheral timers and A/D converters.

This microcontroller has the following number of TAPA and PIC units.

**Table 27.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
<b>TAPA</b>						
Number of Units	1					
Name	TAPAn (n = 0)					
<b>PIC</b>						
Number of Units	1					
Name	PIC0					

**Table 27.2 Indices**

Index	Description
n	Throughout this section, the unit of a timer and A/D converter used by TAPA and the motor control function is identified by the index "n" (n = 0): for example, TAPAnCTL0 is TAPAn control register 0.
m	The channel of a used timer and A/D converter is identified by the index "m". For example, the TAUDn channel is described as CHm.
x	The scan group of an A/D converter is identified by the index "x" (x = 1 to 3).
j	The scan trigger number of an A/D converter is identified by the index "j" (j = 0 to 2).

The following table lists the values indicated by the indices of each product.

**Table 27.3 Indices of Products**

Indices of Each Product
All Products
m = 0 to 15 (e.g. TAUDn)
x = 1 to 3
j = 0 to 2

### 27.1.2 Register Base Address

The base addresses of TAPAn and PIC0 are listed in the following table.

The register addresses of TAPAn and PIC0 are given as offsets from the base addresses.

**Table 27.4 Register Base Addresses**

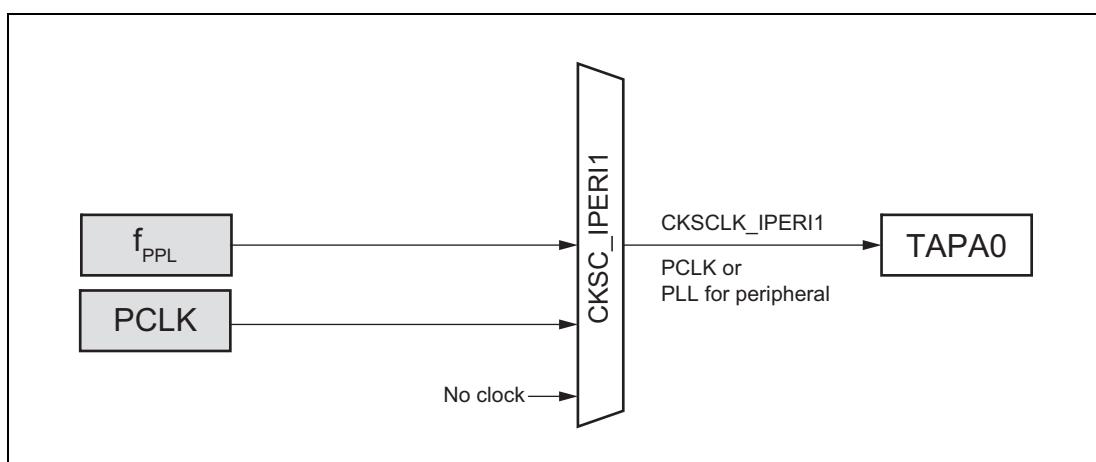
Base Address Name	Base Address
<TAPA0_base>	FFE9 0000 <sub>H</sub>
<PIC0_base>	FFDD 0000 <sub>H</sub>

### 27.1.3 Clock Supply

The TAPAn and PIC0 clock supplies are listed in the following table.

**Table 27.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
TAPAn	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1
PIC0	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1



**Figure 27.1 TAPA Clock Supply**

### 27.1.4 Interrupt Request

The TAPA0 interrupt requests are listed in the following table.

**Table 27.6 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
<b>TAPA0</b>			
TAPATIPEK0	Peak interrupt 0	8, 108	—
TAPATIVLY0	Valley interrupt 0	9, 109	—

### 27.1.5 Reset Sources

The reset sources of TAPAn and PIC0 are listed in the following table. TAPAn and PIC0 are initialized by these reset sources.

**Table 27.7 Reset Sources**

Unit Name	Reset Source
TAPA0	All reset sources (ISORES)
PIC0	All reset sources (ISORES)

### 27.1.6 External Input/Output Signal

The external output signals of TAPAn and PIC0 are listed below.

**Table 27.8 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal
<b>PIC</b>		
TOUTU	Motor control output U phase (positive)	TAPA0UP
TOUTUB	Motor control output U phase (negative)	TAPA0UN
TOUTV	Motor control output V phase (positive)	TAPA0VP
TOUTVB	Motor control output V phase (negative)	TAPA0VN
TOUTW	Motor control output W phase (positive)	TAPA0WP
TOUTWB	Motor control output W phase (negative)	TAPA0WN
<b>TAPA</b>		
TAPATHASIN	Motor control output Hi-Z control input	TAPA0ESO

#### CAUTION

For the port pins that are used as TAPA0UP, TAPA0UN, TAPA0VP, TAPA0VN, TAPA0WP, TAPA0WN, and TAPA0ESO, set the output driving ability to high (PDSCn\_m = 1).

## 27.1.7 Internal Output Signal

The internal output signals of TAPAn and PIC0 are listed below.

**Table 27.9 Internal Output Signals**

Unit Signal Name	Outline	Connected to
<b>TAPA0</b>		
TAPATHZOUT0	TAPA0UP/TAPA0UN output buffers Hi-Z control output <sup>*1</sup>	Port
TAPATHZOUT1	TAPA0VP/TAPA0VN output buffers Hi-Z control output <sup>*1</sup>	Port
TAPATHZOUT2	TAPA0WP/TAPA0WN output buffers Hi-Z control output <sup>*1</sup>	Port
TAPATADOUT0	A/D conversion trigger signal 0 output <sup>*2</sup>	ADCA0 hardware trigger expansion
TAPATADOUT1	A/D conversion trigger signal 1 output <sup>*2</sup>	ADCA0 hardware trigger expansion
<b>PIC0</b>		
TAPATHASIN	TAPA0 asynchronous Hi-Z control signal <sup>*1,*3</sup>	TAPA0
TAPATSIMO	TAUD master channel interrupt signal (TAUD0: INTTAUD0I0, INTTAUD0I2, INTTAUD0I8)	TAPA0
TAPATUDCM0	TAUD master up/down signal (TAUD0: TAUD0UDC0, TAUD0UDC2, TAUD0UDC8)	TAPA0
TAPATCDENS0	TAUD slave 0 match detect <sup>*4</sup> (ADCA0 hardware trigger expansion: ADOPA1ADCATTIN00)	TAPA0
TAPATCDENS1	TAUD slave 1 match detect <sup>*4</sup> (ADCA0 hardware trigger expansion: ADOPA2ADCATTIN00)	TAPA0

Note 1. See **Section 27.4.6, TAPA0 Hi-Z Control Input Selection** for details.

Note 2. These signals can be used to as a trigger source to start the A/D converter. See **Table 29.47, List of A/D Conversion Hardware Triggers**.

Note 3. This input signal is passed through a noise filter. See **Section 2.12, Noise Filter & Edge/Level Detector** and **Section 2.13, Description of Port Noise Filter & Edge/Level Detection**.

Note 4. These signals are selected by the H/W trigger selection bit of the ADCA0 A/D converter. See **Table 29.47, List of A/D Conversion Hardware Triggers**.

## 27.2 Overview

### 27.2.1 Functional Overview

The motor control function provides the following functions by combining the motor control unit (TAPA) and Timer Array Unit D (TAUDn) or A/D (ADCAn):

- Asynchronous Hi-Z control function  
Hi-Z control for TAUDn output can be performed by using pin input or error signals.
- Interrupt signal output function  
Request signals for two types of interrupts, peak interrupts and valley interrupts, can be output by the INTn signals output by TAUDn.
- A/D conversion start trigger selection function  
An A/D conversion start trigger can be output by the INTn signals output by TAUDn.

Additionally, the motor control function provides also the following functions by combining the motor control unit (TAPA) and the peripheral interconnection (PIC):

- Timer simultaneous start trigger function  
The respective channels of TAUD0 and TAUJ1, and the ENCAAn timer can be started simultaneously.
- Trigger and pulse width measuring function  
Measurement of trigger periods can be performed by inputting ENCAAn interrupt signals to TAUDn or TAUJ1.
- A/D trigger encoder capture function  
The value of the ENCAAn counter can be captured at the A/D conversion start trigger timing.
- Three-phase PWM output with dead time / High-accuracy triangle PWM output with dead time  
Three-phase PWM output with dead time can be performed by TAUDn.
- Delay pulse output with dead time  
Delay pulses (with dead time) for the cycle timing can be output.

## 27.2.2 Basic Structure of Motor Control

The peripheral block configuration of the motor control function is shown below.

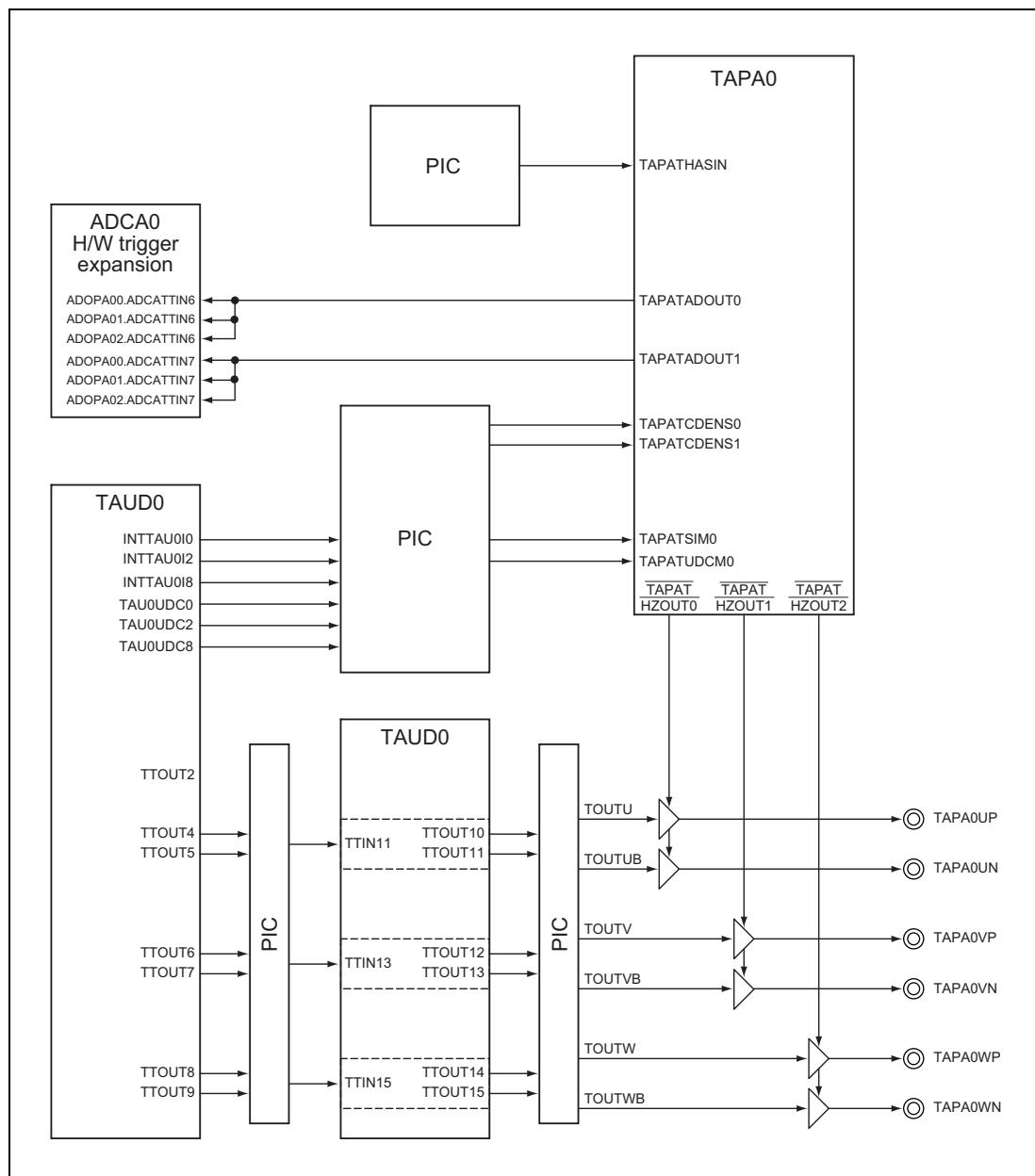


Figure 27.2 Configuration of Motor Control

TAUDn and PIC are used to generate the motor control output signals (three-phase PWM output signals with dead time).

The timer control unit (TAPA) performs Hi-Z control for the motor control output.

Additionally, PIC can provide functions specific to the motor by combining respective channels of TAUDn and TAUJ1, ENCA<sub>n</sub>, and TAPA.

### 27.2.3 Block Diagram

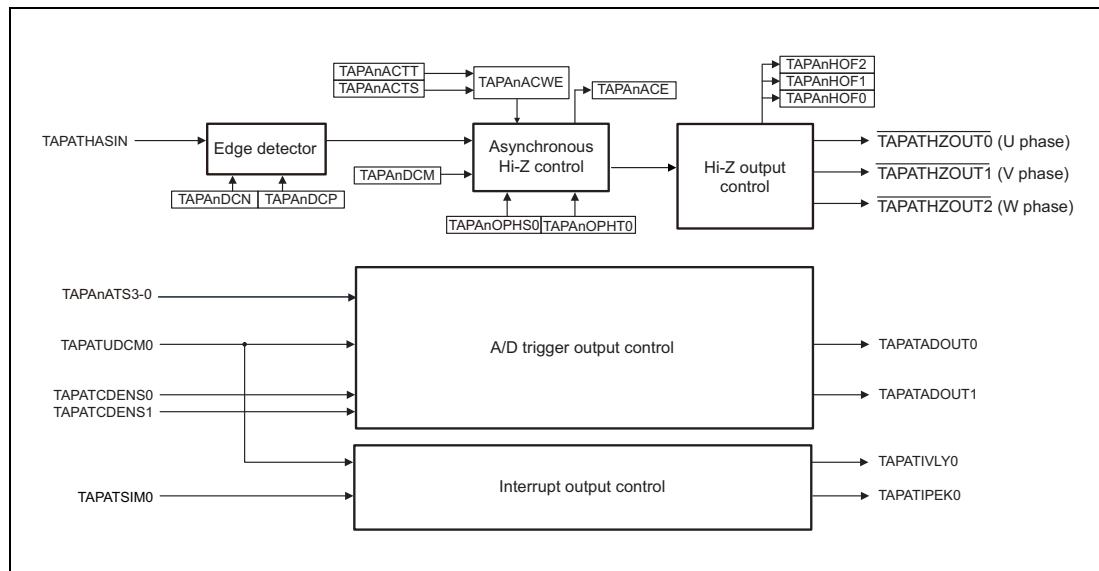


Figure 27.3 TAPA Peripheral Block Diagram

#### NOTE

For the PIC peripheral block diagram, see the respective section describing each function.

### 27.2.4 Definition of Terms

#### Peak and valley interrupts - Peak and valley of timer counter

In this document, the period from a TAUD counting-up status to generation of INT from the master channel is defined as a peak period, and this INT is defined as a peak interrupt.

In contrast, the period from a TAUD counting-down status to generation of INT from the master channel is defined as a valley period, and this INT is defined as a valley interrupt.

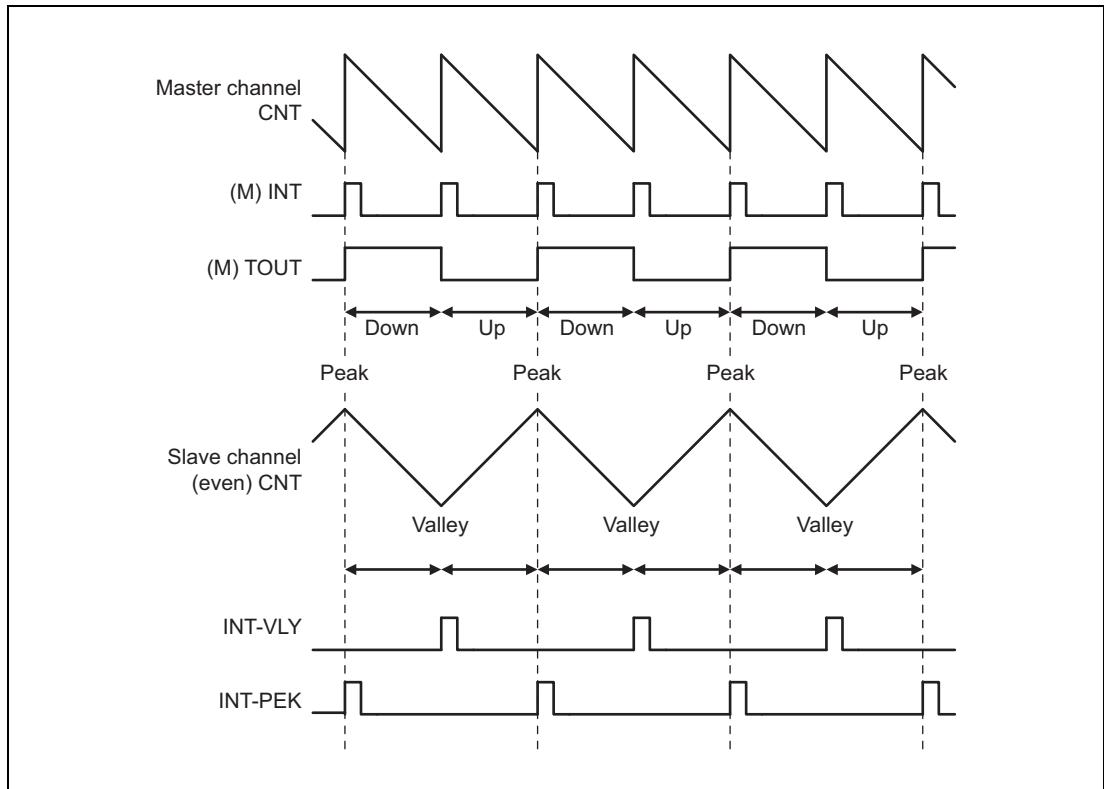


Figure 27.4 Peak and Valley Interrupts

## 27.3 Registers

### 27.3.1 List of Registers

The registers of TAPAn and PIC0 are listed in the following table.

For details about <TAPAn\_base> and <PIC0\_base>, see **Section 27.1.2, Register Base Address**.

Table 27.10 Registers

Module Name	Register Name	Symbol	Address
TAPAn	Control register 0	TAPAnCTL0	<TAPAn_base> + 20 <sub>H</sub>
TAPAn	Control register 1	TAPAnCTL1	<TAPAn_base> + 24 <sub>H</sub>
TAPAn	Flag register	TAPAnFLG	<TAPAn_base> + 00 <sub>H</sub>
TAPAn	Asynchronous Hi-Z control write enable register	TAPAnACWE	<TAPAn_base> + 04 <sub>H</sub>
TAPAn	Asynchronous Hi-Z control start trigger register	TAPAnACTS	<TAPAn_base> + 08 <sub>H</sub>
TAPAn	Asynchronous Hi-Z control stop trigger register	TAPAnACTT	<TAPAn_base> + 0C <sub>H</sub>
TAPAn	Hi-Z start trigger register	TAPAnOPHS	<TAPAn_base> + 14 <sub>H</sub>
TAPAn	Hi-Z stop trigger register	TAPAnOPHT	<TAPAn_base> + 18 <sub>H</sub>
TAPAn	Emulation register	TAPAnEMU	<TAPAn_base> + 28 <sub>H</sub>
PIC0	Simultaneous start trigger control register	PIC0SST	<PIC0_base> + 04 <sub>H</sub>
PIC0	Simultaneous start control register 0	PIC0SSER0	<PIC0_base> + 10 <sub>H</sub>
PIC0	Simultaneous start control register 2	PIC0SSER2	<PIC0_base> + 18 <sub>H</sub>
PIC0	Hi-Z output control register 0	PIC0HIZCEN0	<PIC0_base> + 80 <sub>H</sub>
PIC0	A/D conversion trigger output control register 400	PIC0ADTEN400	<PIC0_base> + 90 <sub>H</sub>
PIC0	A/D conversion trigger output control register 401	PIC0ADTEN401	<PIC0_base> + 94 <sub>H</sub>
PIC0	A/D conversion trigger output control register 402	PIC0ADTEN402	<PIC0_base> + 98 <sub>H</sub>
PIC0	Timer I/O control register 200	PIC0REG200	<PIC0_base> + C0 <sub>H</sub>
PIC0	Timer I/O control register 201	PIC0REG201	<PIC0_base> + C4 <sub>H</sub>
PIC0	Timer I/O control register 202	PIC0REG202	<PIC0_base> + C8 <sub>H</sub>
PIC0	Timer I/O control register 203	PIC0REG203	<PIC0_base> + CC <sub>H</sub>
PIC0	Timer I/O control register 30	PIC0REG30	<PIC0_base> + E8 <sub>H</sub>
PIC0	Timer I/O control register 31	PIC0REG31	<PIC0_base> + EC <sub>H</sub>

#### NOTE

For details about PIC-related registers, see the respective section describing each function.

### 27.3.2 TAPAnCTL0 — TAPA Control Register 0

This register is used to set up the asynchronous Hi-Z control function.

The values of this register can be rewritten only when TAPAnFLG.TAPAnACE is 0 and TAUDnTEM for the corresponding TAUD's master channel is 0.

**Access:** This register can be read or written in 16-bit units.

**Address:** <TAPAn\_base> + 20H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TAPAn DCM	TAPAn DCN	TAPAn DCP	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

**Table 27.11 TAPAnCTL0 Register Contents**

Bit Position	Bit Name	Function															
15 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															
4	TAPAnDCM	Clear Condition Configuration This control bit specifies the clear conditions for Hi-Z control output. 0: Enables manipulation of TAPAnOPHT0 regardless of the TAPATHASIN signal input level. 1: Enables manipulation of TAPAnOPHT0 only if the TAPATHASIN signal input is inactive.															
3, 2	TAPAnDCN, TAPAnDCP	Hi-Z Input Edge Selection These are control bits that specify the valid edge of TAPATHASIN. <table border="1" data-bbox="674 1167 1405 1403"> <thead> <tr> <th>TAPAn DCN</th> <th>TAPAn DCP</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Does not detect valid edges.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Detects a rising edge as the valid edge (active level = high).</td> </tr> <tr> <td>1</td> <td>0</td> <td>Detects a falling edge as the valid edge (active level = low).</td> </tr> <tr> <td>1</td> <td>1</td> <td>Setting prohibited</td> </tr> </tbody> </table>	TAPAn DCN	TAPAn DCP	Description	0	0	Does not detect valid edges.	0	1	Detects a rising edge as the valid edge (active level = high).	1	0	Detects a falling edge as the valid edge (active level = low).	1	1	Setting prohibited
TAPAn DCN	TAPAn DCP	Description															
0	0	Does not detect valid edges.															
0	1	Detects a rising edge as the valid edge (active level = high).															
1	0	Detects a falling edge as the valid edge (active level = low).															
1	1	Setting prohibited															
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.															

### 27.3.3 TAPAnCTL1 — TAPA Control Register 1

This register is used to specify the A/D conversion trigger.

**Access:** This register can be read or written in 8-bit units.

**Address:** <TAPAn\_base> + 24H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAPAnATS3	TAPAnATS2	TAPAnATS1	TAPAnATS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

**Table 27.12 TAPAnCTL1 Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	TAPAnATS3, TAPAnATS2	A/D Conversion Trigger 1 Selection These are control bits that specify A/D conversion trigger output 1 (TAPATADOUT1).
TAPAn ATS3	TAPAn ATS2	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	Setting prohibited
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 signal
1, 0	TAPAnATS1, TAPAnATS0	A/D Conversion Trigger 0 Selection These are control bits that specify the A/D conversion trigger output 0 (TAPATADOUT0).
TAPAn ATS1	TAPAn ATS0	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	Setting prohibited
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 signal

### 27.3.4 TAPAnFLG — TAPA Flag Register

This flag register is for asynchronous Hi-Z control.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <TAPAn\_base> + 00H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	TAPAn HOF2	TAPAn HOF1	TAPAn HOF0	—	—	—	—	—	—	—	TAPAn ACE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 27.13 TAPAnFLG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned.
10	TAPAnHOF2	W phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT2 is high level 1: The present output TAPAnTHZOUT2 is low level.
9	TAPAnHOF1	V phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT1 is high level 1: The present output TAPAnTHZOUT1 is low level.
8	TAPAnHOF0	U phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT0 is high level 1: The present output TAPAnTHZOUT0 is low level.
7 to 1	Reserved	When read, the value after reset is returned.
0	TAPAnACE	Asynchronous Hi-Z Control Enable 0: Indicates that the asynchronous Hi-Z control is stopped. 1: Indicates that the asynchronous Hi-Z control is enabled. The conditions for setting or clearing this bit are as follows: Clear condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1 Set condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1

### 27.3.5 TAPAnACWE — TAPA Asynchronous Hi-Z Control Write Enable Register

This register is used to enable writing for asynchronous Hi-Z control.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<TAPAn_base> + 04H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACWE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 27.14 TAPAnACWE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TAPAnACWE	Asynchronous Control Write Enable This is a write enable bit for asynchronous Hi-Z control. After 1 is written, this bit is automatically cleared to 0 by writing 1 to TAPAnACTS and TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT.

### 27.3.6 TAPAnACTS — TAPA Asynchronous Hi-Z Control Start Trigger Register

This register is used to enable the start trigger for asynchronous Hi-Z control.

<b>Access:</b>	This register is a write-only register that can be written in 8-bit units. This register is always read as 00H.							
<b>Address:</b>	<TAPAn_base> + 08H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.15 TAPAnACTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTS	Asynchronous Hi-Z Control Start Trigger This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Enables asynchronous Hi-Z control when TAPAnACE is 1.

### 27.3.7 TAPAnACTT — TAPA Asynchronous Hi-Z Control Stop Trigger Register

This bit enables the stop trigger for asynchronous Hi-Z control.

	<b>Access:</b>	This register is a write-only register that can be written in 8-bit units. This register is always read as 00H.						
	<b>Address:</b>	<TAPAn_base> + 0C <sub>H</sub>						
	<b>Value after reset:</b>	00H						
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.16 TAPAnACTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTT	Asynchronous Hi-Z Control Stop Trigger This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Disables asynchronous Hi-Z control when TAPAnACE is 0.

### 27.3.8 TAPAnOPHS — TAPA Hi-Z Start Trigger Register

This software trigger register is used to start Hi-Z control for motor control output pins.

	<b>Access:</b>	This register is a write-only register that can be written in 8-bit units. This register is always read as 00H.						
	<b>Address:</b>	<TAPAn_base> + 14 <sub>H</sub>						
	<b>Value after reset:</b>	00H						
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.17 TAPAnOPHS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHS0	Hi-Z Control Start Trigger This bit starts Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Starts Hi-Z control.

### 27.3.9 TAPAnOPHT — TAPA Hi-Z Stop Trigger Register

This software trigger register is used to stop Hi-Z control for motor control output pins.

<b>Access:</b>	This register is a write-only register that can be written in 8-bit units. This register is always read as 00H.							
<b>Address:</b>	<TAPAn_base> + 18H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.18 TAPAnOPHT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHT0	Hi-Z Control Stop Trigger This bit stops Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Stops Hi-Z control. Whether the setting of this bit is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

### 27.3.10 TAPAnEMU — TAPA Emulation Register

This register controls SVSTOP for emulation.

<b>Access:</b>	This register can be read or written in 8-bit units. (when SVSTOP = low, rewritten only)							
<b>Address:</b>	<TAPAn_base> + 28H							
<b>Value after reset:</b>	Reading this register returns always 00H.							
Bit	7	6	5	4	3	2	1	0
	TAPAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 27.19 TAPAnEMU Register Contents

Bit Position	Bit Name	Function
7	TAPAnSVSDIS	This bit is used to control disabling of SVSTOP. 0: SVSTOP is valid. (Sets Hi-Z control output to low level when SVSTOP = H is input). 1: SVSTOP is invalid. (Hi-Z control output level does not change according to the level of SVSTOP input).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

## 27.4 Asynchronous Hi-Z Control Function

If the operation of the timer motor control function controlled by the MCU becomes abnormal, the rotation of the external motor also becomes abnormal. This function can forcibly set the motor control output to the Hi-Z state upon detection of abnormal motor operation, independent of MCU control.

### 27.4.1 Overview

This function forcibly stops TAPAn output through asynchronous Hi-Z control.

- When the TAPATHASIN signal becomes active, the levels of the motor control output pins are set to Hi-Z, and motor control output is forcibly stopped.
- Motor control output in a Hi-Z state can be resumed by writing the Hi-Z stop trigger register (TAPAnOPHT0).
- The Hi-Z state of motor control output can also be specified by writing the Hi-Z control start trigger register (TAPAnOPHS).
- Setting PIC can enable or disable Hi-Z control input when an error occurs.

### 27.4.2 System Configuration Example

A system configuration example is shown below, where an external error detection signal (the TAPA0ESO signal) is used for Hi-Z control of the motor control outputs (TAPA0UP / TAPA0UN / TAPA0VP / TAPA0VN / TAPA0WP / TAPA0WN).

When valid edges of the external error detection signal are detected, the level of the motor control outputs is set to Hi-Z.

Because the microcontroller might freeze when an error occurs, external error detection signals are continuously processed so that the motor control timer outputs can be set to Hi-Z even if no clock is supplied.

Note that an error is detected only when the valid edge of the error detection signal is detected. Therefore, no error is detected if the output level is fixed and the signal level does not change.

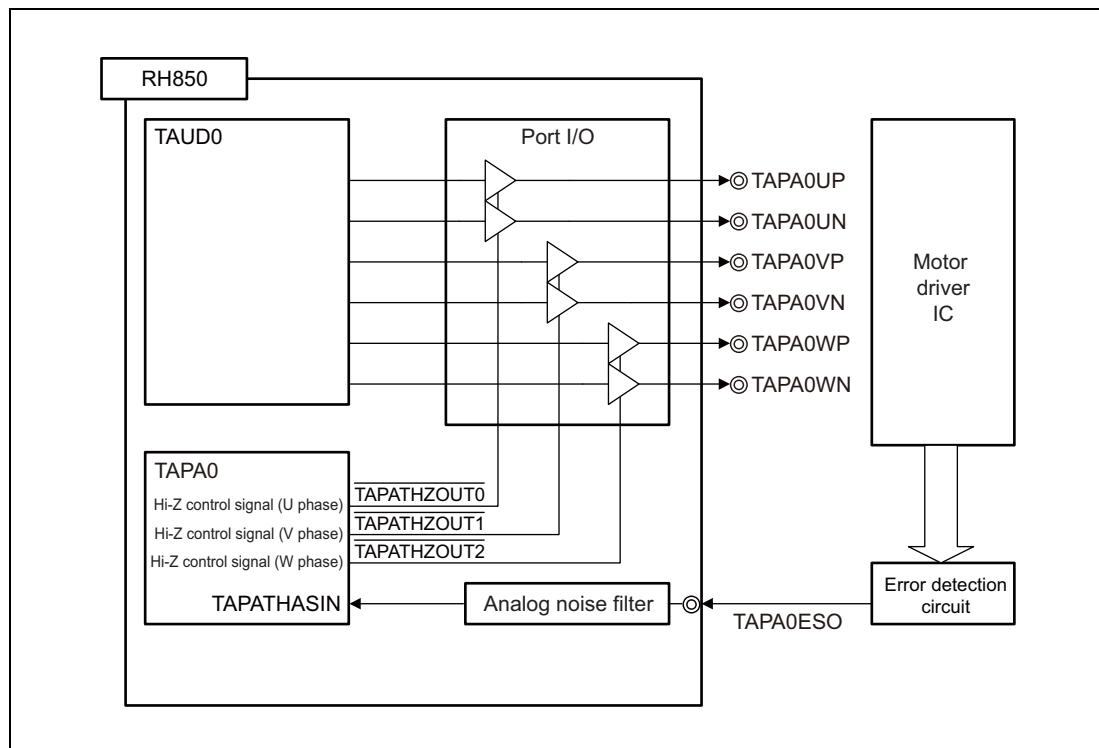


Figure 27.5 System Configuration Example of Asynchronous Hi-Z Control for Pin Input

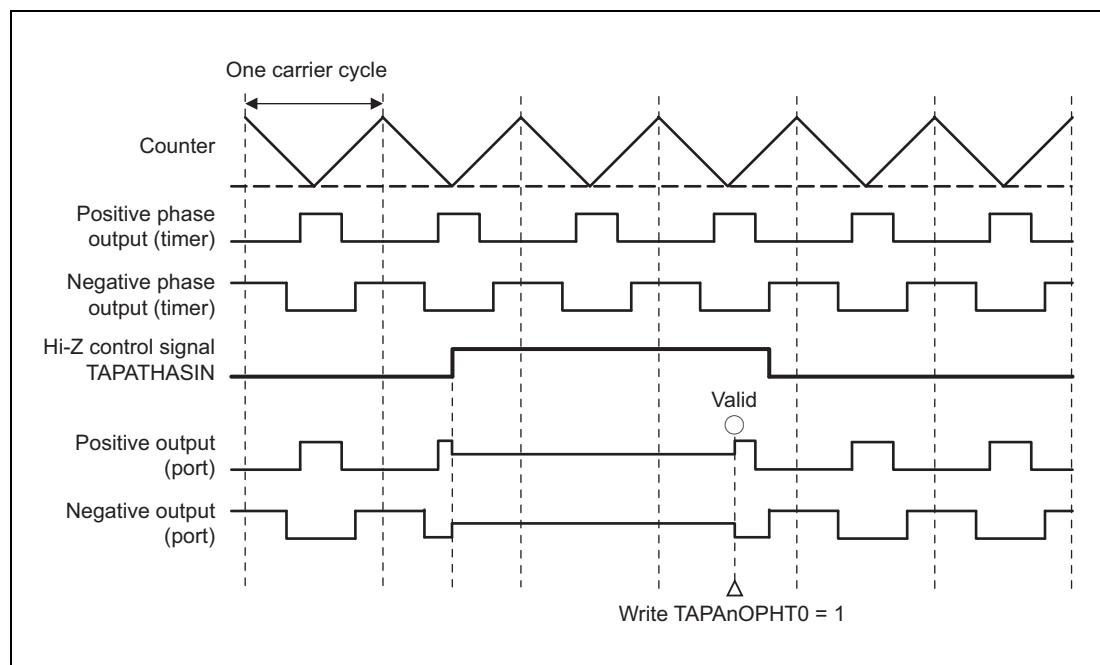
### 27.4.3 Basic Operation

Hi-Z control for motor control output pins can be started as follows:

- Detecting the valid edge of asynchronous Hi-Z control signal (TAPATHASIN)
- Setting the start trigger bit TAPAnOPHS.TAPAnOPHS0 of the Hi-Z control signal

The levels of the motor control output pins are set to Hi-Z until the stop trigger bit of the Hi-Z control signal (TAPAnOPHT.TAPAnOPHT0) is set. Note that whether the setting of TAPAnOPTH0 is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

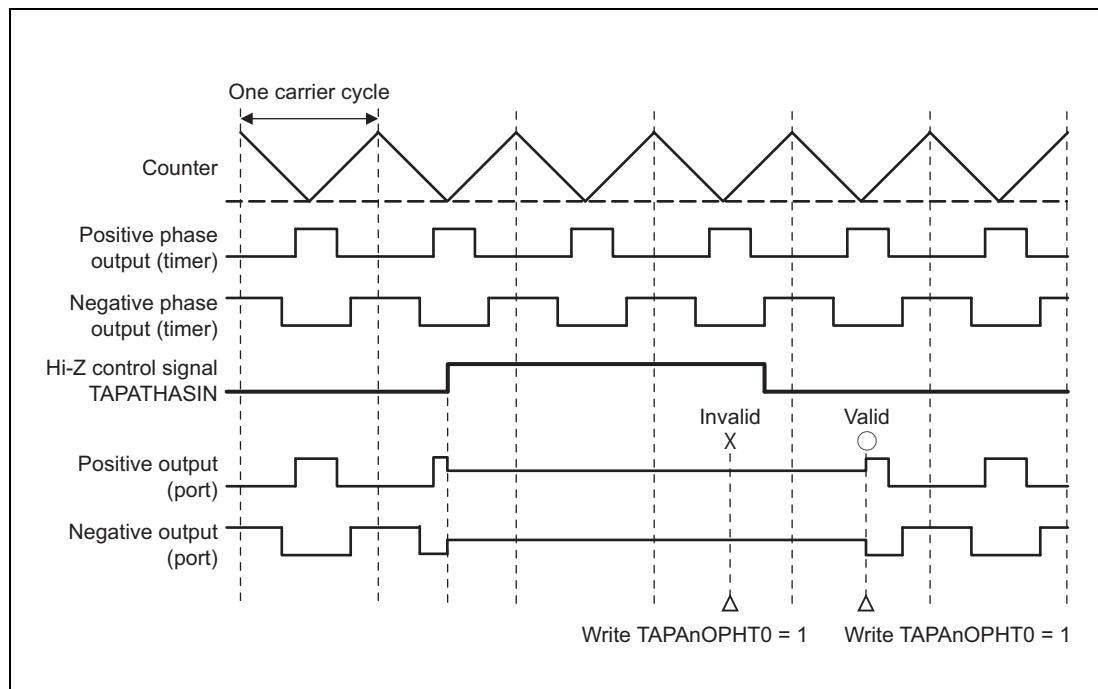
#### (1) Operation when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPATHASIN is detected.

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0, regardless of the level of TAPATHASIN.

## (2) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPATHASIN is detected.

Writing 1 to the stop trigger bit (TAPAnOPHT.TAPAnOPHT0) of the Hi-Z control signal is ignored while TAPATHASIN is active (high level because TAPAnCTL0.TAPAnDCP is 1).

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0 after TAPATHASIN becomes inactive (low level because TAPAnCTL0.TAPAnDCP is 1).

#### 27.4.4 Asynchronous Hi-Z Control Using Software Trigger

Hi-Z control for motor control output is possible by using the Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0 and Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0.

##### (1) Function of Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0

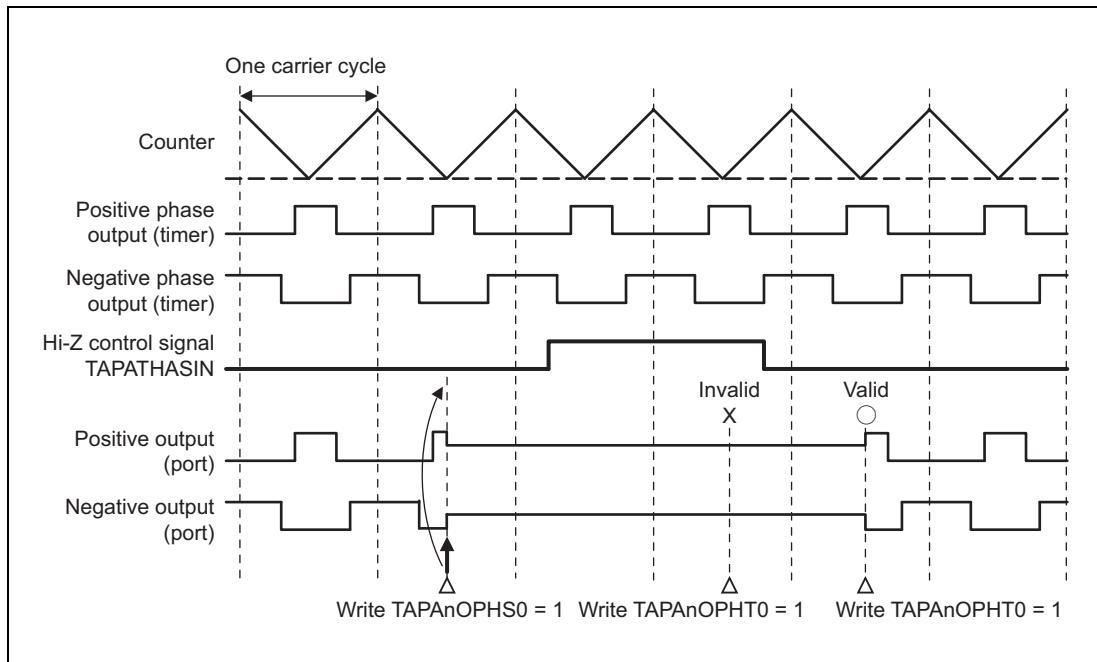
TAPAnDCM	Function
0/1	Writing 1 to TAPAnOPHS0 starts Hi-Z control and forcibly stops the motor control output (Hi-Z output).

##### (2) Function of Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0

Whether the Hi-Z control stop trigger is valid or invalid depends on the conditions below:

TAPAnDCM	Function
0	Writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output.
1	If TAPATHASIN is inactive, writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output. If TAPATHASIN is active, writing 1 to TAPAnOPHT0 is ignored.

##### (3) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control output (Hi-Z output) is forcibly stopped when 1 is written to TAPAnOPHS0.

After that, the level of the motor control output remains Hi-Z even if the rising edge of TAPATHASIN is detected.

Writing to TAPAnOPHT0 is ignored while TAPATHASIN is active (high level because TAPAnDCN is 0 and TAPAnDCP is 1).

After detection of the falling edge of TAPATHASIN, the motor control output restarts when 1 is written to TAPAnOPHT0 while TAPATHASIN is inactive (low level because TAPAnDCN is 0 and TAPAnDCP is 1).

### 27.4.5 Operating Procedure

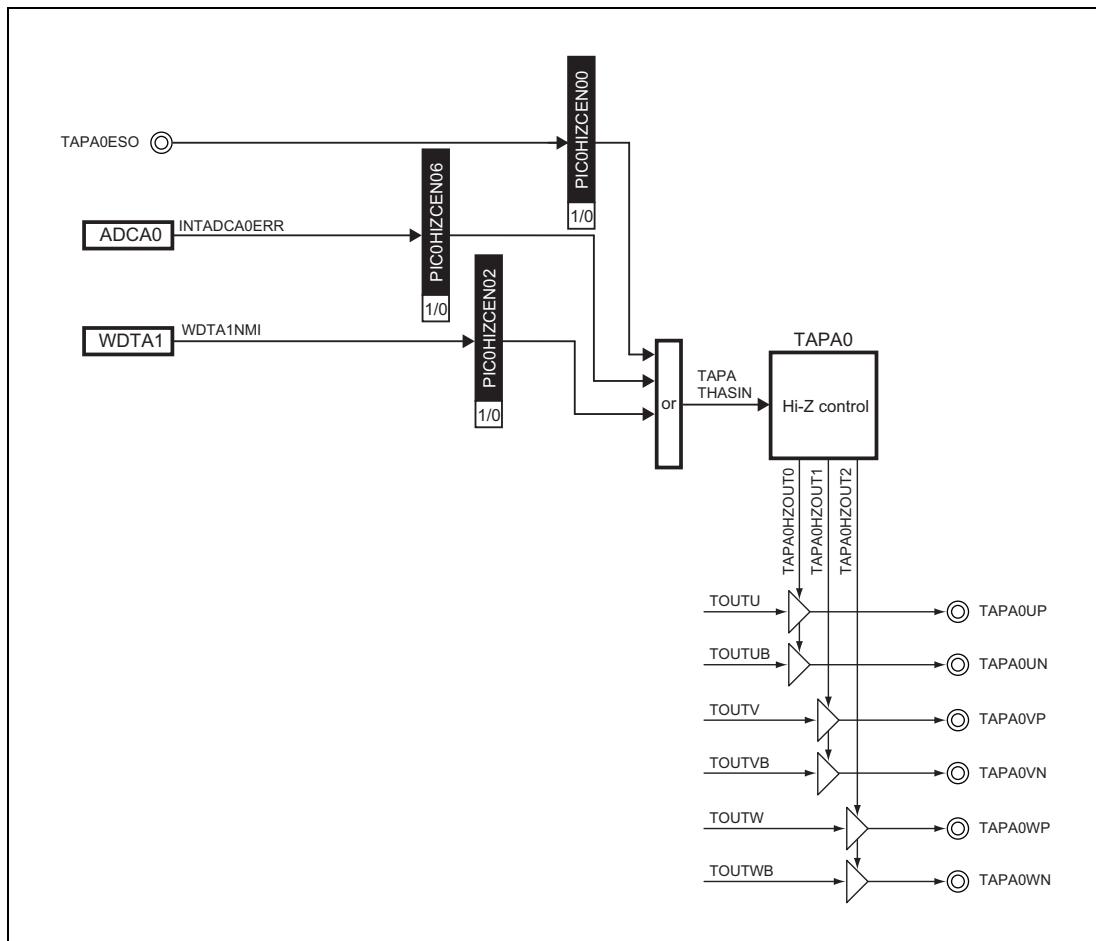
The operating procedure for the asynchronous input Hi-Z control function is shown below:

	<b>Operation</b>	<b>Status of TAPA</b>
Initial setup	Set up the TAPAnCTL0 register. Specify TAPAnDCP and TAPAnDCN to select the input edge. Specify TAPAnDCM to select the clear mode.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)
Start operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1.  Set up the TAPAnACTS register. Set TAPAnACTS to 1.	Writing to TAPAnACTS is enabled.  Asynchronous Hi-Z control enabled (TAPAnFLG.TAPAnACE = 1)
Restart	Hi-Z control for the timer function outputs can be started by controlling the following: <ul style="list-style-type: none"><li>• TAPAnOPHS register</li><li>• Asynchronous Hi-Z control signal (TAPATHASIN)</li></ul> Hi-Z control for the timer function outputs can be stopped by controlling the following: <ul style="list-style-type: none"><li>• TAPAnOPHT register (If TAPAnDCM is 1, control by the TAPAnOPHT register is enabled only while TAPATHASIN is inactive.)</li></ul> The TAPA operating status can always be read using the TAPAnFLG register.	Hi-Z control for the motor control output pins is started by detecting the valid edge of the asynchronous Hi-Z control signal (TAPATHASIN) or by setting the Hi-Z control start trigger bit TAPAnOPHS0 to 1. Hi-Z control for the motor control output pins is stopped by setting the Hi-Z control stop trigger bit TAPAnOPHT0 to 1 according to the operation mode specified by the TAPAnDCM bit.
During		
Stop operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1.  Set up the TAPAnACTT register. Set TAPAnACTT to 1.	Writing to TAPAnACTT is enabled.  Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)

### 27.4.6 TAPA0 Hi-Z Control Input Selection

In order to stop the motor control output in case of errors, error events are selected in PIC and the level of the motor control output is set to Hi-Z in TAPA0, as shown in the diagram below.

The TAPA function can be stopped by setting TAPA0ACTT = 01<sub>H</sub> after setting PIC0HIZCEN0 = 00<sub>H</sub> or TAPA0ACWE = 01<sub>H</sub>.



**Figure 27.6 Hi-Z Control Block Diagram**

Switching into a Hi-Z state can be performed by the following:

- TAPA0ESO pin input
- A/D converter ADCA0 error signal ADCA0ERR
- Window Watchdog Timer WDTA1 non maskable interrupt WDTA1NMI

For details about these signals, see the respective descriptions.

## 27.4.7 Registers

### 27.4.7.1 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

The PIC0HIZCENn register selects the Hi-Z output control signal of TAPAn.

**Access:** This register can be read or written in 8-bit units.

**Address:** <PIC0\_base> + 80<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R	R	R	R/W	R	R/W
Value after reset	0	0	0	0	0	0	0	0

Table 27.20 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Hi-Z Output Control by INTADCA0ERR Interrupt Signal Enable 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Hi-Z Output Control by the TAPA0ESO External Input Enable 0: Disable 1: Enable

## 27.5 INT Signal Output Selection Function

### 27.5.1 Configuration of the INT Signal Output Selection Function

This function generates the peak interrupt TAPATIPEK0 and valley interrupt TAPATIVLY0 by using the TAPATSIM0 signal, which is connected to the INT signal on the TAUD's triangular carrier cycle generation channel (master) and TAPATUDCM0 signal, which is connected to the counter up/down signal.

For the connection destination of TAPATSIM0, see **Section 27.1.7, Internal Output Signal**.

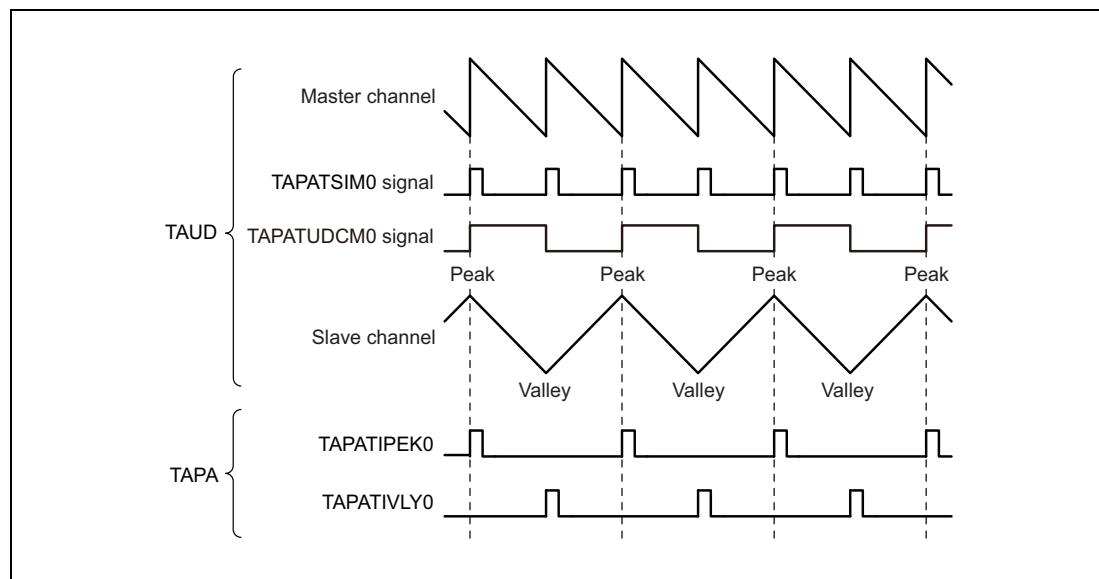


Figure 27.7 Basic Timing of Signals for the INT Signal Output Selection Function

Triangular carrier cycles are generated on the master channel.

The INT signal generated on the master channel in each half triangular carrier cycle is input to TAPAn as the TAPATSIM0 signal. TAPAn generates the TAPATIPEK0 signal (peak interrupt) during high level of the TAPATUDCM0 signal and the TAPATIVLY0 signal (valley interrupt) during low level of the TAPATUDCM0 signal by using the TAPATSIM0 and TAPATUDCM0 input signals.

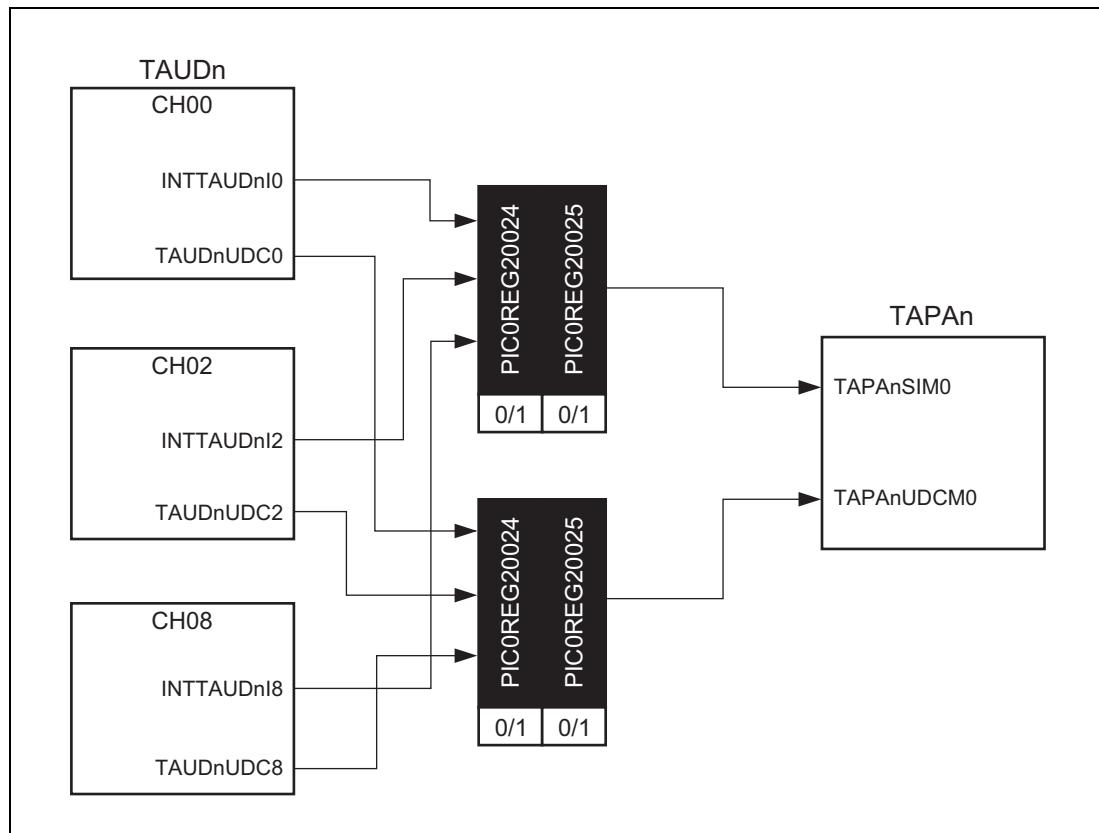
#### CAUTION

**The peak interrupt TAPATIPEK0 and valley interrupt TAPATIVLY0 are generated regardless of the function of the master channel of TAUD.**

**When not using these peak and valley interrupts, mask them by using the ICTAPAnIPEK0 and ICTAPAnIVLY0 registers, respectively.**

### 27.5.2 Block Diagram

TAUDn and TAPAn are connected in the registers shown below by the INT signal output selection function.



## 27.5.3 Registers

### 27.5.3.1 PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0)

This register selects the TAPA0 input.

**Access:** This register can be read or written in 32-bit units.

**Address:** PIC0REG200: FFDD 00C0<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC0RE G2n025	PIC0RE G2n024	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.21 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	*1
25, 24	PIC0REG2n025 PIC0REG2n024	Select the TAUDn channel used by TAPATSIM0 and TAPATUDCM0. 00: Not selected 01: TAUDO channel 0 selected 10: TAUDO channel 2 selected 11: TAUDO channel 8 selected
23 to 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n0 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

## 27.6 A/D Converter Conversion Trigger Selection Function

This function outputs the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1 from the TAPATCDENS0 and TAPATCDENS1 signals, which are connected to a compare match interrupt based on the triangular carrier cycle of TAUD, or a valley interrupt signal (TAPATIVLY0).

### 27.6.1 Configuration of A/D Converter Conversion Trigger Selection Function

**Table 27.22 Signals Used for TAPATADOUT Generation**

Output Signal	Slave Match Detection Signal	Valley Interrupt Signal
TAPATADOUT0	TAPATCDENS0	TAPATIVLY0
TAPATADOUT1	TAPATCDENS1	TAPATIVLY0

**Table 27.23 Operation of TAPATADOUT1 According to the Setting of TAPAnCTL1.TAPAnATS[3:2]**

TAPAnATS3	TAPAnATS2	Description
0	0	Outputs the INT signal from TAPATADOUT1 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPATADOUT1 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPATADOUT1 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 from TAPATADOUT1.

**Table 27.24 Operation of TAPATADOUT0 According to the Setting of TAPAnCTL1.TAPAnATS[1:0]**

TAPAnATS1	TAPAnATS0	Description
0	0	Outputs the INT signal from TAPATADOUT0 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPATADOUT0 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPATADOUT0 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 from TAPATADOUT0.

## 27.6.2 Block Diagram

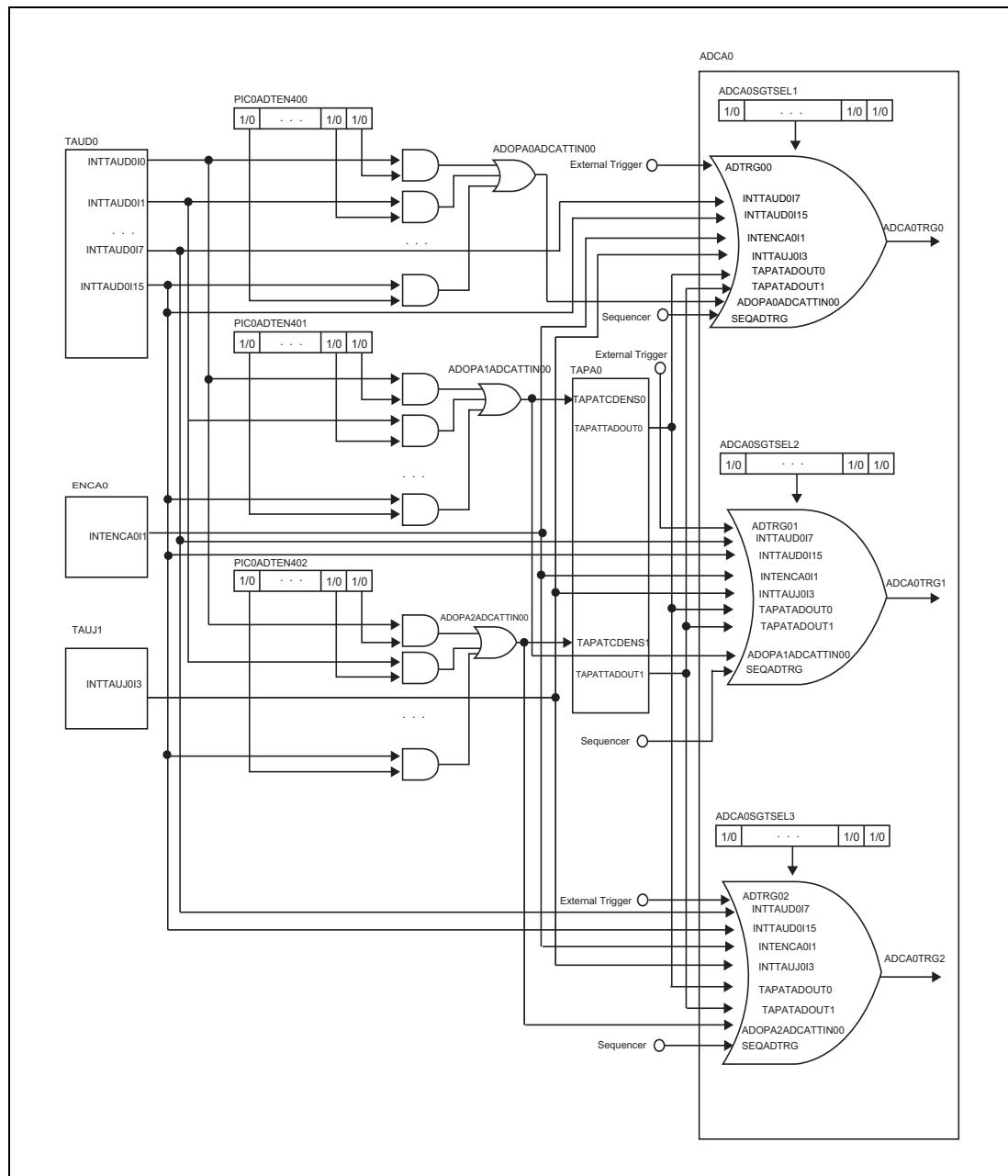
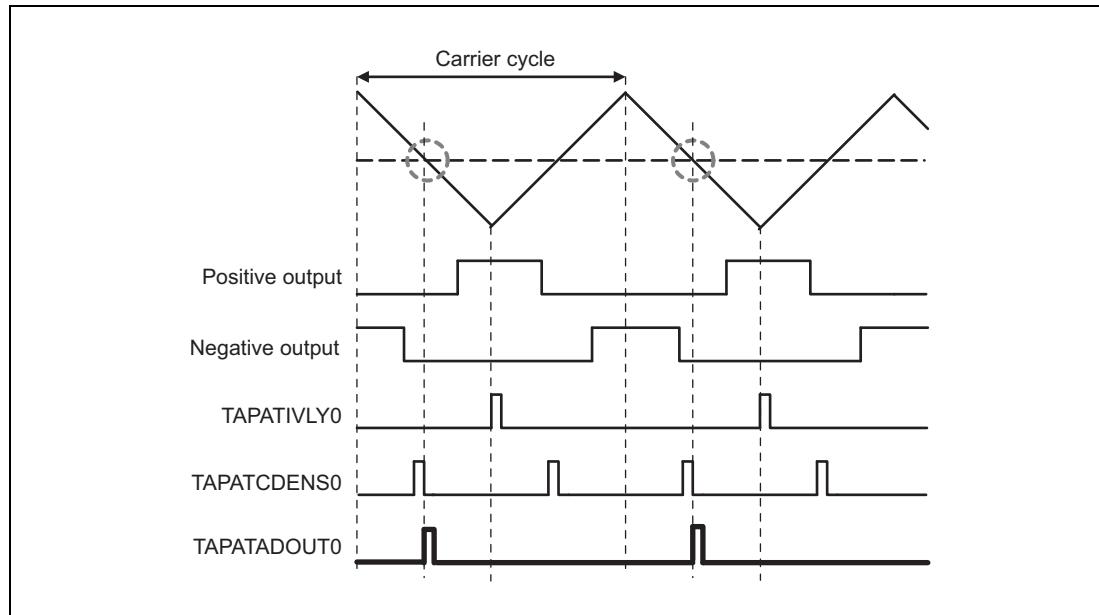


Figure 27.8 Block Diagram of A/D Conversion Trigger Selection Function

### NOTE

See **Section 29.3.4.1, ADCAnSGTSELx — Scan Group x Start Trigger Control Register x** for details on the settings of the ADCA0SGTSEL register.

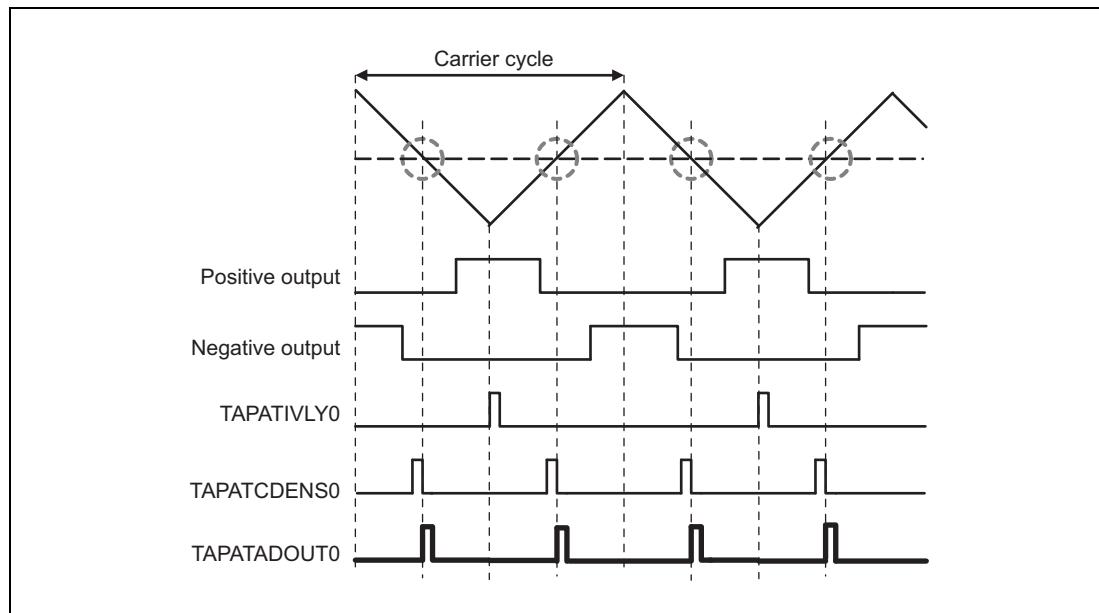
### 27.6.3 Waveforms of A/D Converter Conversion Trigger Output Control Operation in Triangle PWM Mode



**Figure 27.9 TAPAnATS[1:0] bits = 00<sub>B</sub>: Output INT Signal while the Triangle Wave is Falling (Counting Down)**

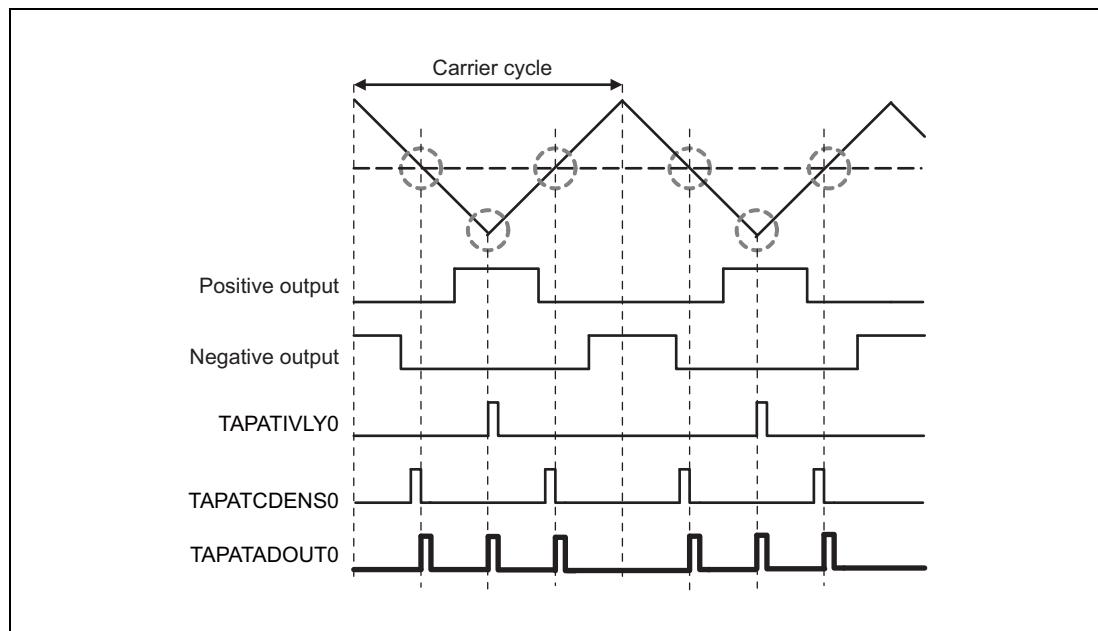
While the triangle wave is falling (counting down), the signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

In this case, no A/D converter conversion trigger signal is output while the triangle wave is rising (counting up).



**Figure 27.10 TAPAnATS[1:0] bits = 10<sub>B</sub>: Output INT Signal while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)**

The signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1.



**Figure 27.11 TAPAnATS[1:0] bits = 11<sub>B</sub>: Output of INT Signal and Valley Interrupt while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)**

The signals TAPATCDENS0 and TAPATCDENS1 and valley interrupt TAPATIVLY0 are output as the A/D converter conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

#### 27.6.4 Operating Procedure for A/D Converter Conversion Trigger Selection Function

The operating procedure for the A/D converter conversion trigger selection function is shown below.

	Operation	Status of TAUD and TAPA
Initial setup	Initialize TAUD. Specify the timer operation mode.  Set up the TAPAnCTL1 register. Specify TAPAnATS[1:0] (TAPATADOUT0 setting). Specify TAPAnATS[3:2] (TAPATADOUT1 setting).  Set up the PIC0ADTEN4nj and PIC0REG2n0 registers according to the signal to be used. Specify PIC0ADTEN4nj (TAPATCDENS0 or TAPATCDENS1 setting). Specify PIC0REG2n0 (TAPATIVLY0 setting).	TAUD and TAPA stop the operation.
Start operation	Start the TAUD operation.	TAUD starts the count operation.
During operation	TAUD operates according to the setting of each function.	The A/D conversion trigger selection function outputs either TAPATADOUT0 according to the setting of TAPAnATS[1:0] or TAPATADOUT1 according to the setting of TAPAnATS[3:2], based on the interrupt TAPATCDENS1 or TAPATCDENS0, which is input from TAUD, and the valley interrupt TAPATIVLY0, which is generated by TAPA.
Stop operation	Stop the TAUD operation.	TAUD stops the count operation.

## 27.7 ADCA Trigger Selection Function

### 27.7.1 Functional Overview

This function generates ADCA hardware trigger signals by using TAUDn channel output.

### 27.7.2 Configuration

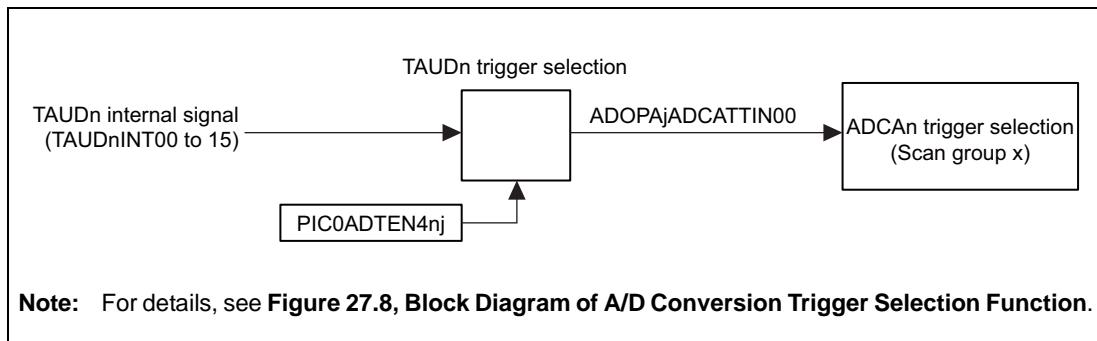


Figure 27.12 Block Diagram of ADCA Trigger Selection Function

## 27.7.3 Registers

### 27.7.3.1 PIC0ADTEN4nj — A/D Conversion Trigger Output Control Register 4nj (n = 0, j = 0 to 2)

This register selects an ADCA0 start trigger source from TAUDn channel m. (m = 0 to 15)

**Access:** This register can be read or written in 16-bit units.

**Address:** <PIC0\_base> + 90H + 4 × j

**Value after reset:** 0000 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC0 ADTEN 4nj15	PIC0 ADTEN 4nj14	PIC0 ADTEN 4nj13	PIC0 ADTEN 4nj12	PIC0 ADTEN 4nj11	PIC0 ADTEN 4nj10	PIC0 ADTEN 4nj09	PIC0 ADTEN 4nj08	PIC0 ADTEN 4nj07	PIC0 ADTEN 4nj06	PIC0 ADTEN 4nj05	PIC0 ADTEN 4nj04	PIC0 ADTEN 4nj03	PIC0 ADTEN 4nj02	PIC0 ADTEN 4nj01	PIC0 ADTEN 4nj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W															

Table 27.25 PIC0ADTEN4nj Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0ADTEN4nj	Sets a trigger source of CHm (m = 0 to 15) in the TAUDn timer.
15 to 00	PIC0ADTEN4nj	0: A/D trigger source of CHm in the TAUDn timer is disabled. 1: A/D trigger source of CHm in the TAUDn timer is enabled.

#### 27.7.4 Example of Operation

- (1) Initial setting: Set the function of each channel of the TAUD0 to be used.
- (2) Setting of the A/D conversion trigger output control register 4nj (PIC0ADTEN4nj):  
Setting the bits of A/D conversion trigger output control register 4nj ((PIC0ADTEN4nj) to 1 enables selection of an interrupt request signal from each channel of the TAUD0 as the trigger of the A/D conversion scan group.
  - Register setting should be performed when A/D conversion is stopped.
- (3) Setting of the A/D conversion trigger selection control register (ADCA0SGTSELx):  
Setting the bits corresponding to each trigger to 1 enables to use the signal generated by executing the logical OR of each trigger as the start trigger of the A/D conversion scan group.
  - Register setting should be performed when the A/D conversion is stopped.
- (4) Enabling of TAUD0 timer operation  
Each channel of the TAUD0 timer set in (1) starts.

#### 27.7.5 Setup Flow

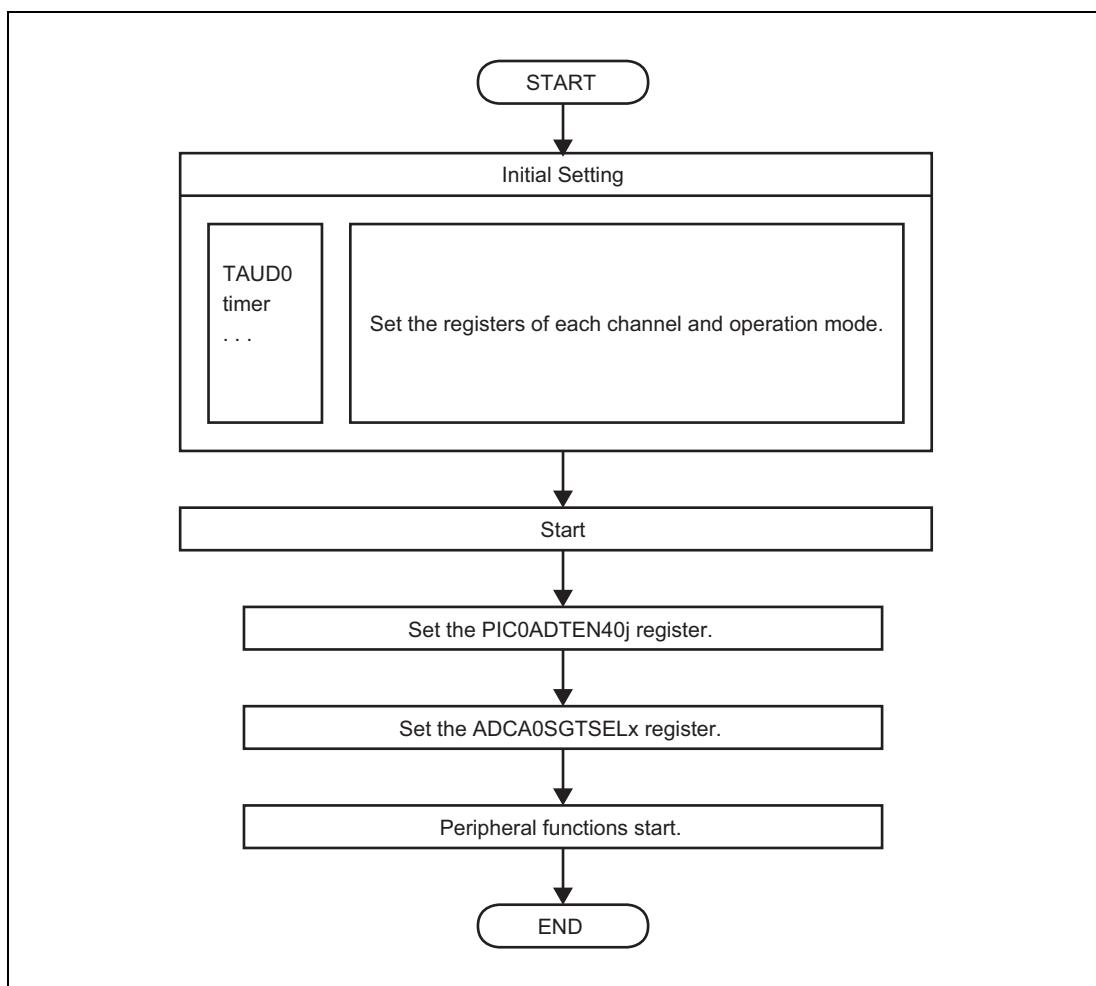


Figure 27.13 Setup Flow (j = 0 to 2)

## 27.8 Simultaneous Start Trigger Function

### 27.8.1 Functional Overview

The timers (TAUD0, TAUJ1, ENCA0) can be simultaneously started in any combination.

### 27.8.2 Configuration

#### (1) Configuration

Table 27.26 Configuration of Simultaneous Start Trigger Function

Configuration/Timer Function	Timer
Configuration of Timer	TAUD0, TAUJ1, ENCA0

#### (2) Block Diagram

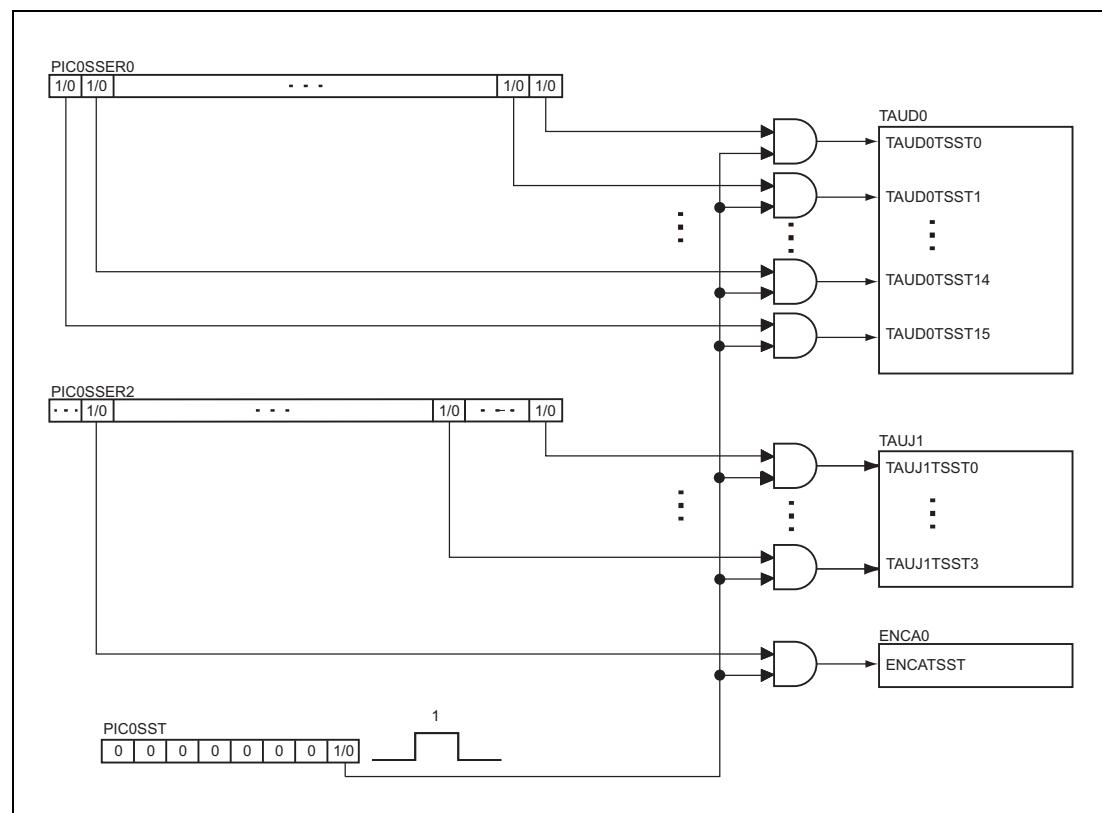


Figure 27.14 Block Diagram of Simultaneous Start Trigger

## 27.8.3 Registers

### 27.8.3.1 PIC0SSER0 — Simultaneous Start Control Register 0

The PIC0SSER0 register enables a start trigger for each channel of the TAUD0.

**Access:** This register can be read or written in 16-bit units

**Address:** <PIC0\_base> + 10H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC0SS ER015	PIC0SS ER014	PIC0SS ER013	PIC0SS ER012	PIC0SS ER011	PIC0SS ER010	PIC0SS ER009	PIC0SS ER008	PIC0SS ER007	PIC0SS ER006	PIC0SS ER005	PIC0SS ER004	PIC0SS ER003	PIC0SS ER002	PIC0SS ER001	PIC0SS ER000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.27 PIC0SSER0 Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0SSER015 to PIC0SSER000	Enable a simultaneous start trigger for the CHm in the TAUD0. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

### 27.8.3.2 PIC0SSER2 — Simultaneous Start Control Register 2

The PIC0SSER2 register enables a start trigger for ENCA0 and TAUJ1.

**Access:** This register can be read or written in 16-bit units

**Address:** <PIC0\_base> + 18H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC0SS ER214( ENCA0)	—	—	—	—	—	—	—	—	—	—	PIC0SS ER203	PIC0SS ER202	PIC0SS ER201	PIC0SS ER200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 27.28 PIC0SSER2 Register Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	PIC0SSER214	Enables a simultaneous start trigger for the ENCA0 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.
13 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	PIC0SSER203 to PIC0SSER200	Set a simultaneous start trigger for the CHm in the TAUJ1 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

### 27.8.3.3 PIC0SST — Simultaneous Start Trigger Control Register

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <PIC0\_base> + 04H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SYNCTRG
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.29 PIC0SST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	SYNCTRG	Generates a start trigger for the timer whose simultaneous start is enabled. When read, this bit is always read as 0. 0: Disabled. 1: Simultaneous start trigger is generated (the pulse in the width of 1PCLK is output).

#### 27.8.4 Example of Operation

- (1) Operation example of timer configuration:  
The timers that operates in operation mode to be selected can be simultaneously started in any combinations.
- (2) Simultaneous start enable:  
Setting the relevant bits in the PIC0SSER0 and PIC0SSER2 registers of the target timers to be simultaneously started to 1 enables these timers to simultaneously start.
- (3) Start trigger output:  
Writing 1 to the SYNCTRG bit in the PIC0SST register enables the target timers set in (2) to simultaneously start.
- (4) Repeating (2) and (3) for the channels that have not started yet enables the different target timers to simultaneously start in multiple batches.

### 27.8.5 Setup Flow

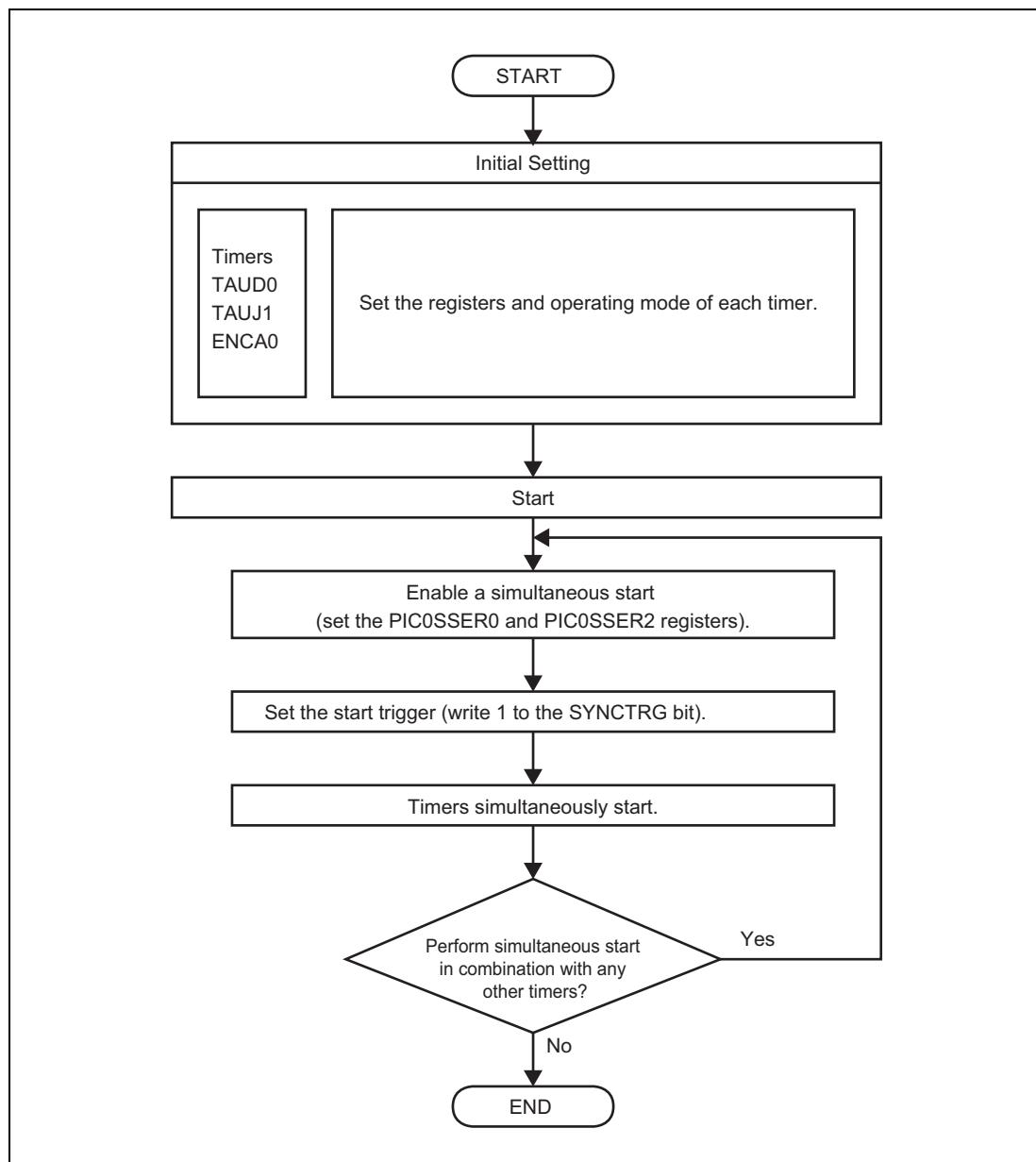


Figure 27.15 Setup Flow

## 27.9 Trigger & Pulse Width Measuring Function

### 27.9.1 Functional Overview

This function allows measurement of trigger periods by inputting the trigger signal output from ENCA0 to TAUJ1 and TAUD0.

### 27.9.2 Configuration

#### (1) Configuration

**Table 27.30 Configuration of Trigger & Pulse Width Measuring Function**

Configuration/Timer Function	Timer
Configuration of Timer	ENCA0, TAUD0, TAUJ1

**Table 27.31 Setting Functions of TAUJ1/TAUD0 Channels**

TAU	Channels	Functions Name	M/S <sup>*1</sup>	Target Trigger of Pulse Width Measurement
TAUJ1	00	TINm input pulse interval measurement function	S	ENCAT0IEC <sup>*2</sup>
	01	TINm input pulse interval measurement function	S	ENCAT0IEC <sup>*2</sup>
TAUD0	00	TINm input pulse interval measurement function	S	ENCAT0EQ0, ENCAT0EQ1
	01	TINm input pulse interval measurement function	S	ENCAT0EQ1
	02	TINm input pulse interval measurement function	S	ENCAT0EQ0

Note 1. M: Master channel, S: Slave channel

Note 2. Read ENCAT0IEC as ENCATICC (encoder clear interrupt) in **Table 26.5, Interrupt Requests**.

## (2) Block Diagram

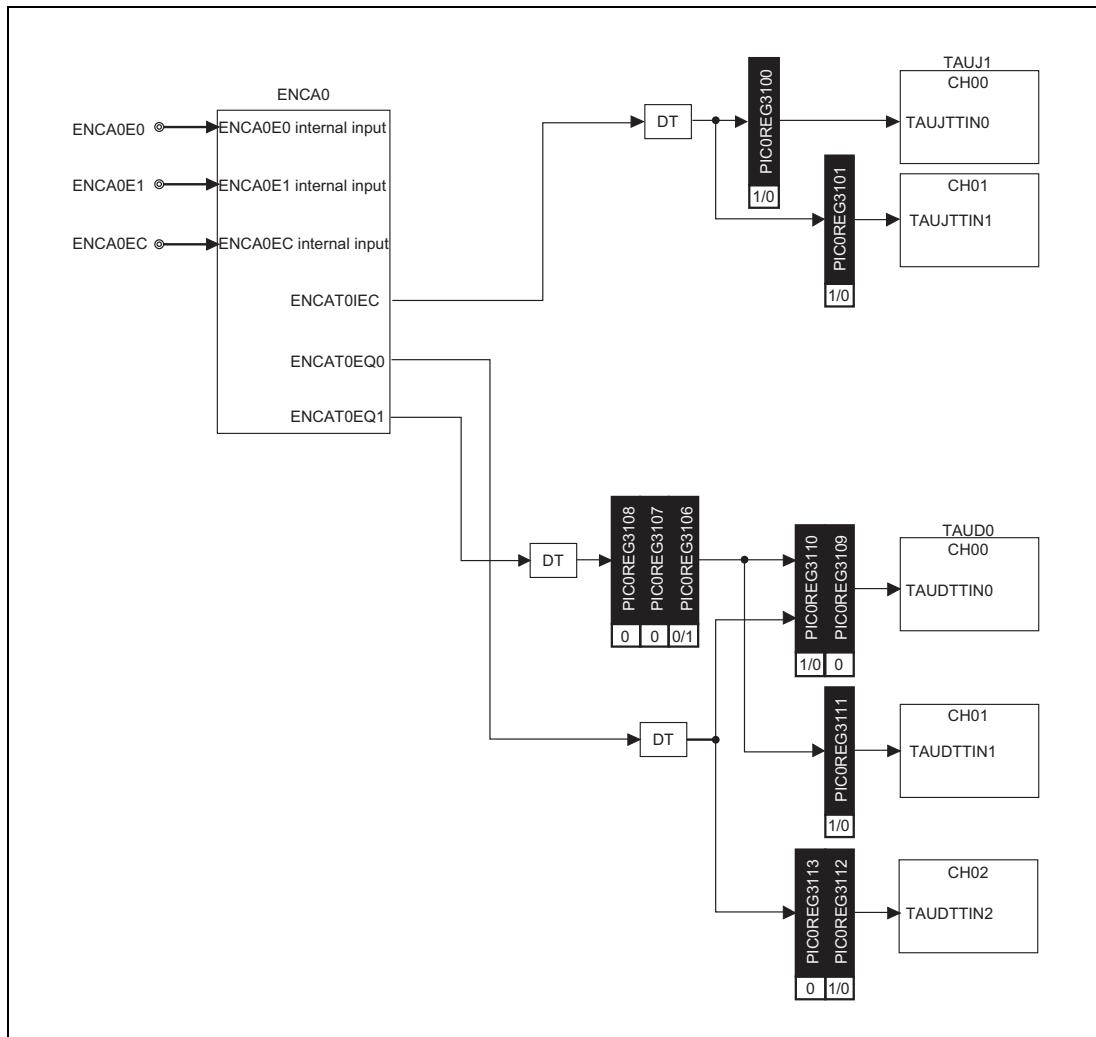


Figure 27.16 Block Diagram of Trigger &amp; Pulse Width Measuring Function

## 27.9.3 Registers

### 27.9.3.1 PIC0REG31 — Timer I/O Control Register 31

**Access:** This register can be read or written in 32-bit units.

**Address:** <PIC0\_base> + EC<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PIC0REG3113	PIC0REG3112	PIC0REG3111	PIC0REG3110	PIC0REG3109	PIC0REG3108	PIC0REG3107	PIC0REG3106	—	—	—	—	PIC0REG3101	PIC0REG3100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R/W	R/W							

Table 27.32 PIC0REG31 Register Contents (1/2)

Bit Position	Bit Name	Function																
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																
13, 12	PIC0REG3113 to PIC0REG3112	Select a TIN input signal to CH2 of TAUD0.  <table border="1"> <thead> <tr> <th>PIC0REG3113</th> <th>PIC0REG3112</th> <th>Input Signal</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CH2 of TAUD0 is not used for trigger width measurement.</td> </tr> <tr> <td>0</td> <td>1</td> <td>DT output signal of ENCATEQ0</td> </tr> <tr> <td>Other than above</td> <td></td> <td>Setting prohibited</td> </tr> </tbody> </table>	PIC0REG3113	PIC0REG3112	Input Signal	0	0	CH2 of TAUD0 is not used for trigger width measurement.	0	1	DT output signal of ENCATEQ0	Other than above		Setting prohibited				
PIC0REG3113	PIC0REG3112	Input Signal																
0	0	CH2 of TAUD0 is not used for trigger width measurement.																
0	1	DT output signal of ENCATEQ0																
Other than above		Setting prohibited																
11	PIC0REG3111	Select a TIN input signal to CH1 of TAUD0. 0: CH1 of TAUD0 is not used for trigger width measurement. 1: Signal selected in PIC0REG3106 to PIC0REG3108 (when measuring the ENCATEQ1 signal)																
10, 9	PIC0REG3110 to PIC0REG3109	Select a TIN input signal to CH0 of TAUD0.  <table border="1"> <thead> <tr> <th>PIC0REG3110</th> <th>PIC0REG3109</th> <th>Input Signal</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Signal selected in PIC0REG3106 to PIC0REG3108</td> </tr> <tr> <td>1</td> <td>0</td> <td>DT output signal of ENCATEQ0</td> </tr> <tr> <td>Other than above</td> <td></td> <td>Setting prohibited</td> </tr> </tbody> </table>	PIC0REG3110	PIC0REG3109	Input Signal	0	0	Signal selected in PIC0REG3106 to PIC0REG3108	1	0	DT output signal of ENCATEQ0	Other than above		Setting prohibited				
PIC0REG3110	PIC0REG3109	Input Signal																
0	0	Signal selected in PIC0REG3106 to PIC0REG3108																
1	0	DT output signal of ENCATEQ0																
Other than above		Setting prohibited																
8 to 6	PIC0REG3108 to PIC0REG3106	Select a TIN input signal to CH0 and CH1 of TAUD0.  <table border="1"> <thead> <tr> <th>PIC0REG3108</th> <th>PIC0REG3107</th> <th>PIC0REG3106</th> <th>Input Signal</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>CH0 of TAUD0 is not used for trigger width measurement.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>DT output signal of ENCATEQ1</td> </tr> <tr> <td>Other than above</td> <td></td> <td></td> <td>Setting prohibited</td> </tr> </tbody> </table>	PIC0REG3108	PIC0REG3107	PIC0REG3106	Input Signal	0	0	0	CH0 of TAUD0 is not used for trigger width measurement.	0	0	1	DT output signal of ENCATEQ1	Other than above			Setting prohibited
PIC0REG3108	PIC0REG3107	PIC0REG3106	Input Signal															
0	0	0	CH0 of TAUD0 is not used for trigger width measurement.															
0	0	1	DT output signal of ENCATEQ1															
Other than above			Setting prohibited															
5 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																

**Table 27.32 PIC0REG31 Register Contents (2/2)**

Bit Position	Bit Name	Function
1	PIC0REG3101	Selects a TIN input signal to CH1 of TAUJ1. 0: CH1 of TAUJ1 is not used for trigger width measurement. 1: DT output signal of ENCAT0IEC
0	PIC0REG3100	Selects a TIN input signal to CH0 of TAUJ1. 0: CH0 of TAUJ1 is not used for trigger width measurement. 1: DT output signal of ENCAT0IEC

### 27.9.4 Example of Operation

The trigger and pulse width measurement function is achieved by combining the ENCA0 trigger signals (ENCAT0IEC, ENC0EQ0, ENC0EQ1) and the following functions of TAUD0 and TAUJ1.

- TAUDTTINm input pulse interval measurement function (TAUD0)
- TAUJTTINm input pulse interval measurement function (TAUJ1)

Also, the following function of PIC is used to convert the trigger signal input to TINm into a level-sensitive toggle signal.

- DT circuit

The trigger and pulse width measurement function implements measurement of the ENCA0 output trigger signal interval using the TAUDTTINm input pulse interval measurement function of TAUD0 and the TAUJTTINm input pulse interval measurement function of TAUJ1.

#### (1) TAUDTTINm input pulse interval measurement function, TAUJTTINm input pulse interval measurement function

When the valid TINm edge of TAUD0 or TAUJ1 is detected, the CNTm value is captured into CDRm and CNTm is cleared.

##### **CAUTION**

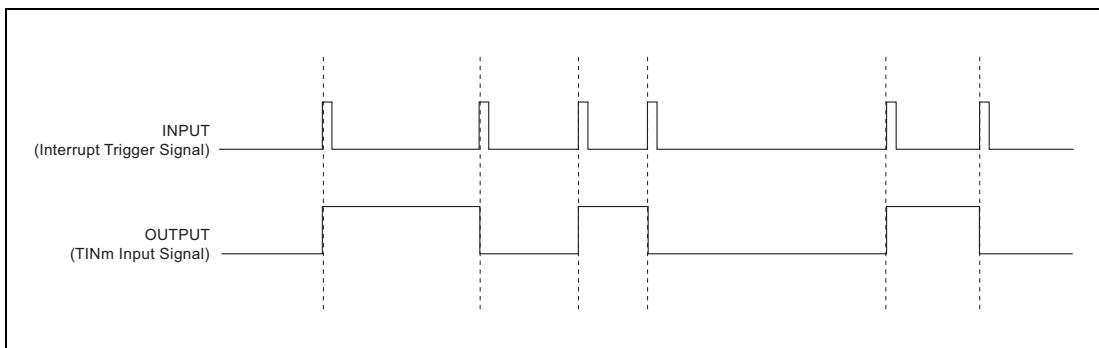
**Set both edges (rising and falling edges) of TINm to be detected as valid (TAUD0CMURm.TAUD0TIS[1:0] = 10<sub>B</sub>, TAUJ1CMURm.TAUJ1TIS[1:0] = 10<sub>B</sub>) for this function.**

For details of the TAUD and TAUJ functions, see the corresponding sections.

#### (2) DT circuit

The DT circuit is used to convert the trigger signal output from ENCA0 into a level-sensitive toggle signal.

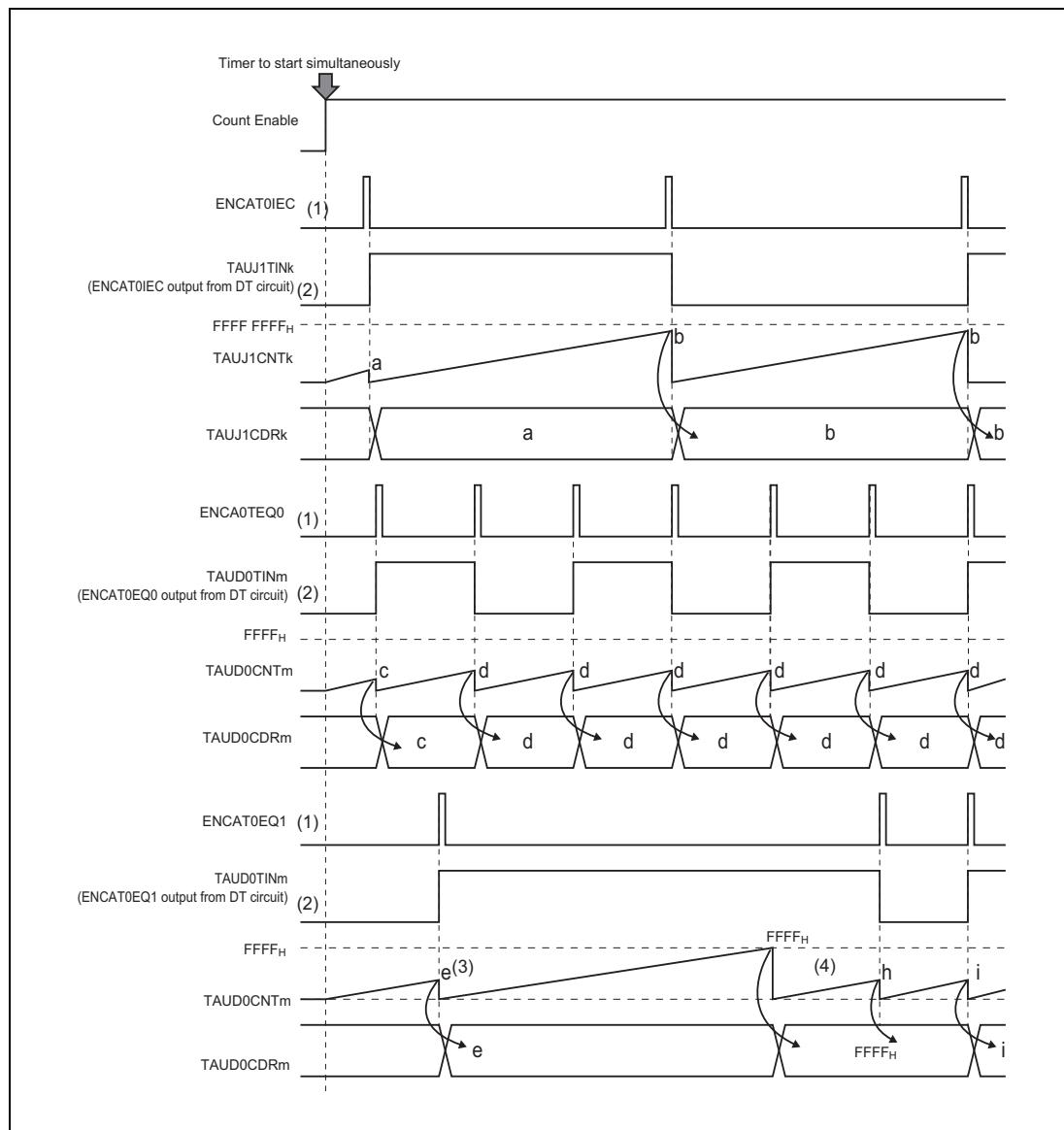
As shown in **Figure 27.17, Operation of DT Circuit**, the output signal is toggled upon each input trigger signal generation.



**Figure 27.17 Operation of DT Circuit**

PIC provides input signal conversion and signal connection to TAUD0 and TAUJ1 to measure the generation interval of trigger signals from ENCA0.

The timing chart of the trigger and pulse width measurement function is shown below.



**Figure 27.18 Operation Example of the Function of Trigger and Pulse Width Measurement ( $m = 0$  to  $2$ ,  $k = 0, 1$ )**

- (1) The following signals are output from ENCA0 as triggers:
  - ENCAT0IEC (interrupt trigger signal output when timer counter value is cleared by ENCA0EC input)
  - ENCA0TEQ0 (trigger signal output according to timing of a match of timer counter value and value of compare register 0)
  - ENCA0EQ1 (trigger signal output according to timing of a match of timer counter value and value of compare register 1)
- (2) The trigger signal output from ENCA0 is converted to a level-sensitive toggle signal by the DT circuit and is output to TINm of TAUD0 and TAUJ1.
- (3) By setting both TINm edges (rising and falling edges) of TAUD0 and TAUJ1 as valid, the CNTm value is captured into CDRm at the TINm toggle timing and cleared to 0000<sub>H</sub>. This operation is repeated.

The first captured value (shown as “a” in the figure) from the start of operation indicates the interval from the start of TAUJ operation until trigger input.

- (4) When an overflow occurs, the count value  $\text{FFFF}_H$  ( $\text{FFFF FFFF}_H$  for TAUJ) is captured but the count value is not captured at the first trigger after the overflow.

With the above operation, the trigger generation interval can be measured.

The following table shows the combinations of the trigger signals and measurement timers, and the bit settings of the pertinent PIC registers for setting the signal paths and the I/O selection registers.

Appropriately set these bits according to the trigger signal to be measured and the measurement timer to be used.

**Table 27.33 Combinations of Trigger Signals and Measurement Timers**

Interrupt Trigger Signal	Measurement Timer	PIC Register Bit Setting	
ENCAT0IEC	TAUJ1 CH0	PIC0REG3100 = 1	
	TAUJ1 CH1	PIC0REG3101 = 1	
ENCAT0EQ0	TAUD0 CH0	PIC0REG3109 = 0	PIC0REG3110 = 1
	TAUD0 CH2	PIC0REG3112 = 1	PIC0REG3113 = 0
ENCAT0EQ1	TAUD0 CH0	PIC0REG3106 = 1	PIC0REG3109 = 0 PIC0REG3110 = 0
	TAUD0 CH1	PIC0REG3108 = 0	PIC0REG3111 = 1

## 27.9.5 Setup Flow

The setup flow in this section shows the general setup flow to measure the pulse interval, which applies to all the following combinations. For the combinations of the trigger signals and measurement timers, see **Table 27.33, Combinations of Trigger Signals and Measurement Timers**.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	ENCAT0IEC	TAUJ1 CH0, TAUJ1 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1

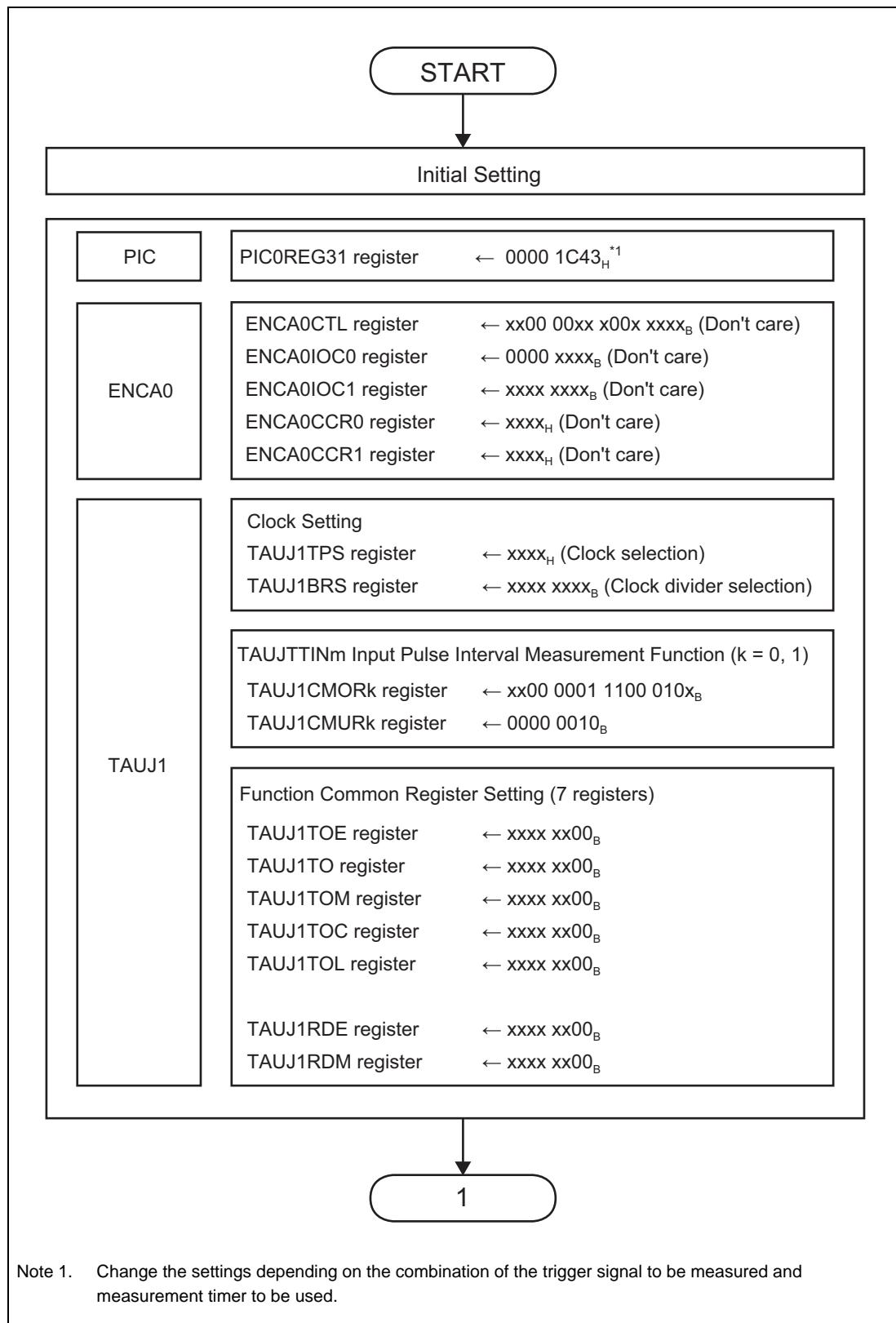
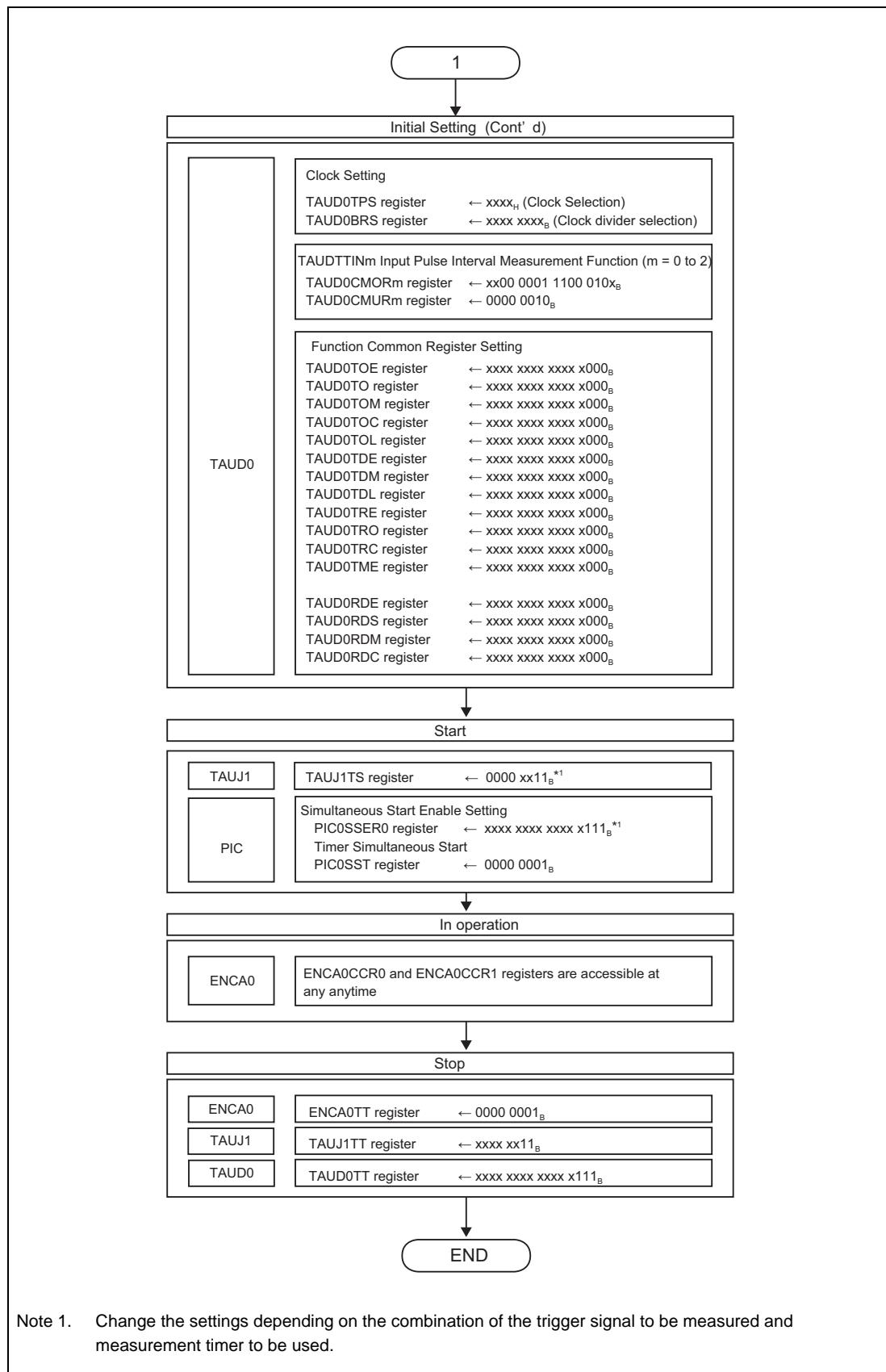


Figure 27.19 Setup Flow

**Figure 27.20 Setup Flow (Cont'd)**

### 27.9.6 Setting Examples for Operation Functions

This section provides example settings for each register.

The setup example shown in this section describes how to set up measurement of the pulse interval for all the combinations below. For the combinations of the trigger signals and measurement timers, see **Table 27.33, Combinations of Trigger Signals and Measurement Timers**.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	ENCAT0IEC	TAUJ1 CH0, TAUJ1 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1

Table 27.34 ENCA0 Setting

Register	Bit Position	Bit Name	Setting Value	Note
ENCA0CTL	15	ENCA0CME	Don't care	Enables or disables compare match interrupt detection mask
	14	ENCA0MCS	Don't care	Selects a cancelation trigger for compare match interrupt detection mask
	13 to 10		0	Fixed to 0
	9	ENCA0CRM1	Don't care	Selects the ENCA0CCR1 register function
	8	ENCA0CRM0	Don't care	Selects the ENCA0CCR0 register function
	7	ENCA0CTS	Don't care	Selects trigger for capture operation of ENCA0CCR1.
	6, 5		0	Fixed to 0
	4	ENCA0LDE	Don't care	Enables or disables reload operation when underflow is generated
	3	ENCA0ECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR1
	2	ENCA0ECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR0
ENCA0IOC0	1, 0	ENCA0UDS[1:0]	Don't care	Selects the counter up/down control by ENCA0E0 and ENCA0E1
	7 to 4		0	Fixed to 0
	3, 2	ENCA0TIS[3:2]	Don't care	Selects the valid edge for capture trigger 1 (ENCA0I1)
ENCA0IOC1	1, 0	ENCA0TIS[1:0]	Don't care	Selects the valid edge for capture trigger 0 (ENCA0I0)
	7	ENCA0SCE	Don't care	Enables encoder special-clear
	6	ENCA0ZCL	Don't care	Selects the clear level of Z phase for a encoder special-clear
	5	ENCA0BCL	Don't care	Selects the clear level of B phase for a encoder special-clear
	4	ENCA0ACL	Don't care	Selects the clear level of A phase for a encoder special-clear
	3, 2	ENCA0ECS[1:0]	Don't care	Selects encoder clear input (Z phase) edge
	1, 0	ENCA0EIS[1:0]	Don't care	Selects encoder input (A or B phase) edge

**Table 27.35 TAUJ1 Setting (k = 0, 1)  
TAUJ1 (TAUDTTINm Input Pulse Interval Measurement Function)**

Register	Bit Position	Bit Name	Setting Value	Note
TAUJ1CMORk	15,14	TAUJ1CKS[1:0]	Don't care	Operation Clock Setting
	13,12	TAUJ1CCS[1:0]	00	
	11	TAUJ1MAS	0	
	10, 9, 8	TAUJ1STS[2:0]	001	
	7, 6	TAUJ1COS[1:0]	11	
	5		0	Fixed to 0
	4, 3, 2, 1	TAUJ1MD[4:1]	0010	
	0	TAUJ1MD0	Don't care	
	1, 0	TAUJ1TIS[1:0]	10	

#### NOTE

When TAUJ1CMORk is used for the TAUDTTINm input pulse interval measurement function, the TAUJ1CKS[1:0] (operating clock selection) and TAUJ1MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Although the TAUJ1COS[1:0] (overflow mode selection) bits can also be set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 24, Timer Array Unit J (TAUJ)**.

For TAUJ common registers (TAUJ1TOE, TAUJ1TO, TAUJ1TOM, TAUJ1TOC, TAUJ1TOL, TAUJ1RDE, and TAUJ1RDM), only set the bits corresponding to the used channels to 0.

**Table 27.36 TAUD0 Setting (m = 0 to 2)  
TAUD0 (TAUDTTINm Input Pulse Interval Measurement Function)**

Register	Bit Position	Bit Name	Setting Value	Note
TAUD0CMORm	15, 14	TAUD0CKS[1:0]	Don't care	Operation Clock Setting
	13, 12	TAUD0CCS[1:0]	00	
	11	TAUD0MAS	0	
	10 to 8	TAUD0STS[2:0]	001	
	7, 6	TAUD0COS[1:0]	11	
	5		0	Fixed to 0
	4 to 1	TAUD0MD[4:1]	0010	
	0	TAUD0MD0	Don't care	
TAUD0CMURm	1, 0	TAUD0TIS[1:0]	10	

#### NOTE

When TAUD0CMORm is used for the TAUDTTINm input pulse interval measurement function, the TAUD0CKS[1:0] (operating clock selection) and TAUD0MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Although the TAUD0COS[1:0] (overflow mode selection) bits can also be set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 23, Timer Array Unit D (TAUD)**

For TAUD common registers (TAUD0TOE, TAUD0TO, TAUD0TOM, TAUD0TOC, TAUD0TOL, TAUD0TDE, TAUD0TDM, TAUD0TDL, TAUD0TRE, TAUD0TRO, TAUD0TRC, TAUD0TME, TAUD0RDE, TAUD0RDS, TAUD0RDM, and TAUD0RDC), only set the bits corresponding to the used channels to 0.

**Table 27.37 PIC Setting**

Register	Bit Position	Bit Name	Setting Value	Note
PIC0REG31	13, 12	PIC0REG3113	0	Selects the DT output signal from ENC0EQ0 as TAUD0TTIN2 input signal
		PIC0REG3112	1	
	11	PIC0REG3111	1	Selects the signal selected with PIC0REG3106 to PIC0REG3108 (DT output signal from ENC0EQ1) as TAUD0TTIN1 input signal
	10, 9	PIC0REG3110	1	Selects the DT output signal from ENC0EQ0 as TAUD0TTIN0 input signal
		PIC0REG3109	0	
	8 to 6	PIC0REG3108	0	Selects the DT output signal from ENC0EQ1 as TAUD0TTIN1 or TAUD0TTIN0 input signal
		PIC0REG3107	0	
		PIC0REG3106	1	
1	PIC0REG3101		1	Selects the DT output signal from ENC0IEC as TAUJ1TTIN1 input signal
0	PIC0REG3100		1	Selects the DT output signal from ENC0IEC as TAUJ1TTIN0 input signal

## 27.10 A/D Trigger Encoder Capture Function

### 27.10.1 Functional Overview

The value of the encoder counter synchronized with A/D conversion can be obtained by using an A/D conversion trigger signal as a capture signal of ENCA0.

### 27.10.2 Configuration

#### (1) Configuration

Table 27.38 Configuration of A/D Trigger Encoder Capture Function

A/D Converter	Encoder Timer
ADCA0	ENCA0

#### (2) Block Diagram

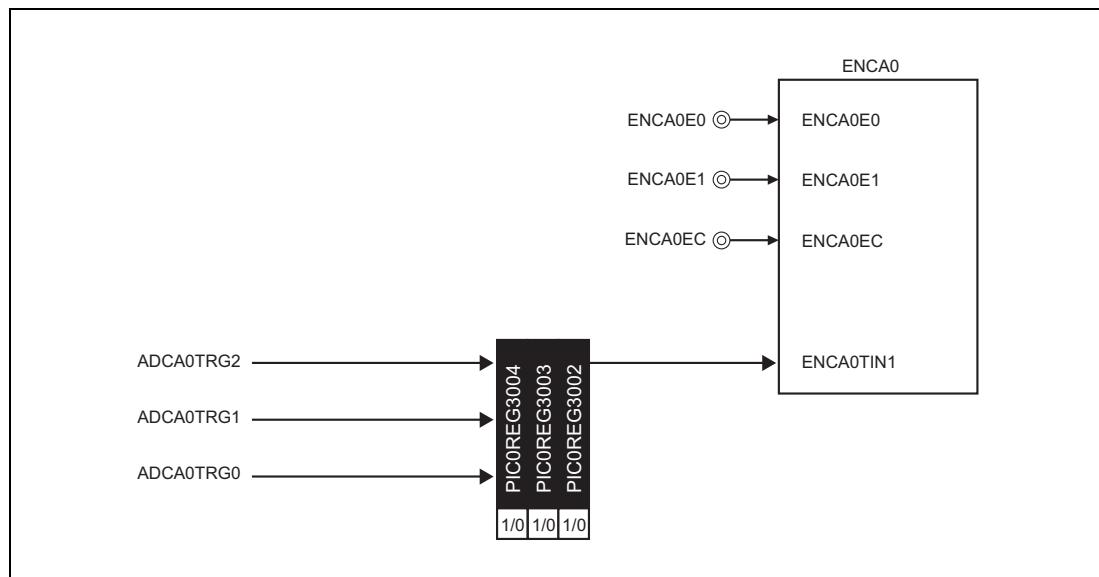


Figure 27.21 Block Diagram of A/D Trigger Encoder Capture Function

#### CAUTIONS

1. It takes ENCA0 one additional clock cycle to capture the input signal of the ADCA0TRG0, ADCA0TRG1, and ADCA0TRG2 pins compared with the number of clock cycles it takes to capture the input signal of the ENCA0TIN1 pin when using CKSCLK\_IPERI1, and three additional clock cycles when using CKSCLK\_AADCA. Be sure to take this into account when configuring your system.
2. Configure the edge detection function by using the edge detection function registers of the digital noise filter, which are FCLA0CTL0\_ADC0, FCLA0CTL1\_ADC0, and FCLA0CTL2\_ADC0 (see Section 2.12.1.4, Input Pins that Incorporate Digital Filter Type D, for details), and specify “rising edge” for edge detection of the ENCA0TIN1 capture trigger input of ENCA0 (ENCA0IOC0.ENCA0TIS [3:2] = 01<sub>B</sub>). Do not set ENCA0IOC0.ENCA0TIS [3:2] to 10<sub>B</sub> (falling edge) or 11<sub>B</sub> (both edges).

### 27.10.3 Registers

#### 27.10.3.1 PIC0REG30 — Timer I/O Control Register 30

**Access:** This register can be read or written in 32-bit units.

**Address:** <PIC0\_base> + E8H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 27.39 PIC0REG30 Register Contents

Bit Position	Bit Name	Function																								
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																								
4 to 2	PIC0REG3004 to PIC0REG3002	Selects an input signal to ENCA0TIN1.																								
		<table border="1"> <thead> <tr> <th>PIC0REG 3004</th> <th>PIC0REG 3003</th> <th>PIC0REG 3002</th> <th>Input signal</th> </tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>Capture is not performed by an A/D trigger signal in ENCA0.</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>ADCA0TRG2</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>ADCA0TRG1</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>ADCA0TRG0</td></tr> <tr> <td colspan="3">Other than above</td><td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG 3004	PIC0REG 3003	PIC0REG 3002	Input signal	0	0	0	Capture is not performed by an A/D trigger signal in ENCA0.	0	1	0	ADCA0TRG2	0	1	1	ADCA0TRG1	1	0	0	ADCA0TRG0	Other than above			Setting prohibited
PIC0REG 3004	PIC0REG 3003	PIC0REG 3002	Input signal																							
0	0	0	Capture is not performed by an A/D trigger signal in ENCA0.																							
0	1	0	ADCA0TRG2																							
0	1	1	ADCA0TRG1																							
1	0	0	ADCA0TRG0																							
Other than above			Setting prohibited																							
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.																								

#### 27.10.4 Example of Operation

The A/D trigger encoder capture function is implemented by connecting A/D conversion trigger signal ADCAnTRGi ( $n = 0, i = 0$  to 2) to ENCA0.

##### CAUTION

**When using this function, the ENCA0 interrupt signal ENCA0INT1 should not be selected as the A/D converter trigger. If selected, the correct operation cannot be performed because the following loop occurs: ADCAnTRG1 generation → ENCA0 capture operation → ENCA0INT1 generation by capture operation → ADCAnTRG1 generation.**

The following shows a timing chart of the A/D trigger encoder capture function using the ADCA0TRG1 as a trigger.

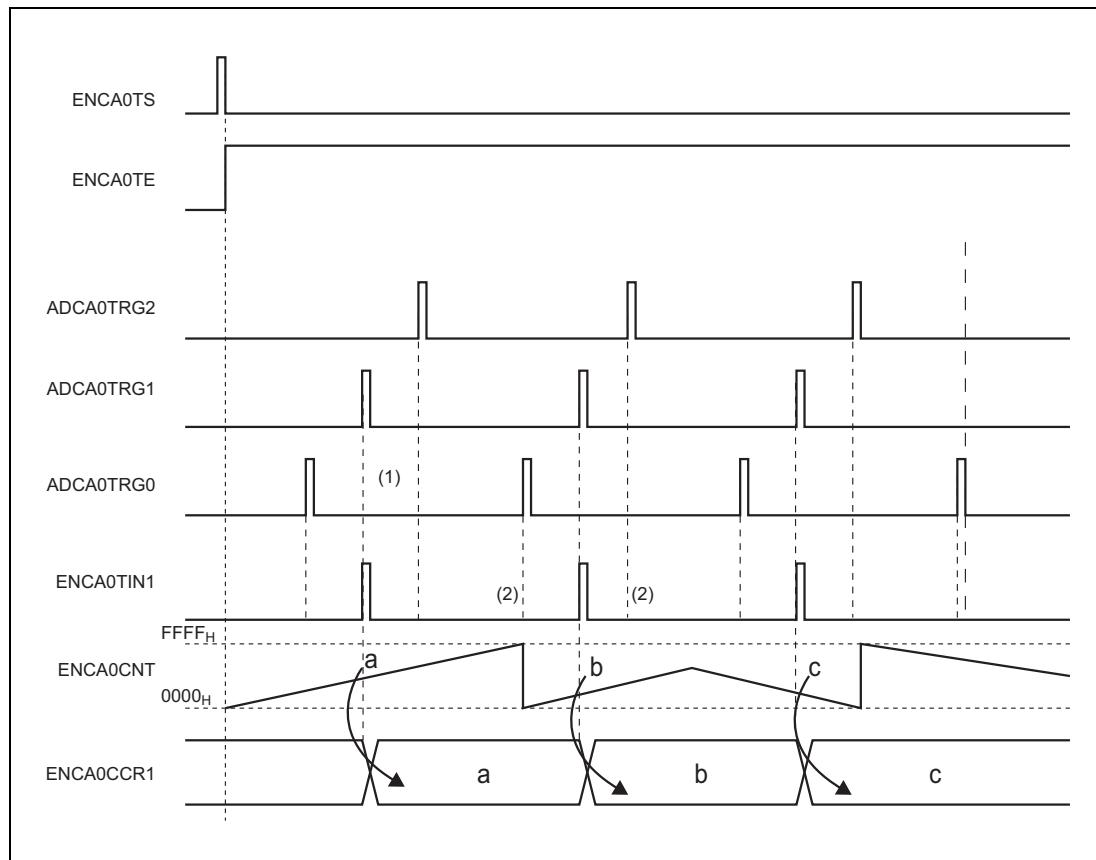


Figure 27.22 Operation Example of Trigger Encoder Capture function

- (1) When ADCA0TRG1 is selected as ENCA0 capture trigger 1 signal ENCA0TIN1, the valid ADCA0TRG1 is input to ENCA0 as the ENCA0TIN1 signal and ENCA0 is captured.
- (2) When a hardware trigger signal (ADCA0TRG0, ADCA0TRG2) other than ADCA0TRG1 is generated, the ENCA0TIN1 signal is not generated and ENCA0 is not captured.

### 27.10.5 Setup Flow

The setup flow in this section shows the general setup flow to perform the capture operation of encoder timer ENCA0 based on the ADCA0TRG1 signal.

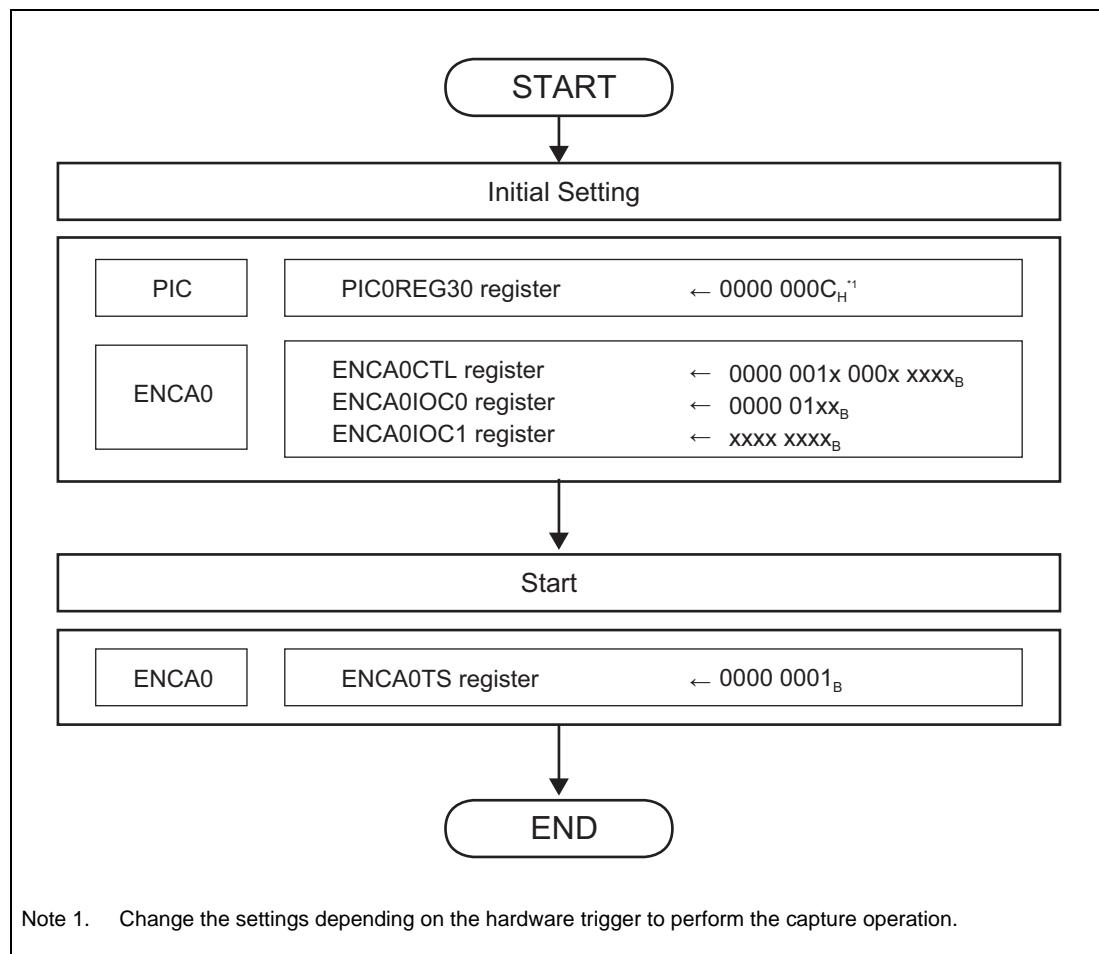


Figure 27.23 Setup Flow

### 27.10.6 Setting Examples for Operation Functions

This section provides example settings for each register.

The setup example shown in this section describes how to set up capture operation of encoder timer ENCA0 based on the ADCA0TRG1 signal. Change the settings depending on the hardware trigger to perform the capture operation.

**Table 27.40 ENCA<sub>n</sub> Setting**

Register	Bit Position	Bit Name	Setting Value	Remark
ENCA <sub>n</sub> CTL	15	ENCA <sub>n</sub> CME	0	Disables compare match interrupt detection masking
	14	ENCA <sub>n</sub> MCS	0	Selects release trigger for compare match interrupt detection masking
	13 to 10		0	Fixed to 0
	9	ENCA <sub>n</sub> CRM1	1	Sets the ENCA <sub>n</sub> CCR1 register for capture operation
	8	ENCA <sub>n</sub> CRM0	Don't care	Selects the function of ENCA <sub>n</sub> CCR0 register
	7	ENCA <sub>n</sub> CTS	0	Selects ENCATTIN1 as trigger for capture operation
	6, 5		0	Fixed to 0
	4	ENCA <sub>n</sub> LDE	Don't care	Enables or disables reload operation when ENCA <sub>n</sub> CCR0 register underflow occurs
	3	ENCA <sub>n</sub> ECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCA <sub>n</sub> CCR1 register
	2	ENCA <sub>n</sub> ECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCA <sub>n</sub> CCR0 register
ENCA <sub>n</sub> IOC0	1, 0	ENCA <sub>n</sub> UDS[1:0]	Don't care	Select the counter up/down control by ENCA <sub>n</sub> E0 and ENCA <sub>n</sub> E1
	7 to 4		0	Fixed to 0
	3, 2	ENCA <sub>n</sub> TIS[3:2]	0 <sup>*1</sup> 1 <sup>*1</sup>	Select the valid edge of capture trigger 1 (ENCATTIN1) for the rising edge detection
ENCA <sub>n</sub> IOC1	1, 0	ENCA <sub>n</sub> TIS[1:0]	Don't care	Select the valid edge of capture trigger 0 (ENCATTIN0)
	7	ENCA <sub>n</sub> SCE	Don't care	Enables encoder special-clear
	6	ENCA <sub>n</sub> ZCL	Don't care	Selects the clear level (input level) of Z phase for encoder special-clear
	5	ENCA <sub>n</sub> BCL	Don't care	Selects the clear level (input level) of B phase for encoder special-clear
	4	ENCA <sub>n</sub> ACL	Don't care	Selects the clear level (input level) of A phase for encoder special-clear
	3, 2	ENCA <sub>n</sub> ECS[1:0]	Don't care	Select encoder clear input (Z phase) edge
	1, 0	ENCA <sub>n</sub> EIS[1:0]	Don't care	Select encoder input (A or B phase) edge
PIC0REG30	4	PIC0REG3004	Don't care	Selects ADCA0 trigger signal of ENCA0TIN1
	3	PIC0REG3003	Don't care	
	2	PIC0REG3002	Don't care	

Note 1. Change the setting depending on the hardware trigger to perform the capture operation.

#### NOTE

Bits ENCA0CRM1 and ENCA0CTS in ENCA0CTL are fixed: ENCA0CRM1 = 1 (ENCA0CCR1 register function) and ENCA0CTS = 0 (trigger source of capture to the ENCA0CCR1 register). All the other bits can be set arbitrarily.

## 27.11 Three-Phase PWM Output with Dead Time

### 27.11.1 Functional Overview

This feature generates each of the set signals (assert timing signals) and clear signals (deassert timing signals) once or less per cycle and then uses the results to output a three-phase PWM waveform with dead time.

For the PWM output feature of TAUD, only the clear timing used during each cycle is specified by specifying the duty value, but for the feature described here, the set timing can also be specified, which makes more flexible PWM output with dead time possible.

### 27.11.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0)

**Table 27.41 Configuration of Three-Phase PWM Output with Dead Time**

Timer	Timer Motor Control Function
TAUD0 CH2, CH4 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

#### (1) TAUDn configuration

Because CH10, CH12, and CH14 are only used for TOOUTm, these channels can be used for features that do not use TOOUTm (m = 10, 12, 14).

**Table 27.42 TAUDn Configuration (1/2)**

CH	Function Name	M/S	CDR Setting	Description
2	PWM output (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U phase signal setting)	
5		S	Duty (U phase signal clearing)	
6		S	Duty (V phase signal setting)	
7		S	Duty (V phase signal clearing)	
8		S	Duty (W phase signal setting)	
9		S	Duty (W phase signal clearing)	
10	Any feature that does not use TOOUT10	S		TOOUT10: U phase output
11	One-phase PWM output	S	Dead time (U phase)	TOOUT11: UB phase output
12	Any feature that does not use TOOUT12	S		TOOUT12: V phase output

Table 27.42 TAUDn Configuration (2/2)

CH	Function Name	M/S	CDR Setting	Description
13	One-phase PWM output	S	Dead time (V phase)	TOUT13: VB phase output
14	Any feature that does not use TOUT14	S		TOUT14: W phase output
15	One-phase PWM output	S	Dead time (W phase)	TOUT15: WB phase output

Note: M: Master channel, S: Slave channel

## (2) Block diagram

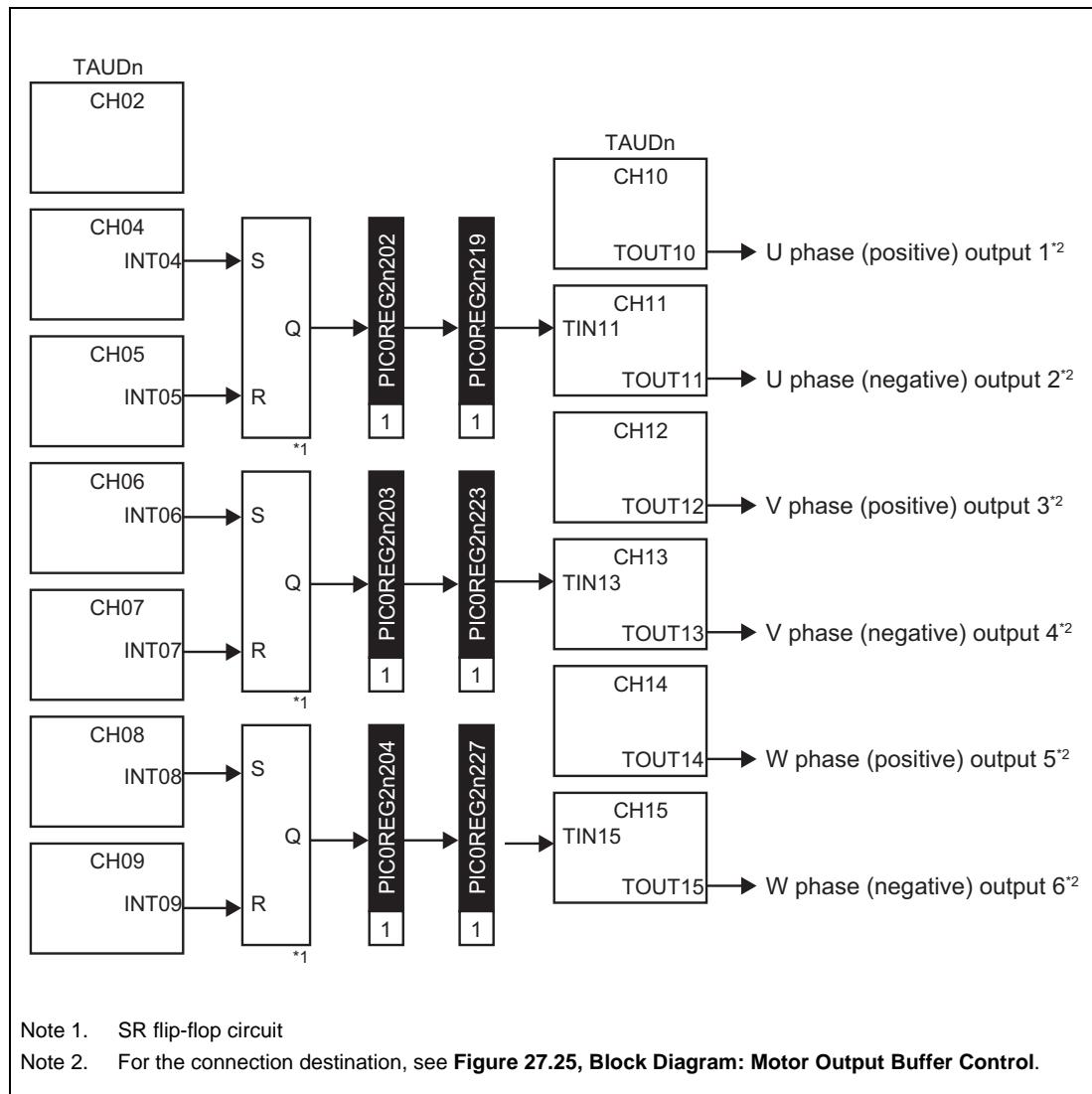


Figure 27.24 Block Diagram: Three-Phase PWM Output with Dead Time

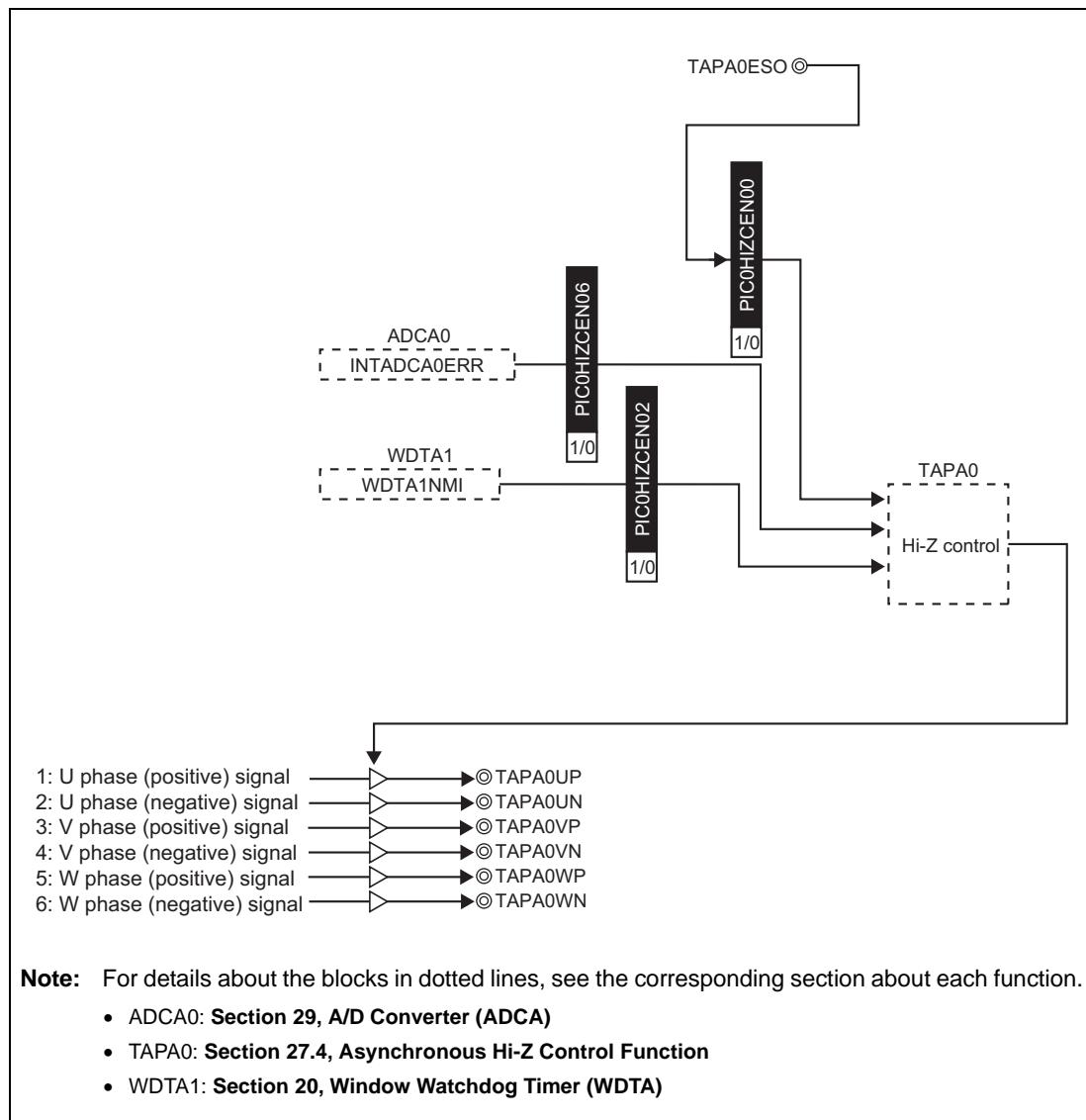


Figure 27.25 Block Diagram: Motor Output Buffer Control

### 27.11.3 Registers

#### 27.11.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

**Access:** This register can be read or written in 32-bit units.

**Address:** PIC0REG202: FFDD 00C8<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0RE G2n227	—	—	—	PIC0RE G2n223	—	—	—	PIC0RE G2n219	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0RE G2n204	PIC0RE G2n203	PIC0RE G2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 27.43 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27	PIC0REG2n227	Selects the signal input to TAUDTTIN15.
	PIC0REG2n227	Input signal
	1	Signal selected by the PIC0REG2n204 bit.
	Other than above	Setting prohibited
26 to 24	Reserved	*1
23	PIC0REG2n223	Selects the signal input to TAUDTTIN13.
	PIC0REG2n223	Input signal
	1	Signal selected by the PIC0REG2n203 bit.
	Other than above	Setting prohibited
22 to 20	Reserved	*1
19	PIC0REG2n219	Selects the signal input to TAUDTTIN11.
	PIC0REG2n219	Input signal
	1	Signal selected by the PIC0REG2n202 bit
	Other than above	Setting prohibited
18 to 5	Reserved	*1
4	PIC0REG2n204	Selects the TIN input signal to TAUDTTIN15. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDn18 and INTTAUDn19.
3	PIC0REG2n203	Selects the TIN input signal to TAUDTTIN13. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDn16 and INTTAUDn17.
2	PIC0REG2n202	Selects the TIN input signal to TAUDTTIN11. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDn14 and INTTAUDn15.
1, 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

### 27.11.3.2 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

The PIC0HIZCENn register selects the Hi-Z output control input signal of TAPAn.

**Access:** This register can be read or written in 8-bit units.

**Address:** <PIC0\_base> + 80H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 27.44 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Selects whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Selects whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

## 27.11.4 Operation Example

This example shows how to generate each of the set signals and clear signals once or less per cycle and then use the results to output a three-phase PWM waveform with dead time.

This is achieved by combining the following TAUD features:

- PWM output
- One-phase PWM output

In addition, the following peripheral interconnections are used to create the PWM waveform supplied from the set and clear signals generated during PWM output to the input TINm signal ( $m = 11, 13$ , or  $15$ ) of one-phase PWM output:

- SR flip-flop circuit

Three-phase PWM output is achieved by assigning the one-phase PWM output with dead time achieved using the above features to the U, V, and W phases. Therefore, the set and clear signals of PWM output can be freely specified for each PWM phase. Because the only difference among phases is the assigned channel, only one phase (the U phase) is described below.

### 27.11.4.1 Pwm Output

PWM output uses a combination of CH2, CH4, and CH5.

By specifying the cycle for CDR02, the U phase set value for CDR04, and the U phase clear value for CDR05, a set/clear signal is generated for the SR flip-flop circuit that generates the input TIN11 signal of one-phase PWM output from INT04 and INT05.

Instead of CH4 and CH5, which are used for the above described U phase set/clear signal generation, the V phase uses CH6 and CH7, and the W phase uses CH8 and CH9.

### 27.11.4.2 One-Phase PWM Output

One-phase PWM output is generated from TOUT10 and TOUT11 by using a combination of CH10 and CH11.

By specifying the dead time value for CDR11, a one-phase PWM signal with dead time is output for the TIN11 input.

Similarly, the V phase uses CH12 and CH13 to output a one-phase PWM signal with dead time, while the W phase uses CH14 and CH15.

#### CAUTION

**Specify the same clock for each TAUDn channel that uses the PWM output and one-phase PWM output features.**

For details about the TAUD functions, see **Section 23, Timer Array Unit D (TAUD)**.

### 27.11.4.3 SR Flip-Flop Circuit

The PWM waveform supplied to input TIN11 of one-phase PWM output is generated by using the U phase set signal generated by CH4 of TAUD and the U phase clear signal generated by CH5.

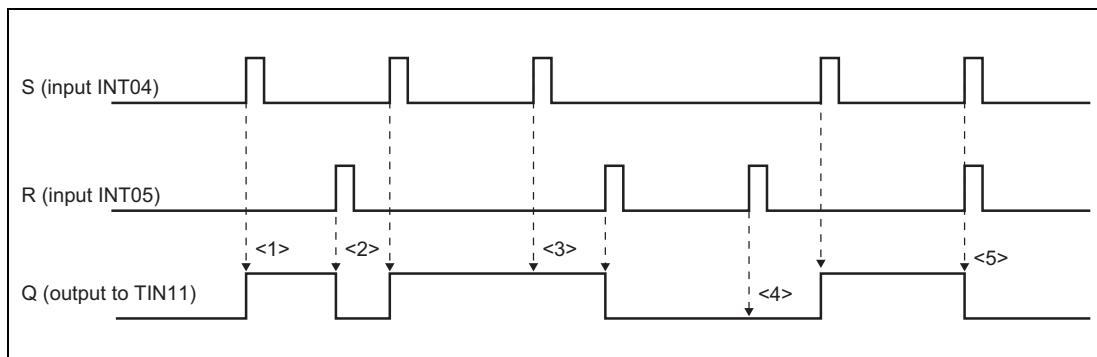


Figure 27.26 SR Flip-Flop Circuit Operation Timing Chart (U phase example)

- (1) When a signal is input to input S, output Q goes to the high level at the rising edge of S.
- (2) When a signal is input to input R, output Q goes to the low level at the rising edge of R.
- (3) If a signal is input to input S while output Q is at the high level, output Q is not affected.
- (4) If a signal is input to input R while output Q is at the low level, output Q is not affected.
- (5) If a signal is input to input S and input R at the same time, input R is prioritized and output Q goes to the low level at the rising edge of R.

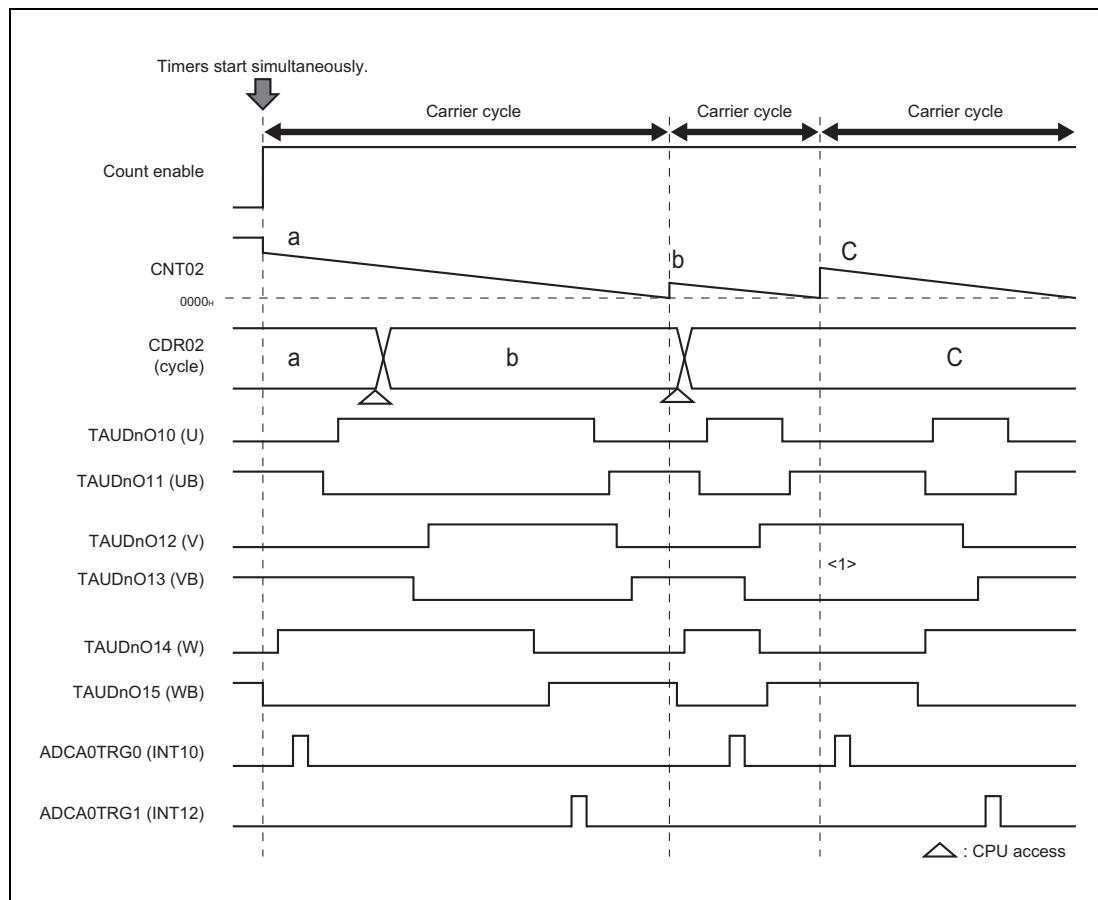
The V phase uses INT06 and INT07 as input to supply a PWM waveform to TIN13, and the W phase uses INT08 and INT09 as input to supply a PWM waveform to TIN15.

The output change timing of the PWM waveform generated during one-phase PWM output is based on PWM output.

The active level output timing set signal and inactive level output timing clear signal of PWM are generated during PWM output. By inputting these signals to the SR flip-flop circuit, a PWM signal that can be changed at any time is generated.

A one-phase PWM signal is output by generating a positive or negative PWM waveform and then adding dead time to it according to changes in the generated PWM signal.

PIC is used to set/clear signal generated during PWM output as the TIN input for one-phase PWM output through the SR flip-flop circuit.



**Figure 27.27 Example of Three-Phase PWM (U/UB, V/VB, W/WB) Output with Dead Time**

**Figure 27.27** shows a typical example of three-phase PWM output with dead time.

By appropriately setting up the set/clear signal output timing, PWM output that extends across carrier cycles (point <1>) and other types of output are also possible.

In this example, ADCA0TRG0 and ADCA0TRG1 (which are at the bottom) use the CNT and INT signals of CH10 and CH12, which are not used for one-phase PWM output, and the A/D trigger signal is output by performing type-1 A/D trigger output.

In this way, because only the TOUTm signal that performs signal output for the channel performing positive phase output is used during one-phase PWM output, any feature that uses CNTm, CDRm, or INTm can be specified. For details, see **Section 23, Timer Array Unit D (TAUD)** ( $m = 10, 12$ , or  $14$ ).

The following figures show timing charts for outputting a three-phase PWM signal with dead time.

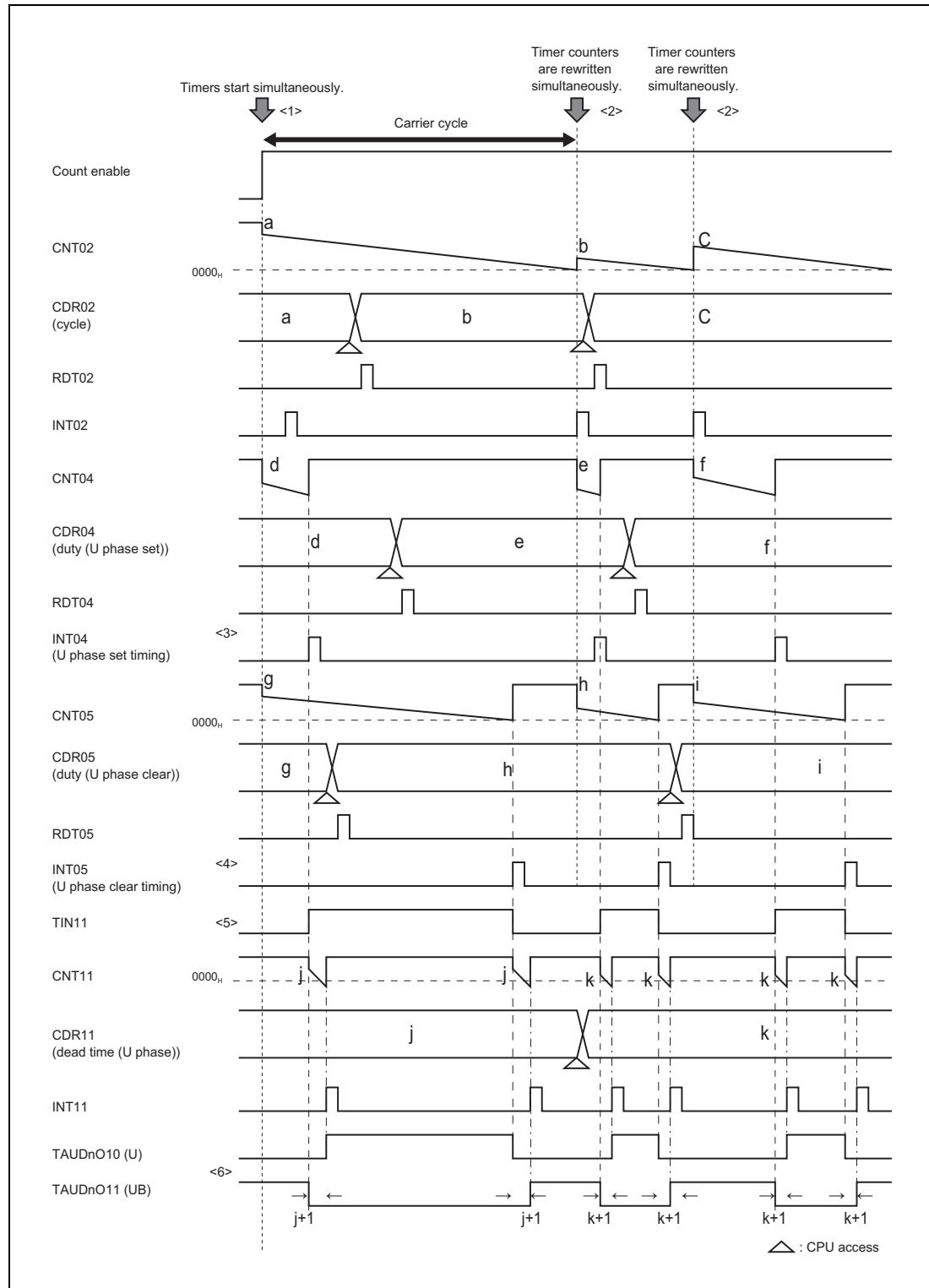
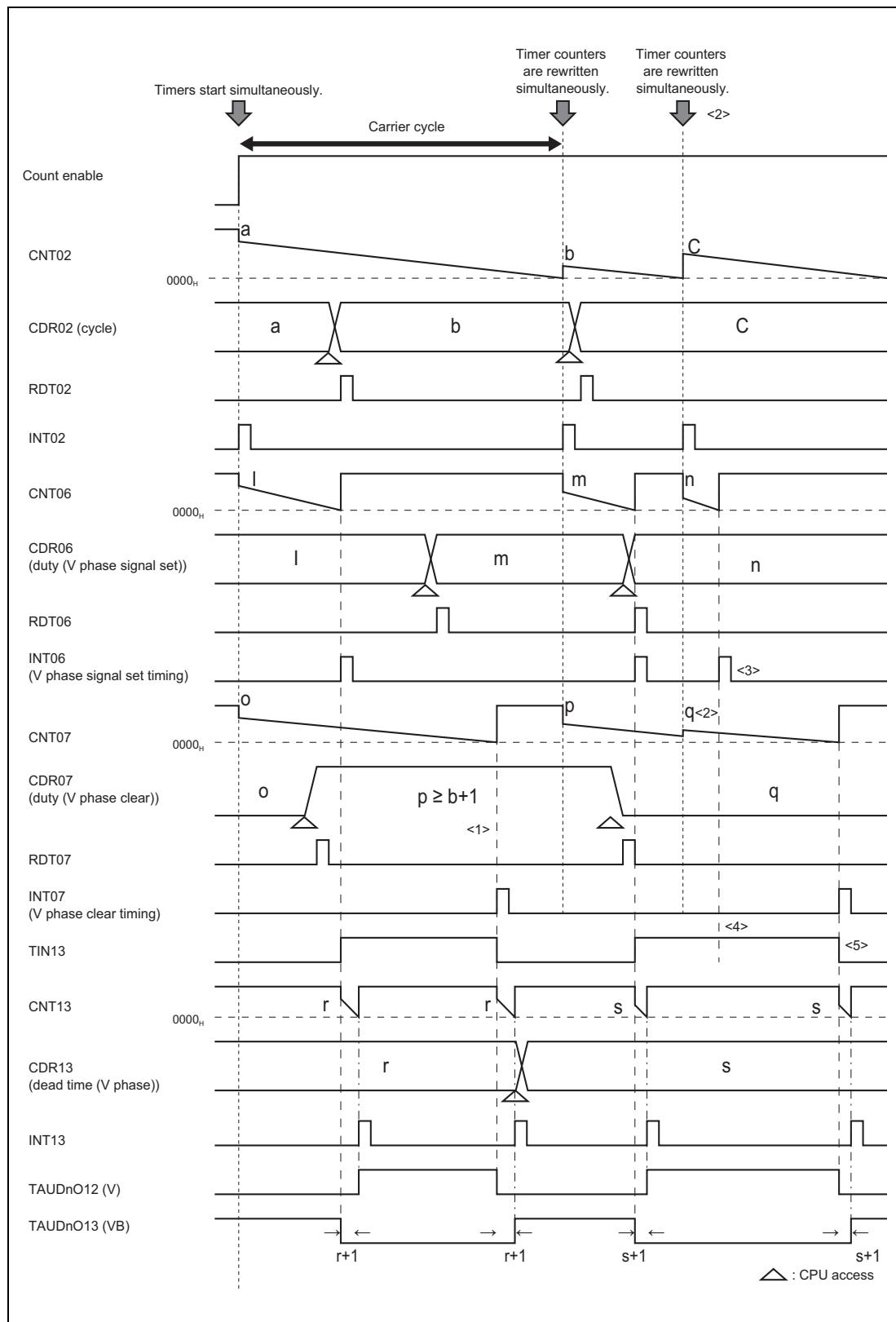


Figure 27.28 Example of One-Phase PWM (U phase, UB phase) Output with Dead Time

An operation example of the timer configuration for performing the U phase PWM output in **Figure 27.28** is provided below.

- (1) By simultaneously starting timers, CH2 (the carrier cycle timer), CH4 (the U phase set signal output timing timer), and CH5 (the U phase clear signal output timing timer) are started simultaneously.  
The CH11 timer is also enabled, but until a TIN11 edge is detected, which is the count start timing, counting is not performed.
- (2) For CH4 and CH5, when there is a CH2 underflow, the settings from CDR04 and CDR05 are reloaded to CNT04 and CNT05.
- (3) When there is a CH4 underflow, the U phase set timing signal (INT04) is generated.
- (4) When there is a CH5 underflow, the U phase clear timing signal (INT05) is generated.
- (5) The peripheral interconnections supply the output of the SR flip-flop circuit that uses INT04 (the set timing signal) and INT05 (the clear timing signal) as input to the input TIN11 signal of one-phase PWM output.
- (6) During one-phase PWM output, a PWM waveform with dead time is output by detecting a TIN11 edge.



**Figure 27.29 Example of One-Phase PWM (V phase, VB phase) Output with Dead Time**

An operation example of the timer configuration for performing the V phase PWM output in **Figure 27.29** is provided below.

For details about the operations from when timers are simultaneously started until a one-phase PWM signal is output, see the U phase operation example.

- (1) If the setting of CH7 (the V phase clear signal output timing timer), which generates the V phase clear timing signal (INT07), is greater than the CH2 (the carrier cycle timer) setting.
- (2) Before a V phase clear timing signal (INT07) is generated by a CH7 underflow, a CH2 (carrier cycle timer) underflow occurs, and the CH7 setting is reloaded.
- (3) It causes the V phase clear timing signal (INT07) not to be generated, resulting in consecutive generation of the V phase set timing signal (INT06).
- (4) In this case, because the V phase set timing signal (INT06) is ignored by the SR flip-flop circuit, there is no effect on the PWM output waveform. Therefore, a PWM waveform that extends across carrier cycles is output.
- (5) The PWM output is changed at the timing of the next V phase clear timing signal (INT07).

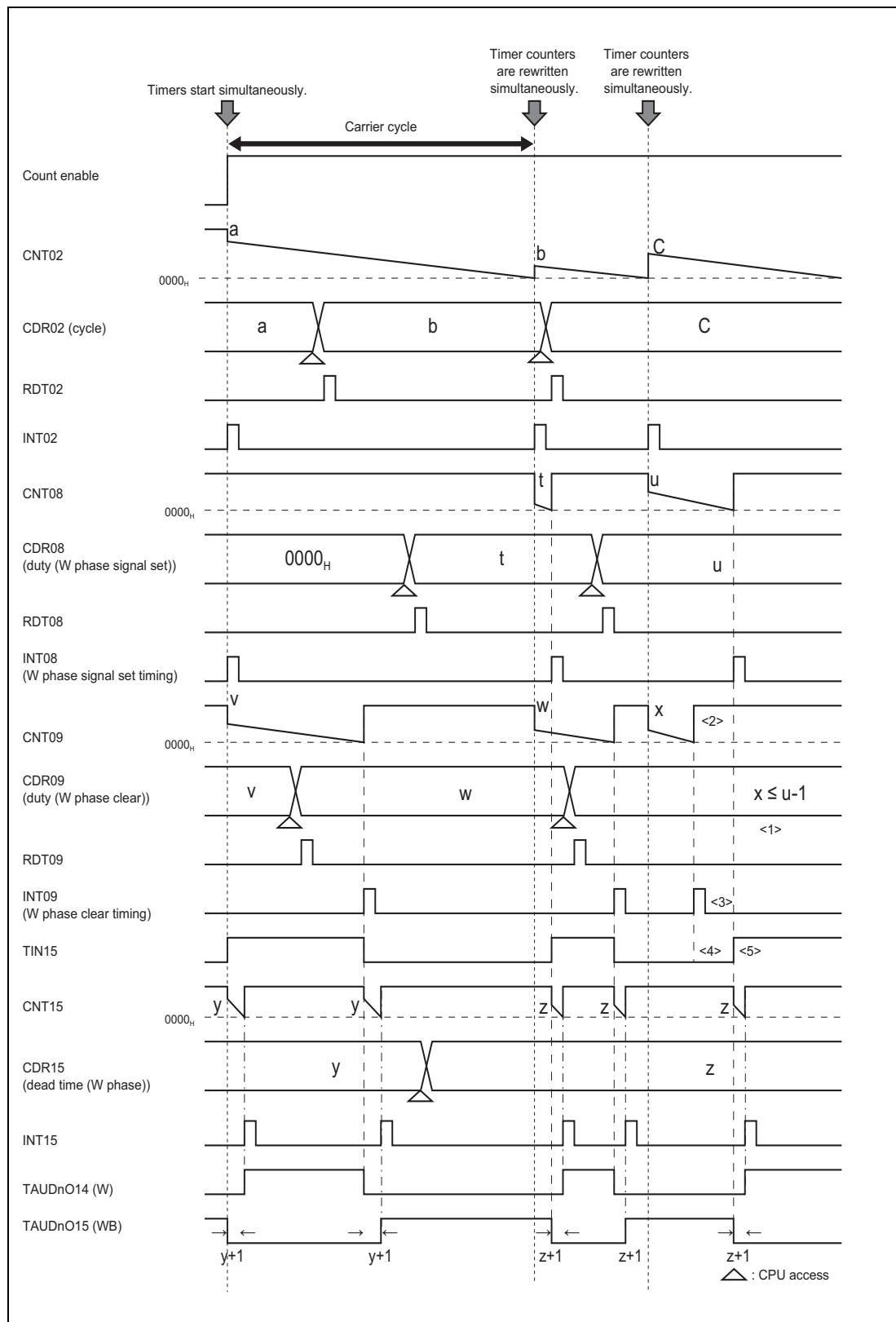


Figure 27.30 Example of One-Phase PWM (W phase, WB phase) Output with Dead Time

An operation example of the timer configuration for performing the W phase PWM output in **Figure 27.30** is provided below.

For details about the operations from when timers are simultaneously started until a one-phase PWM signal is output, see the U phase operation example.

- (1) If the setting of CH9 (the W phase clear signal output timing timer), which generates the W phase clear timing signal (INT09), is less than the CH8 (the W phase set signal output timing timer) setting.
- (2) Before a W phase set timing signal (INT08) is generated by a CH8 underflow, a CH9 (W phase clear signal output timing timer) underflow occurs, and the W phase clear timing signal (INT09) is generated.
- (3) This results in consecutive W phase clear timing signals (INT09) being generated.
- (4) In this case, because the consecutively generated W phase clear timing signals (INT09) are ignored by the SR flip-flop circuit, there is no effect on the PWM output waveform.
- (5) The PWM output is changed at the timing of the next W phase set timing signal (INT08).

### 27.11.5 Setup Flow

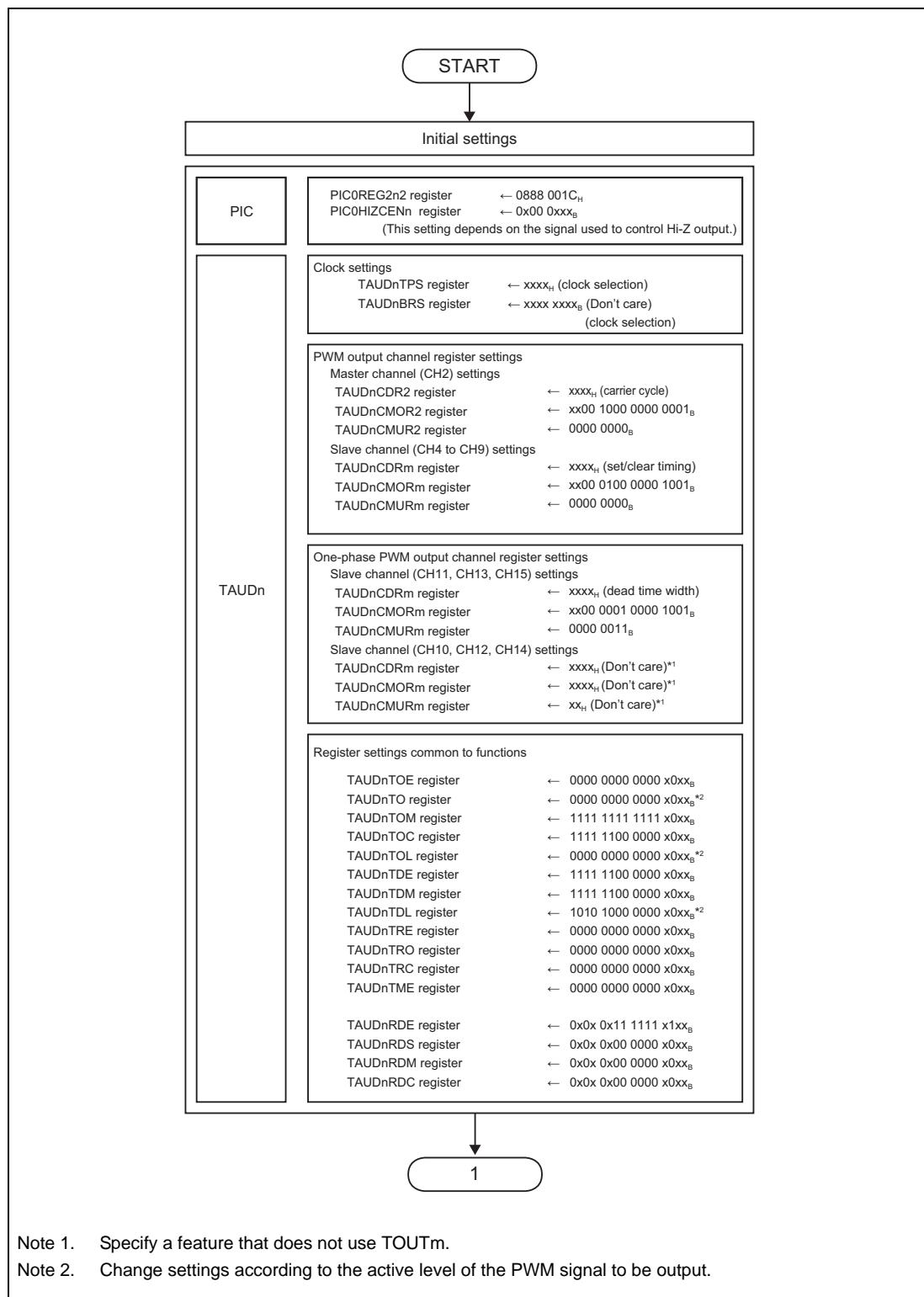


Figure 27.31 Setup Flow (Active High Example)

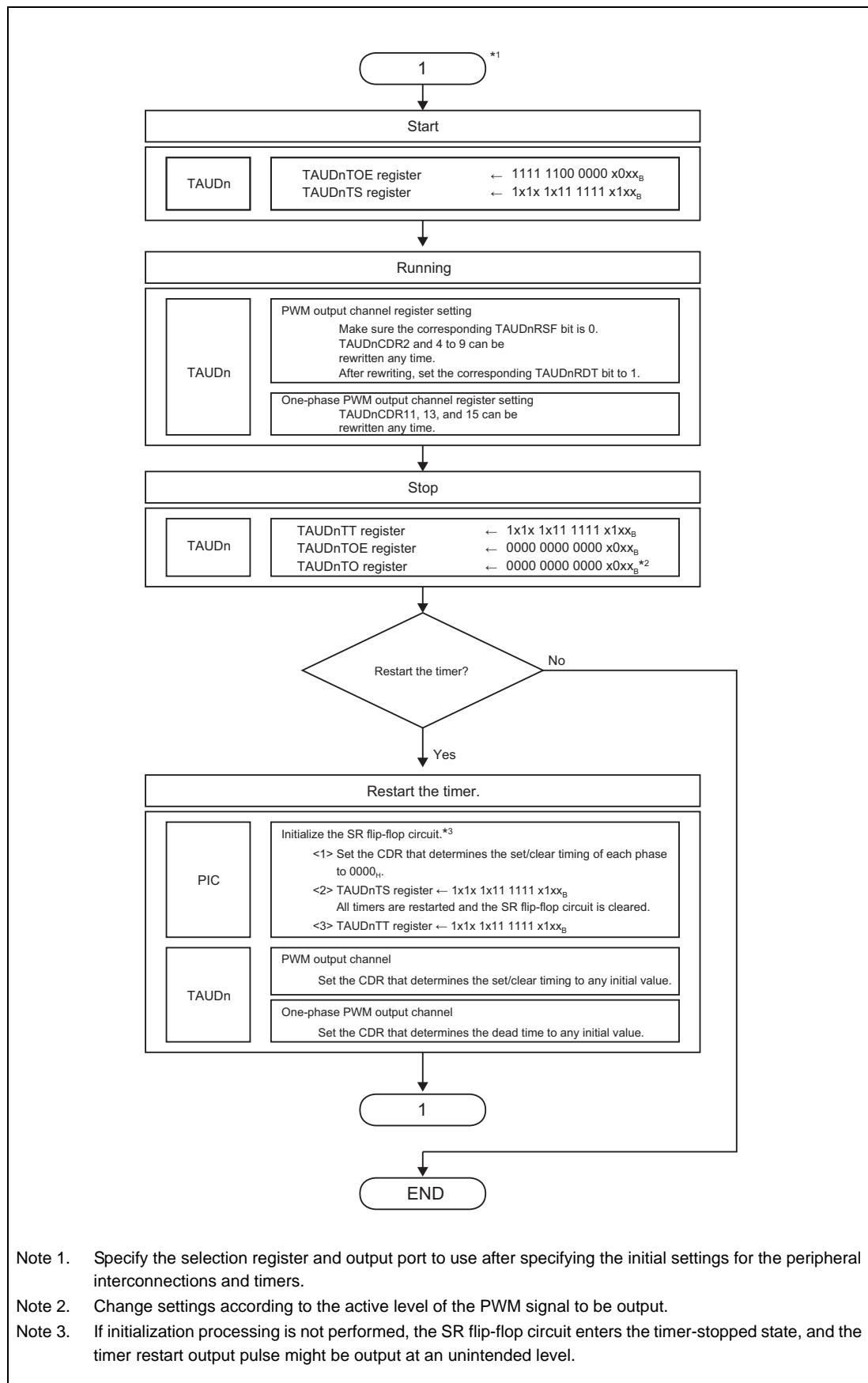


Figure 27.32 Setup Flow (Active High Example) (continued)

## 27.11.6 Setting Examples for Operation Functions

This section provides example settings for each register.

### 27.11.6.1 TAUDn Settings (Active High Example)

Table 27.45 TAUDn: CH2-related (PWM Output Master Channel<sup>\*1</sup>)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 27.46 TAUDn: CH4 to CH9-related (PWM Output Slave Channel<sup>\*1</sup>) ( $m = 4$  to  $9$ )

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Any <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

#### NOTE

For the TAUDnCMORm register used during PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

**Table 27.47 TAUDn: CH11, CH13, and CH15-related (One-phase PWM Output) ( $m = 11, 13$ , or  $15$ )**

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*1</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

#### NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output). For details, see **Section 23, Timer Array Unit D (TAUD)**.

**Table 27.48 Common TAUDn Channel Settings (1/4)**

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0	These are fixed to 0 because TOUT09 to TOUT04 are not used.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	
	15 to 10	TAUDnTO15 to TAUDnTO10	0 <sup>*1</sup>	Output a low-level signal to TOUT15 to TOUT10.
	9 to 4	TAUDnTO09 to TAUDnTO04	0	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't care	
TAUDnTO	2	TAUDnTO02	0	Output a low-level signal to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	
	15 to 4	TAUDnTOM15 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't care	
	2	TAUDnTOM02	0	Independent operation mode
TAUDnTOM	1, 0	TAUDnTOM01 TAUDnTOM00	Don't care	

Table 27.48 Common TAUDn Channel Settings (2/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	0	Synchronous operation mode 1
	3	TAUDnTOC03	Don't care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don't care	
TAUDnTOL	15 to 4	TAUDnTOL15 to TAUDnTOL04	0 <sup>*1</sup>	Positive logic output (active high)
	3	TAUDnTOL03	Don't care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don't care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enable dead time control. <sup>*2</sup>
	9 to 4	TAUDnTDE09 to TAUDnTDE04	0	Disable dead time control.
	3	TAUDnTDE03	Don't care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 4	TAUDnTDM09 to TAUDnTDM04	0	Invalid because dead time control is disabled.
	3	TAUDnTDM03	Don't care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don't care	
TAUDnTDL	15	TAUDnTDL15	1 <sup>*1</sup>	Dead time is in the negative segment of the W phase output
	14	TAUDnTDL14	0 <sup>*1</sup>	Dead time is in the positive segment of the W phase output
	13	TAUDnTDL13	1 <sup>*1</sup>	Dead time is in the negative segment of the V phase output
	12	TAUDnTDL12	0 <sup>*1</sup>	Dead time is in the positive segment of the V phase output
	11	TAUDnTDL11	1 <sup>*1</sup>	Dead time is in the negative segment of the U phase output
	10	TAUDnTDL10	0 <sup>*1</sup>	Dead time is in the positive segment of the U phase output
	9 to 4	TAUDnTDL09 to TAUDnTDL04	0	Invalid because dead time control is disabled.
	3	TAUDnTDL03	Don't care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't care	

Table 27.48 Common TAUDn Channel Settings (3/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Stop real-time output.
	3	TAUDnTRE03	Don't care	
	2	TAUDnTRE02	0	Stop real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't care	
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don't care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't care	
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't care	
	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't care	
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't care	
	9 to 4	TAUDnRDE09 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't care	

Table 27.48 Common TAUDn Channel Settings (4/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't care	
	9 to 4	TAUDnRDS09 to TAUDnRDS04	0	Enable simultaneous rewriting by using a master channel.
	3	TAUDnRDS03	Don't care	
	2	TAUDnRDS02	0	Enable simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't care	
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't Care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't Care	
	9 to 4	TAUDnRDM09 to TAUDnRDM04	0	Load the signal when the master channel starts counting.
	3	TAUDnRDM03	Don't Care	
	2	TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't Care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't Care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't Care	
	9 to 4	TAUDnRDC09 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't Care	
	2	TAUDnRDC02	1	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't Care	

- Note 1. Change the setting according to the used system.
- Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 23, Timer Array Unit D (TAUD)**.

### 27.11.6.2 PIC Settings

**Table 27.49 PIC Settings**

Register	Bit Position	Bit Name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Select the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Select the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Select the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	1	Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
	3	PIC0REG2n203	1	Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
	2	PIC0REG2n202	1	Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.

## 27.12 High-accuracy Triangle PWM Output with Dead Time

### 27.12.1 Functional Overview

Compared to the triangle PWM output with dead time of TAUD, this makes it possible to control the variable dead time areas near duties of 100% and 0%. This makes more accurate triangle PWM output possible.

For the triangle PWM output with dead time feature of TAUD, it is not possible to output a UB phase dead time pulse, such as when transitioning to U phase 0% triangular wave output. (See **Figure 27.33**)

For this feature, a pulse is generated in combination with the TAUD timer output, and a pseudo dead time pulse is added.

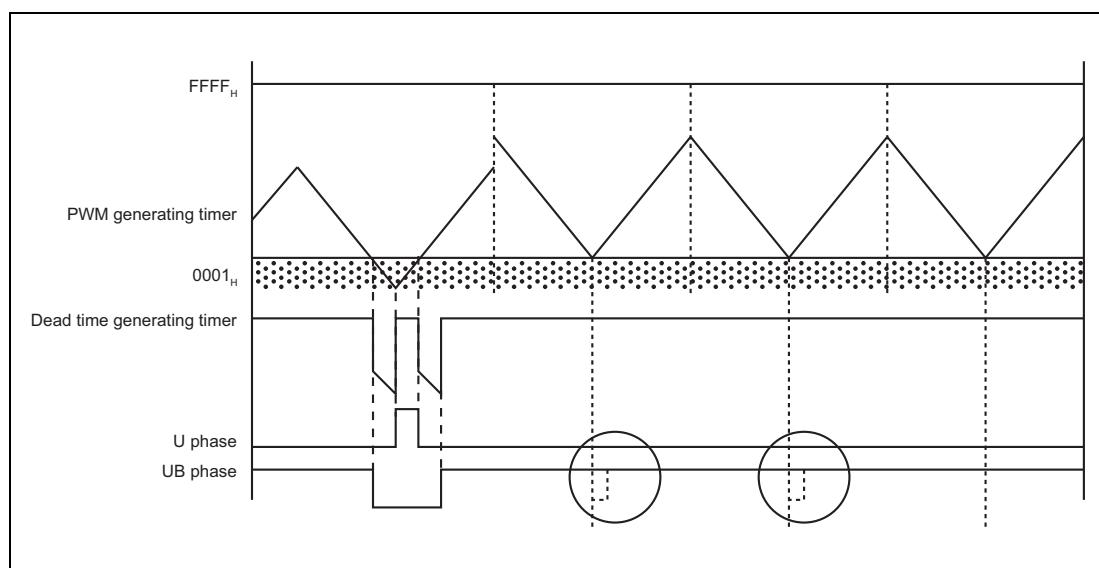


Figure 27.33 Timing of Dead Time Output by the TAUD Feature for Outputting a Triangle PWM Signal with Dead Time

## 27.12.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0)

**Table 27.50 Configuration of Delay Pulse Output with Dead Time**

Timer	Timer Motor Control Function
TAUD0 CH2, CH4 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

### (1) TAUDn configuration

**Table 27.51 TAUD Configuration**

CH	Function Name	M/S <sup>*1</sup>	CDR Setting	Description
2	Triangle PWM output with dead time (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U phase)	
5		S	Dead time (U phase)	
6		S	Duty (V phase)	
7		S	Dead time (V phase)	
8		S	Duty (W phase)	
9		S	Dead time (W phase)	
10	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for U phase PWM.
11		S	Pulse width	
12	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for V phase PWM.
13		S	Pulse width	
14	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for W phase PWM.
15		S	Pulse width	

Note 1. M: Master channel, S: Slave channel

## (2) Block diagram

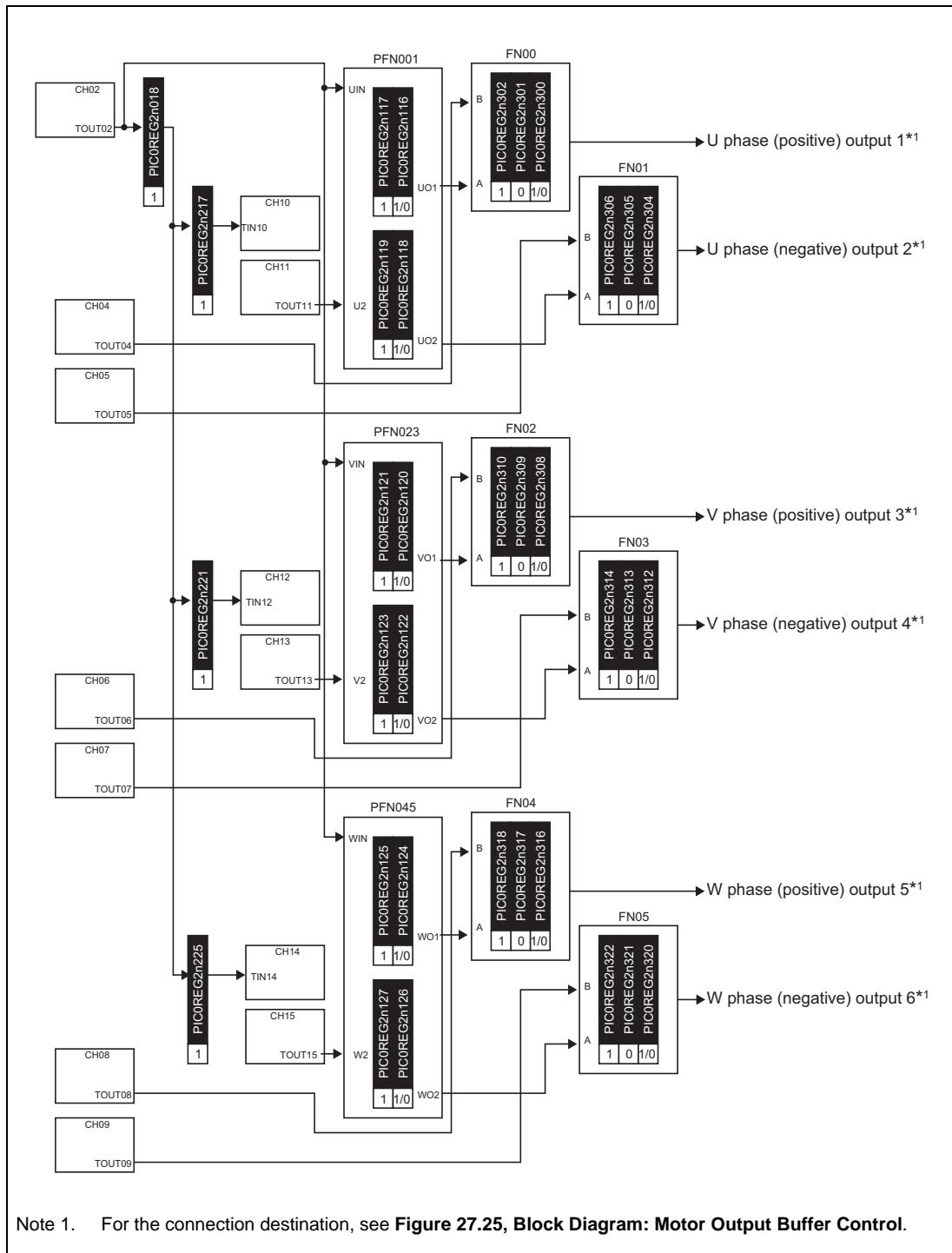


Figure 27.34 Block Diagram: High-Accuracy Triangle PWM Output with Dead Time

### 27.12.3 Registers

#### 27.12.3.1 PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0)

This register selects TAUDn input.

**Access:** This register can be read or written in 32-bit units.

**Address:** PIC0REG200: FFDD 00C0<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC0RE G2n018	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.52 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	*1
18	PIC0REG2n018	Select the TIN input signal to TAUDTTIN10, TAUDTTIN12, and TAUDTTIN14. 0: Setting prohibited 1: Select TAUDTTOUT2.
17 to 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n0 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

### 27.12.3.2 PIC0REG2n1 — Timer I/O Control Register 2n1 (n = 0)

This register selects the logic of a combination circuit.

**Access:** This register can be read or written in 32-bit units.

**Address:** PIC0REG201: FFDD 00C4<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0RE G2n127	PIC0RE G2n126	PIC0RE G2n125	PIC0RE G2n124	PIC0RE G2n123	PIC0RE G2n122	PIC0RE G2n121	PIC0RE G2n120	PIC0RE G2n119	PIC0RE G2n118	PIC0RE G2n117	PIC0RE G2n116
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.53 PIC0REG2n1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27, 26	PIC0REG2n127 PIC0REG2n126	Select the FN05 A input signal according to the output logic specified for CH9 of TAUDn.
	<b>PIC0REG2n127</b>	<b>PIC0REG2n126</b>
	1	0
		Combination circuit output (Select this when the active high setting is specified (TAUDnTOL09 = 0).)
	1	1
		Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL09 = 1).)
	Other than above	Setting prohibited
25, 24	PIC0REG2n125 PIC0REG2n124	Select the FN04 A input signal according to the output logic specified for CH8 of TAUDn.
	<b>PIC0REG2n125</b>	<b>PIC0REG2n124</b>
	1	0
		Combination circuit output (Select this when the active high setting is specified (TAUDnTOL08 = 0).)
	1	1
		Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL08 = 1).)
	Other than above	Setting prohibited
23, 22	PIC0REG2n123 PIC0REG2n122	Select the FN03 A input signal according to the output logic specified for CH7 of TAUDn.
	<b>PIC0REG2n123</b>	<b>PIC0REG2n122</b>
	1	0
		Combination circuit output (Select this when the active high setting is specified (TAUDnTOL07 = 0).)
	1	1
		Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL07 = 1).)
	Other than above	Setting prohibited

Table 27.53 PIC0REG2n1 Register Contents (2/2)

Bit Position	Bit Name	Function												
21, 20	PIC0REG2n121 PIC0REG2n120	Select the FN02 A input signal according to the output logic specified for CH6 of TAUDn.												
		<table border="1"> <thead> <tr> <th>PIC0REG2n121</th><th>PIC0REG2n120</th><th>Input Signal</th></tr> </thead> <tbody> <tr> <td>1</td><td>0</td><td>Combination circuit output (Select this when the active high setting is specified (TAUDnTOL06 = 0).)</td></tr> <tr> <td>1</td><td>1</td><td>Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL06 = 1).)</td></tr> <tr> <td></td><td>Other than above</td><td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG2n121	PIC0REG2n120	Input Signal	1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL06 = 0).)	1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL06 = 1).)		Other than above	Setting prohibited
PIC0REG2n121	PIC0REG2n120	Input Signal												
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL06 = 0).)												
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL06 = 1).)												
	Other than above	Setting prohibited												
19, 18	PIC0REG2n119 PIC0REG2n118	Select the FN01 A input signal according to the output logic specified for CH5 of TAUDn.												
		<table border="1"> <thead> <tr> <th>PIC0REG2n119</th><th>PIC0REG2n118</th><th>Input Signal</th></tr> </thead> <tbody> <tr> <td>1</td><td>0</td><td>Combination circuit output (Select this when the active high setting is specified (TAUDnTOL05 = 0).)</td></tr> <tr> <td>1</td><td>1</td><td>Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL05 = 1).)</td></tr> <tr> <td></td><td>Other than above</td><td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG2n119	PIC0REG2n118	Input Signal	1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL05 = 0).)	1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL05 = 1).)		Other than above	Setting prohibited
PIC0REG2n119	PIC0REG2n118	Input Signal												
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL05 = 0).)												
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL05 = 1).)												
	Other than above	Setting prohibited												
17, 16	PIC0REG2n117 PIC0REG2n116	Select the FN00 A input signal according to the output logic specified for CH4 of TAUDn.												
		<table border="1"> <thead> <tr> <th>PIC0REG2n117</th><th>PIC0REG2n116</th><th>Input Signal</th></tr> </thead> <tbody> <tr> <td>1</td><td>0</td><td>Combination circuit output (Select this when the active high setting is specified (TAUDnTOL04 = 0).)</td></tr> <tr> <td>1</td><td>1</td><td>Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL04 = 1).)</td></tr> <tr> <td></td><td>Other than above</td><td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG2n117	PIC0REG2n116	Input Signal	1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL04 = 0).)	1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL04 = 1).)		Other than above	Setting prohibited
PIC0REG2n117	PIC0REG2n116	Input Signal												
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL04 = 0).)												
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL04 = 1).)												
	Other than above	Setting prohibited												
15 to 0	Reserved	*1												

Note 1. Some of the bits defined as 0 in the PIC0REG2n1 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

### 27.12.3.3 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

**Access:** This register can be read or written in 32-bit units.

**Address:** PIC0REG202: FFDD 00C8<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC0RE G2n225	—	—	—	PIC0RE G2n221	—	—	—	PIC0RE G2n217	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 27.54 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function						
31 to 26	Reserved	*1						
25	PIC0REG2n225	Select the TIN input signal to TAUDTTIN14.						
		<table border="1"> <thead> <tr> <th>PIC0REG2n225</th> <th>Input signal</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)</td></tr> <tr> <td>Other than above</td> <td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG2n225	Input signal	1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)	Other than above	Setting prohibited
PIC0REG2n225	Input signal							
1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)							
Other than above	Setting prohibited							
24 to 22	Reserved	*1						
21	PIC0REG2n221	Select the TIN input signal to TAUDTTIN12.						
		<table border="1"> <thead> <tr> <th>PIC0REG2n221</th> <th>Input signal</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)</td></tr> <tr> <td>Other than above</td> <td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG2n221	Input signal	1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)	Other than above	Setting prohibited
PIC0REG2n221	Input signal							
1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)							
Other than above	Setting prohibited							
20 to 18	Reserved	*1						
17	PIC0REG2n217	Select the TIN input signal to TAUDTTIN10.						
		<table border="1"> <thead> <tr> <th>PIC0REG2n217</th> <th>Input signal</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)</td></tr> <tr> <td>Other than above</td> <td>Setting prohibited</td></tr> </tbody> </table>	PIC0REG2n217	Input signal	1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)	Other than above	Setting prohibited
PIC0REG2n217	Input signal							
1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)							
Other than above	Setting prohibited							
16 to 0	Reserved	*1						

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

### 27.12.3.4 PIC0REG2n3 — Timer I/O Control Register 2n3 (n = 0)

This register selects the logic of a combination circuit.

**Access:** This register can be read or written in 32-bit units.

**Address:** PIC0REG203: FFDD 00CC<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PIC0REG G2n322	PIC0REG G2n321	PIC0REG G2n320	—	PIC0REG G2n318	PIC0REG G2n317	PIC0REG G2n316
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC0REG G2n314	PIC0REG G2n313	PIC0REG G2n312	—	PIC0REG G2n310	PIC0REG G2n309	PIC0REG G2n308	—	PIC0REG G2n306	PIC0REG G2n305	PIC0REG G2n304	—	PIC0REG G2n302	PIC0REG G2n301	PIC0REG G2n300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W												

Table 27.55 PIC0REG2n3 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	*1
22	PIC0REG2n322	Select the logical operation to perform on input signals A and B according to the output logic specified for CH9 of TAUDn.
21	PIC0REG2n321	
20	PIC0REG2n320	
		<b>PIC0REG 2n322    PIC0REG 2n321    PIC0REG 2n320    Input signal</b>
		1        0        0        A and B (Select this when the active high setting is specified (TAUDnTOL09 = 0).)
		1        0        1        A or B (Select this when the active low setting is specified (TAUDnTOL09 = 1).)
		Other than above              Setting prohibited
19	Reserved	*1
18	PIC0REG2n318	Select the logical operation to perform on input signals A and B according to the output logic specified for CH8 of TAUDn.
17	PIC0REG2n317	
16	PIC0REG2n316	
		<b>PIC0REG 2n318    PIC0REG 2n317    PIC0REG 2n316    Input signal</b>
		1        0        0        A and B (Select this when the active high setting is specified (TAUDnTOL08 = 0).)
		1        0        1        A or B (Select this when the active low setting is specified (TAUDnTOL08 = 1).)
		Other than above              Setting prohibited
15	Reserved	*1

Table 27.55 PIC0REG2n3 Register Contents (2/2)

Bit Position	Bit Name	Function
14	PIC0REG2n314	Select the logical operation to perform on input signals A and B according to the output logic specified for CH7 of TAUDn.
13	PIC0REG2n313	
12	PIC0REG2n312	
		<b>PIC0REG 2n314</b> <b>PIC0REG 2n313</b> <b>PIC0REG 2n312</b> <b>Input signal</b>
		1      0      0      A and B (Select this when the active high setting is specified (TAUDnTOL07 = 0).)
		1      0      1      A or B (Select this when the active low setting is specified (TAUDnTOL07 = 1).)
		Other than above      Setting prohibited
11	Reserved	*1
10	PIC0REG2n310	Select the logical operation to perform on input signals A and B according to the output logic specified for CH6 of TAUDn.
9	PIC0REG2n309	
8	PIC0REG2n308	
		<b>PIC0REG 2n310</b> <b>PIC0REG 2n309</b> <b>PIC0REG 2n308</b> <b>Input signal</b>
		1      0      0      A and B (Select this when the active high setting is specified (TAUDnTOL06 = 0).)
		1      0      1      A or B (Select this when the active low setting is specified (TAUDnTOL06 = 1).)
		Other than above      Setting prohibited
7	Reserved	*1
6	PIC0REG2n306	Select the logical operation to perform on input signals A and B according to the output logic specified for CH5 of TAUDn.
5	PIC0REG2n305	
4	PIC0REG2n304	
		<b>PIC0REG 2n306</b> <b>PIC0REG 2n305</b> <b>PIC0REG 2n304</b> <b>Input signal</b>
		1      0      0      A and B (Select this when the active high setting is specified (TAUDnTOL05 = 0).)
		1      0      1      A or B (Select this when the active low setting is specified (TAUDnTOL05 = 1).)
		Other than above      Setting prohibited
3	Reserved	*1
2	PIC0REG2n302	Select the logical operation to perform on input signals A and B according to the output logic specified for CH4 of TAUDn.
1	PIC0REG2n301	
0	PIC0REG2n300	
		<b>PIC0REG 2n302</b> <b>PIC0REG 2n301</b> <b>PIC0REG 2n300</b> <b>Input signal</b>
		1      0      0      A and B (Select this when the active high setting is specified (TAUDnTOL04 = 0).)
		1      0      1      A or B (Select this when the active low setting is specified (TAUDnTOL04 = 1).)
		Other than above      Setting prohibited

Note 1. Some of the bits defined as 0 in the PIC0REG2n3 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

### 27.12.3.5 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

This register selects the Hi-Z output control input signal of TAPAn.

**Access:** This register can be read or written in 8-bit units.

**Address:** PIC0HIZCEN0: FFDD 0080<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 27.56 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Select whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Select whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

## 27.12.4 Operation Example

This is achieved by combining the following TAUD features:

- Triangle PWM output with dead time
- One-shot pulse output

In addition, the following PIC is also used because the pulse to be inserted into the variable dead time area is generated for the positive or negative phase:

- Combination circuit (PFN001, PFN023, and PFN045)

In addition, the following peripheral interconnections are also used because the pulse to be inserted into the variable dead time area is combined with the triangle PWM output waveform:

- Logical operation circuit (FN0i) ( $i = 0$  to 5)

A high-accuracy triangle PWM signal with dead time is output by assigning the PWM output achieved using the above features to the U, V, and W phases. Therefore, the PWM output dead time can be freely specified for the PWM signal of each phase. Because the only difference among phases is the assigned channel, only one phase (the U phase) is described below.

### 27.12.4.1 Triangle PWM Output with Dead Time

A triangle PWM signal with dead time is output from TOUT04 and TOUT05 by using CH2, CH4, and CH5 in combination.

### 27.12.4.2 One-shot Pulse Output

A CDR11 pulse for which the width is delayed by the delay time (CDR10) from the valid edge of the TIN10 (TOUT02) signal of CH10 is output as TOUT11 by using CH10 and CH11 in combination.

This pulse is used as the variable dead time area pulse used near duties of 100% and 0%.

#### CAUTION

Specify each CDR setting for one-shot pulse output such that the following condition is satisfied:  $CDR05 \geq (CDR10 + CDR11)$

If a value that does not satisfy the above condition is specified, the output waveform might be affected. To minimize this effect, in addition to satisfying the above setting condition, leave CDR11 set to  $0000_H$  until the variable dead time area pulse is required. Detect both rising and falling edges as the valid TIN10 (TOUT02) edge, and set TAUDnTOL11 to 1 (active low).

Specify the same operation clock for each TAUDn channel used for outputting a triangle PWM signal with dead time or a one-shot pulse.

For details about the TAUD functions, see **Section 23, Timer Array Unit D (TAUD)**.

### 27.12.4.3 U phase Combination Circuit (PFN001)

This circuit generates a variable dead time area pulse (FN00 A, FN01 A) for adding a generated one-shot pulse to a generated triangle PWM signal with dead time.

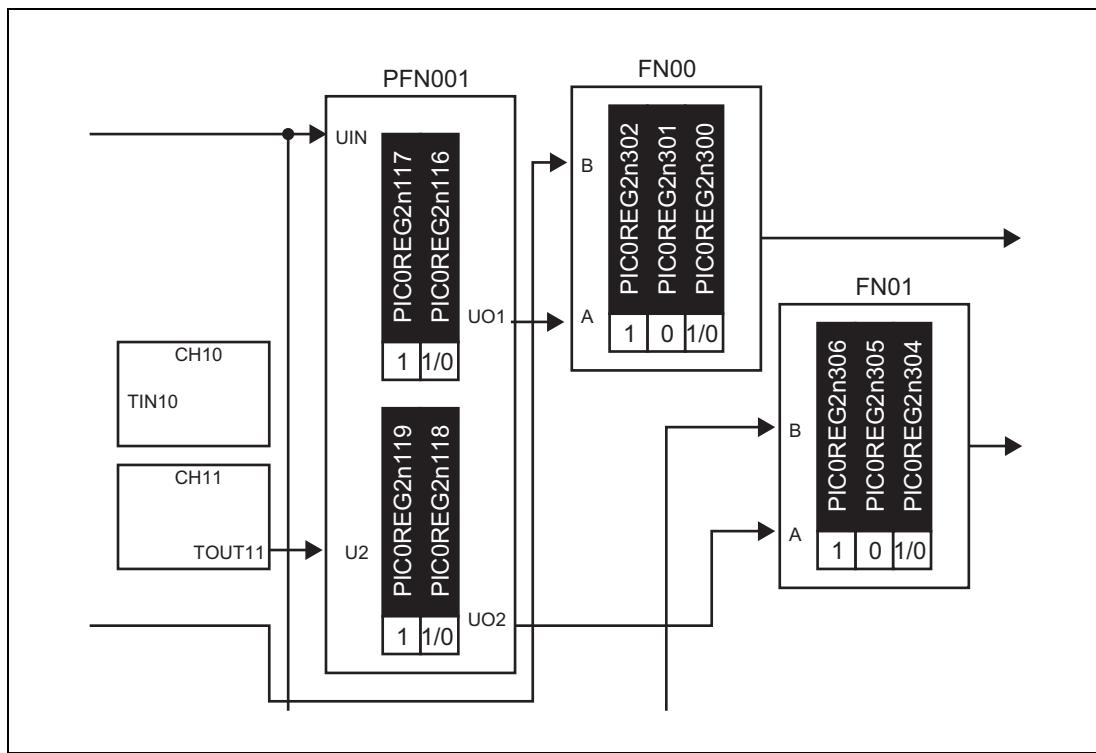


Figure 27.35 Block Diagram Excerpt (PFN001, FN00, and FN01)

The table below shows the relationships between combination circuit input (UIN, U2) and output (UO1, UO2).

**Table 27.57 U and UB Phase Combination Circuit (PFN001) I/O Table**

- **UO1 (U phase variable dead time area pulse) output**

UIN (TOUT02)	U2 (TOUT11)	UO1	
		PIC0REG2n117, 16 = 10 <sub>B</sub> U phase output active high (TAUDnTOL04 = 0)	PIC0REG2n117, 16 = 11 <sub>B</sub> U phase output active low (TAUDnTOL04 = 1)
0	0	1	0
0	1	1	0
1	0	0	1
1	1	1	0

- **UO2 (UB phase variable dead time area pulse) output**

UIN (TOUT02)	U2 (TOUT11)	UO2	
		PIC0REG2n119, 18 = 10 <sub>B</sub> UB phase output active high (TAUDnTOL05 = 0)	PIC0REG2n119, 18 = 11 <sub>B</sub> UB phase output active low (TAUDnTOL05 = 1)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

#### NOTE

The PIC0REG2n116, PIC0REG2n117, PIC0REG2n118, and PIC0REG2n119 settings change depending on the active U phase and UB phase levels of the generated triangle PWM signal with dead time.

#### 27.12.4.4 Logical Operation Circuit (FN0i) (i = 0 or 1)

This circuit combines an output triangle PWM signal with dead time (TOUT04, TOUT05) with combination circuit output (UO1 and UO2 of PFN001) and generates a PWM signal to which a variable dead time area pulse has been added.

The combination logic for the logical operation circuit is switched according to the PIC0REG2n3 register setting. (Bits 0 to 2 are specified for U phase output, and bits 4 to 6 are specified for UB phase output.)

Set up the logical operation circuit as shown in the table below. The combined signal is output from the TAPAnUP and TAPAnUM pins according to the specified combination logic.

**Table 27.58 Logical Operation Circuit (FN0i) (i = 0 or 1) Settings and TAPAnUP and TAPAnUM Pin Output**

- **U phase output (TOUT04)**

Active level	PIC0REG2n302 to 00	TAPAnUP pin output waveform
Active high (TAUDnTOL04 = 0)	100B	AND of FN00 B (TOUT04) and FN00 A (UO1)
Active low (TAUDnTOL04 = 1)	101B	OR of FN00 B (TOUT04) and FN00 A (UO1)

- **UB phase output (TOUT05)**

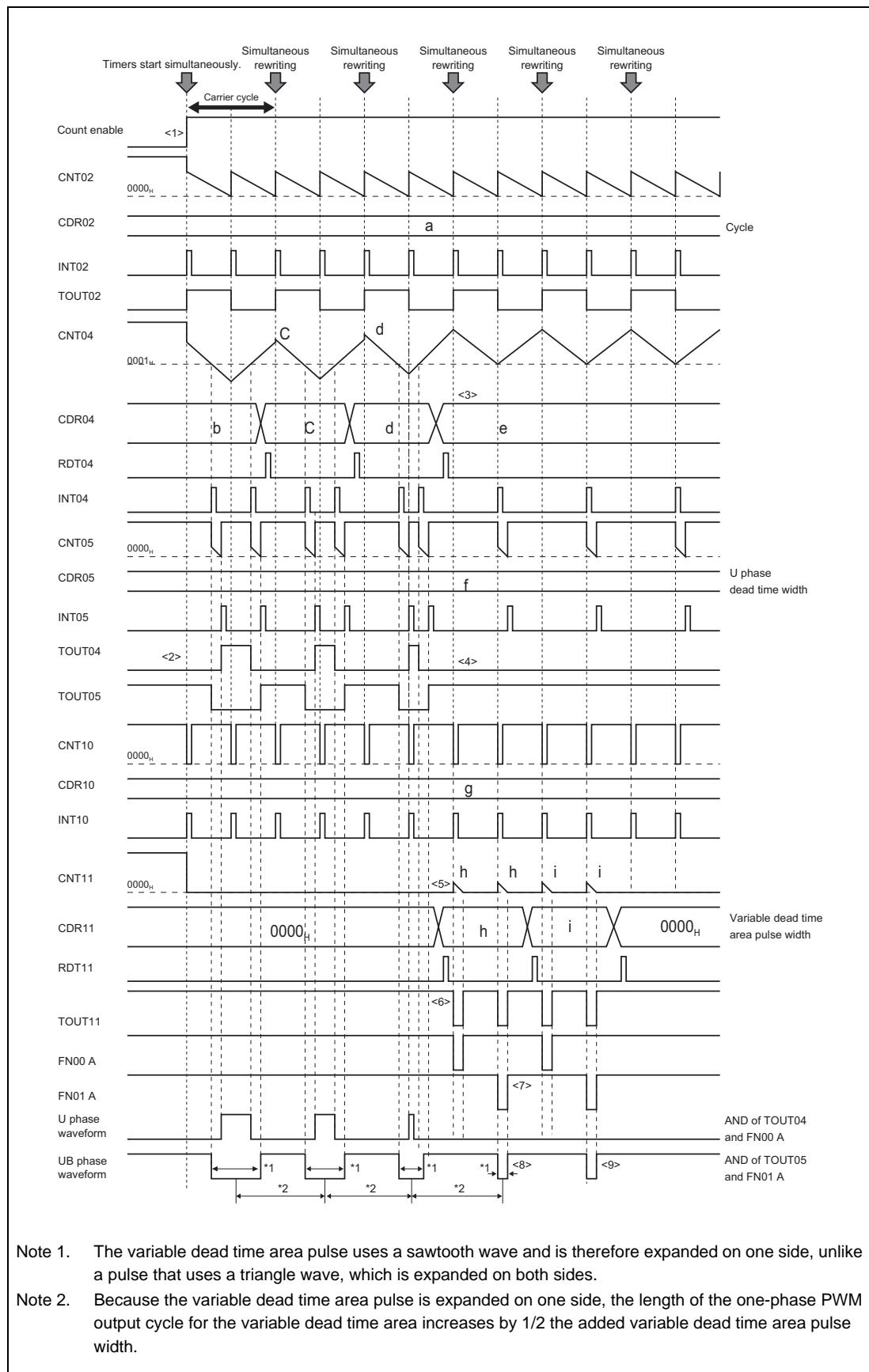
Active level	PIC0REG2n306 to 04	TAPAnUM pin output
Active high (TAUDnTOL05 = 0)	100B	AND of FN01 B (TOUT05) and FN01 A (UO2)
Active low (TAUDnTOL05 = 1)	101B	OR of FN01 B (TOUT05) and FN01 A (UO2)

Because the above makes variable dead time control possible to ensure output accuracy near duties of 0% and 100% even for TAUD, a more accurate triangle PWM signal can be output than that output using the TAUD feature for outputting a triangle PWM signal with dead time.

For the V/VB phase and W/WB phase, the used channels and register bits differ, but the settings are the same, as shown in **Figure 27.34, Block Diagram: High-Accuracy Triangle PWM Output with Dead Time.**

The peripheral interconnections provide a connection for adding the pulse generated during one-shot pulse output to the PWM signal generated during output of a triangle PWM signal with dead time by using the combination circuit and logical operation circuit of the peripheral interconnections.

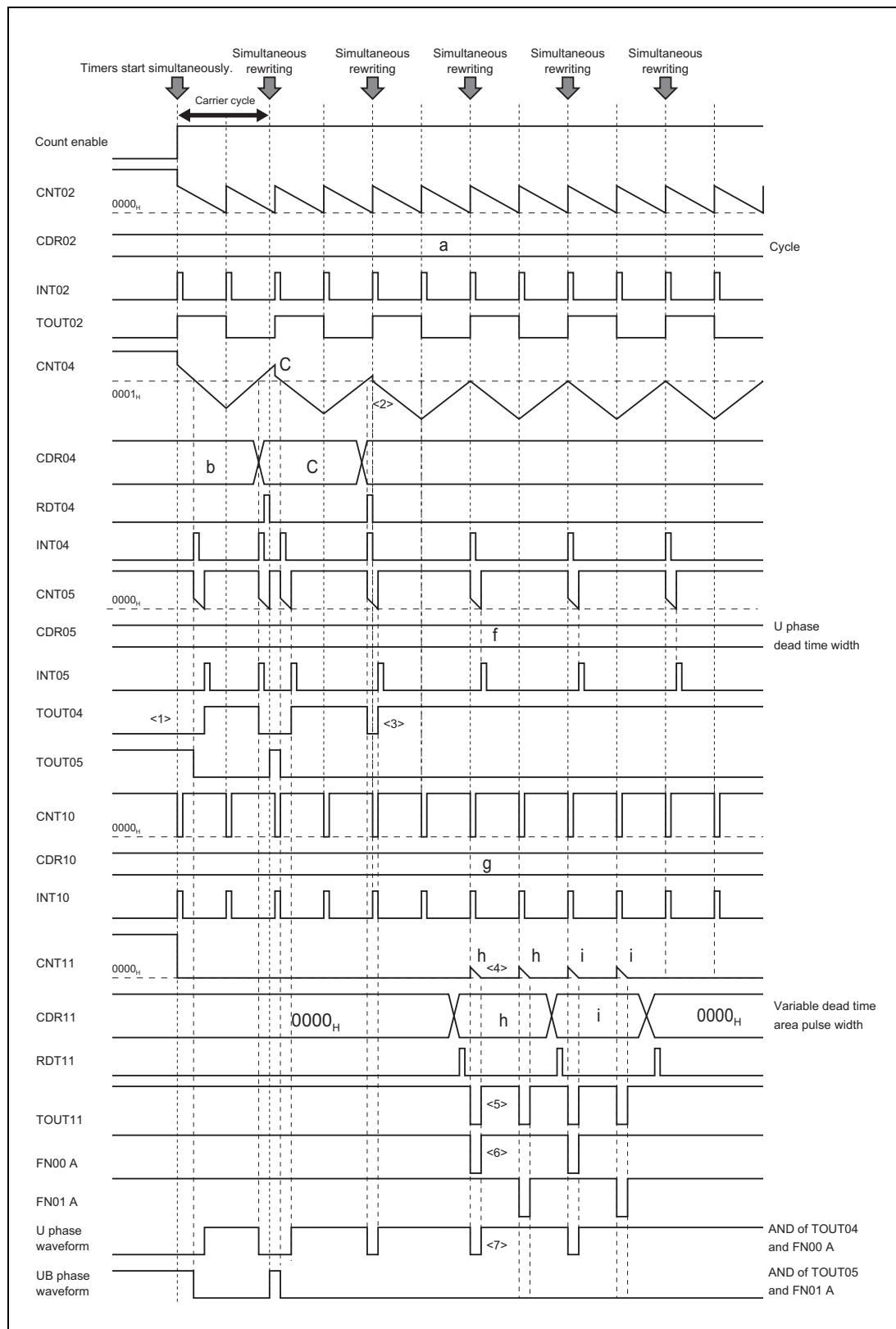
The following figures show timing charts for outputting a high-accuracy triangle PWM signal with dead time.



**Figure 27.36 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**

An operation example in which the system transitions to a U phase of 0% and UB phase of 100% in the timer configuration for performing the U phase PWM output shown in **Figure 27.36** is provided below. Output of a triangle PWM signal with dead time is active high.

- (1) When timer operation is started, output of a triangle PWM with dead time is started by the CH2, CH4, and CH5 channels of TAUDn.
- (2) A triangle PWM waveform with dead time is generated from TOUT04 and TOUT05.
- (3) A U phase duty output value of 0% is specified for CDR04.
- (4) Due to the setting in <3>, TOUT04 output is the inactive level, and TOUT05 output is the active level. However, no variable dead time area pulse is output during this operation.
- (5) To create a variable dead time area pulse, the value to be used as the pulse width is specified for CDR11 when specifying the 0% U phase duty in <3>. For this example, the CDR11 setting is fixed to 0000<sub>H</sub> until the system enters the variable dead time area to prevent adverse effects on the output PWM signal.
- (6) The variable dead time area pulse is output as a pulse that has the width specified for CDR11 after the delay time specified for CDR10 elapses, starting at the TOUT02 edge.
- (7) The pulse output in <6> is converted to a variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A) by the combination circuit (PFN001).
- (8) The pulse generated in <7> is combined with the TOUT04 and TOUT05 output waveforms by using the logical operation circuits (FN00, FN01), and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output).
- (9) By later changing the CDR11 setting, which specifies the width of the variable dead time area pulse, the desired variable dead time area pulse can be added.



**Figure 27.37 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**

An operation example in which the system transitions to a U phase of 100% and UB phase of 0% in the timer configuration for performing the U phase PWM output shown in **Figure 27.37** is provided below. Output of a triangle PWM signal with dead time is active high.

- (1) The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same.
- (2) A U phase duty output value of 100% ( $CDR04 = 0000_H$ ) is specified for CDR04.
- (3) Due to the setting in <2>, TOUT04 output is the active level, and TOUT05 output is the inactive level. However, no variable dead time area pulse is output during this operation.
- (4) To create a variable dead time area pulse, the value to be used as the pulse width is specified for CDR11 one cycle after specifying the 100% U phase duty setting in <2>. For this example, the CDR11 setting is fixed to  $0000_H$  until the system enters the variable dead time area to prevent adverse effects on the output PWM signal.
- (5) The variable dead time area pulse is output as a pulse that has the width specified for CDR11 after the delay time specified for CDR10 elapses, starting at the TOUT02 edge.
- (6) The pulse output in <5> is converted to a variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A) by the combination circuit (PFN001).
- (7) The pulse generated in <6> is combined with the TOUT04 and TOUT05 output waveforms by using the logical operation circuits (FN00, FN01), and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output).

---

**CAUTION**

If the 100% U phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the variable dead time area pulse is affected by the amount shown by <2> for the last PWM signal output from TOUT04 and shown by feature specification <1>, as shown in Figure 27.38.

To cancel this effect, the CDR11 setting is delayed one cycle.

---

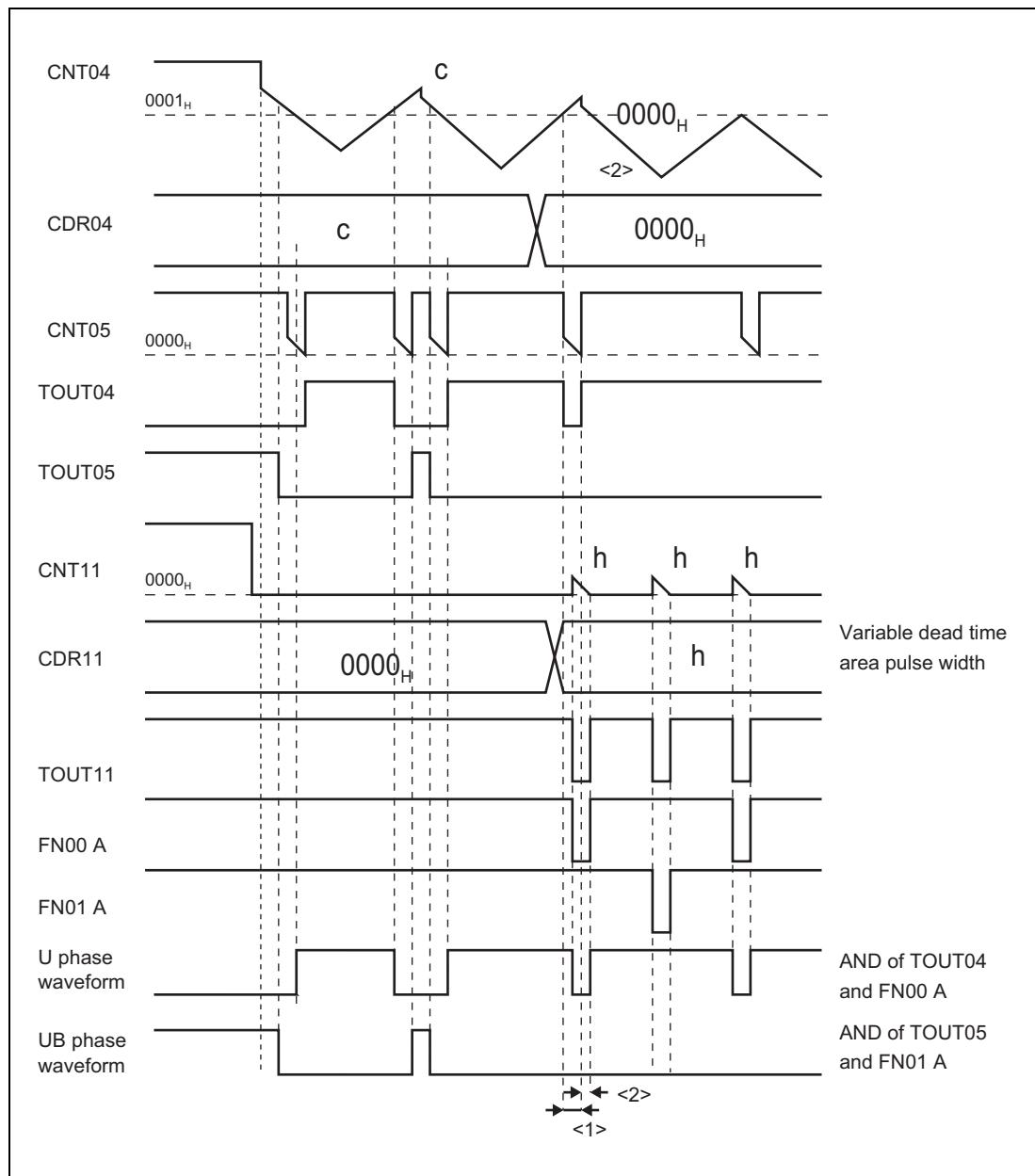
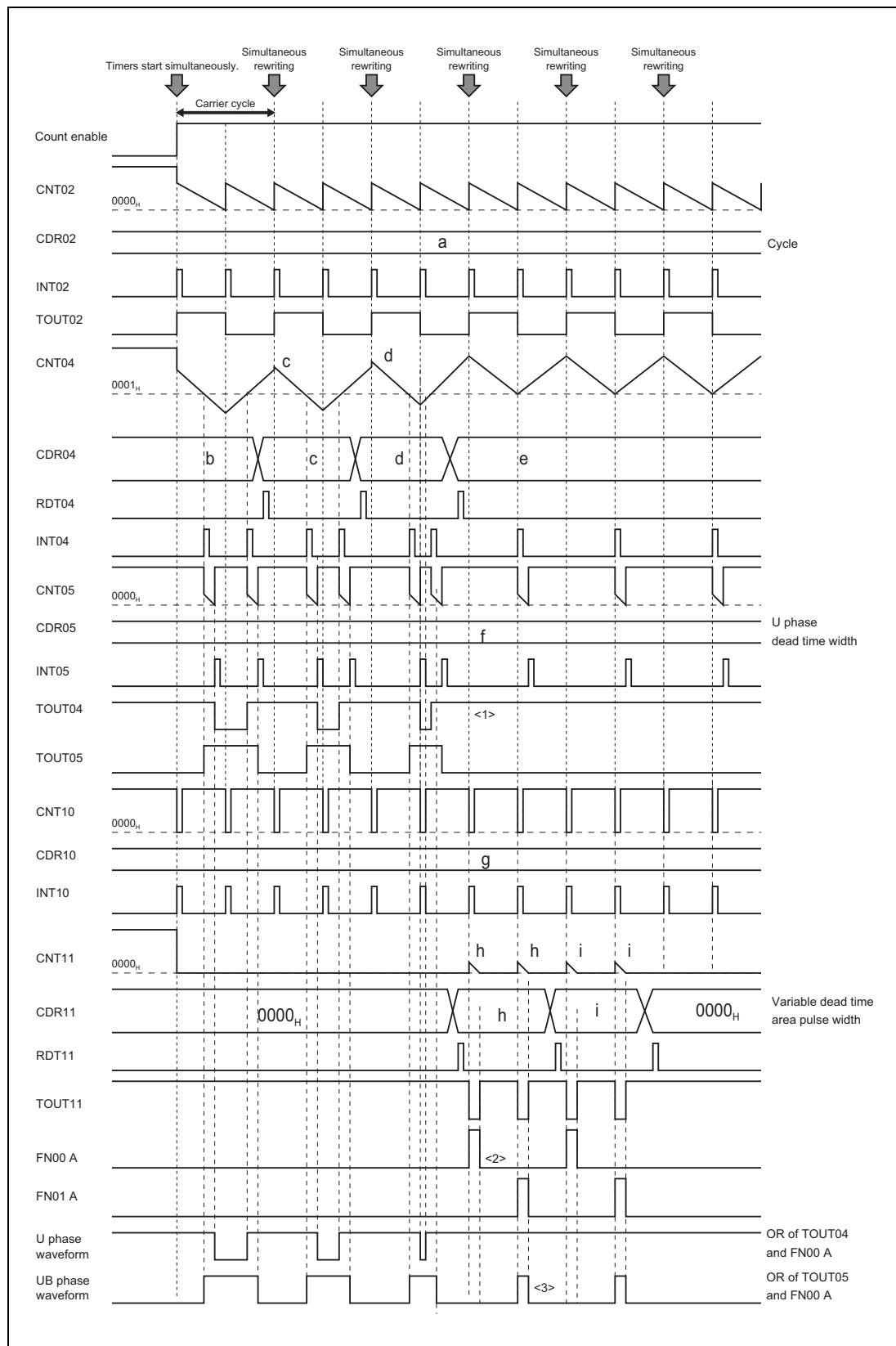


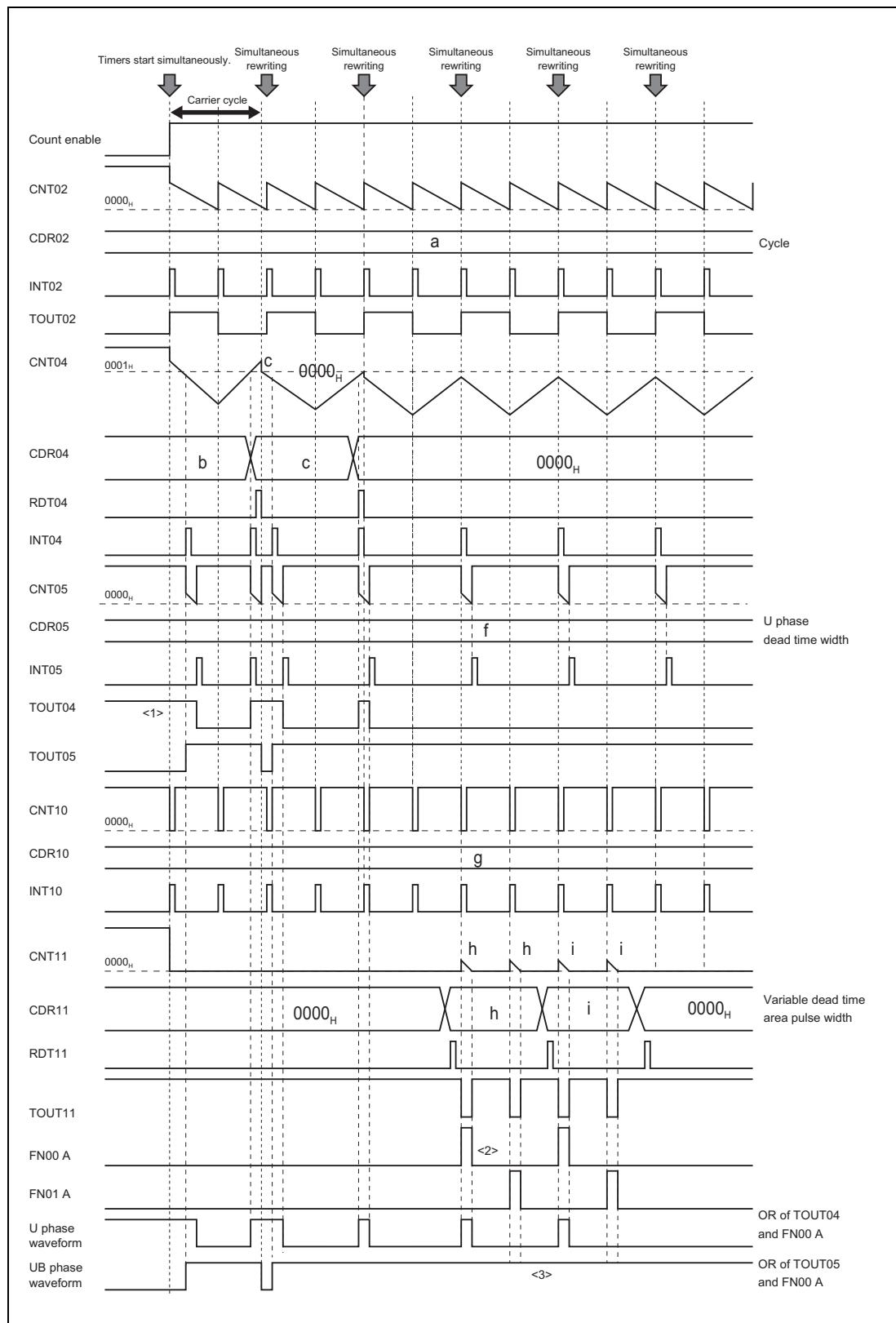
Figure 27.38 Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse



**Figure 27.39 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (TAUDnTOL04 = 1 (Active Low), TAUDnTOL05 = 1 (Active Low))**

An operation example in which the system transitions to a U phase of 100% and UB phase of 0% in the timer configuration for performing the U phase PWM output shown in **Figure 27.39** is provided below. Output of a triangle PWM signal with dead time is active low.

- (1) The timer operation from the start of timer operation until the output of a triangle PWM with dead time is the same as in **Figure 27.36, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**. However, an active low PWM signal is output from TOUT04 and TOUT05.
- (2) Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).
- (3) In addition, active low output that corresponds with PWM output is also specified as the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304). The pulse generated in <2> is combined with the TOUT04 and TOUT05 output waveforms, and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output) as an active low PWM signal.



**Figure 27.40 Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 0%, UB Phase: 100%) (when TAUDnTOL04 = 0 (Active Low) and TAUDnTOL05 = 0 (Active Low))**

An operation example in which the system transitions to a U phase of 0% and UB phase of 100% in the timer configuration for performing the U phase PWM output shown in **Figure 27.40** is provided below. Output of a triangle PWM signal with dead time is active low.

- (1) The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same as in **Figure 27.37, Example of a High-Accuracy PWM Signal Output with Dead Time (U Phase: 100%, UB Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**. However, an active low PWM signal is output.
- (2) Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).
- (3) In addition, active low output that corresponds with PWM output is also specified as the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304). The pulse generated in <2> is combined with the TOUT04 and TOUT05 output waveforms, and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB phase output) as an active low PWM signal.

---

**CAUTION**

If the 100% U phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the last PWM signal output from TOUT04 is adversely affected due to the feature specifications.

To cancel this effect, the CDR11 setting is delayed one cycle.

For details, see **Figure 27.38, Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse**.

---

### 27.12.5 Setup Flow

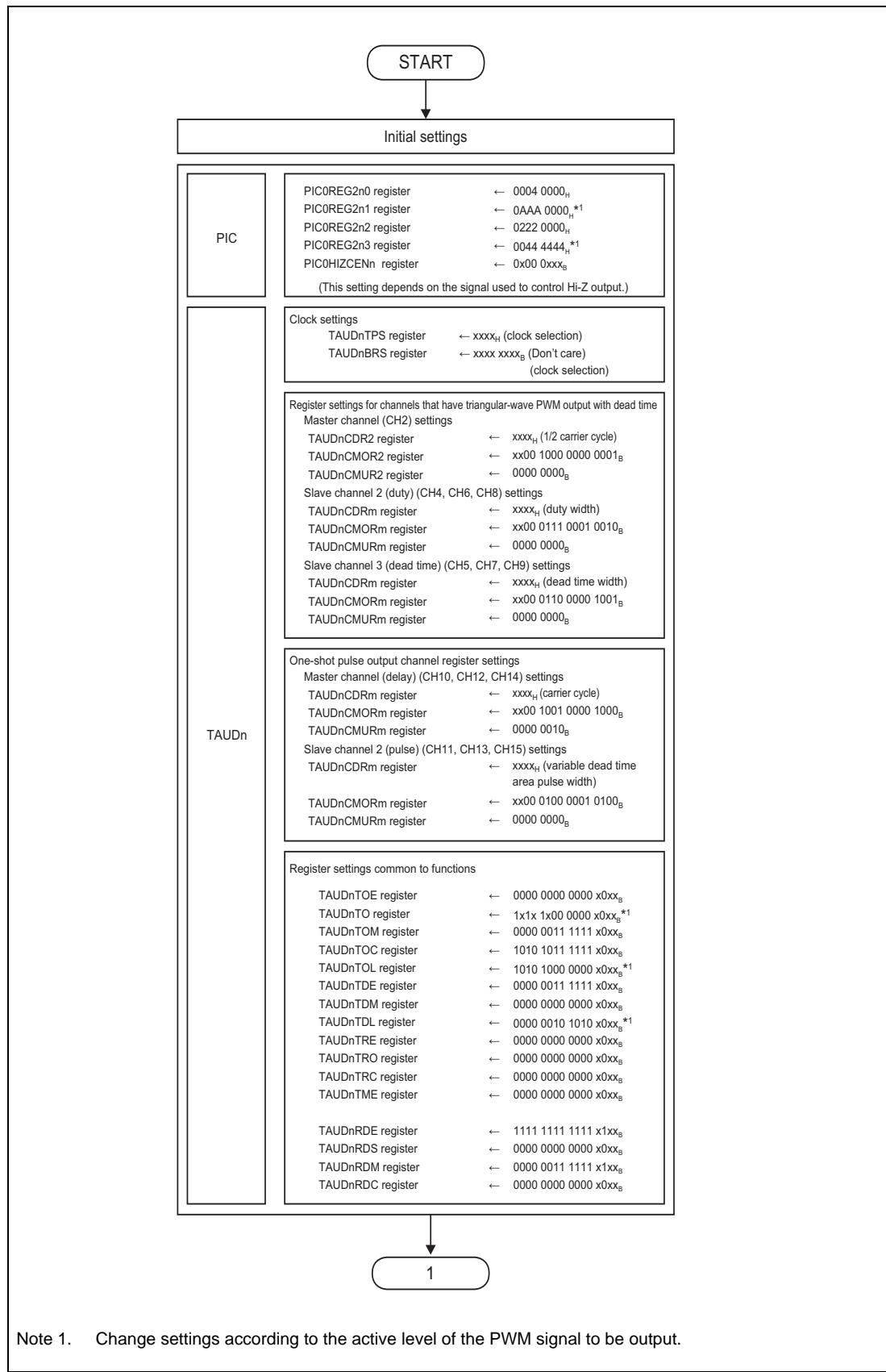


Figure 27.41 Setup Flow (Active High Example)

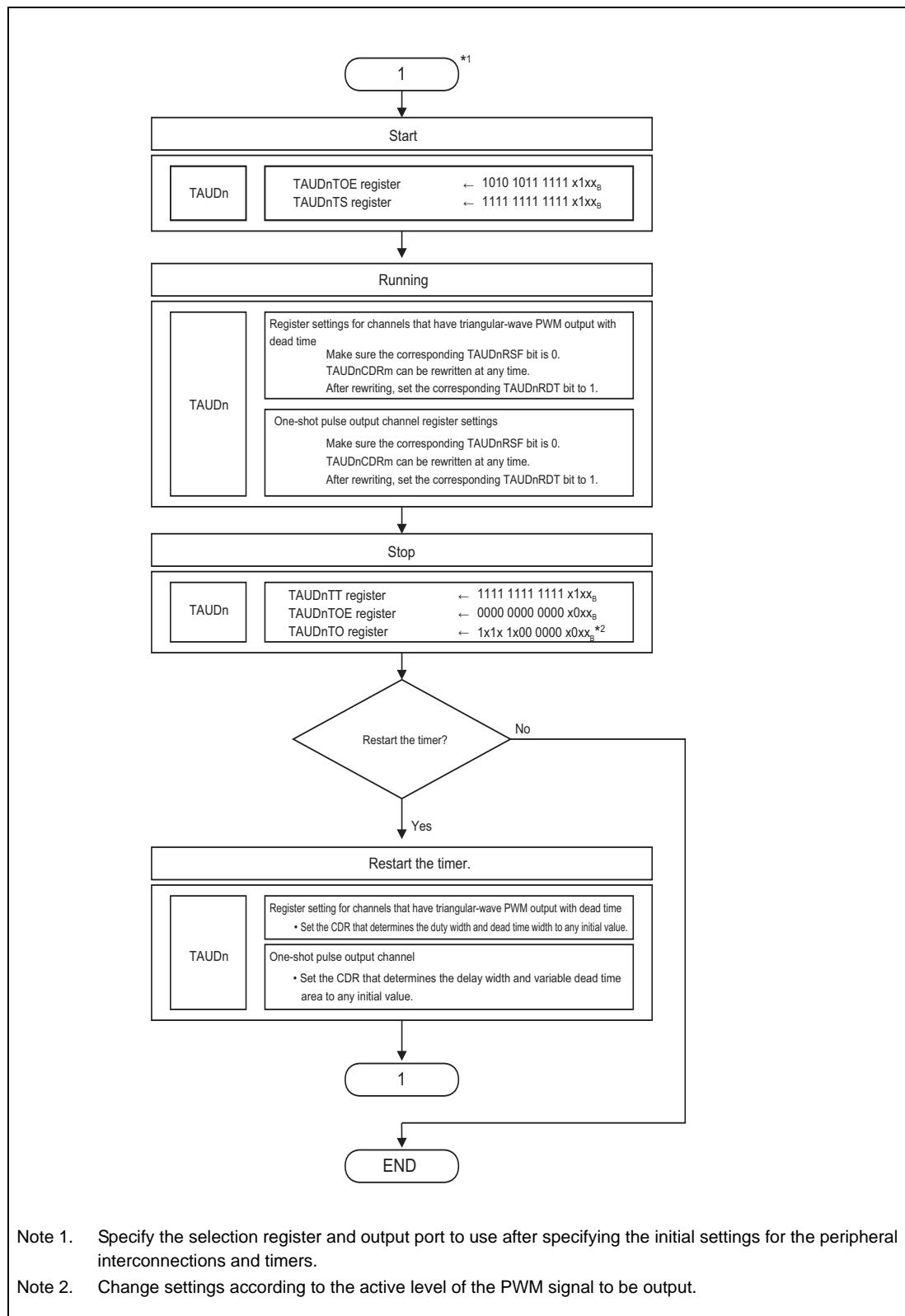


Figure 27.42 Setup Flow (Active High Example) (continued)

## 27.12.6 Setting Examples for Operation Functions

This section provides example settings for each register.

### 27.12.6.1 TAUDn settings (active high example)

**Table 27.59 TAUDn: CH2-related (Master Channel Used To Output A Triangle PWM Signal with Dead Time<sup>\*1</sup>)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	At the start of operation, output INTm and toggle TOUTm.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	Fixed

Note 1. The master channel and slave channel names are defined for TAUD triangle PWM output with dead time. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

#### NOTE

For the TAUDnCMORm register of the master channel used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

For this feature, set TAUDnMD0 to 1.

**Table 27.60 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Triangle PWM Signal with Dead Time<sup>\*1</sup>) (m = 4, 6, or 8)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	111	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1001	
	0	TAUDnMD0	0	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.

For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

**Table 27.61 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Triangle PWM Signal with Dead Time<sup>\*1</sup>) (m = 5, 7, or 9)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	110	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.  
 For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

**Table 27.62 TAUDn: CH10, CH12, and CH14-related (Master Channel used to Output a One-shot Pulse<sup>\*1</sup>) (m = 10, 12, or 14)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	0	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	10	Detect both rising and falling edges as valid.

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

**Table 27.63 TAUDn: CH11, CH13, and CH15-related (Slave Channel used to Output a One-Shot Pulse<sup>\*1</sup>) (m = 11, 13, or 15)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	0	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel. Specify the same clock setting as for the master channel (CH2) used to output a triangle PWM signal with dead time.

#### NOTE

For the TAUDnCMORm register used during one-shot pulse output, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

For this feature clear TAUDnMD0 to 0.

**Table 27.64 Common TAUDn Channel Settings (1/4)**

Register	Bit position	Bit name	Setting	Remark
TAUDnTOE	15	TAUDnTOE15	0	Disable the timer.
			1	Enable the timer.
	14	TAUDnTOE14	0	
	13	TAUDnTOE13	0	Disable the timer.
			1	Enable the timer.
	12	TAUDnTOE12	0	
	11	TAUDnTOE11	0	Disable the timer.
			1	Enable the timer.
	10	TAUDnTOE10	0	
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0	Disable the timer.
			1	Enable the timer.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0	Disable the timer.
			1	Enable the timer.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	

Table 27.64 Common TAUDn Channel Settings (2/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTO	15	TAUDnTO15	1 <sup>*1</sup>	Output a high-level signal to TOUT15.
	14	TAUDnTO14	Don't Care	
	13	TAUDnTO13	1 <sup>*1</sup>	Output a high-level signal to TOUT13.
	12	TAUDnTO12	Don't Care	
	11	TAUDnTO11	1 <sup>*1</sup>	Output a high-level signal to TOUT11.
	10	TAUDnTO10	Don't Care	
	9 to 4	TAUDnTO09 to TAUDnTO04	0 <sup>*1</sup>	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't Care	
	2	TAUDnTO02	0	Output a low-level signal to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't Care	
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	0	Independent operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't Care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don't Care	
TAUDnTOC	15	TAUDnTOC15	1	Operation mode 2
	14	TAUDnTOC14	0	Operation mode 1
	13	TAUDnTOC13	1	Operation mode 2
	12	TAUDnTOC12	0	Operation mode 1
	11	TAUDnTOC11	1	Operation mode 2
	10	TAUDnTOC10	0	Operation mode 1
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1	Operation mode 2
	3	TAUDnTOC03	Don't Care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don't Care	
TAUDnTOL	15	TAUDnTOL15	1 <sup>*1</sup>	Inverted logic output (active low)
	14	TAUDnTOL14	Don't Care	
	13	TAUDnTOL13	1 <sup>*1</sup>	Inverted logic output (active low)
	12	TAUDnTOL12	Don't Care	
	11	TAUDnTOL11	1 <sup>*1</sup>	Inverted logic output (active low)
	10	TAUDnTOL10	Don't Care	
	9 to 4	TAUDnTOL09 to TAUDnTOL04	0 <sup>*1</sup>	Positive logic output (active high)
	3	TAUDnTOL03	Don't Care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don't Care	

**Table 27.64 Common TAUDn Channel Settings (3/4)**

<b>Register</b>	<b>Bit position</b>	<b>Bit name</b>	<b>Setting</b>	<b>Remark</b>
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	0	Disable dead time control.
	9 to 4	TAUDnTDE09 to TAUDnTDE04	1	Enable dead time control. <sup>*2</sup>
	3	TAUDnTDE03	Don't Care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't Care	
TAUDnTDM	15 to 9	TAUDnTDM15 to TAUDnTDM09	0	
	3	TAUDnTDM03	Don't Care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don't Care	
TAUDnTDL	15 to 10	TAUDnTDL15 to TAUDnTDL10	0	Invalid because dead time control is disabled.
	9	TAUDnTDL09	1 <sup>*1</sup>	Dead time is in the negative segment of the W phase output
	8	TAUDnTDL08	0 <sup>*1</sup>	Dead time is in the positive segment of the W phase output
	7	TAUDnTDL07	1 <sup>*1</sup>	Dead time is in the negative segment of the V phase output
	6	TAUDnTDL06	0 <sup>*1</sup>	Dead time is in the positive segment of the V phase output
	5	TAUDnTDL05	1 <sup>*1</sup>	Dead time is in the negative segment of the U phase output
	4	TAUDnTDL04	0 <sup>*1</sup>	Dead time is in the positive segment of the U phase output
	3	TAUDnTDL03	Don't Care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't Care	
	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Disable real-time output.
	3	TAUDnTRE03	Don't Care	
TAUDnTRO	2	TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't Care	
	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don't Care	
TAUDnTRC	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't Care	
	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't Care	
TAUDnTRC	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't Care	

**Table 27.64 Common TAUDn Channel Settings (4/4)**

<b>Register</b>	<b>Bit position</b>	<b>Bit name</b>	<b>Setting</b>	<b>Remark</b>
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't Care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15 to 4	TAUDnRDE15 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't Care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't Care	
TAUDnRDS	15 to 4	TAUDnRDS15 to TAUDnRDS04	0	Do not enable simultaneous rewriting by using another upper channel.
	3	TAUDnRDS03	Don't Care	
	2	TAUDnRDS02	0	Do not enable simultaneous rewriting by using another upper channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't Care	
TAUDnRDM	15 to 10	TAUDnRDM15 to TAUDnRDM10	0	Perform simultaneous rewriting when the master channel starts counting.
	9 to 4	TAUDnRDM09 to TAUDnRDM04	1	Perform simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	3	TAUDnRDM03	Don't Care	
	2	TAUDnRDM02	1	Perform simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
TAUDnRDC	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	
	15 to 4	TAUDnRDC15 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't Care	
	2	TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
TAUDnRDC	1, 0	TAUDnRDC01 TAUDnRDC00	Don't Care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 23, Timer Array Unit D (TAUD)**.

### 27.12.6.2 PIC Settings (Active High Example)

Table 27.65 PIC Settings

Register	Bit position	Bit name	Setting	Remark
PIC0REG2n0	18	PIC0REG2n018	1	Select the TOUT signal of CH2 of TAUDn.
PIC0REG2n1	27, 26	PIC0REG2n127 PIC0REG2n126	1 0	Negative W phase active high combination circuit output
	25, 24	PIC0REG2n125 PIC0REG2n124	1 0	Positive W phase active high combination circuit output
	23, 22	PIC0REG2n123 PIC0REG2n122	1 0	Negative V phase active high combination circuit output
	21, 20	PIC0REG2n121 PIC0REG2n120	1 0	Positive V phase active high combination circuit output
	19, 18	PIC0REG2n119 PIC0REG2n118	1 0	Negative U phase active high combination circuit output
	17, 16	PIC0REG2n117 PIC0REG2n116	1 0	Positive U phase active high combination circuit output
PIC0REG2n2	25	PIC0REG2n225	1	Select the input selected by the PIC0REG2n018 bit.
	21	PIC0REG2n221	1	Select the input selected by the PIC0REG2n018 bit.
	17	PIC0REG2n217	1	Select the input selected by the PIC0REG2n018 bit.
PIC0REG2n3	22, 21, 20	PIC0REG2n322 PIC0REG2n321 PIC0REG2n320	1 0 0	Negative W phase active high logical operation circuit output
	18, 17, 16	PIC0REG2n318 PIC0REG2n317 PIC0REG2n316	1 0 0	Positive W phase active high logical operation circuit output
	14, 13, 12	PIC0REG2n314 PIC0REG2n313 PIC0REG2n312	1 0 0	Negative V phase active high logical operation circuit output
	10, 9, 8	PIC0REG2n310 PIC0REG2n309 PIC0REG2n308	1 0 0	Positive V phase active high logical operation circuit output
	6, 5, 4	PIC0REG2n306 PIC0REG2n305 PIC0REG2n304	1 0 0	Negative U phase active high logical operation circuit output
	2, 1, 0	PIC0REG2n302 PIC0REG2n301 PIC0REG2n300	1 0 0	Positive U phase active high logical operation circuit output

## 27.13 Delay Pulse Output with Dead Time

### 27.13.1 Functional Overview

This feature outputs a three-phase PWM with dead time that is later than the cycle timing by an amount equal to the delay amount.

Unlike the function of three-phase PWM output with dead time, a PWM signal that has a reset in the next cycle can be output.

### 27.13.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0, m = 0 to 15)

**Table 27.66 Configuration of Delay Pulse Output with Dead Time**

Timer	Timer motor control function
TAUD0 CH2 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

### 27.13.2.1 TAUDn configuration

Because the CDRm value of CH3 does not affect TOUT0 to TOUT15, the INTm signal of CH3 can also be used for other purposes such as A/D conversion trigger generation.

**Table 27.67 TAUDn configuration**

CH	Function name	M/S <sup>*1</sup>	CDR setting	Description
2	Delay pulse output function (CH2 is the master channel for CH3 to CH9.)	M	Cycle	
3		S		Reserved
4		S	Delay (U phase)	
5		S	Pulse width (U phase)	
6		S	Delay (V phase)	
7		S	Pulse width (V phase)	
8		S	Delay (W phase)	
9		S	Pulse width (W phase)	
10	Any feature that does not use TOUTm	S		TOUT: U phase output
11	One-phase PWM output	S	Dead time (U phase)	TOUT: UB phase output
12	Any feature that does not use TOUTm	S		TOUT: V phase output
13	One-phase PWM output	S	Dead time (V phase)	TOUT: VB phase output
14	Any feature that does not use TOUTm	S		TOUT: W phase output
15	One-phase PWM output	S	Dead time (W phase)	TOUT: WB phase output

Note 1. M: Master channel, S: Slave channel

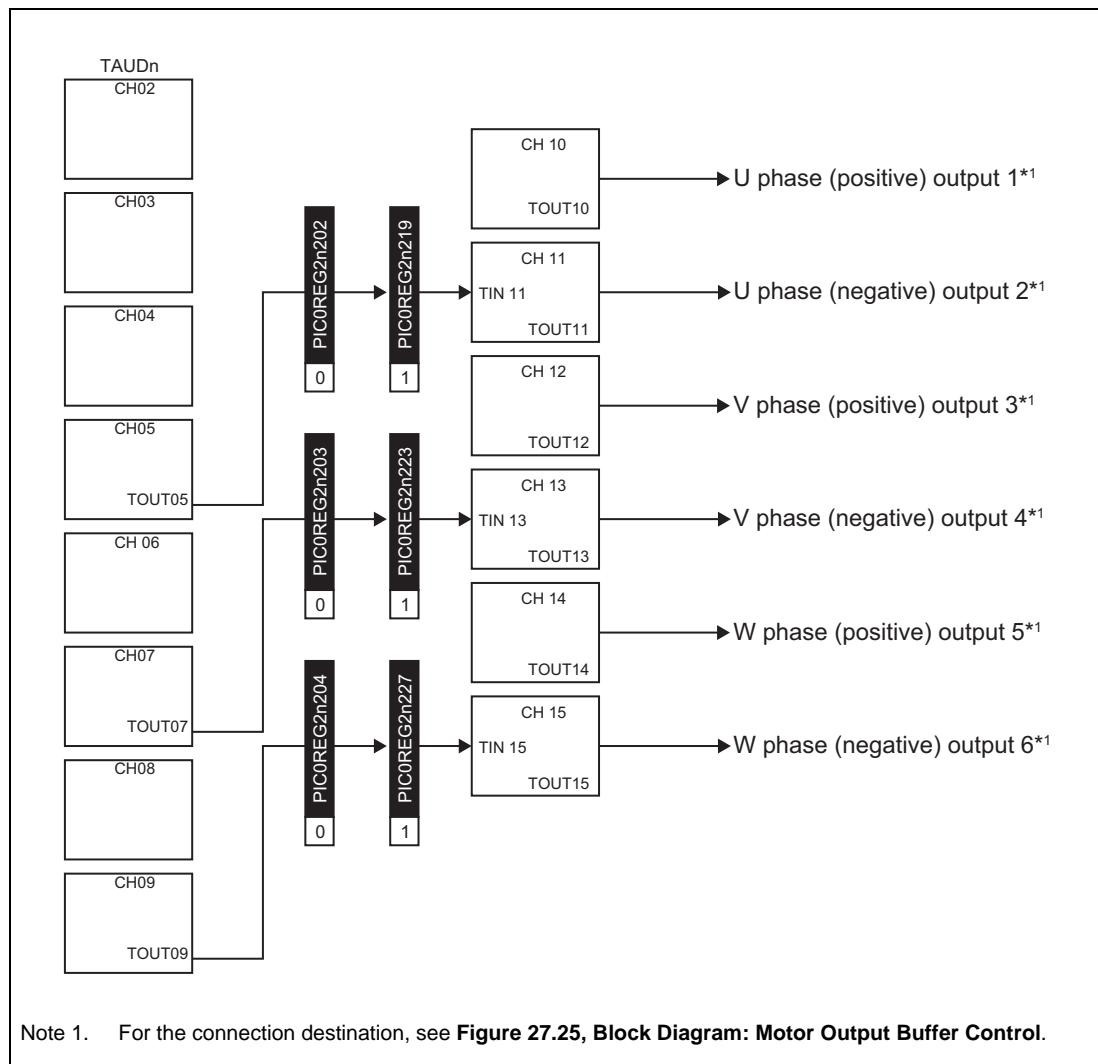


Figure 27.43 Block Diagram: Delay Pulse Output with Dead Time

### 27.13.3 Registers

#### 27.13.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

This register selects CHm input signals of the TAUDn timer. This section describes the bits to be used in the delay pulse output with dead time.

<b>Access:</b>	This register can be read or written in 32-bit units.															
<b>Address:</b>	PIC0REG202: FFDD 00C8H															
<b>Value after reset:</b>	0000 0000H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0RE G2n227	—	—	—	PIC0RE G2n223	—	—	—	PIC0RE G2n219	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0RE G2n204	PIC0RE G2n203	PIC0RE G2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 27.68 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27	PIC0REG2n227	Select the TIN input signal to TAUDTTIN15.
	PIC0REG2n227	Input signal
	1	Signal selected by the PIC0REG2n204 bit
	Other than above	Setting prohibited
26 to 24	Reserved	*1
23	PIC0REG2n223	Select the TIN input signal to TAUDTTIN13.
	PIC0REG2n223	Input signal
	1	Signal selected by the PIC0REG2n203 bit
	Other than above	Setting prohibited
22 to 20	Reserved	*1
19	PIC0REG2n219	Select the TIN input signal to TAUDTTIN11.
	PIC0REG2n219	Input signal
	1	Signal selected by the PIC0REG2n202 bit
	Other than above	Setting prohibited
18 to 5	Reserved	*1
4	PIC0REG2n204	Select the signal supplied to TAUDTTOUT9. 0: Select TAUDTTOUT9. 1: Setting prohibited
3	PIC0REG2n203	Select the signal supplied to TAUDTTIN13. 0: Select TAUDTTOUT7. 1: Setting prohibited
2	PIC0REG2n202	Select the signal supplied to TAUDTTIN11. 0: Select TAUDTTOUT5. 1: Setting prohibited
1, 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

### 27.13.3.2 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

This register selects the Hi-Z output control input signal of TAPAn.

**Access:** This register can be read or written in 8-bit units.

**Address:** PIC0HIZCEN0: FFDD 0080<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R	R	R	R/W	R	R/W
Value after reset	0	0	0	0	0	0	0	0

Table 27.69 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Select whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

## 27.13.4 Operation Example

This is achieved by combining the following TAUD features:

- Delay pulse output function
- One-phase PWM output

The delay pulse output feature generates a PWM signal that is later than the cycle timing by an amount equal to the delay amount. Next, a one-phase PWM signal to which dead time has been added is output for the delayed PWM signal by one-phase PWM output feature.

A delay pulse with dead time is output by assigning the PWM output achieved using the above features to the U, V, and W phases. Therefore, the PWM output dead time can be freely specified for the PWM signal of each phase. Because the only difference between phases is the assigned channel, only one phase (the U phase) is described below.

### 27.13.4.1 Delay pulse output function

By using a combination of CH2, CH4, and CH5, a basic PWM signal for one-phase PWM is output from TOUT05 delayed by the amount specified by CH4 with respect to the cycle specified by CH2.

Note that CH3 is a reserved timer for achieving this feature, so do not use it for other features.

---

**CAUTION**

---

**Do not specify a delay amount that exceeds the cycle.**

---

### 27.13.4.2 One-phase PWM output

One-phase PWM output is generated from TOUT10 and TOUT11 by using a combination of CH10 and CH11.

By specifying the dead time value for CDR11, a one-phase PWM with dead time is output for the TIN11 input.

Similarly, the V phase uses CH12 and CH13 to output a one-phase PWM with dead time, while the W phase uses CH14 and CH15.

---

**CAUTION**

---

**Specify the same clock for each TAUDn channel that uses the delay pulse output and one-phase PWM output features.**

---

For details about the TAUD functions, see **Section 23, Timer Array Unit D (TAUD)**.

The differences between the delay pulse output with dead time and the three-phase PWM output with dead time are described below.

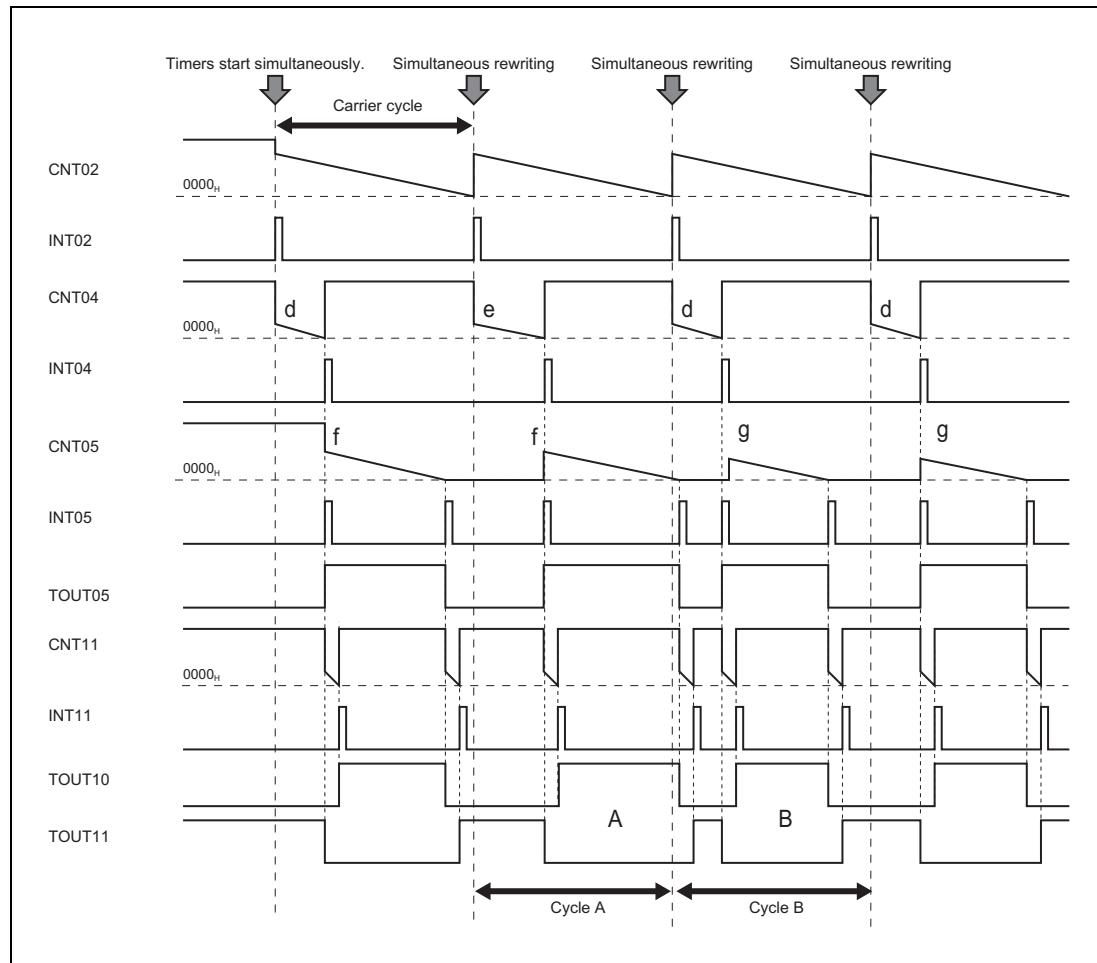


Figure 27.44 PWM Output by Outputting a Delay Pulse with Dead Time

In **Figure 27.44**, PWM waveform A is supposed to be output before cycle A ends, but because the delay timing is too long, the PWM clear position is after the end of cycle A. Next, PWM waveform B, which is for cycle B, is output.

The operations shown below occur when an attempt is made to achieve the operations shown in **Figure 27.44, PWM Output by Outputting a Delay Pulse with Dead Time** by the three-phase PWM output with dead time.

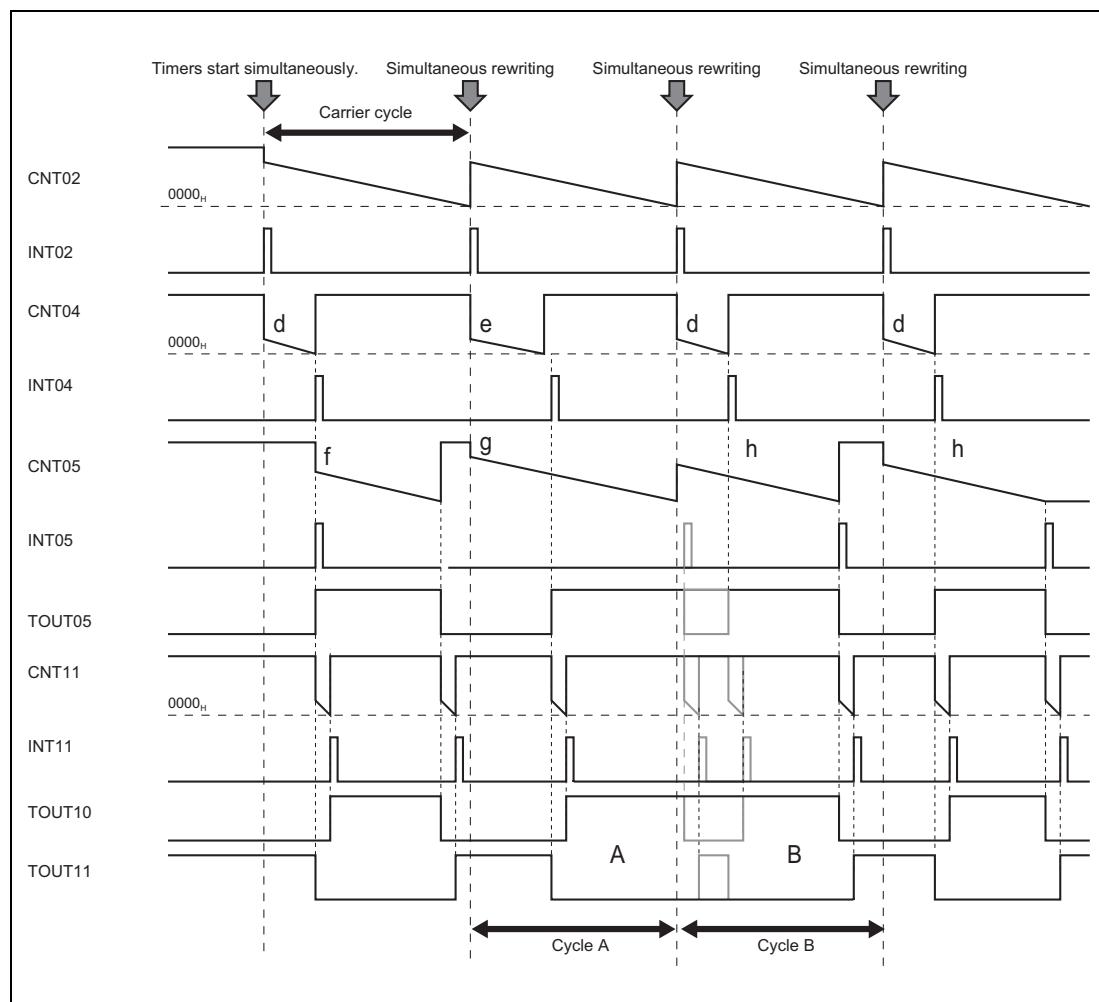


Figure 27.45 Output of a Three-Phase PWM Signal with Dead Time (1)

**Figure 27.45** shows an example in which the output PWM signal does not end before carrier cycle A because the set timing for outputting a three-phase PWM signal with dead time is delayed and the clear timing is after the end of the carrier cycle.

For cycle A, the set timing of PWM waveform A is the same as that in the figure on the previous page, but because the clear timing is after the end of cycle A, a reload operation occurs in cycle A before PWM waveform A is cleared, and the clear timing for PWM waveform A does not occur.

In addition, the set timing of PWM waveform B for cycle B is ignored because a PWM waveform is already set. The result is that there is no PWM waveform change until the clear timing of cycle B, and a waveform that combines PWM waveform A and PWM waveform B is output.

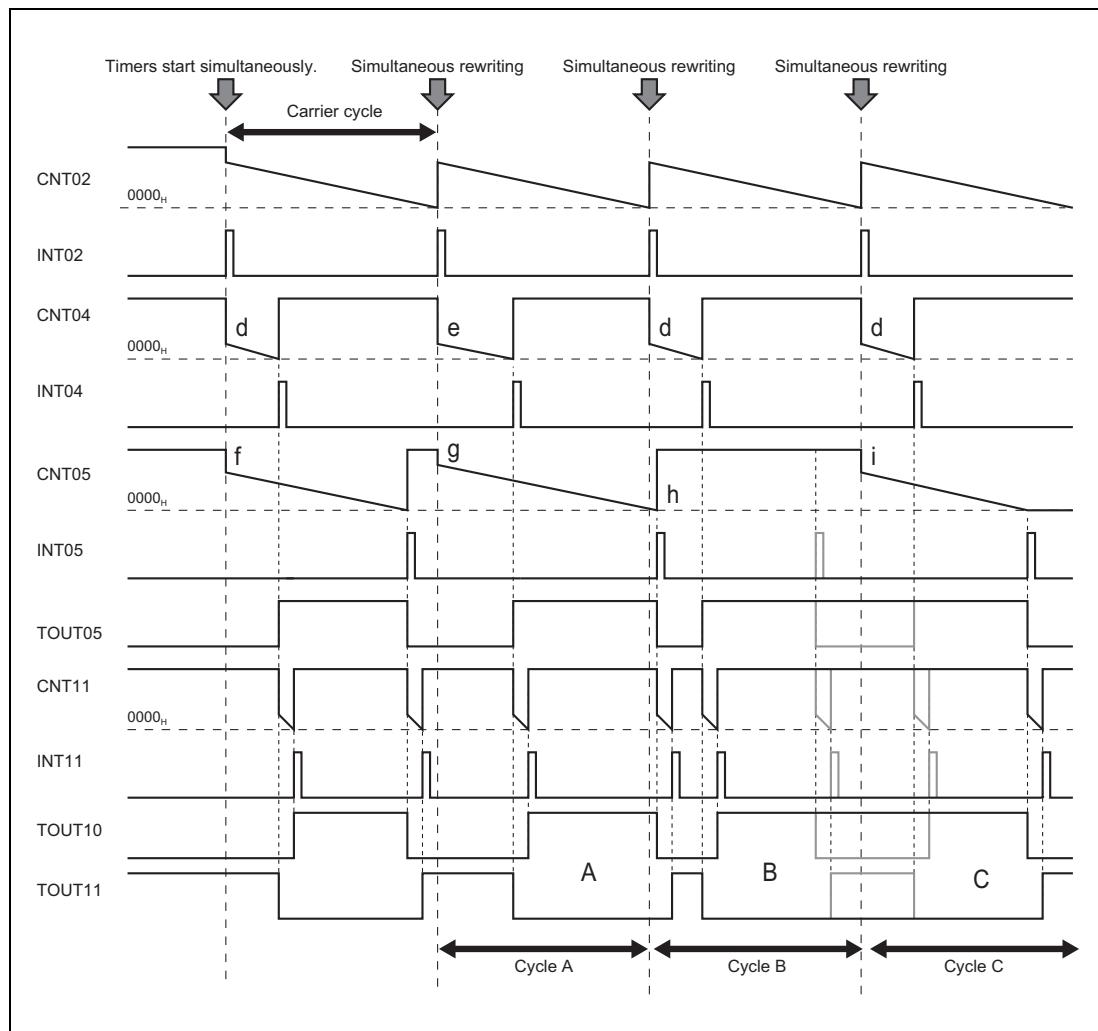


Figure 27.46 Output of a Three-Phase PWM Signal with Dead Time (2)

**Figure 27.46** shows an example of outputting a three-phase PWM signal with dead time in which counter operation for which the clear timing is longer than cycle A is continued in cycle B, and PWM output A is cleared at the beginning of cycle B.

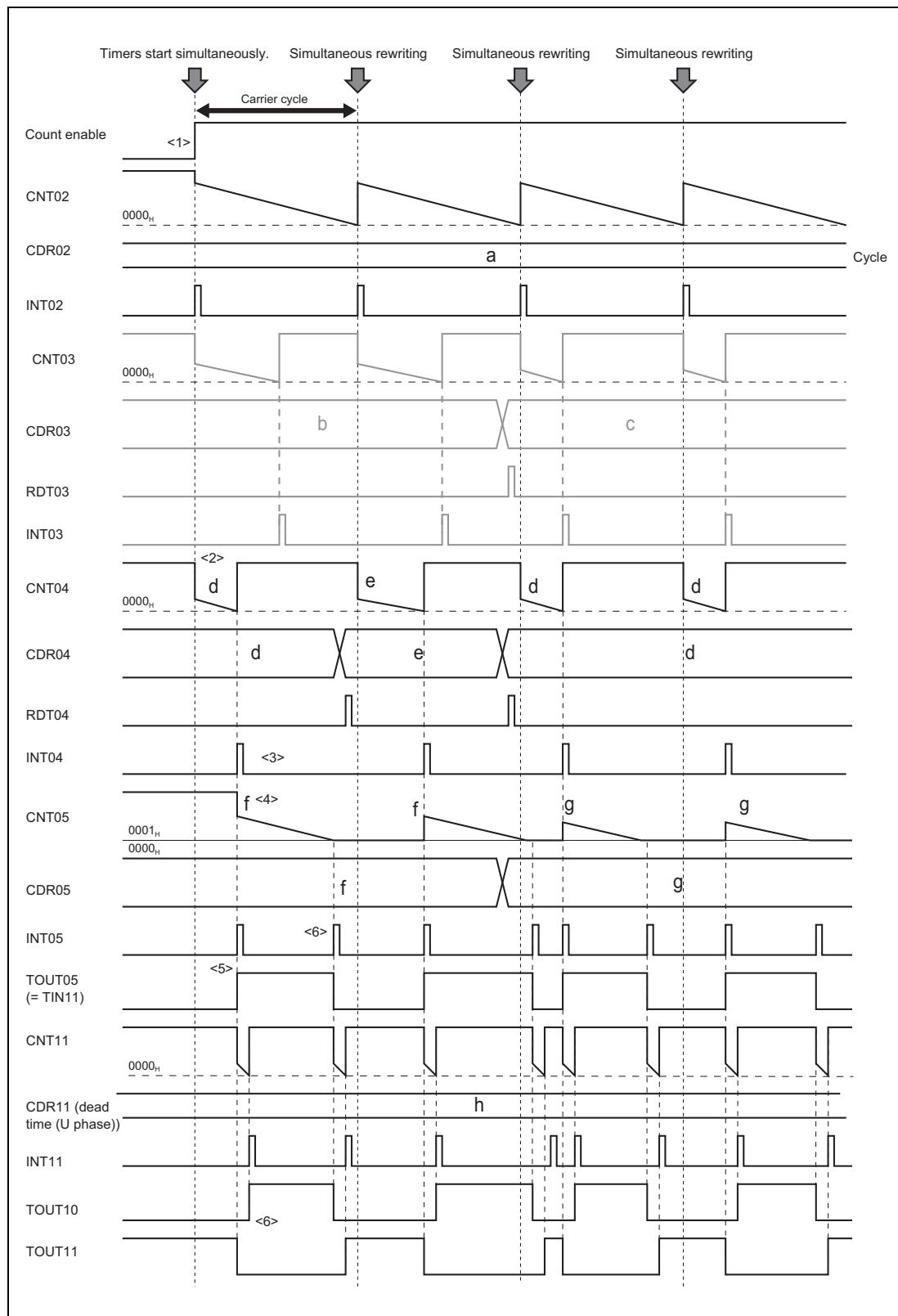
The output of PWM waveform A for cycle A is the same as the output of a delay pulse with dead time, but because the clear timing is used at the beginning of cycle B, the clear timing of PWM output B, which is supposed to be output during cycle B, does not occur.

In addition, the set timing of PWM waveform C for cycle C is ignored because a PWM waveform is already set. The result is that there is no PWM waveform change until the clear timing of cycle C, and a waveform that combines PWM waveform B and PWM waveform C is output.

In this way, it is possible to achieve freer PWM output timing when outputting a delay pulse with dead time than when outputting a three-phase PWM signal with dead time.

The peripheral interconnections provide a connection for using the PWM output timing of delay pulse output as input for one-phase PWM output.

**Figure 27.47** shows a timing chart for outputting a delay pulse with dead time.



**Figure 27.47** Output of a Delay Pulse with Dead Time

The output of a delay pulse with dead time shown in **Figure 27.47** is described below.

- (1) CH2 (the carrier cycle timer) and CH4 (the delay timing timer) are started simultaneously by starting timers simultaneously.  
CH5 (the PWM duty timer) and CH11 (the dead time timer) are also enabled, but no counting operations are performed until the edges of INT04, which indicates the count start timing for CH5, and TIN11, which indicates the count start timing for CH11, are detected.  
Because CH3 does not affect PWM output for this function, the channel is not described.
- (2) For CH4, when there is a CH2 underflow, the settings from CDR04 are reloaded to CNT04.
- (3) The CH4 underflow generates the delay timing signal (INT04).
- (4) When INT04 is generated, the settings from CDR05 are reloaded to CNT05, and then the CH5 (the PWM duty timer) operation starts.
- (5) At this time, INT05 is generated and the TOUT05 output level changes to the active level.
- (6) Due to the CH5 underflow, INT05 is generated again, and TOUT05 changes to the inactive level.  
TOUT05, which is changed by the CH4 and CH5 underflow, is supplied to the TIN11 input of one-phase PWM output.
- (7) During one-phase PWM output, a PWM waveform with dead time is output by detecting a TIN11 edge.

### 27.13.5 Setup Flow

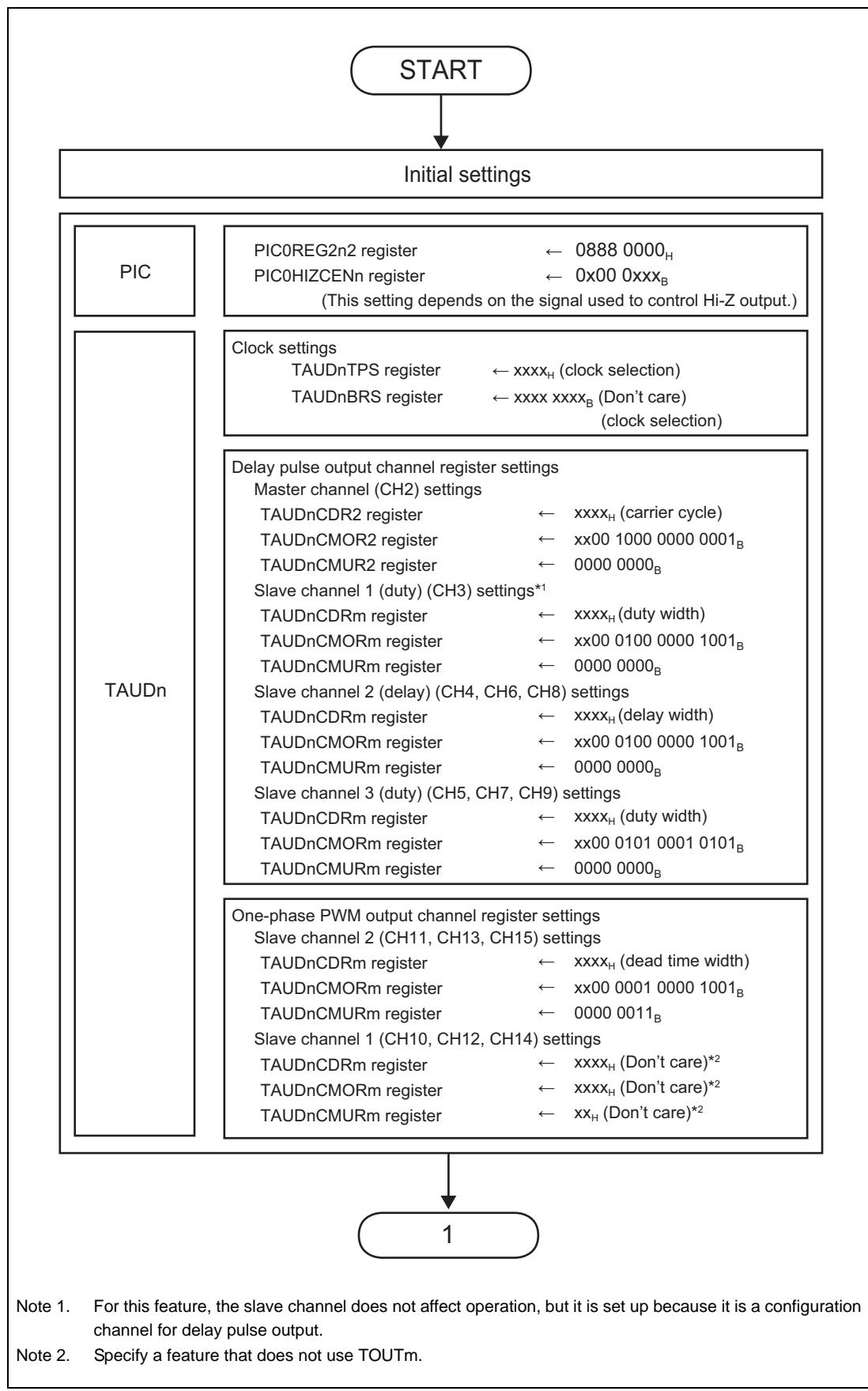


Figure 27.48 Setup Flow (Active High Example)

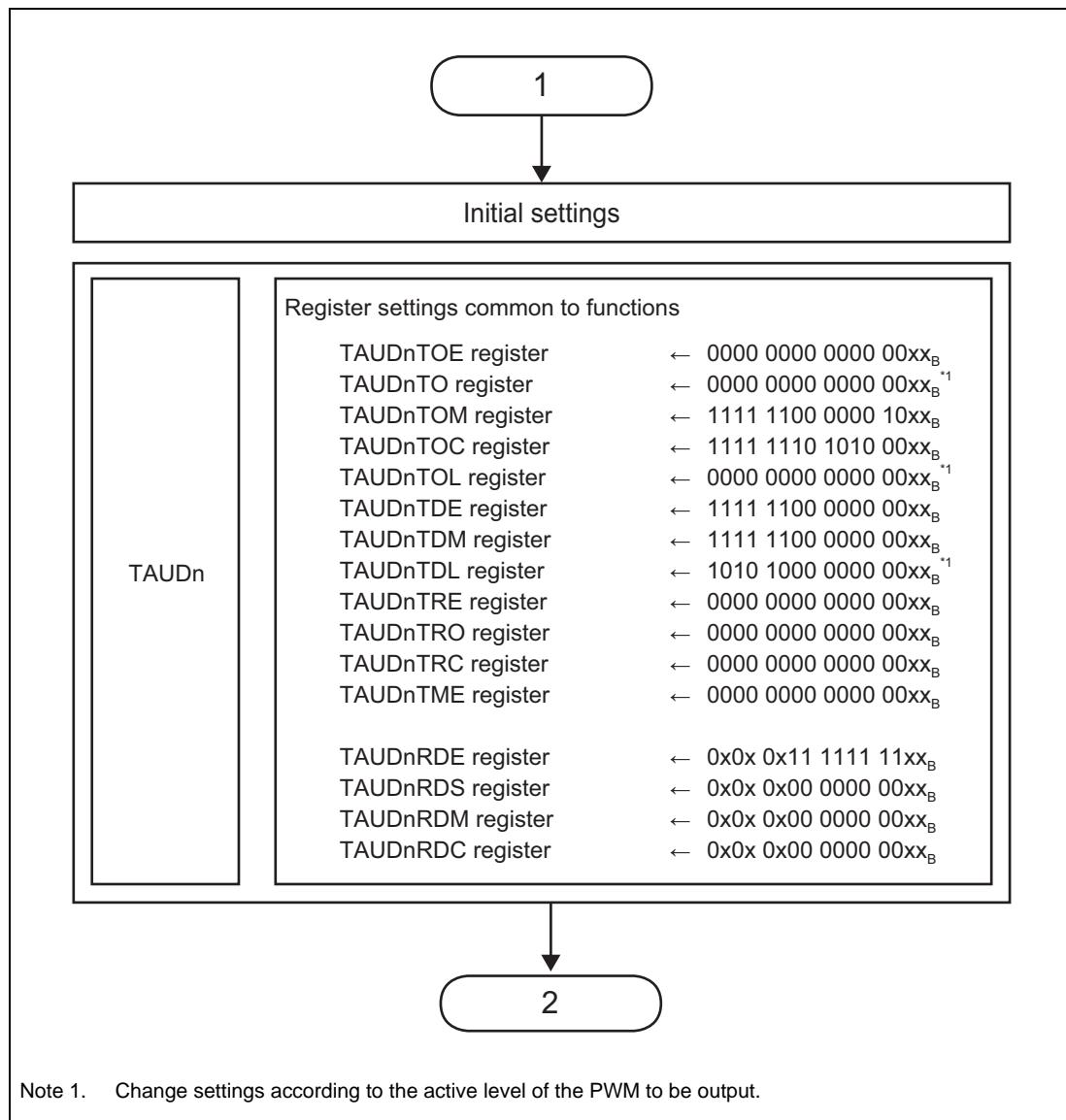


Figure 27.49 Setup Flow (Active High Example) (continued)

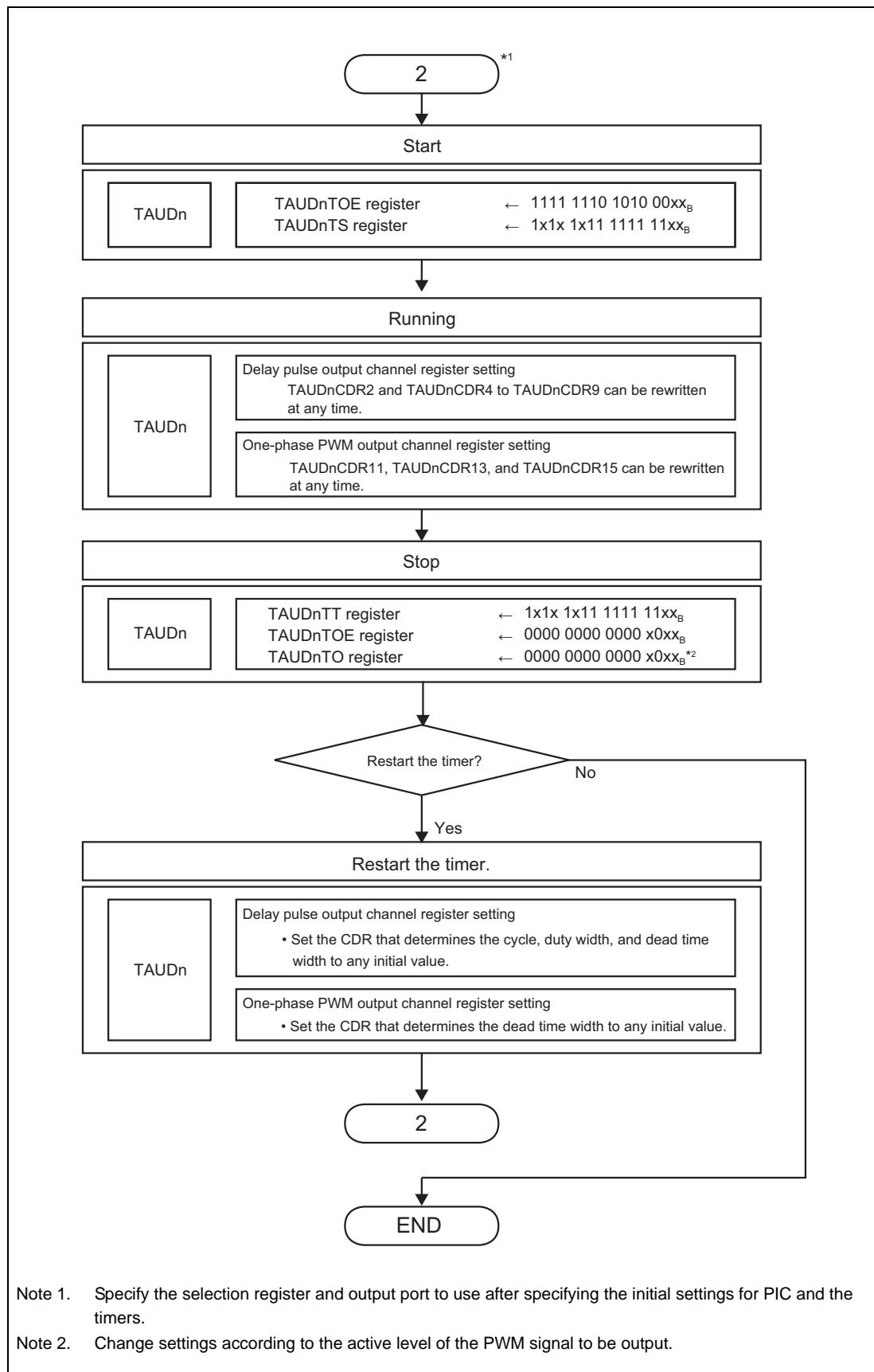


Figure 27.50 Setup Flow (Active High Example) (continued)

## 27.13.6 Setting Examples for Operation Functions

This section provides example settings for each register.

### 27.13.6.1 TAUDn Settings

**Table 27.70 TAUDn: CH2-related (Master Channel used to Output a Delay Pulse<sup>\*1</sup>)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	Output INTm at the start of operation.
	1, 0	TAUDnTIS[1:0]	00	
TAUDnCMUR2				

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

**Table 27.71 TAUDn: CH3-related (Slave Channel used to Output a Delay Pulse<sup>\*1\*2</sup>)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMOR3	15, 14	TAUDnCKS[1:0]	Don't care <sup>*3</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
	1, 0	TAUDnTIS[1:0]	00	
TAUDnCMUR3				

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Note 3. For this feature, the channel does not affect operation, but it is set up because it is a configuration channel for delay pulse output.

#### NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

**Table 27.72 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Delay Pulse<sup>\*1</sup>) (m = 4, 6, or 8)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the slave channel and master channel.

#### NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

**Table 27.73 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Delay Pulse<sup>\*1</sup>) (m = 5, 7, or 9)**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*2</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	101	Start trigger: INTm detection on an upper channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the slave channel and master channel.

#### NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

**Table 27.74 TAUDn: CH11, CH13, and CH15-related (One-Phase PWM Output) ( $m = 11, 13$ , or  $15$ )**

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care <sup>*1</sup>	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	Start trigger: Detection of a TINm-input valid edge
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

#### NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 23, Timer Array Unit D (TAUD)**.

CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output).

**Table 27.75 Common TAUDn Channel Settings (1/4)**

Register	Bit position	Bit name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9	TAUDnTOE09	0 1	Disable the timer. Enable the timer.
	8	TAUDnTOE08	0	This is fixed to 0 because TOUT08 is not used.
	7	TAUDnTOE07	0 1	Disable the timer. Enable the timer.
	6	TAUDnTOE06	0	This is fixed to 0 because TOUT06 is not used
	5	TAUDnTOE05	0 1	Disable the timer. Enable the timer.
	4	TAUDnTOE04	0	This is fixed to 0 because TOUT04 is not used
	3	TAUDnTOE03	0	This is fixed to 0 because TOUT03 is not used
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0 <sup>*1</sup>	Output a low-level signal to TOUT15 to TOUT10.
	9 to 2	TAUDnTO09 to TAUDnTO02	0	Output a low-level signal to TOUT09 to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	

**Table 27.75 Common TAUDn Channel Settings (2/4)**

<b>Register</b>	<b>Bit position</b>	<b>Bit name</b>	<b>Setting</b>	<b>Remark</b>
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	1	Synchronous operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	0	Independent operation mode
	3	TAUDnTOM03	1	Synchronous operation mode
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don't Care	
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1, 0, 1, 0, 1, 0	CH5, CH7, CH9: Operation mode 2 CH4, CH6, CH8: Operation mode 1
	3	TAUDnTOC03	0	Operation mode 1
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don't Care	
TAUDnTOL	15 to 10	TAUDnTOL15 to TAUDnTOL10	0* <sup>1</sup>	Positive logic output (active high)
	9 to 2	TAUDnTOL09 to TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don't Care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enable dead time control.* <sup>2</sup>
	9 to 2	TAUDnTDE09 to TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't Care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 2	TAUDnTDM09 to TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don't Care	
TAUDnTDL	15	TAUDnTDL15	1* <sup>1</sup>	Add dead time to the negative W phase period.
	14	TAUDnTDL14	0* <sup>1</sup>	Add dead time to the positive W phase period.
	13	TAUDnTDL13	1* <sup>1</sup>	Add dead time to the negative V phase period.
	12	TAUDnTDL12	0* <sup>1</sup>	Add dead time to the positive V phase period.
	11	TAUDnTDL11	1* <sup>1</sup>	Add dead time to the negative U phase period.
	10	TAUDnTDL10	0* <sup>1</sup>	Add dead time to the positive U phase period.
	9 to 2	TAUDnTDL09 to TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't Care	
TAUDnTRE	15 to 2	TAUDnTRE15 to TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't Care	

**Table 27.75 Common TAUDn Channel Settings (3/4)**

<b>Register</b>	<b>Bit position</b>	<b>Bit name</b>	<b>Setting</b>	<b>Remark</b>
TAUDnTRO	15 to 2	TAUDnTRO15 to TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't Care	
TAUDnTRC	15 to 2	TAUDnTRC15 to TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't Care	
TAUDnTME	15 to 2	TAUDnTME15 to TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't Care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't Care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't Care	
	9 to 2	TAUDnRDE09 to TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't Care	
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't Care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't Care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't Care	
	9 to 2	TAUDnRDS09 to TAUDnRDS02	0	Enable simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't Care	
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't Care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't Care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't Care	
	9 to 2	TAUDnRDM09 to TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	

**Table 27.75 Common TAUDn Channel Settings (4/4)**

Register	Bit position	Bit name	Setting	Remark
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't Care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't Care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't Care	
	9 to 2	TAUDnRDC09 to TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't Care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 23, Timer Array Unit D (TAUD)**.

### 27.13.6.2 Peripheral Interconnections Settings

**Table 27.76 Peripheral Interconnections Settings**

Register	Bit position	Bit name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Select the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Select the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Select the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	0	Select TAUDTTOUT9.
	3	PIC0REG2n203	0	Select TAUDTTOUT7.
	2	PIC0REG2n202	0	Select TAUDTTOUT5.

## Section 28 PWM Output/Diagnostic (PWM-Diag)

This section contains a generic description of the PWM output/diagnostic function (PWM-Diag).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the units constituting PWM-Diag.

### 28.1 Features of RH850/F1L PWM-Diag

#### 28.1.1 Number of Units and Channels

The PWM-Diag unit consists of a PWBA block for generating clock signals, PWGA blocks that generate PWM signals, and a PWSA block for generating triggers for A/D conversion. The numbers of individual units are listed below.

Each PWGA unit has one PWM channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

**Table 28.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
<b>PWBA</b>						
Number of Units	1					
Name	PWBAn (n = 0)					
<b>PWGA</b>						
Number of Units	13	24	24	48	64	72
Name	PWGAn (n = 0 to 12)	PWGAn (n = 0 to 23)	PWGAn (n = 0 to 23)	PWGAn (n = 0 to 47)	PWGAn (n = 0 to 63)	PWGAn (n = 0 to 71)
<b>PWSA</b>						
Number of Units	1					
Name	PWSAn (n = 0)					

**Table 28.2 Indices**

Index	Description
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”, for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock is identified by the index “m” (m = 0 to 3); for example, PWBAnBRSm indicates the PWMCLKm clock cycle configuration register.
x, y	An A/D converter configuration register number corresponding to a PWM-Diag channel is identified by the index “x, y”; for example, PWSAnPVCRx_y (x, y = 00_01, 02_03, ..., 70_71).
j	Registers storing trigger channel numbers (encoded value) from PWGAn are identified by the index “j”; for example, the PWSAnQUEj register (j = 0 to 7).
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGAk register (k = 0 to 2).

The following table shows values indicated by the indices of each product.

**Table 28.3 Indices of Products**

Indices of each product				
48 pins	64 pins, 80 pins	100 pins	144 pins	176 pins
x = 00, 02, ..., 12 y = 01, 03, ..., 13 <sup>*1</sup>	x = 00, 02, ..., 22 y = 01, 03, ..., 23	x = 00, 02, ..., 46 y = 01, 03, ..., 47	x = 00, 02, ..., 62, y = 01, 03, ..., 63	x = 00, 02, ..., 70 y = 01, 03, ..., 71
j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0	k = 0	k = 0, 1	k = 0, 1	k = 0 to 2

Note 1. Channels 0 to 12 are provided in PWM-Diag.

### 28.1.2 Register Base Address

PWM-Diag base addresses are listed in the following table.

PWM-Diag register addresses are given as offsets from the base addresses.

**Table 28.4 Register Base Addresses**

Base Address Name	Base Address
<PWBAn_base>	FFE7 2800 <sub>H</sub>
<PWGAn_base>	FFE7 1000 <sub>H</sub> + 40 <sub>H</sub> × n
<PWSAn_base>	FFE7 0000 <sub>H</sub>

### 28.1.3 Clock Supply

The PWM-Diag clock supply is shown in the following table.

**Table 28.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
PWBAn	PCLK	CKSCLK_IPERI2
PWGAn	PCLK	CKSCLK_IPERI2
PWSAn	PCLK	CKSCLK_IPERI2

### 28.1.4 Interrupt Requests

PWM-Diag interrupt requests are listed in the following table.

**Table 28.6 Interrupt Requests (1/2)**

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT0	PWGA0 interrupt	84	—
PWGA_INT1	PWGA1 interrupt	85	—
PWGA_INT2	PWGA2 interrupt	86	—
PWGA_INT3	PWGA3 interrupt	87	—
PWGA_INT4	PWGA4 interrupt	77	—
PWGA_INT5	PWGA5 interrupt	78	—
PWGA_INT6	PWGA6 interrupt	79	—
PWGA_INT7	PWGA7 interrupt	80	—
PWGA_INT8	PWGA8 interrupt	88	—
PWGA_INT9	PWGA9 interrupt	89	—
PWGA_INT10	PWGA10 interrupt	90	—
PWGA_INT11	PWGA11 interrupt	91	—
PWGA_INT12	PWGA12 interrupt	92	—
PWGA_INT13	PWGA13 interrupt	93	—
PWGA_INT14	PWGA14 interrupt	94	—
PWGA_INT15	PWGA15 interrupt	95	—
PWGA_INT16	PWGA16 interrupt	137	—
PWGA_INT17	PWGA17 interrupt	139	—
PWGA_INT18	PWGA18 interrupt	141	—
PWGA_INT19	PWGA19 interrupt	143	—
PWGA_INT20	PWGA20 interrupt	116	—
PWGA_INT21	PWGA21 interrupt	117	—
PWGA_INT22	PWGA22 interrupt	118	—
PWGA_INT23	PWGA23 interrupt	119	—
PWGA_INT24	PWGA24 interrupt	176	—
PWGA_INT25	PWGA25 interrupt	177	—
PWGA_INT26	PWGA26 interrupt	145	—
PWGA_INT27	PWGA27 interrupt	178	—
PWGA_INT28	PWGA28 interrupt	179	—
PWGA_INT29	PWGA29 interrupt	180	—
PWGA_INT30	PWGA30 interrupt	147	—
PWGA_INT31	PWGA31 interrupt	149	—
PWGA_INT32	PWGA32 interrupt	181	—
PWGA_INT33	PWGA33 interrupt	182	—
PWGA_INT34	PWGA34 interrupt	183	—
PWGA_INT35	PWGA35 interrupt	184	—
PWGA_INT36	PWGA36 interrupt	185	—
PWGA_INT37	PWGA37 interrupt	186	—
PWGA_INT38	PWGA38 interrupt	187	—
PWGA_INT39	PWGA39 interrupt	188	—

Table 28.6 Interrupt Requests (2/2)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT40	PWGA40 interrupt	189	—
PWGA_INT41	PWGA41 interrupt	190	—
PWGA_INT42	PWGA42 interrupt	191	—
PWGA_INT43	PWGA43 interrupt	192	—
PWGA_INT44	PWGA44 interrupt	193	—
PWGA_INT45	PWGA45 interrupt	194	—
PWGA_INT46	PWGA46 interrupt	195	—
PWGA_INT47	PWGA47 interrupt	196	—
PWGA_INT48	PWGA48 interrupt	232	—
PWGA_INT49	PWGA49 interrupt	233	—
PWGA_INT50	PWGA50 interrupt	234	—
PWGA_INT51	PWGA51 interrupt	235	—
PWGA_INT52	PWGA52 interrupt	236	—
PWGA_INT53	PWGA53 interrupt	237	—
PWGA_INT54	PWGA54 interrupt	238	—
PWGA_INT55	PWGA55 interrupt	239	—
PWGA_INT56	PWGA56 interrupt	240	—
PWGA_INT57	PWGA57 interrupt	241	—
PWGA_INT58	PWGA58 interrupt	242	—
PWGA_INT59	PWGA59 interrupt	243	—
PWGA_INT60	PWGA60 interrupt	244	—
PWGA_INT61	PWGA61 interrupt	245	—
PWGA_INT62	PWGA62 interrupt	246	—
PWGA_INT63	PWGA63 interrupt	247	—
PWGA_INT64	PWGA64 interrupt	269	—
PWGA_INT65	PWGA65 interrupt	270	—
PWGA_INT66	PWGA66 interrupt	271	—
PWGA_INT67	PWGA67 interrupt	272	—
PWGA_INT68	PWGA68 interrupt	273	—
PWGA_INT69	PWGA69 interrupt	274	—
PWGA_INT70	PWGA70 interrupt	275	—
PWGA_INT71	PWGA71 interrupt	276	—
PWSA_INT_QFULL	PWSA queue full interrupt	83	—

### 28.1.5 Reset Sources

PWM-Diag reset sources are listed in the following table. The individual PWM-Diag units are initialized by these reset sources.

**Table 28.7 Reset Sources**

Unit Name	Reset Source
PWBAn	All reset sources (ISORES)
PWGAn	
PWSAn	

### 28.1.6 External Input/Output Signals

External input/output signals of the PWM-Diag are listed below.

**Table 28.8 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal
PWGA_TOUTn (Unit: PWGA)	PWGA unit n output	PWGAnO

**CAUTION**

When port P0\_0 is used as PWGA10O, the P0\_0 pin outputs a low-level RESETOUT signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2.11.1.1, P0\_0: RESETOUT**.

### 28.1.7 Internal Signals

The I/O signals for connecting two PWM-Diag channels or a PWM-Diag and another function are listed below.

**Table 28.9 Internal Output Signals**

Unit Signal Name	Outline	Connected to
<b>PWBA0</b>		
PWMCLK0	PWGA count clock 0	PWGAn
PWMCLK1	PWGA count clock 1	PWGAn
PWMCLK2	PWGA count clock 2	PWGAn
PWMCLK3	PWGA count clock 3	PWGAn
<b>PWGAn</b>		
PWGA_TRGOUT	PWGAn trigger	PWSA0
<b>PWSA0</b>		
PWSA_ADTRG[1:0]	A/D converter unit select signal	ADCA0, ADCA1
PWSA_PVCR_VALUE[11:0]	A/D converter control signal	ADCA0, ADCA1
<b>ADCAn</b>		
ADC_CONV_ENDn	A/D conversion completion signal	PWSA0

## 28.1.8 Functional Overview

This function is comprised of four types of units: clock divider (PWBA), PWM generator (PWGA), A/D conversion trigger select function (PWSA), and A/D converter (ADCA).

### PWBA

- Clock divider

PWBA generates a PWMCLKm count clock signal by frequency division of PCLK and supplies it to the PWM generator PWGA.

The cycle of the PWMCLKm count clock signal can be calculated from the setting of the PWBAAnBRSm register by the equation below.

$$\text{PWMCLKm count clock cycle} = (\text{PWBAAnBRSm value} \times 2) \times \text{PCLK cycle}$$

In addition, PWBA can control operation when the on-chip debugger is in use by using the PWBAAnEMU register.

### PWGA

PWGA outputs PWM waveforms and A/D conversion trigger to PWSA by using the clock PWMCLKm input from PWBA.

- PWM waveform output PWGA\_TOUTn

This generator outputs PWM waveforms from the PWGA\_TOUTn pin. The PWM cycle is the full count cycle of the PWGAnCNT register (12-bit free-running counter). Set the high-level period of PWM output in the PWGAnCSDR and PWGAnCRDR registers.

The PWM waveform cycle and duty can be calculated by the equations below.

$$\begin{aligned}\text{PWM waveform cycle} &= \text{PWGAnCNT} \text{ (12-bit full count: } \text{FFF}_H + 1) \\ &\quad \times \text{Count clock cycle} \\ &= 4096 \times \text{PWMCLKm count clock cycle}\end{aligned}$$

When PWGAnCRDR[11:0] > PWGAnCSDR[11:0],

High-level period of PWM waveform =

$$\begin{aligned}&(\text{PWGAnCRDR register value} - \text{PWGAnCSDR register value}) \\ &\quad \times \text{PWMCLKm count clock cycle}\end{aligned}$$

$$\text{PWM waveform duty (\%)} = \frac{\text{High-level period of PWM waveform}}{\text{PWM waveform cycle}} \times 100 = \frac{(\text{PWGAnCRDR register value} - \text{PWGAnCSDR register value})}{4096 \times 100}$$

Note that the PWM output is fixed to the low level when the PWGAnCRDR register value is equal to the PWGAnCSDR register value.

When 1xxx<sub>H</sub> is set in the PWGAnCRDR register (i.e. bit 12 is set to 1), the PWM output is fixed to the high level.

- A/D conversion trigger output PWGA\_TRGOUTn

The A/D conversion trigger signal PWGA\_TRGOUTn for PWSA is generated when the PWGAnCTDR register value and the PWGAnCNT register value match while the PWM output PWGA\_TOUTn is at the high level.

The timing can be calculated by the equation below.

$$\begin{aligned}\text{A/D conversion trigger signal generation timing} &= \text{PWGAnCTDR register value} \\ &\quad \times \text{PWMCLKm count clock cycle}\end{aligned}$$

- PWGA interrupt request signal PWGA\_INTn

PWGA generates the interrupt request signal PWGA\_INTn at the falling edge of the PWM output PWGA\_TOUTn.

When the PWM output is fixed to the low level, PWGA\_INTn is generated when the PWGAnCRDR register value and the PWGAnCNT register value (free-running counter value) match; when the PWM output is fixed to the high level, PWGA\_INTn is generated at the timing of overflow of the PWGAnCNT register.

### **PWSA**

PWSA transmits the required setting information to the A/D converter and outputs the A/D conversion start trigger, based on the A/D conversion trigger signal PWGA\_TRGOUTn from the PWM generator (PWGA).

- A/D conversion control by PWSA

PWSA outputs the information required for the A/D conversion, which is set in the corresponding PWSAnPVCRx\_y register for the channel number of the trigger input from PWGAn, (i.e., information on ADC physical channel, external MPX control, and error detection level selection) to the A/D converter.

At the same timing, A/D conversion trigger (PWSA\_ADTRG) is output to ADCA0 or ADCA1. (A maximum of eight input trigger signal PWGA\_TRGOUTn data received during A/D conversion are stored and kept in PWSAnQUE.)

The setting information to be output to the A/D converter is kept until the next trigger is generated.

When the A/D conversion triggered by the PWM-Diag function is completed in the A/D converter, PWSA triggers the next A/D conversion based on the data stored in the PWSAnQUE register.

- Queuing of A/D conversion triggers from PWGA

The A/D conversion trigger signal (PWGA\_TRGOUTn) input from PWGAn is stored in the PWSAnQUEj register as a channel number. The PWSAnQUEj register stores a maximum of eight channel numbers of the A/D conversion trigger signal PWGA\_TRGOUTn received during A/D conversion in a queue structure.

A PWSA queue full interrupt occurs in the following states, when the queue of the PWSAnQUEj register becomes full:

- A trigger number is written to PWSAnQUE7
- A trigger number has already been written to PWSAnQUE7 and cannot be written when PWGA\_TRGOUTn is input.

### **ADCA**

A/D conversion is executed upon receipt of information required for A/D conversion and A/D conversion trigger from PWSA.

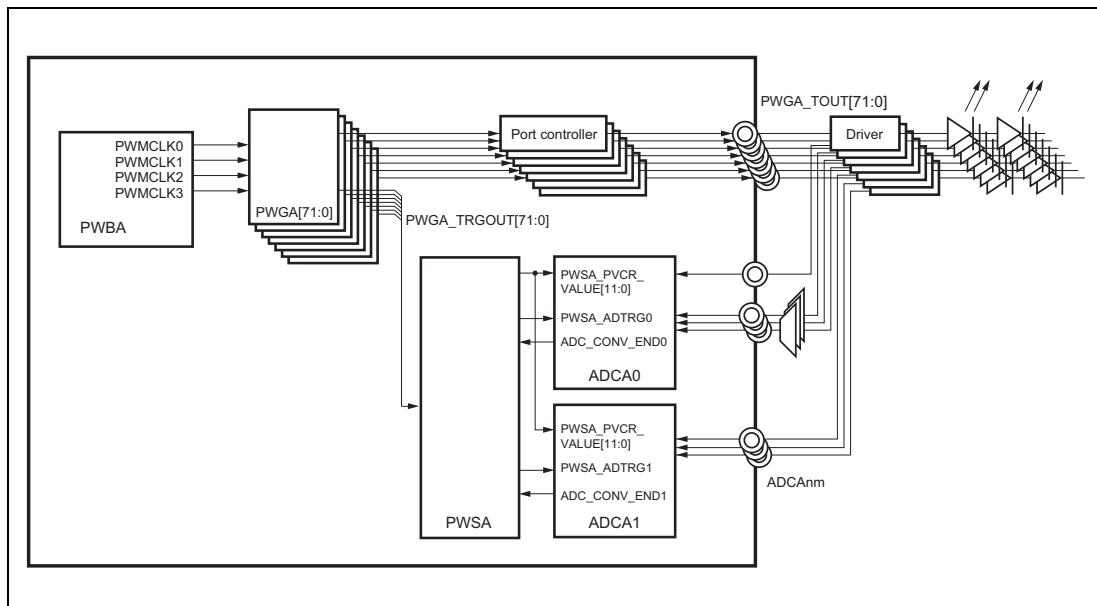
A/D conversion is executed using the PWM-Diag-dedicated scan group; on completion of the A/D conversion, it is reported to the PWSA.

For the basic operation of the A/D converter, see **Section 29, A/D Converter (ADCA)**.

For the A/D converter operation with the PWM-Diag function, see **Section 29.4.7.1, A/D Conversion with PWM-Diag Enabled**.

### 28.1.9 Block Diagram

The following figure shows an example of connecting the LED control circuit combining the PWM-Diag and the A/D converter.



**Figure 28.1 Example of Connecting the LED Control Circuit using the PWM-Diag and the A/D Converter**

## 28.2 Registers

### 28.2.1 List of Registers

PWM Output/Diagnostic registers are listed in the following table.

<PWBAn\_base>, <PWSAn\_base>, and <PWGAn\_base> are defined in **Section 28.1.2, Register Base Address**.

**Table 28.10 List of PWM Output/Diagnostic Registers**

Module name	Register name	Symbol	Address
PWBAn	PWMCLKm cycle configuration register	PWBAnBRSm	<PWBAn_base> + 0004 <sub>H</sub> × m
	PWMCLKm enable status register	PWBAnTE	<PWBAn_base> + 0010 <sub>H</sub>
	PWMCLKm start trigger register	PWBAnTS	<PWBAn_base> + 0014 <sub>H</sub>
	PWMCLKm stop trigger register	PWBAnTT	<PWBAn_base> + 0018 <sub>H</sub>
	PWBA emulation register	PWBAnEMU	<PWBAn_base> + 001C <sub>H</sub>
PWGAn	PWM output set condition register	PWGAnCSDR	<PWGAn_base> + 0000 <sub>H</sub>
	PWM output reset condition register	PWGAnCRDR	<PWGAn_base> + 0004 <sub>H</sub>
	PWGA_TRGOUTn generation condition register	PWGAnCTDR	<PWGAn_base> + 0008 <sub>H</sub>
	Buffer register reload trigger register	PWGAnRDT	<PWGAn_base> + 000C <sub>H</sub>
	Buffer register reload status register	PWGAnRSF	<PWGAn_base> + 0010 <sub>H</sub>
	PWM cycle count register	PWGAnCNT	<PWGAn_base> + 0014 <sub>H</sub>
	PWGA control register	PWGAnCTL	<PWGAn_base> + 0020 <sub>H</sub>
	PWGAnCSDR buffer register	PWGAnCSBR	<PWGAn_base> + 0024 <sub>H</sub>
	PWGAnCRDR buffer register	PWGAnCRBR	<PWGAn_base> + 0028 <sub>H</sub>
	PWGAnCTDR buffer register	PWGAnCTBR	<PWGAn_base> + 002C <sub>H</sub>
—	PWGA synchronous trigger register	SLPWGAK	FFBC 1000 <sub>H</sub> + k × 4 <sub>H</sub>
PWSAn	PWSA control register	PWSAnCTL	<PWSAn_base> + 0000 <sub>H</sub>
	Trigger queue status register	PWSAnSTR	<PWSAn_base> + 0004 <sub>H</sub>
	Trigger queue status clear register	PWSAnSTC	<PWSAn_base> + 0008 <sub>H</sub>
	Trigger queue register	PWSAnQUEj	<PWSAn_base> + 0020 <sub>H</sub> + j × 4 <sub>H</sub>
	PWM-Diag mode A/D setting register	PWSAnPVCRx_y	<PWSAn_base> + 0040 <sub>H</sub> + x × 2 <sub>H</sub>
	PWSA emulation control register	PWSAnEMU	<PWSAn_base> + 000C <sub>H</sub>

### 28.2.1.1 PWBAnBRSm Register

This register sets the clock cycle of PWMCLKm.

**Access:** This register can be read or written in 16-bit units.

**Address:** <PWBAn\_base> + 0004<sub>H</sub> × m

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	—	—	—	—	—	PWBAnBRSm[10:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				

**Table 28.11 PWBAnBRSm Register Contents**

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	PWBAnBRSm [10:0]	Register for setting the clock cycle of PWMCLKm. – PWBAnBRSm = 0: PWMCLKm = PCLK – PWBAnBRSm = 1: PWMCLKm = PCLK / 2 × 1 – PWBAnBRSm = 2: PWMCLKm = PCLK / 2 × 2 ... – PWBAnBRSm = n: PWMCLKm = PCLK / 2 × n (n = 1 to 2047) These bits can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).

### 28.2.1.2 PWBAnTE Register

This is a status register that indicates the output status of PWMCLKm (m = 0 to 3).

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <PWBAn\_base> + 0010H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTE3	PWBAnTE2	PWBAnTE1	PWBAnTE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 28.12 PWBAnTE Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	PWBAnTE3	A status flag indicating the operation status of PWMCLK3 0: Not operating 1: Operating
2	PWBAnTE2	A status flag indicating the operation status of PWMCLK2 0: Not operating 1: Operating
1	PWBAnTE1	A status flag indicating the operation status of PWMCLK1 0: Not operating 1: Operating
0	PWBAnTE0	A status flag indicating the operation status of PWMCLK0 0: Not operating 1: Operating

### 28.2.1.3 PWBAnTS Register

This register is a start trigger register for PWMCLKm (m = 0 to 3).

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <PWBAn\_base> + 0014H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTS3	PWBAnTS2	PWBAnTS1	PWBAnTS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

**Table 28.13 PWBAnTS Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTS3	Start Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Starts the output of PWMCLK3.
2	PWBAnTS2	Start Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Starts the output of PWMCLK2.
1	PWBAnTS1	Start Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Starts the output of PWMCLK1.
0	PWBAnTS0	Start Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Starts the output of PWMCLK0.

### 28.2.1.4 PWBAnTT Register

This register is a stop trigger register for PWMCLKm (m = 0 to 3).

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <PWBAn\_base> + 0018H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTT3	PWBAnTT2	PWBAnTT1	PWBAnTT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

**Table 28.14 PWBAnTT Register Contents**

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTT3	Stop Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Stops the output of PWMCLK3.
2	PWBAnTT2	Stop Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Stops the output of PWMCLK2.
1	PWBAnTT1	Stop Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Stops the output of PWMCLK1.
0	PWBAnTT0	Stop Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Stops the output of PWMCLK0.

### 28.2.1.5 PWBAAnEMU Register

This register sets the operation during emulation.

**Access:** This register can be read or written in 8-bit units.

**Address:** <PWBAAn\_base> + 001CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	PWBAAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

**Table 28.15 PWBAAnEMU Register Contents**

Bit Position	Bit Name	Function
7	PWBAAnSVSDIS	(When the EPC.SVSTOP bit = 0) The count clock is provided continuously when the debugger is controlling the microcontroller (by using break points, etc.), regardless of the value of this bit (1 or 0).  (When the EPC.SVSTOP bit = 1) 0: The count clock is stopped when the debugger is controlling the microcontroller (by using break points, etc.). 1: The count clock is provided continuously when the debugger is controlling the microcontroller (by using break points, etc.). This bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAAnTE.PWBATEm = 0).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

### 28.2.1.6 PWGAnCTL — PWGA Control Register

PWGAnCTL is used to select the count clock from PWBA.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<PWGAn_base> + 0020H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnCKS[1:0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 28.16 PWGAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PWGAnCKS [1:0]	Count Clock Enable Input PWMCLK3 to PWMCLK0 Select 00: Uses PWMCLK0 as count clock 01: Uses PWMCLK1 as count clock 10: Uses PWMCLK2 as count clock 11: Uses PWMCLK3 as count clock These bits can only be rewritten when the PWGAn operation is stopped (SLPWGAk.SLPWGA[31:0] = 0).

### 28.2.1.7 PWGAnCNT — PWM Cycle Count Register

This is a count register.

<b>Access:</b>	This register is a read-only register that can be read in 16-bit units.															
<b>Address:</b>	<PWGAn_base> + 0014H															
<b>Value after reset:</b>	0FFFH															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—												PWGAnCNT[11:0]
Value after reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.17 PWGAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCNT [11:0]	12-bit counter value

### 28.2.1.8 PWGAnCSDR — PWM Output Set Condition Register

This register sets the setting condition for PWGA\_TOUTn output.

**Access:** This register can be read or written in 16-bit units.

**Address:** <PWGAn\_base> + 0000H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	—	—	—	—	PWGAnCSDR[11:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

**Table 28.18 PWGAnCSDR Register Contents**

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAnCSDR [11:0]	These bits set the setting condition for PWM output. The set value is reflected to the PWGAnCSBR register at the start of PWGAn operation (SLPWGAK.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

### 28.2.1.9 PWGAnCRDR — PWM Output Reset Condition Register

This register sets the reset condition for PWGA\_TOUTn output.

**Access:** This register can be read or written in 16-bit units.

**Address:** <PWGAn\_base> + 0004H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	—	—	—	PWGAnCRDR[12:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

**Table 28.19 PWGAnCRDR Register Contents**

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 0	PWGAnCRDR [12:0]	These bits set the reset condition for PWM output. The set value is reflected to the PWGAnCRBR register at the start of PWGAn operation (SLPWGAK.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

### 28.2.1.10 PWGAnCTDR — PWGA\_TRGOUTn Generation Condition Register

This register sets the generation condition for PWGA\_TRGOUTn.

**Access:** This register can be read or written in 16-bit units.

**Address:** <PWGAn\_base> + 0008H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	—	—	—	—	PWGAnCTDR[11:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			

**Table 28.20 PWGAnCTDR Register Contents**

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAnCTDR [11:0]	These bits set the A/D conversion trigger generation condition for PWSAn. The set value is reflected to the PWGAnCTBR register at the start of PWGAn operation (SLPWGAK.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

### 28.2.1.11 PWGAnCSBR — PWGAnCSDR Buffer Register

This is a buffer register for the PWGAnCSDR register.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <PWGAn\_base> + 0024H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	—	—	—	—	PWGAnCSBR[11:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R			

**Table 28.21 PWGAnCSBR Register Contents**

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCSBR [11:0]	The set value is reflected to the PWGAnCSDR register at the start of PWGAn operation (SLPWGAK.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven high.

### 28.2.1.12 PWGAnCRBR — PWGAnCRDR Buffer Register

This is a buffer register for the PWGA\_TOUTn reset condition.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <PWGAn\_base> + 0028H

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	PWGAnCRBR[12:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

**Table 28.22 PWGAnCRBR Register Contents**

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12 to 0	PWGAnCRBR [12:0]	The set value is reflected to the PWGAnCRDR register at the start of PWGAn operation (SLPWGAK.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven low.

### 28.2.1.13 PWGAnCTBR — PWGAnCTDR Buffer Register

This is a buffer register for the PWGA\_TRGOUTn generation condition.

**Access:** This register is a read-only register that can be read in 16-bit units.

**Address:** <PWGAn\_base> + 002CH

**Value after reset:** 0000H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	—	—	—	—	PWGAnCTBR[11:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R			

**Table 28.23 PWGAnCTBR Register Contents**

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCTBR [11:0]	The set value is reflected to the PWGAnCTDR register at the start of PWGAn operation (SLPWGAK.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, a trigger is transmitted to PWSAn.

### 28.2.1.14 PWGAnRSF — Buffer Register Reload Status Register

This register is a status register for simultaneous rewrite control.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** <PWGAn\_base> + 0010H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnRSF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 28.24 PWGAnRSF Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	PWGAnRSF	Simultaneous Rewrite Control Status 0: Simultaneous rewrite is enabled. This value indicates the completion of simultaneous rewrite after the generation of a simultaneous rewrite trigger signal. 1: Simultaneous rewrite is in progress. This value indicates the waiting state for completion.

### 28.2.1.15 PWGAnRDT — Buffer Register Reload Trigger Register

This is a simultaneous rewrite request trigger register.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** <PWGAn\_base> + 000CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnRDT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

**Table 28.25 PWGAnRDT Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	PWGAnRDT	Simultaneous Rewrite Request Trigger 0: Writing 0 has no effect. 1: Triggers the simultaneous rewrite request for the compare registers (PWGAnCSDR, PWGAnCRDR, and PWGAnCTDR), and sets PWGAnRSF.PWGAnRSF to 1.

### 28.2.1.16 SLPWGAk — PWGA Synchronous Trigger Register ( $k = 0$ to 2)

This register triggers start and stop for multiple channels simultaneously.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFBC 1000<sub>H</sub> +  $k \times 4<sub>H</sub>$

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLPWGA[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLPWGA[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 28.26 SLPWGAk Register Contents**

Bit Position	Bit Name	Function
31 to 0	SLPWGA [31:0]	<p>Trigger start and stop to multiple channels simultaneously.            0: Stops the corresponding channels.            1: Starts the corresponding channels.</p> <p>The bits correspond to the following channels.            SLPWGA0.SLPWGA[31:0]: PWGA31 - PWGA0            SLPWGA1.SLPWGA[31:0]: PWGA63 - PWGA32            SLPWGA2.SLPWGA[7:0]: PWGA71 - PWGA64</p>

### 28.2.1.17 PWSAnCTL Register

This register is used to control operations of PWSA.

<b>Access:</b>	This register can be read or written in 8-bit units.							
<b>Address:</b>	<PWSAn_base> + 0000H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWSAnENBL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

**Table 28.27 PWSAnCTL Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSAnENBL	Operation Permission Control 0: Operation is prohibited (initial state). Writing 0 initializes PWSAnSTR and PWSAnQUEj. 1: Operation is enabled.

### 28.2.1.18 PWSAnSTR Register

This is a status register that indicates whether the number of a channel for which an A/D conversion trigger has been generated is stored in a PWSAnQUEj register.

<b>Access:</b>	This register is a read-only register that can be read in 8-bit units.							
<b>Address:</b>	<PWSAn_base> + 0004H							
<b>Value after reset:</b>	00H							
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSAnQFL	PWSAnQNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 28.28 PWSAnSTR Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	PWSAnQFL	Indicates the queuing state of the A/D conversion trigger. 0: Some PWSAnQUEj registers do not store a channel number. 1: All of the PWSAnQUEj registers store a channel number.
0	PWSAnQNE	Bit indicating that there is a trigger in the trigger queue 0: A channel number is not stored in a PWSAnQUEj register, or A/D conversion is in progress while only PWSAnQUE0 stores a channel number. 1: The number of the channel waiting for conversion is stored in j = 1 and subsequent PWSAnQUEj registers.

### 28.2.1.19 PWSAnSTC Register

This register clears the status of the PWSAnSTR register.

	<b>Access:</b>	This register is a write-only register that can be written in 8-bit units.						
	<b>Address:</b>	<PWSAn_base> + 0008H						
	<b>Value after reset:</b>	00H						
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSAnCLFL	PWSAnCLNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

**Table 28.29 PWSAnSTC Register Contents**

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	PWSAnCLFL	PWSAnQFL Clear 0: PWSAnQFL retains the status (Writing 0 has no effect). 1: PWSAnQFL is cleared to 0.
0	PWSAnCLNE	PWSAnQNE Clear 0: PWSAnQNE retains the status (Writing 0 has no effect). 1: PWSAnQNE is cleared to 0.

### 28.2.1.20 PWSAnQUEj (j = 0 to 7) Register

This register stores the channel number that received the trigger from PWGAn.

	<b>Access:</b>	This register is a read-only register that can be read in 8-bit units.						
	<b>Address:</b>	<PWSAn_base> + 0020H + j × 4H						
	<b>Value after reset:</b>	RH850/F1L 48, 64, 80 and 100 pin products: 3FH RH850/F1L 144 and 176 pin products: 7FH						
Bit	7	6	5	4	3	2	1	0
	—	PWSAnQUEj[6:0]						
Value after reset	0	0/1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R

**Table 28.30 PWSAnQUEj Register Contents**

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	PWSAnQUEj [6:0]	These bits hold the channel number (0 to 71) of the PWGA for which a trigger was generated in order from PWSAnQUE0 to PWSAnQUE7. After the A/D conversion of PWSAnQUE0 is completed, the values in PWSAnQUE1 to PWSAnQUE7 shift to PWSAnQUE0 to PWSAnQUE6.

#### NOTE

If a trigger occurs simultaneously for multiple channels, the trigger with the smaller channel number has priority.

### 28.2.1.21 PWSAnPVCRx\_y (x = 00, 02, 04 ... 70, y = 01, 03, 05 ... 71) Register

This register is used to set the corresponding A/D converter for each channel.

Two consecutive channels are set such as PWSA0PVCR02\_03, and the 16 higher-order bits of each register correspond to an odd-numbered channel while the 16 lower-order bits correspond to an even-numbered channel.

At the generation of a trigger, the set value is transmitted to the ADCAnPWDVCR register of the A/D converter.

For the ADCAnPWDVCR register, see **Section 29, A/D Converter (ADCA)**.

**Access:** This register can be read or written in 32-bit units.

**Address:** <PWSAn\_base> + 0040H + x × 2H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	PWSAnSLADy	PWSAnVRDTy [27]	PWSAnVRDTy[26:24]			PWSAnVRDTy [23:22]	PWSAnVRDTy[21:16]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWSAnSLADx	PWSAnVRDTx [11]	PWSAnVRDTx[10:8]			PWSAnVRDTx [7:6]	PWSAnVRDTx[5:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 28.31 PWSAnPVCRx\_y Register Contents (1/2)**

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	PWSAnSLADy	RH850/F1L 48, 64, 80 and 100 pin products: ADCA Select (odd-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.
27	PWSAnVRDTy [27]	This bit indicates the set value of the ADCAnPWDVCR.MPXE bit (odd-numbered channel).
26 to 24	PWSAnVRDTy [26:24]	These bits indicate the set value of the ADCAnPWDVCR.MPXV[2:0] bits (odd-numbered channel).
23, 22	PWSAnVRDTy [23:22]	These bits indicate the set value of the ADCAnPWDVCR.ULS[1:0] bits (odd-numbered channel).
21 to 16	PWSAnVRDTy [21:16]	These bits indicate the set value of the ADCAnPWDVCR.GCTRL[5:0] bits (odd-numbered channel).
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	PWSAnSLADx	RH850/F1L 48, 64, 80 and 100 pin products: When writing, write the value after reset.  RH850/F1L 144 and 176 pin products: ADCA Select (even-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.

**Table 28.31 PWSAnPVCRx\_y Register Contents (2/2)**

Bit Position	Bit Name	Function
11	PWSAnVRDTx [11]	This bit indicates the set value of the ADCAnPWDVCR.MPXE bit. (even-numbered channel)
10 to 8	PWSAnVRDTx [10:8]	These bits indicate the set value of the ADCAnPWDVCR.MPXV[2:0] bits. (even-numbered channel)
7, 6	PWSAnVRDTx [7:6]	These bits indicate the set value of the ADCAnPWDVCR.ULS[1:0] bits. (even-numbered channel)
5 to 0	PWSAnVRDTx [5:0]	These bits indicate the set value of the ADCAnPWDVCR.GCTRL[5:0] bits. (even-numbered channel)

**CAUTION**

For the RH850/F1L 48 pin products, set the initial value to the 16 higher-order bits of PWSA0PVCR12\_13.

**28.2.1.22 PWSAnEMU — Emulation Control Register**

This register is used to set the operation for emulation.

**Access:** This register can be read or written in 8-bit units.

**Address:** <PWSAn\_base> + 000C<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWSAnSVSDIS
Value after reset	0	0	0	0	0	0	0	0

R/W	0	0	0	0	0	0	0	R/W
	R	R	R	R	R	R	R	R/W

**Table 28.32 PWSAnEMU Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSAnSVSDIS	(When the EPC.SVSTOP bit = 0) The operation continues when the debugger is controlling the microcontroller (by using break points, etc.), regardless of the value of this bit (1 or 0).  (When the EPC.SVSTOP bit = 1) 0: When the debugger is controlling the microcontroller (by using break points, etc.); – The output state to A/D is retained, the ADC_CONV_ENDn input at a break point is internally retained, and PWSAnQUEj is updated after break release. – The PWGA_TRGOUT input is accepted even at a break, and PWSA_INT_QFULL is also output. – Reading and writing to the register is possible.  1: The operation continues when the debugger is controlling the microcontroller (by using break points, etc.).  The above bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0), the operation of all channels PWGAn has stopped (SLPWGAK.SLPWGA), and no trigger has been generated from any of the channels PWGAn (PWSAnQUE0 is the value after reset).

## 28.3 Operating Procedure

Procedures for setting when starting and stopping operation of PWM-Diag are illustrated below.

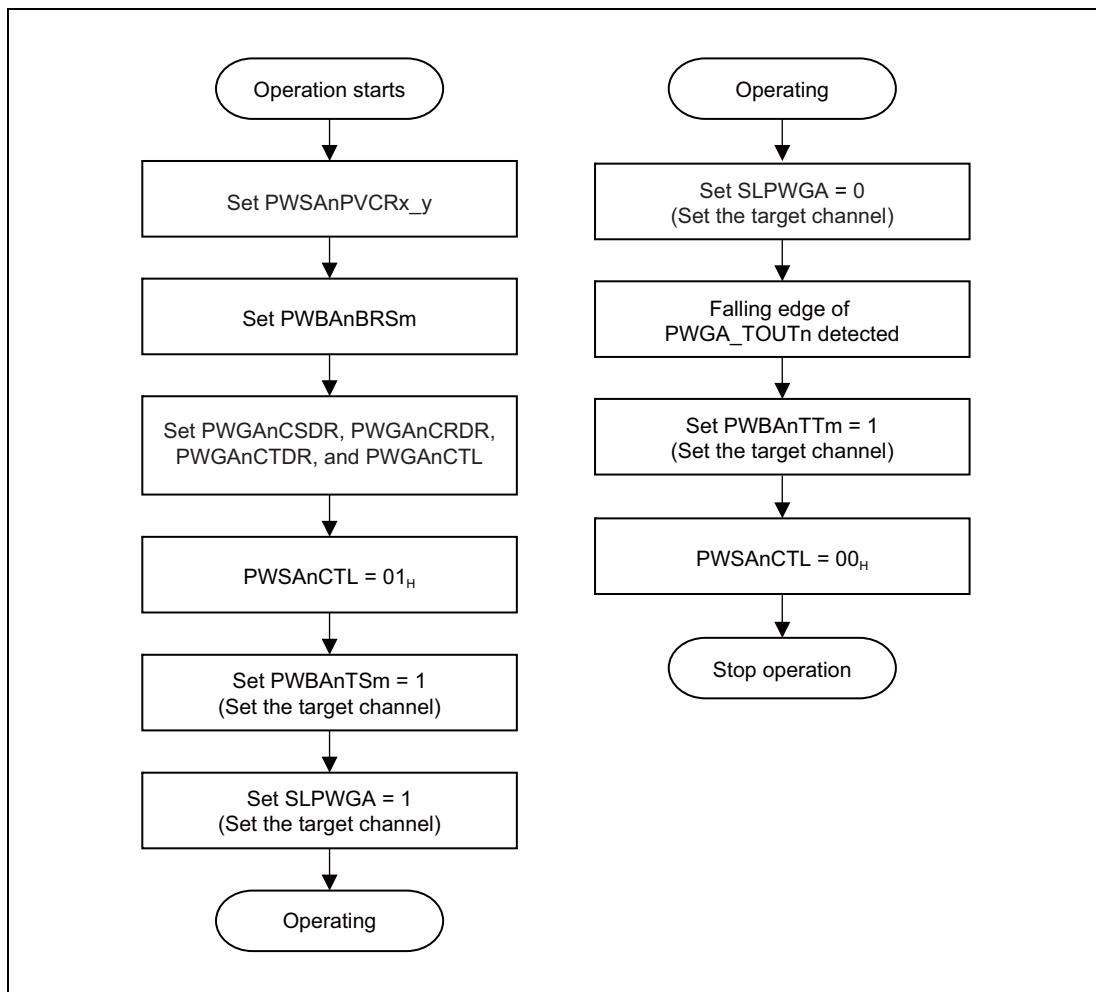


Figure 28.2 PWM-Diag Operating Procedure

Procedures for simultaneous rewrite of PWGA are illustrated below.

The described term “compare register” indicates PWGAnCSDR, PWGAnCRDR, or PWGAnCTDR.

In addition, the described term “buffer register” indicates PWGAnCSBR, PWGAnCRBR, or PWGAnCTBR.

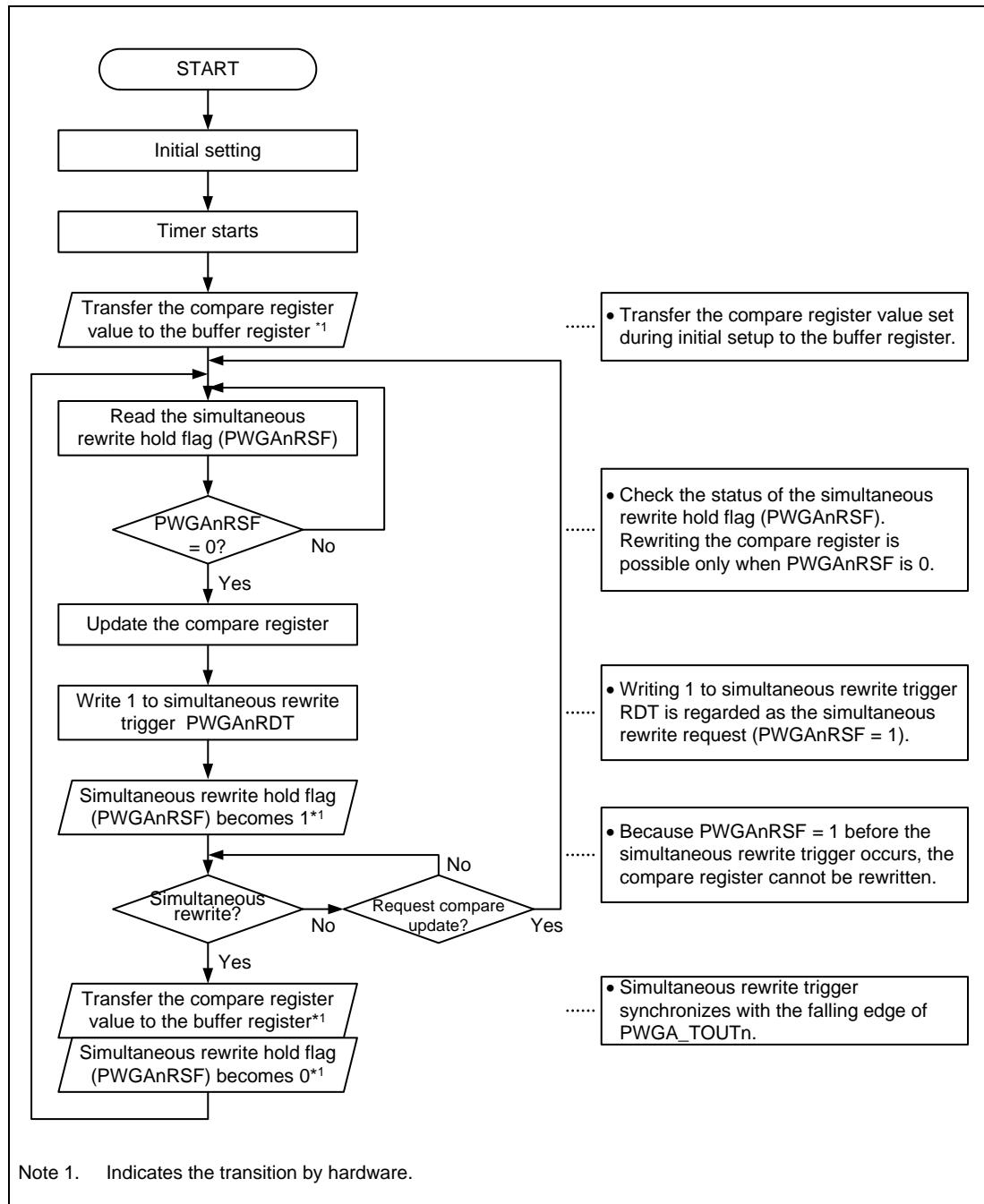


Figure 28.3    Simultaneous Rewrite Procedure

## 28.4 Operation Waveform of PWM-Diag

### 28.4.1 PWM Waveform Output by PWGA and Operation Waveform for A/D Conversion Trigger Output

#### 28.4.1.1 Basic Operation Waveform of PWGA

The basic operation waveforms of PWGA are illustrated below.

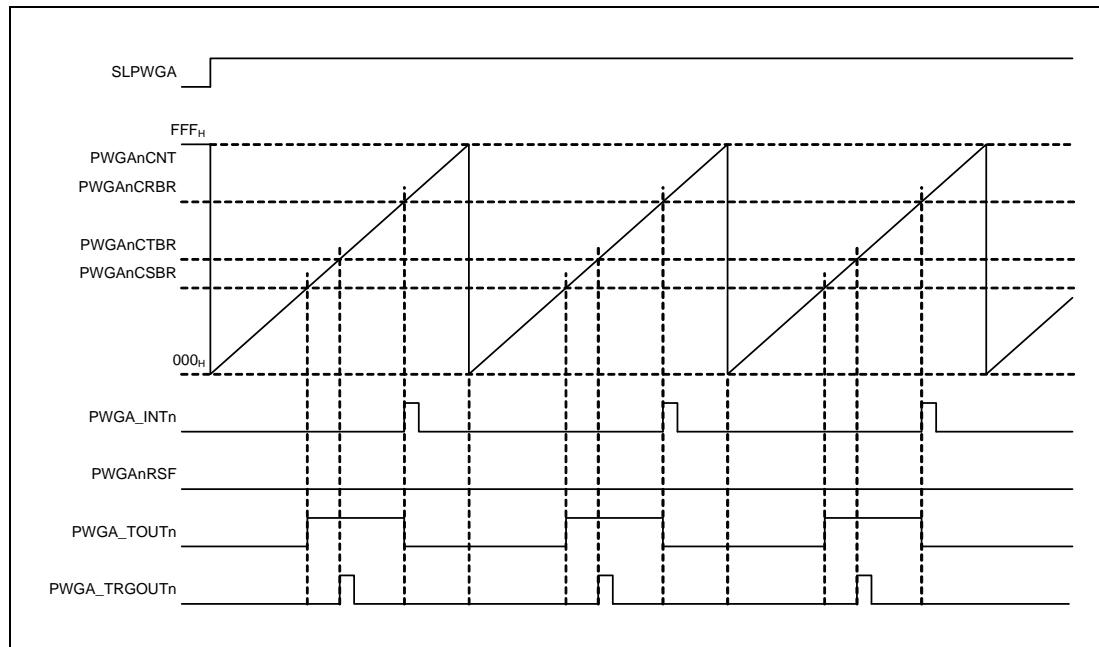
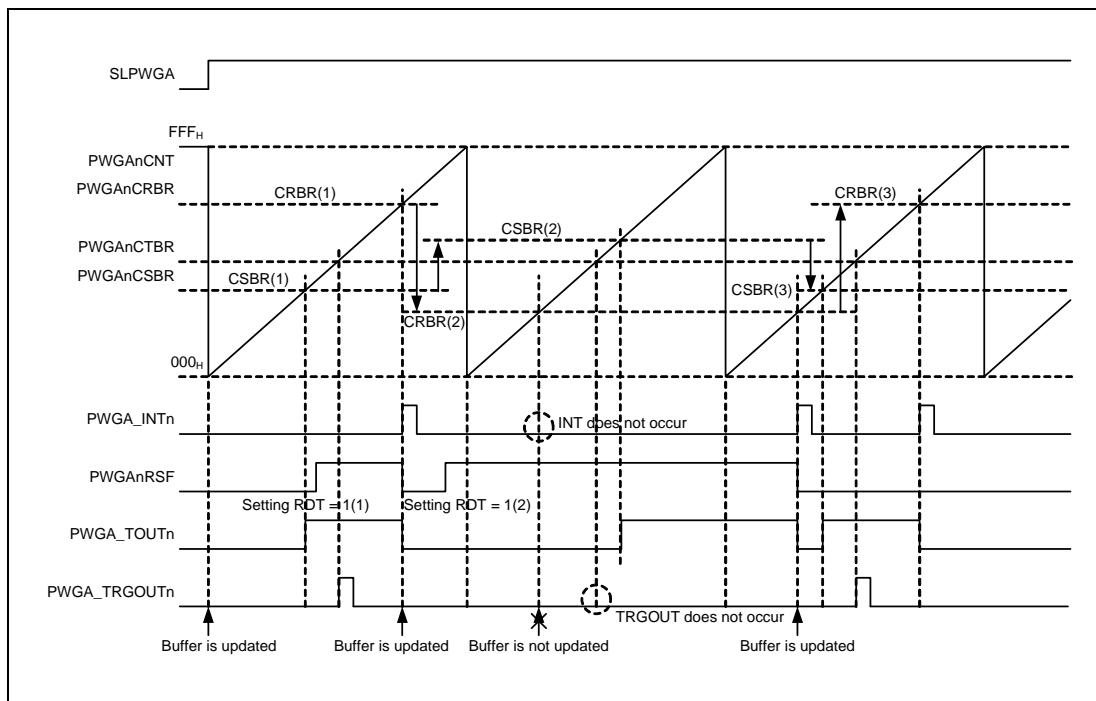


Figure 28.4 Basic Waveform

### 28.4.1.2 Operation Waveform when Simultaneous Rewrite for PWGA is Executed

The following figure illustrates the operation waveforms when simultaneous rewrite for PWGA is executed.



**Figure 28.5    Waveform when Simultaneous Rewrite is Executed**

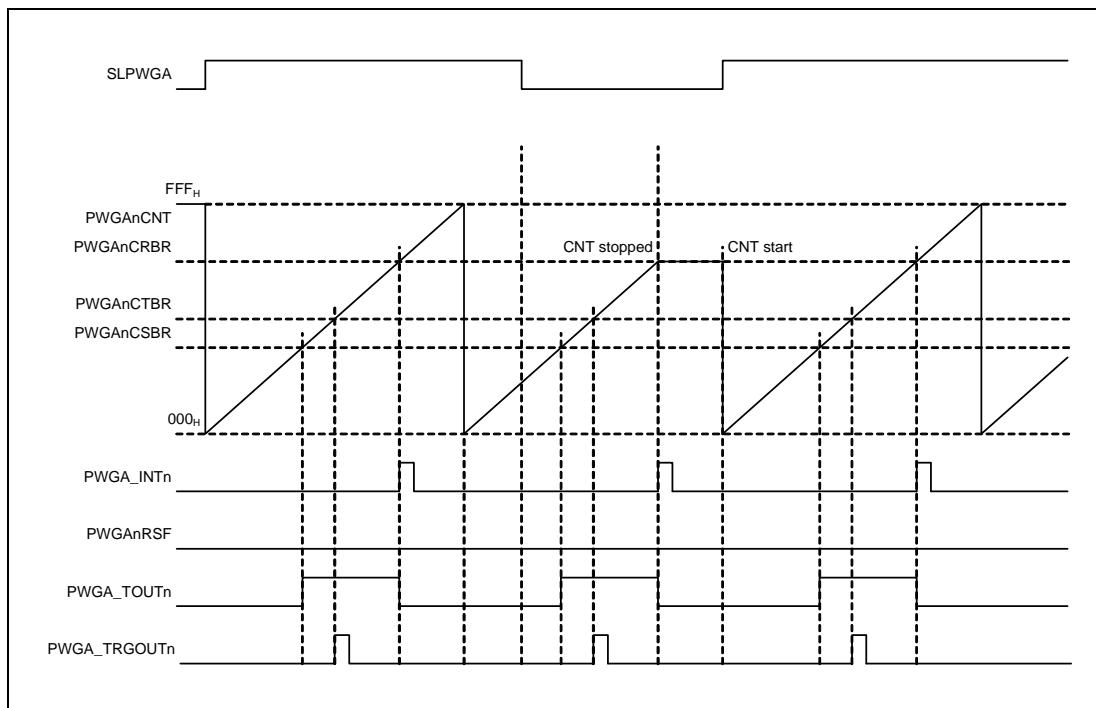
Simultaneous rewrite is executed by re-setting the PWGAnCSDR and PWGAnCRDR registers, then setting either the PWGAnRDT or SLPWGAK register.

Moreover, if the relationship between set values in one interval is PWGAnCSDR > PWGAnCRDR, a falling edge in that interval is meaningless, and the falling edge in the next interval is valid.

Moreover, PWGA\_TRGOUTn does not become valid unless PWGA\_TOUTn is at the high level.

### 28.4.1.3 Operation Waveform when Stopping and Resuming PWGA Operation

The following figure illustrates the operation waveforms when stopping and resuming PWGA operation.



**Figure 28.6    Stopping and Resuming Operation (1)**

After the setting of SLPWGA has been changed from 1 to 0, PWGAnCNT stops operation because PWGA\_INTn is generated.

After PWGA\_INTn has been generated, by changing the setting of SLPWGA from 0 to 1, PWGAnCNT resumes counting from 000H.

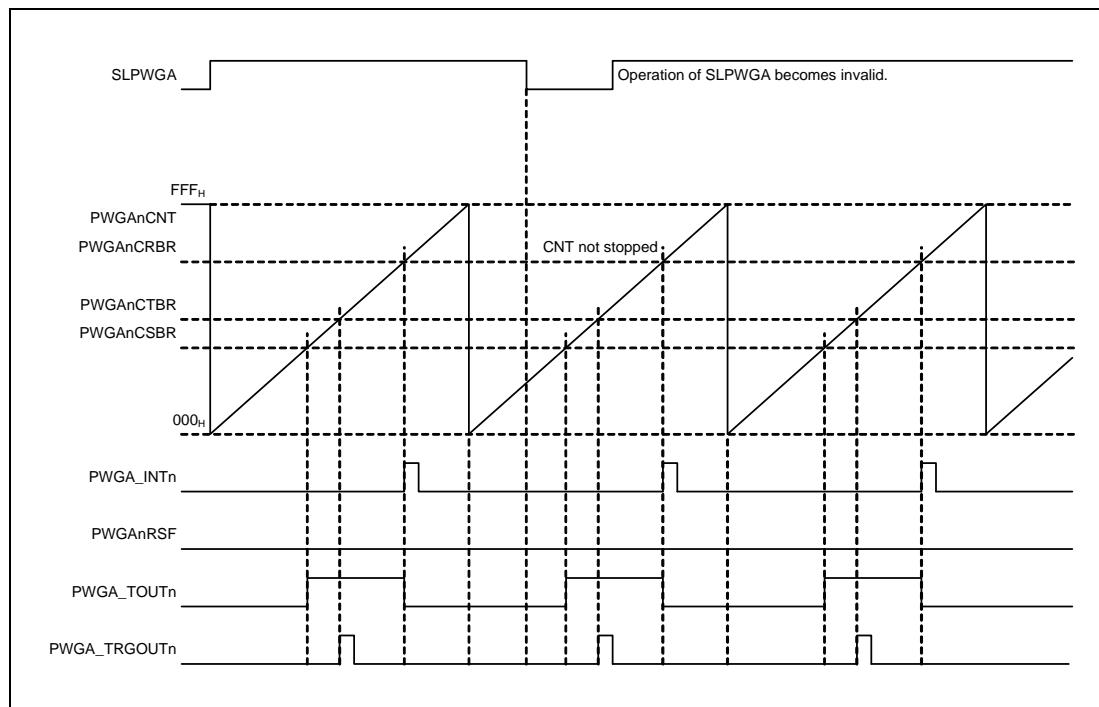
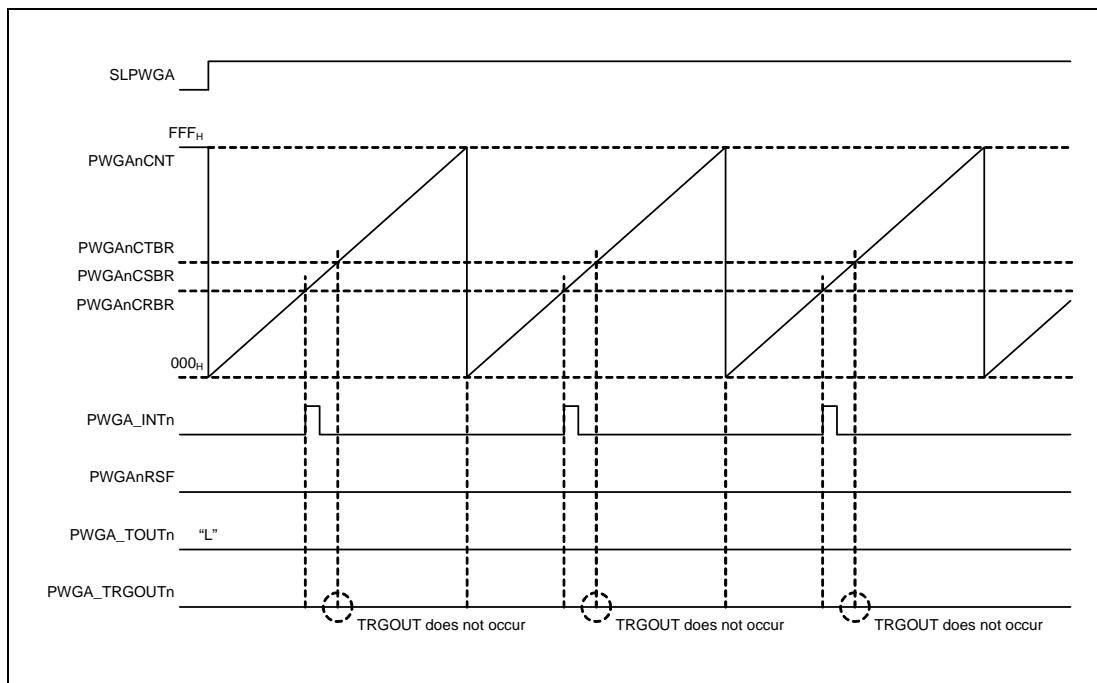


Figure 28.7 Stopping and Resuming Operation (2)

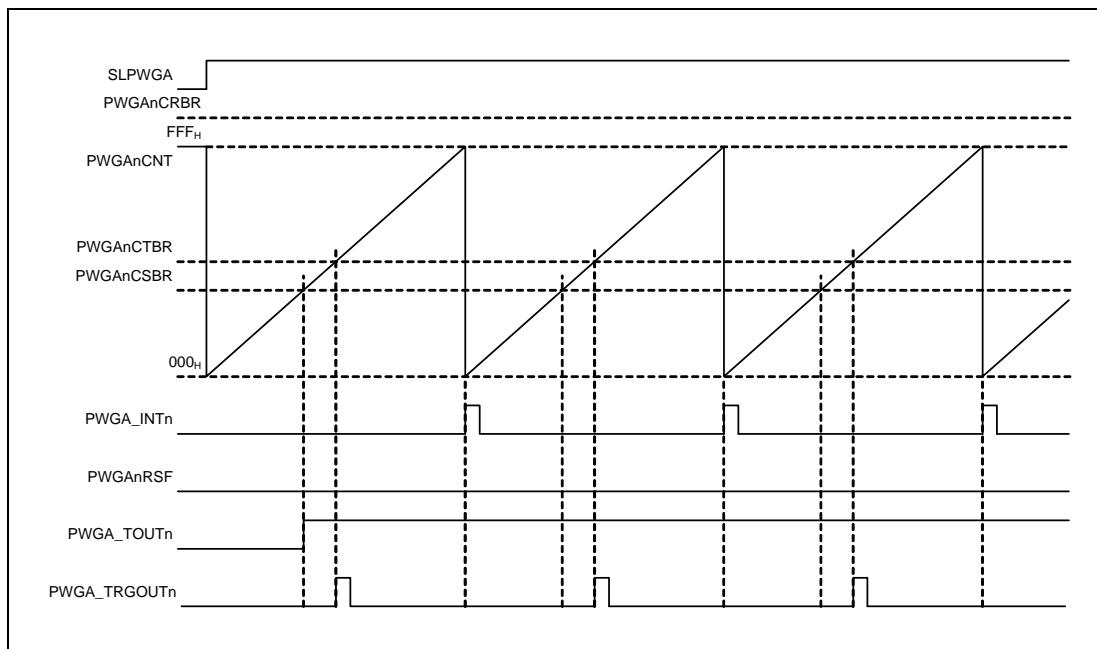
After the setting of SLPWGA has been changed from 1 to 0, if the setting of SLPWGA is changed from 0 to 1 before PWGA\_INTn is generated, operations of SLPWGA become invalid, and PWGAnCNT continues counting.

#### 28.4.1.4 Waveforms of PWGA Operation with Specific Settings

The following figures illustrate the waveforms of PWGA operation with specific settings.



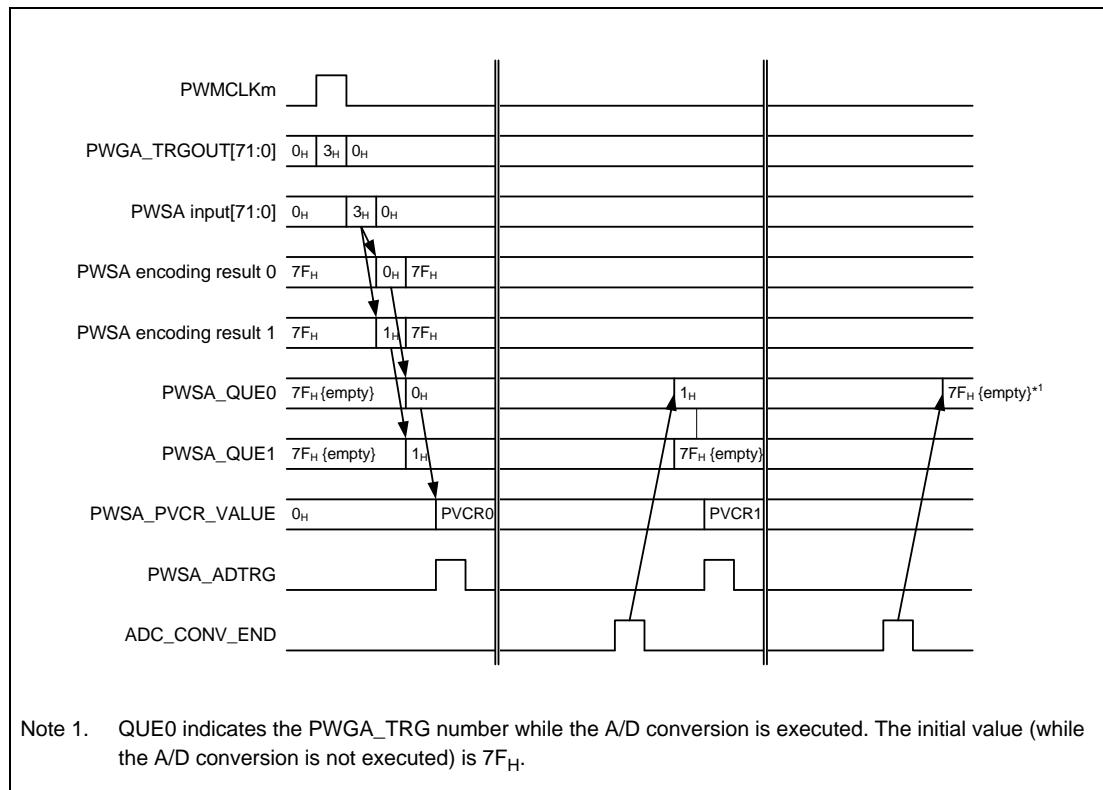
**Figure 28.8** PWGA\_TOOUTn = 0% Output Waveform



**Figure 28.9** PWGA\_TOOUTn = 100% Output Waveform

### 28.4.2 Operation Waveform when A/D Conversion Trigger Occurs in PWSA

An example of the PWSA operation is shown below.



**Figure 28.10 Example of PWSA Operation**

- (1) Triggers occur simultaneously in channels 0 and 1 of PWGA. Channel 0 with the smaller channel number is stored in PWSAnQUE0, and channel 1 with the larger channel number is stored in PWSAnQUE1. The lower 16-bit data of PWSAnPVCR00\_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter. At this time, as the A/D conversion for channel 1 is in the waiting state, the PWSAnSTR.PWSAnQNE bit is set.
- (2) On completion of A/D conversion executed in step (1), the channel number of PWSAnQUE1 shifts to PWSAnQUE0 and PWSAnQUE1 enters the empty state. After that, as similar to step (1), the upper 16-bit data of PWSAnPVCR00\_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter.
- (3) On completion of A/D conversion executed in step (2), PWSAnQUE0 enters the empty state.

## 28.5 PWM-Diag Related Functions in A/D Converter (ADCA)

This section describes the A/D converter used for the PWM-Diag function.

### 28.5.1 ADCA registers when the PWM-Diag function is used

- Before starting PWSA operation, the A/D converter must be set using the following register.
  - PWM-Diag scan group control register (ADCAnPWDMSGCR)
- When the PWM-Diag is running, the PWSAnPVCRx\_y value corresponding to the channel under conversion is set in the following register of the A/D converter.
  - PWM-Diag virtual channel register (ADCAnPWDVCR)
- After completion of A/D conversion, the conversion result can be checked by reading the following registers.
  - PWM-Diag data register (ADCAnPWDTSNDR)
  - PWM-Diag data supplementary information register (ADCAnPWDDIR)
- When A/D conversion result is outside the expected range, it can be confirmed using the upper/lower limit error detection function. The upper/lower limit error detection function is set by the following register.
  - Upper limit/lower limit error register (ADCAnULER)
- The scan end flag of the PWM-Diag scan group can be cleared using the following register.
  - PWM-Diag scan end flag clear register (ADCAnPWDMSGSEFCR)

## Section 29 A/D Converter (ADCA)

This section contains a generic description of the A/D Converter (ADCA).

The first part of this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the ADCA.

### 29.1 Features of RH850/F1L ADCA

#### 29.1.1 Number of Units and Channels

This microcontroller has the following number of ADCA units.

**Table 29.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	2	2
Name	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0, 1)	ADCAn (n = 0, 1)

An ADCAn unit has the same number of physical channels as the number of A/D input pins and the same number of virtual channels as the number of addresses where the results of A/D conversion will be stored. The numbers of channels on individual products are as listed below.

**Table 29.2 Unit Configurations and Physical Channels**

Unit Name (Number of Channels) ADCAn	RH850/F1L 48 pins (12 ch)	RH850/F1L 64 pins (21 ch)	RH850/F1L 80 pins (25 ch)	RH850/F1L 100 pins (36 ch)	RH850/F1L 144 pins (48 ch)	RH850/F1L 176 pins (60 ch)
ADCA0	12 bit pin for conversion* <sup>1</sup>	8	10	11	16	16
	10 bit pin for conversion* <sup>2</sup>	4	11	14	20	20
ADCA1	12 bit pin for conversion* <sup>1</sup>				8	16
	10 bit pin for conversion* <sup>2</sup>				4	8

Note 1. When 10-bit mode is selected, this pin can be used for 10-bit conversion.

Note 2. When 12-bit mode is selected but a pin is for 10-bit conversion, the 2 low-order bits of the result of conversion must be masked before use.

**Table 29.3 Unit Configurations and Virtual Channels**

Unit Name (Number of Channels) ADCAn	RH850/F1L 48 pins (20 ch)	RH850/F1L 64 pins (29 ch)	RH850/F1L 80 pins (37 ch)	RH850/F1L 100 pins (48 ch)	RH850/F1L 144 pins (62 ch)	RH850/F1L 176 pins (74 ch)
ADCA0	20	29	37	48	50	50
ADCA1	—	—	—	—	12	24

**Table 29.4 Indices**

Index	Description
n	Throughout this section, the individual ADCA units are identified by the index "n" (n = 0, 1); for example, ADCAnPWDVCR indicates the PWM-Diag virtual channel register.
m	Throughout this section, the individual physical channels (channels in the unit) of ADCAn are identified by the index "m"; for example, ANInm.
j	Throughout this section, the individual virtual channels of ADCAn are identified by the index "j"; for example, ADCAnVCRj indicates the virtual channel register.
x	Throughout this section, the individual scan groups (SG) of ADCAn are identified by the index "x" (x = 1 to 3); for example, ADCAnSGSTCRx indicates the scan group x start control register.
k	Throughout this section, the individual physical channel numbers for T&H are identified by the index "k" (k = 0 to 5); for example, THkE is T&H enable bit of the T&H enable register (ADCAnTHER).

The following table shows values indicated by the indices of each product.

**Table 29.5 Indices of Products**

Indices of Each Product					
48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
m = 0 to 11 (ADCA0)	m = 0 to 20 (ADCA0)	m = 0 to 24(ADCA0)	m = 0 to 35(ADCA0)	m = 0 to 35 (ADCA0) m = 0 to 11 (ADCA1)	m = 0 to 35 (ADCA0) m = 0 to 23 (ADCA1)
j = 00 to 19* <sup>1</sup> (ADCA0)	j = 00 to 28* <sup>1</sup> (ADCA0)	j = 00 to 36 (ADCA0)	j = 00 to 47 (ADCA0)	j = 00 to 49 (ADCA0) j = 00 to 11 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 23 (ADCA1)
x = 1 to 3	x = 1 to 3	x = 1 to 3	x = 1 to 3	x = 1 to 3	x = 1 to 3
k = 0, 2, 4	k = 0, 2, 4	k = 0, 2, 4	k = 0 to 5	k = 0 to 5	k = 0 to 5

Note 1. j = 33 to 35 are supported only in diagnosis of T&H circuit.

### 29.1.2 Register Base Address

ADCAn base addresses are listed in the following table.

ADCAn register addresses are given as offsets from the base addresses in general.

**Table 29.6 Register Base Addresses**

Base Address Name	Base Address
<ADCA0_base>	FFF2 0000 <sub>H</sub>
<ADCA1_base>	FFF2 1000 <sub>H</sub>

### 29.1.3 Clock Supply

The ADCAn clock supply is shown in the following table.

**Table 29.7 Clock Supply**

<b>Unit Name</b>	<b>Unit Clock Name</b>	<b>Supply Clock Name</b>
ADCA0	Register access clock ADCLK	CKSCLK_AADCA CKSCLK_IADCA
ADCA1	Register access clock ADCLK	CPUCLK2 CKSCLK_IADCA

### 29.1.4 Interrupt Requests

ADCAn interrupt requests are listed in the following table.

Table 29.8 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number	Other Trigger Signals
<b>ADCA0</b>				
INT_ADE	A/D error interrupt	47	—	Motor control
INT_SG1	Scan group 1 (SG1) end interrupt	10	4 (Channels 0 to 7)	LPS
INT_SG2	Scan group 2 (SG2) end interrupt	11	5 (Channels 0 to 7)	LPS
INT_SG3	Scan group 3 (SG3) end interrupt	12	6 (Channels 0 to 7)	LPS
ADC_CONV_END0	Scan group 4 (SG4) A/D conversion end signal	—	7 (Channels 0 to 7)	—
<b>ADCA1</b>				
INT_ADE	A/D error interrupt	204	—	—
INT_SG1	Scan group 1 (SG1) end interrupt	205	39 (Channels 8 to 15)	—
INT_SG2	Scan group 2 (SG2) end interrupt	206	40 (Channels 8 to 15)	—
INT_SG3	Scan group 3 (SG3) end interrupt	207	41 (Channels 8 to 15)	—
ADC_CONV_END1	Scan group 4 (SG4) A/D conversion end signal	—	42 (Channels 8 to 15)	—

### 29.1.5 Reset Sources

ADCAn reset sources are listed in the following table. ADCAn is initialized by these reset sources.

Table 29.9 Reset Sources

Unit Name	Reset Source
ADCA0	Reset sources other than transition to DeepSTOP mode (AWORES)
ADCA1	All reset sources (ISORES)

### 29.1.6 External Input/Output Signals

External input/output signals of ADCAn are listed below.

Table 29.10 ADCA0 External Input/Output Signals (1/2)

Unit Signal Name	Outline	Alternative Port Pin Signal
<b>ADCA0</b>		
ANI000	12-bit resolution analog input pin (for T&H)	ADCA0I0
ANI001	12-bit resolution analog input pin (for T&H)* <sup>1</sup>	ADCA0I1
ANI002	12-bit resolution analog input pin (for T&H)	ADCA0I2
ANI003	12-bit resolution analog input pin (for T&H)* <sup>1</sup>	ADCA0I3
ANI004	12-bit resolution analog input pin (for T&H)	ADCA0I4
ANI005	12-bit resolution analog input pin (for T&H)* <sup>1</sup>	ADCA0I5
ANI006	12-bit resolution analog input pin	ADCA0I6
ANI007	12-bit resolution analog input pin	ADCA0I7
ANI008	12-bit resolution analog input pin	ADCA0I8
ANI009	12-bit resolution analog input pin	ADCA0I9
ANI010	12-bit resolution analog input pin	ADCA0I10

Table 29.10 ADCA0 External Input/Output Signals (2/2)

Unit Signal Name	Outline	Alternative Port Pin Signal
ANI011	12-bit resolution analog input pin	ADCA0I11
ANI012	12-bit resolution analog input pin	ADCA0I12
ANI013	12-bit resolution analog input pin	ADCA0I13
ANI014	12-bit resolution analog input pin	ADCA0I14
ANI015	12-bit resolution analog input pin	ADCA0I15
ANI016	10-bit resolution analog input pin	ADCA0I0S
ANI017	10-bit resolution analog input pin	ADCA0I1S
ANI018	10-bit resolution analog input pin	ADCA0I2S
ANI019	10-bit resolution analog input pin	ADCA0I3S
ANI020	10-bit resolution analog input pin	ADCA0I4S
ANI021	10-bit resolution analog input pin	ADCA0I5S
ANI022	10-bit resolution analog input pin	ADCA0I6S
ANI023	10-bit resolution analog input pin	ADCA0I7S
ANI024	10-bit resolution analog input pin	ADCA0I8S
ANI025	10-bit resolution analog input pin	ADCA0I9S
ANI026	10-bit resolution analog input pin	ADCA0I10S
ANI027	10-bit resolution analog input pin	ADCA0I11S
ANI028	10-bit resolution analog input pin	ADCA0I12S
ANI029	10-bit resolution analog input pin	ADCA0I13S
ANI030	10-bit resolution analog input pin	ADCA0I14S
ANI031	10-bit resolution analog input pin	ADCA0I15S
ANI032	10-bit resolution analog input pin	ADCA0I16S
ANI033	10-bit resolution analog input pin	ADCA0I17S
ANI034	10-bit resolution analog input pin	ADCA0I18S
ANI035	10-bit resolution analog input pin	ADCA0I19S
ADCA0TRG0	External trigger pin (scan group 1)*2	ADCA0TRG0
ADCA0TRG1	External trigger pin (scan group 2)*2	ADCA0TRG1
ADCA0TRG2	External trigger pin (scan group 3)*2	ADCA0TRG2
ADCA0SEL0	External analog multiplexer (MPX) output pin 0	ADCA0SEL0
ADCA0SEL1	External analog multiplexer (MPX) output pin 1	ADCA0SEL1
ADCA0SEL2	External analog multiplexer (MPX) output pin 2	ADCA0SEL2

Note 1. The track and hold circuit (T&H) for ANI0m (m = 01, 03, 05) is only included on F1L devices with 100, 144 or 176 pins.

Note 2. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2.12, Noise Filter & Edge/Level Detector**.

Table 29.11 ADCA1 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
<b>ADCA1</b>		
ANI100	12-bit resolution analog input pin	ADCA1I0
ANI101	12-bit resolution analog input pin	ADCA1I1
ANI102	12-bit resolution analog input pin	ADCA1I2
ANI103	12-bit resolution analog input pin	ADCA1I3
ANI104	12-bit resolution analog input pin	ADCA1I4
ANI105	12-bit resolution analog input pin	ADCA1I5
ANI106	12-bit resolution analog input pin	ADCA1I6
ANI107	12-bit resolution analog input pin	ADCA1I7
ANI108	12-bit resolution analog input pin	ADCA1I8
ANI109	12-bit resolution analog input pin	ADCA1I9
ANI110	12-bit resolution analog input pin	ADCA1I10
ANI111	12-bit resolution analog input pin	ADCA1I11
ANI112	12-bit resolution analog input pin	ADCA1I12
ANI113	12-bit resolution analog input pin	ADCA1I13
ANI114	12-bit resolution analog input pin	ADCA1I14
ANI115	12-bit resolution analog input pin	ADCA1I15
ANI116	10-bit resolution analog input pin	ADCA1I0S
ANI117	10-bit resolution analog input pin	ADCA1I1S
ANI118	10-bit resolution analog input pin	ADCA1I2S
ANI119	10-bit resolution analog input pin	ADCA1I3S
ANI120	10-bit resolution analog input pin	ADCA1I4S
ANI121	10-bit resolution analog input pin	ADCA1I5S
ANI122	10-bit resolution analog input pin	ADCA1I6S
ANI123	10-bit resolution analog input pin	ADCA1I7S
ADCA1TRG0	External trigger pin (scan group 1)* <sup>1</sup>	ADCA1TRG0
ADCA1TRG1	External trigger pin (scan group 2)* <sup>1</sup>	ADCA1TRG1
ADCA1TRG2	External trigger pin (scan group 3)* <sup>1</sup>	ADCA1TRG2

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2.12, Noise Filter & Edge/Level Detector.**

## 29.2 Overview

### 29.2.1 Functional Overview

ADCA has the following features.

- 10-bit/12-bit resolution
- Successive approximation conversion method
- Number of A/D input channels

A/D conversion is available for a maximum of 36 ADCA0 channels and 24 ADCA1 channels.

Additionally, ADCA0 supports the connection of an external analog multiplexer (MPX) to extend the number of analog input channels.

- Internal track and hold (T&H) circuit

ANI000 to ANI005 (ADCA0I0 to ADCA0I5) of ADCA0 include the track and hold circuit. The track and hold circuit can sample up to 6 channels of analog input simultaneously.

- A/D conversion control by scan groups

The A/D conversion channel or conversion mode (scan mode) can be set for each scan group.

- Two scan modes

Multi-cycle scan mode: Specified number of scans are executed.

Continuous scan mode: Scans are executed repeatedly and continuously.

- Asynchronous/synchronous suspend and resume function

A processing for a scan group can be interrupted to run the processing for another scan group.

- Start trigger for each scan group

Software, hardware, and external trigger can start processing of each scan group.

- Scan end interrupt and DMA transfer are supported.

For each scan group, an interrupt request to INTC can be issued or DMA transfer can be started, each time a processing for the virtual channel indicated by the end virtual channel pointer ends, or a virtual channel ends.

- A/D conversion channel repeat function

A/D conversion is performed for the same channel two or four times sequentially, and the result is stored in the data register.

- Abundant safety functions

Abundant safety functions are provided, such as A/D converter diagnosis, diagnosis of the channel multiplexer, diagnosis of open pins, diagnosis of the T&H circuit, upper limit/lower limit check for the A/D converter, overwrite check for data registers, and read and clear function for data registers.

- Shortest A/D conversion time per channel

1.15  $\mu$ s (when MPX is not used)

2.30  $\mu$ s (when MPX is used)

**NOTE**

- Physical channel (ANI<sub>m</sub>)

Each A/D input channel of ADCA0 and ADCA1 units is called a physical channel. The physical channel of each unit is represented as ANI<sub>0m</sub> ( $m = 0$  to 35) for ADCA0 and ANI<sub>1m</sub> ( $m = 0$  to 23) for ADCA1.

In RH850/F1L, the alternative port pins for 12-bit resolution A/D input channel and 10-bit resolution A/D input channel are represented as ADCAn<sub>m</sub> and ADCAn<sub>mS</sub>, respectively. In this section, the physical channels and the corresponding alternative port pins are listed together.

- Virtual channel (ADCAnVCR<sub>j</sub>)

ADCA0 has a maximum of 50 virtual channels, and ADCA1 has a maximum of 24 virtual channels. The virtual channel specifies the physical channel to be scanned.

Scans are executed in sequence from the smallest virtual channel number. The scan order can be arbitrarily-specified by using virtual channels. In addition, the scanned result is stored in the data register (ADCAnDR<sub>j</sub>) corresponding to the virtual channel.

- Scan group (SG<sub>x</sub>)

ADCA has three scan groups (SG1, SG2, SG3) and one PWM-Diag group (SG4). A/D conversion is executed in scan group unit. The channel to be scanned can be selected for each group by specifying the scan range, that is, the conversion start virtual channel and the conversion end virtual channel.

## 29.2.2 Block Diagram

The block diagram of ADCA0 is shown in **Figure 29.1**. The block diagram of ADCA1 is shown in **Figure 29.2**.

### (1) Configuration of ADCA0

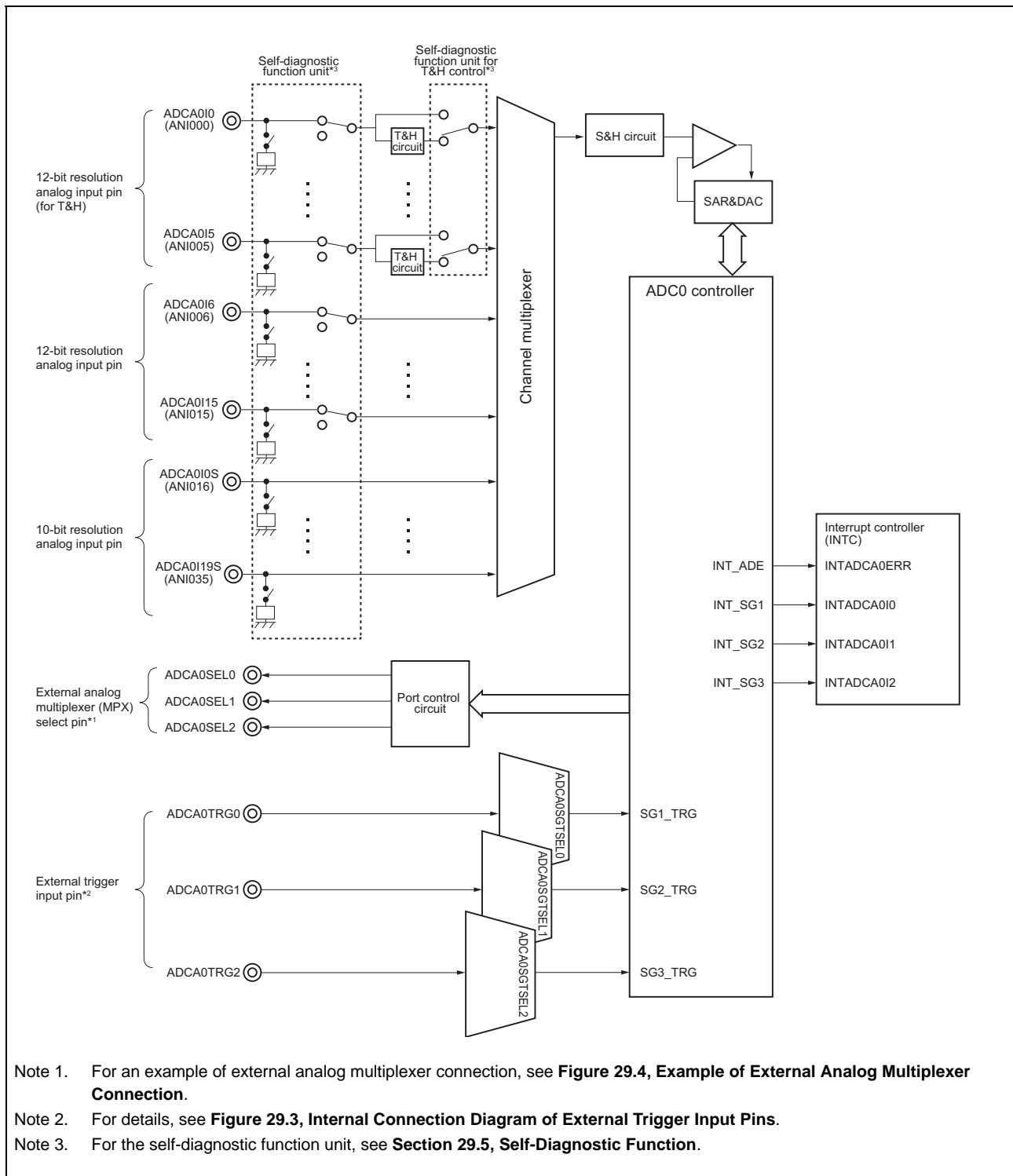


Figure 29.1 ADCA0 Block Diagram (RH850/F1L with 176 Pins)

## (2) Configuration of ADCA1

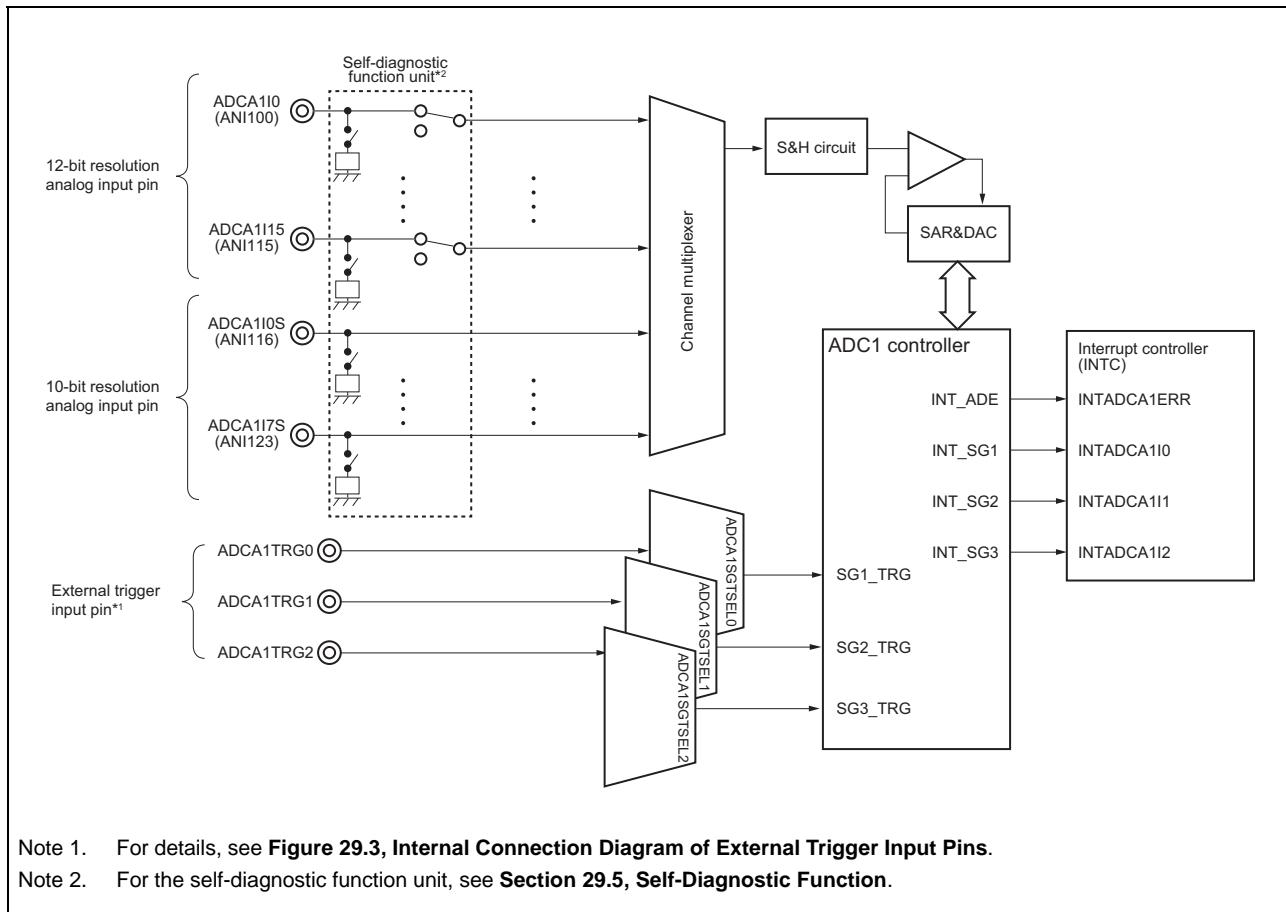
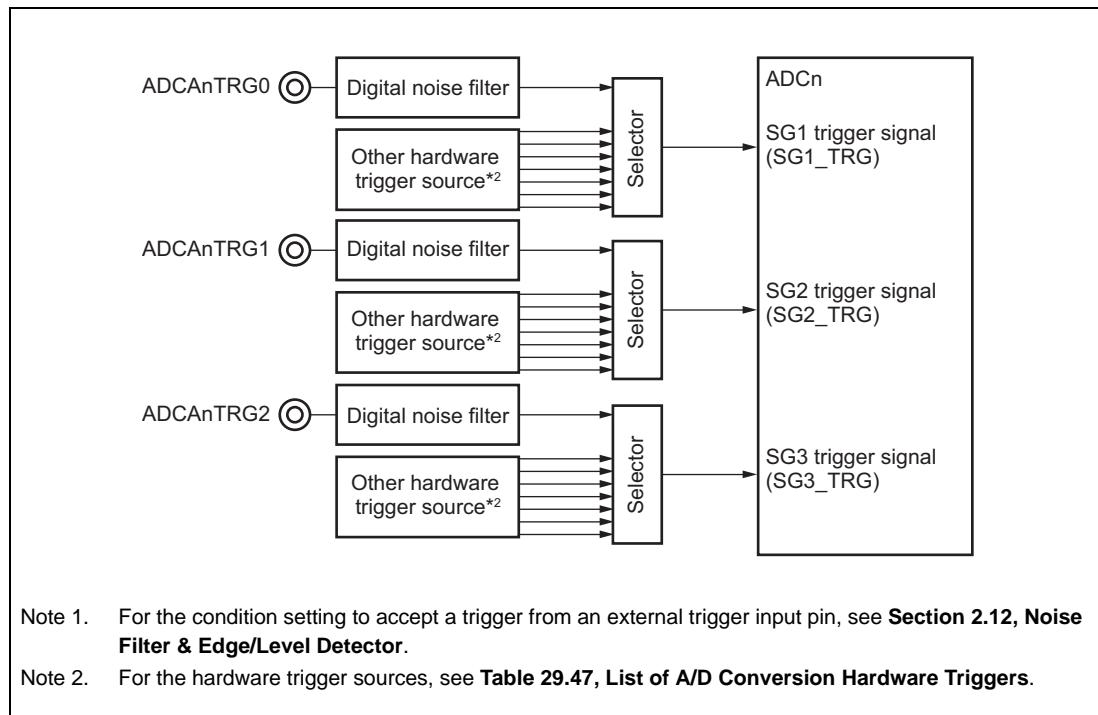


Figure 29.2 ADCA1 Block Diagram (RH850/F1L with 176 Pins)

### (3) Configuration of external trigger input pins

An external trigger input pin is a hardware trigger source to activate ADCAn.

The configuration of external trigger input pins is shown below.



**Figure 29.3 Internal Connection Diagram of External Trigger Input Pins**

#### (4) Configuration of external analog multiplexer (MPX)

The external analog multiplexer (MPX) can be connected to any input signal pins ADCA0I0 to ADCA0I19S. An example of the external analog multiplexer connection is shown below.

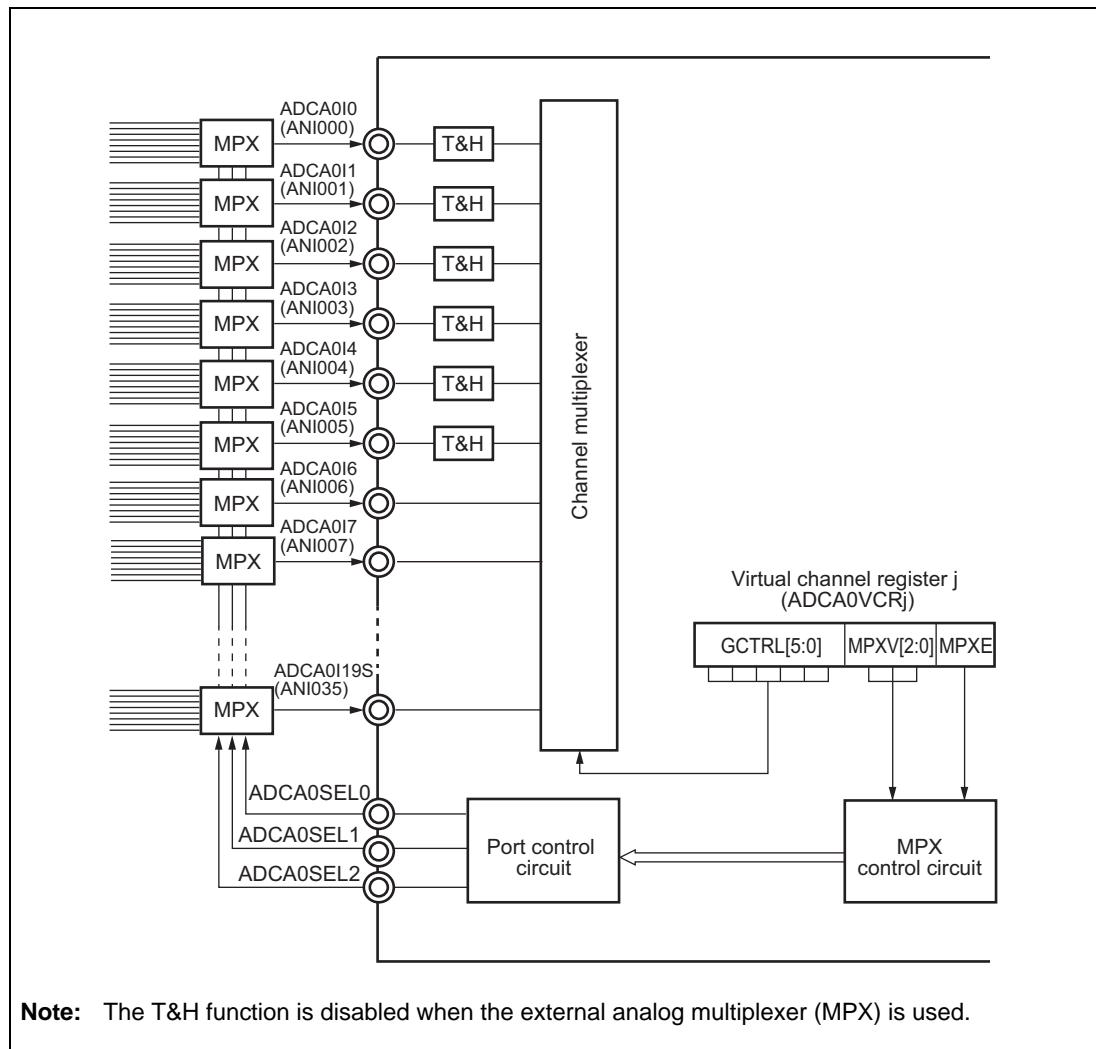


Figure 29.4 Example of External Analog Multiplexer Connection

## (5) Virtual channel

The virtual channel specifies the physical address to be scanned.

The virtual channel is controlled by the ADCAnVCRj register.

A usage example of the virtual channel is shown below.

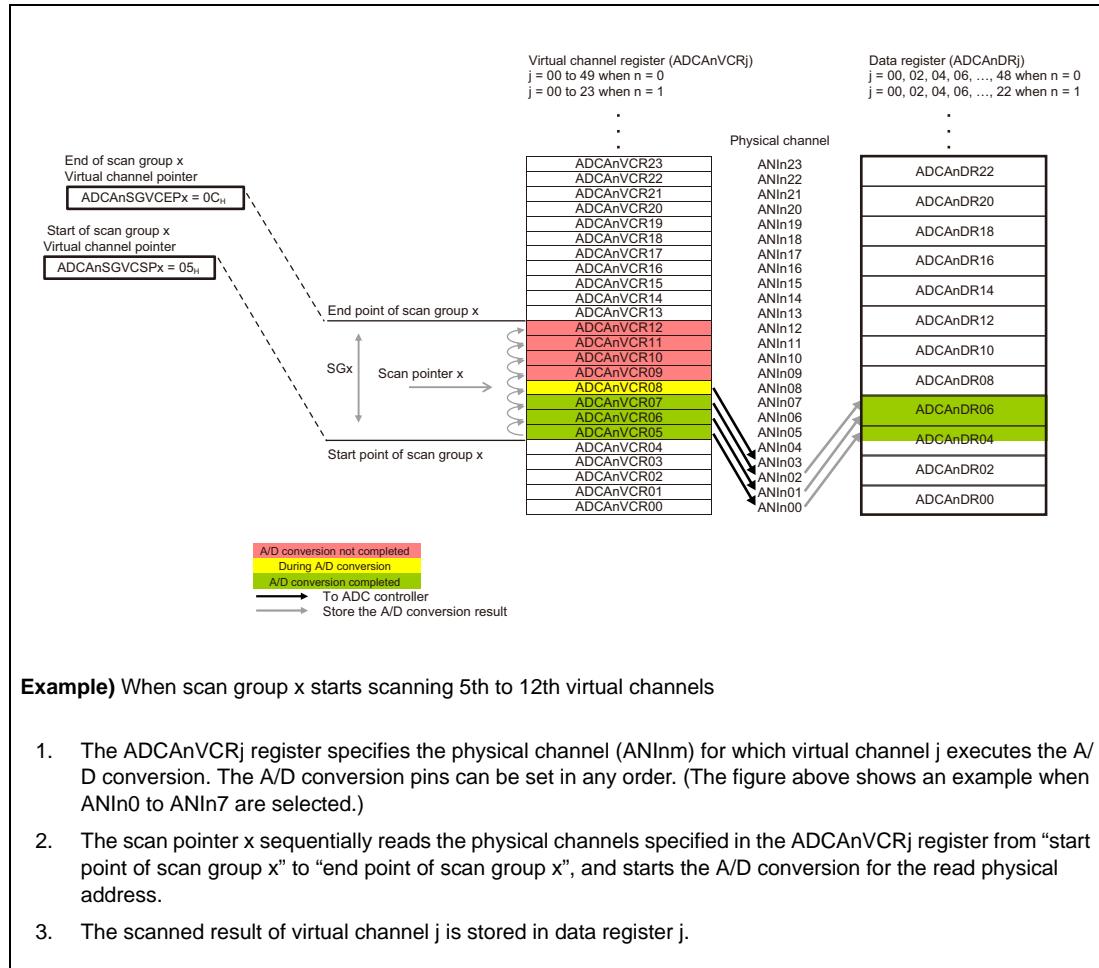


Figure 29.5 Usage Example of Virtual Register

## 29.3 Registers

### 29.3.1 List of Registers

The ADCA registers are listed in the following table.

For details about <ADCAn\_base>, see **Section 29.1.2, Register Base Address**.

**Table 29.12 List of Registers (1/2)**

Module Name	Register Name	Symbol	Address
<b>ADCA Specific Registers (Virtual Channel)</b>			
ADCAn	Virtual Channel Register j	ADCAnVCRj	<ADCAn_base> + j × 4H
ADCAn	PWM-Diag Virtual Channel Register	ADCAnPWDVCR	<ADCAn_base> + 0F4H
ADCAn	Data Register j	ADCAnDRj	<ADCAn_base> + 100H + j × 2H
ADCAn	Data Supplementary Information Register j	ADCAnDIRj	<ADCAn_base> + 200H + j × 4H
ADCAn	PWM-Diag data register	ADCAnPWDTSNDR	<ADCAn_base> + 178H
ADCAn	PWM-Diag Data Supplementary Information Register	ADCAnPWDDIR	<ADCAn_base> + 2F4H
<b>ADCA Specific Registers (Control)</b>			
ADCAn	A/D Force Halt Register	ADCAnADHALTR	<ADCAn_base> + 300H
ADCAn	A/D Control Register	ADCAnADCR	<ADCAn_base> + 304H
ADCAn	MPX Current Register	ADCAnMPXCURR	<ADCAn_base> + 30CH
ADCAn	T&H Sampling Start Control Register	ADCAnTHSMPSTCR	<ADCAn_base> + 314H
ADCAn	T&H Control Register	ADCAnTHCR	<ADCAn_base> + 318H
ADCAn	T&H Group A Hold Start Control Register	ADCAnTHAHLDCSTR	<ADCAn_base> + 31CH
ADCAn	T&H Group B Hold Start Control Register	ADCAnTHBHLDCSTR	<ADCAn_base> + 320H
ADCAn	T&H Group A Control Register	ADCAnTHACR	<ADCAn_base> + 324H
ADCAn	T&H Group B Control Register	ADCAnTHBCR	<ADCAn_base> + 328H
ADCAn	T&H Enable Register	ADCAnTHER	<ADCAn_base> + 32CH
ADCAn	T&H Group Select Register	ADCAnTHGSR	<ADCAn_base> + 330H
ADCAn	Sampling Control Register	ADCAnSMPPCR	<ADCAn_base> + 380H
<b>ADCA Specific Registers (Safety-related)</b>			
ADCAn	Safety Control Register	ADCAnSFTCR	<ADCAn_base> + 334H
ADCAn	Upper Limit/Lower Limit Table Register 0	ADCAnULLMTBR0	<ADCAn_base> + 338H
ADCAn	Upper Limit/Lower Limit Table Register 1	ADCAnULLMTBR1	<ADCAn_base> + 33CH
ADCAn	Upper Limit/Lower Limit Table Register 2	ADCAnULLMTBR2	<ADCAn_base> + 340H
ADCAn	Error Clear Register	ADCAnECR	<ADCAn_base> + 344H
ADCAn	Upper Limit/Lower Limit Error Register	ADCAnULER	<ADCAn_base> + 348H
ADCAn	Overwrite Error Register	ADCAnOWER	<ADCAn_base> + 34CH
<b>Scan Group Specific Registers</b>			
ADCAn	Scan Group x Start Control Register	ADCAnSGSTCRx	<ADCAn_base> + x × 40H + 400H
ADCAn	PWM-Diag Scan Group Control Register	ADCAnPWDSGCR	<ADCAn_base> + 508H
ADCAn	Scan Group x Control Register	ADCAnSGCRx	<ADCAn_base> + x × 40H + 408H
ADCAn	Scan Group x Start Virtual Channel Pointer	ADCAnSGVCSPx	<ADCAn_base> + x × 40H + 40CH
ADCAn	Scan Group x End Virtual Channel Pointer	ADCAnSGVCEPx	<ADCAn_base> + x × 40H + 410H
ADCAn	Scan Group x Multicycle Register	ADCAnSGMCYCRx	<ADCAn_base> + x × 40H + 414H
ADCAn	PWM-Diag Scan End Flag Clear Register	ADCAnPWDGSEFCR	<ADCAn_base> + 518H
ADCAn	Scan Group x Scan End Flag Clear Register	ADCAnSGSEFCRx	<ADCAn_base> + x × 40H + 418H
ADCAn	Scan Group Status Register	ADCAnSGSTR	<ADCAn_base> + 308H
<b>H/W Trigger Specific Register</b>			
ADCAn	Scan Group x Start Trigger Control Register	ADCAnSGTSELx	<ADCAn_base> + x × 40H + 41CH

Table 29.12 List of Registers (2/2)

Module Name	Register Name	Symbol	Address
<b>Self-Diagnosis Specific Registers</b>			
ADCAn	Self-Diagnostic Control Register 0	ADCAnDGCTL0	<ADCAn_base> + 350 <sub>H</sub>
ADCAn	Self-Diagnostic Control Register 1	ADCAnDGCTL1	<ADCAn_base> + 354 <sub>H</sub>
ADCAn	Pull Down Control Register 1	ADCAnPDCTL1	<ADCAn_base> + 358 <sub>H</sub>
ADCAn	Pull Down Control Register 2	ADCAnPDCTL2	<ADCAn_base> + 35C <sub>H</sub>
<b>Emulation Specific Register</b>			
ADCAn	Emulation Control Register	ADCAnEMU	<ADCAn_base> + 388 <sub>H</sub>

## 29.3.2 ADCA Specific Registers

This section describes the registers that are equipped in each of ADCA0 and ADCA1.

### 29.3.2.1 ADCAnVCRj — Virtual Channel Register j

This register is used to control the virtual channel.

**Access:** ADCAnVCRj register can be read or written in 32-bit units.  
 ADCAnVCRjL register can be read or written in 16-bit units.  
 ADCAnVCRjLL and ADCAnVCRjLH registers can be read or written in 8-bit units.

**Address:** ADCAnVCRj: <ADCAn\_base> + j × 4<sub>H</sub>  
 ADCAnVCRjL: <ADCAn\_base> + j × 4<sub>H</sub>  
 ADCAnVCRjLL: <ADCAn\_base> + j × 4<sub>H</sub>  
 ADCAnVCRjLH: <ADCAn\_base> + j × 4<sub>H</sub> + 1<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPXE <sup>*1</sup>	MPXV[2:0] <sup>*1</sup>			—	—	CNVCLS <sup>*2</sup>	ADIE	ULS[1:0]		GCTRL[5:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.13 ADCAnVCRj Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	MPXE <sup>*1</sup>	MPX Enable 0: The use of MPX is prohibited. No wait is inserted before A/D conversion is performed. 1: The use of MPX is permitted. The MPXV[2:0] bits are output from ADCAnSEL0 to ADCAnSEL2 when the virtual channel starts, and a wait of one A/D-conversion time is inserted before A/D conversion is performed.
14 to 12	MPXV[2:0] <sup>*1</sup>	These bits are used to set the MPX value to be transferred to an external analog multiplexer.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	CNVCLS <sup>*2</sup>	A/D Conversion Type Select for Self-Diagnosis 0: A/D conversion of the hold value is performed during a self-diagnosis. 1: Normal A/D conversion is performed during a self-diagnosis. When normal A/D conversion is performed during a self-diagnosis and MPX is in use (MPXE is set), however, a wait of one A/D conversion time is inserted before A/D conversion is performed. On the other hand, MPX cannot be used when A/D conversion of the hold value is performed during a self-diagnosis.
8	ADIE	A/D Conversion End Interrupt Enable 0: A scan group x end interrupt (INT_SGx) is not generated when A/D conversion for virtual channel j ends in SGx. 1: A scan group x end interrupt (INT_SGx) is generated when A/D conversion for virtual channel j ends in SGx.
7, 6	ULS[1:0]	Upper Limit/Lower Limit Table Select 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCAnULLMTBR0. 10: Upper limit and lower limit are checked for ADCAnULLMTBR1. 11: Upper limit and lower limit are checked for ADCAnULLMTBR2.

**Table 29.13 ADCA<sub>n</sub>VCRj Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
5 to 0	GCTRL[5:0]	<p>Physical Channel Select  <math>0_H</math> to <math>23_H</math>: Corresponding ANIn<sub>m</sub> is selected.  <math>24_H</math>: Diagnosis channel for A/D converter is selected.  Others: Setting prohibit</p> <p><b>Note:</b> These bits shall set the physical channel which is supported. See <b>Table 29.5, Indices of Products</b>, <b>Table 29.10, ADCA0 External Input/Output Signals</b> and <b>Table 29.11, ADCA1 External Input/Output Signals</b>.</p>

Note 1. These bits are only supported for ADCA0. For ADCA1, when writing, write the value after reset.

Note 2. This bit is only supported when  $j = 33$  to  $35$ . Otherwise, when writing, write the value after reset.

#### CAUTION

To prevent malfunction, ADCA<sub>n</sub>VCRj should be set when SGACT of applicable scan groups is 0 (before scan groups are started) and TRGMD of applicable scan groups is 0.

**Table 29.14 Selection of Physical Channels (1/2)**

<b>GCTRL5</b>	<b>GCTRL4</b>	<b>GCTRL3</b>	<b>GCTRL2</b>	<b>GCTRL1</b>	<b>GCTRL0</b>	<b>Analog Input Pin to be Selected</b>
0	0	0	0	0	0	ADCAnI0 (Physical channel ANIn00)
0	0	0	0	0	1	ADCAnI1 (Physical channel ANIn01)
0	0	0	0	1	0	ADCAnI2 (Physical channel ANIn02)
0	0	0	0	1	1	ADCAnI3 (Physical channel ANIn03)
0	0	0	1	0	0	ADCAnI4 (Physical channel ANIn04)
0	0	0	1	0	1	ADCAnI5 (Physical channel ANIn05)
0	0	0	1	1	0	ADCAnI6 (Physical channel ANIn06)
0	0	0	1	1	1	ADCAnI7 (Physical channel ANIn07)
0	0	1	0	0	0	ADCAnI8 (Physical channel ANIn08)
0	0	1	0	0	1	ADCAnI9 (Physical channel ANIn09)
0	0	1	0	1	0	ADCAnI10 (Physical channel ANIn10)
0	0	1	0	1	1	ADCAnI11 (Physical channel ANIn11)
0	0	1	1	0	0	ADCAnI12 (Physical channel ANIn12)
0	0	1	1	0	1	ADCAnI13 (Physical channel ANIn13)
0	0	1	1	1	0	ADCAnI14 (Physical channel ANIn14)
0	0	1	1	1	1	ADCAnI15 (Physical channel ANIn15)
0	1	0	0	0	0	ADCAnI0S (Physical channel ANIn16)
0	1	0	0	0	1	ADCAnI1S (Physical channel ANIn17)
0	1	0	0	1	0	ADCAnI2S (Physical channel ANIn18)
0	1	0	0	1	1	ADCAnI3S (Physical channel ANIn19)
0	1	0	1	0	0	ADCAnI4S (Physical channel ANIn20)
0	1	0	1	0	1	ADCAnI5S (Physical channel ANIn21)
0	1	0	1	1	0	ADCAnI6S (Physical channel ANIn22)
0	1	0	1	1	1	ADCAnI7S (Physical channel ANIn23)
0	1	1	0	0	0	ADCAnI8S (Physical channel ANIn24)* <sup>1</sup>
0	1	1	0	0	1	ADCAnI9S (Physical channel ANIn25)* <sup>1</sup>
0	1	1	0	1	0	ADCAnI10S (Physical channel ANIn26)* <sup>1</sup>
0	1	1	0	1	1	ADCAnI11S (Physical channel ANIn27)* <sup>1</sup>
0	1	1	1	0	0	ADCAnI12S (Physical channel ANIn28)* <sup>1</sup>
0	1	1	1	0	1	ADCAnI13S (Physical channel ANIn29)* <sup>1</sup>

**Table 29.14 Selection of Physical Channels (2/2)**

GCTRL5	GCTRL4	GCTRL3	GCTRL2	GCTRL1	GCTRL0	Analog Input Pin to be Selected
0	1	1	1	1	0	ADCAnI14S (Physical channel ANIn30)* <sup>1</sup>
0	1	1	1	1	1	ADCAnI15S (Physical channel ANIn31)* <sup>1</sup>
1	0	0	0	0	0	ADCAnI16S (Physical channel ANIn32)* <sup>1</sup>
1	0	0	0	0	1	ADCAnI17S (Physical channel ANIn33)* <sup>1</sup>
1	0	0	0	1	0	ADCAnI18S (Physical channel ANIn34)* <sup>1</sup>
1	0	0	0	1	1	ADCAnI19S (Physical channel ANIn35)* <sup>1</sup>
1	0	0	1	0	0	Diagnosis channel for A/D converter
Other than above						Setting prohibited

Note 1. This setting only applies to ADCA0. Setting is prohibited in ADCA1 (RH850/F1L with 176 pins).

### 29.3.2.2 ADCAnPWDVCR — PWM-Diag Virtual Channel Register

This register is used to indicate virtual channel setting (PWSAnPVCRx\_y register setting) of the PWM-Diag (SG4).

**Access:** ADCAnPWDVCR register is a read-only register that can be read in 32-bit units.  
 ADCAnPWDVCRL register is a read-only register that can be read in 16-bit units.  
 ADCAnPWDVCRL register is a read-only register that can be read in 8-bit units.  
 ADCAnPWDVCRLH register is a read-only register that can be read in 8-bit units.

**Address:** ADCAnPWDVCR: <ADCAn\_base> + 0F4<sub>H</sub>  
 ADCAnPWDVCRL: <ADCAn\_base> + 0F4<sub>H</sub>  
 ADCAnPWDVCRL: <ADCAn\_base> + 0F4<sub>H</sub>  
 ADCAnPWDVCRLH: <ADCAn\_base> + 0F4<sub>H</sub> + 1<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXE* <sup>1</sup>	MPXV[2:0]* <sup>1</sup>				—	—	—	ULS[1:0]	GCTRL[5:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 29.15 ADCAnPWDVCR Register Contents**

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	MPXE* <sup>1</sup>	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[27] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[11] (even channel) bit.  MPX Enable Set this bit to 1 when an external analog multiplexer is used. 0: The use of MPX is prohibited. 1: The use of MPX is permitted. The MPXV[2:0] bits are output from ADCAnSEL0 to ADCAnSEL2 when the virtual channel starts, and a wait of one A/D-conversion time is inserted before A/D conversion is performed.
14 to 12	MPXV[2:0] <sup>1</sup>	These bits are used to set the MPX value to be transferred to an external analog multiplexer by using the PWSAnPVCRx_y.PWSAnVRDTy[26:24] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[10:8] (even channel) bit.
11 to 8	Reserved	When read, the value after reset is returned.

**Table 29.15 ADCA<sub>n</sub>PWDVCR Register Contents**

Bit Position	Bit Name	Function
7 to 6	ULS[1:0]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[23:22] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[7:6] (even channel) bit. Upper Limit/Lower Limit Table Select 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCA <sub>n</sub> ULLMTBR0. 10: Upper limit and lower limit are checked for ADCA <sub>n</sub> ULLMTBR1. 11: Upper limit and lower limit are checked for ADCA <sub>n</sub> ULLMTBR2.
5 to 0	GCTRL[5:0]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[21:16] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[5:0] (even channel) bit.  Physical Channel Select These bits are used to specify a physical channel to be assigned to virtual channel j. For the selection of the channel, see <b>Table 29.14, Selection of Physical Channels</b> .

Note 1. These bits are only supported for ADCA0. For ADCA1, when read, the value after reset is returned.

### 29.3.2.3 ADCA<sub>n</sub>DR<sub>j</sub> — Data Register j

This register is a 32-/16-bit read-only register that stores the A/D conversion results corresponding to ADCA<sub>n</sub>VCR<sub>j</sub> and ADCA<sub>n</sub>VCR(j+1). As the A/D conversion results, the conversion result for ADCA<sub>n</sub>VCR(j+1) is stored in the upper bits (ADCA<sub>n</sub>DR(j+1)), and the conversion result for ADCA<sub>n</sub>VCR<sub>j</sub> is stored in the lower bits (ADCA<sub>n</sub>DR<sub>j</sub>).

**Access:** ADCA<sub>n</sub>DR<sub>j</sub> register is a read-only register that can be read in 32-bit units.  
ADCA<sub>n</sub>DR<sub>j</sub>L and ADCA<sub>n</sub>DR<sub>j</sub>H registers are the read-only registers that can be read in 16-bit units.

**Address:** ADCA<sub>n</sub>DR<sub>j</sub>: <ADCA<sub>n</sub>\_base> + 100<sub>H</sub> + j × 2<sub>H</sub>  
ADCA<sub>n</sub>DR<sub>j</sub>L: <ADCA<sub>n</sub>\_base> + 100<sub>H</sub> + j × 2<sub>H</sub>  
ADCA<sub>n</sub>DR<sub>j</sub>H: <ADCA<sub>n</sub>\_base> + 100<sub>H</sub> + j × 2<sub>H</sub> + 2<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR(j+1)[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR <sub>j</sub> [15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 29.16 ADCA<sub>n</sub>DR<sub>j</sub> Register Contents**

Bit Position	Bit Name	Function
31 to 16	DR(j+1)[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCA <sub>n</sub> VCR(j+1) are transferred.)
15 to 0	DR <sub>j</sub> [15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCA <sub>n</sub> VCR <sub>j</sub> are transferred.)

**CAUTION**

If the number of channels is odd, the higher-order bits (DR(j+1)[15:0]) in the ADCAnDRj register cannot be used. If virtual channels 33, 34, and 35 are used exclusively for self-diagnosis, the lower-order bits (DRj[15:0]) for channel 32 cannot be used.

**NOTES**

1. j = 00, 02, ..., 46, 48 (for ADCA0 of the RH850/F1L with 176 pins)  
j = 00, 02, ..., 20, 22 (for ADCA1 of the RH850/F1L with 176 pins)
2. By controlling ADCAnADCR.CRAC and ADCAnADCR.CTYP, the data format of this register becomes as follows:
  - ADCAnADCR.CTYP = 0 and ADCAnADCR.CRAC = 0 → Right alignment is used.  
→The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 27 to 16, and the A/D conversion result for ADCAnVCRj is transferred to bits 11 to 0.
  - ADCAnADCR.CTYP = 0 and ADCAnADCR.CRAC = 1 → Left alignment is used.  
→The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 31 to 20, and the A/D conversion result for ADCAnVCRj is transferred to bits 15 to 4.
  - ADCAnADCR.CTYP = 1 and ADCAnADCR.CRAC = 0 → Right alignment is used.  
→The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 25 to 16, and the A/D conversion result for ADCAnVCRj is transferred to bits 9 to 0.
  - ADCAnADCR.CTYP = 1 and ADCAnADCR.CRAC = 1 → Left alignment is used.  
→The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 31 to 22, and the A/D conversion result for ADCAnVCRj is transferred to bits 15 to 6.

### 29.3.2.4 ADCAnDIRj — Data Supplementary Information Register j

This register is a 32-bit read-only register that stores the A/D conversion result for ADCAnDRj and information incidental to the A/D converted value.

As the A/D conversion result, the ADCAnDRj value is transferred. As information incidental to the A/D converted value, information about the write flag (WFLG), the MPX value (MPXV[2:0]), and the physical channel (ID[5:0]) is transferred. The data format of the A/D conversion result stored in ADCAnDIRj is the same as that for the ADCAnDRj register.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <ADCAn\_base> + 200H + j × 4H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE*1	MPXV[2:0]*1			—	—	WFLG	—	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.17 ADCAnDIRj Register Contents

Bit Position	Bit Name	Function
31	MPXE*1	MPX Enable Flag 0: MPX function is not used. 1: MPX function is used.
30 to 28	MPXV[2:0]*1	These bits are used to store the MPX value. The MPX value to be stored is the MPX value of the most recent conversion result.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: ADCAnDRj or ADCAnDIRj is read (cleared when read). 1: A/D converted value is stored in ADCAnDRj (set when the value is stored).
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	These bits store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the number corresponding to the most recent conversion result.
15 to 0	DR[15:0]	These bits are used to store the A/D conversion result.

Note 1. These bits are only supported by ADCA0.  
For ADCA1, when read, the value after reset is returned.

### 29.3.2.5 ADCAnPWDTSNDR — PWM-Diag Data Register

This register is a 32-/16-bit read-only register that stores the A/D conversion results corresponding to the PWM-Diag. As the A/D conversion results, the conversion result for the PWM-Diag (PWDDR) is stored in the upper bits.

**Access:** ADCAnPWDTSNDR register is a read-only register that can be read in 32-bit units.  
ADCAnPWDTSNDRH register is a read-only register that can be read in 16-bit units.

**Address:** ADCAnPWDTSNDR: <ADCAn\_base> + 178H  
ADCAnPWDTSNDRH: <ADCAn\_base> + 17AH

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWDDR[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.18 ADCAnPWDTSNDR Register Contents

Bit Position	Bit Name	Function
31 to 16	PWDDR[15:0]	These bits are used to store the A/D conversion result data for the PWM-Diag.
15 to 0	Reserved	When read, the value after reset is returned.

#### NOTE

The data format of this register is controlled by ADCAnADCR.CRAC and ADCAnADCR.CTYP, as shown below.

- ADCAnADCR.CTYP = 0, ADCAnADCR.CRAC = 0 → Right alignment is used.  
→ The A/D conversion result for ADCAnPWDVCR is transferred to bits 27 to 16.
- ADCAnADCR.CTYP = 0, ADCAnADCR.CRAC = 1 → Left alignment is used.  
→ The A/D conversion result for ADCAnPWDVCR is transferred to bits 31 to 20.
- ADCAnADCR.CTYP = 1, ADCAnADCR.CRAC = 0 → Right alignment is used.  
→ The A/D conversion result for ADCAnPWDVCR is transferred to bits 25 to 16.
- ADCAnADCR.CTYP = 1, ADCAnADCR.CRAC = 1 → Left alignment is used.  
→ The A/D conversion result for ADCAnPWDVCR is transferred to bits 31 to 22.

### 29.3.2.6 ADCAnPWDDIR — PWM-Diag Data Supplementary Information Register

This register is a 32-bit read-only register that stores the A/D conversion result when PWM-Diag is used, and information incidental to the A/D converted value.

As the A/D conversion result, the ADCAnPWDSNDR.PWDDR[15:0] value is transferred. As supplementary information to the A/D converted value, the write flag (WFLG), MPX value (MPXV[2:0]), and physical channel (ID[5:0]) are transferred. The data format of the A/D conversion result stored in ADCAnPWDDIR is the same as that for the ADCAnPWDTSNDR register.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** <ADCAn\_base> + 2F4H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE*1	MPXV[2:0]*1			—	—	WFLG	—	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.19 ADCAnPWDDIR Register Contents

Bit Position	Bit Name	Function
31	MPXE*1	MPX Enable Flag 0: The MPX function is not used. 1: The MPX function is used.
30 to 28	MPXV[2:0]*1	These bits are used to store the MPX value. The MPX value to be stored is the MPX value of the most recent conversion result.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: ADCAnPWDTSNDR or ADCAnPWDDIR is read (cleared when read). 1: The A/D converted value is stored in ADCAnPWDTSNDR (set when the value is stored).
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	These bits are used to store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the physical channel number corresponding to the most recent conversion result.
15 to 0	PWDDR[15:0]	These bits are used to store the A/D conversion result for PWM-Diag.

Note 1. These bits are only supported for ADCA0.  
For ADCA1, when read, the value after reset is returned.

### 29.3.2.7 ADCAnADHALTR — A/D Force Halt Register

This register is used to halt conversion for all SGs of ADCAn. The read value is always 0.

**Access:** ADCAnADHALTR register is a write-only register that can be written in 32-bit units.  
 ADCAnADHALTRL register is a write-only register that can be written in 16-bit units.  
 ADCAnADHALTRLL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnADHALTR:<ADCAn\_base> + 300H  
 ADCAnADHALTRL:<ADCAn\_base> + 300H  
 ADCAnADHALTRLL:<ADCAn\_base> + 300H

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HALT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 29.20 ADCAnADHALTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HALT	ADCA Force Halt Trigger All scan groups are halted and initialized, and ADCA becomes idle state. Writing of 0: No effect Writing of 1: Scan groups are halted.

### 29.3.2.8 ADCAnADCR — A/D Control Register

This register is used for ADCAn common control.

**Access:** ADCAnADCR register can be read or written in 32-bit units.  
 ADCAnADCRL register can be read or written in 16-bit units.  
 ADCAnADCRLL register can be read or written in 8-bit units.

**Address:** ADCAnADCR: <ADCAn\_base> + 304H  
 ADCAnADCRL: <ADCAn\_base> + 304H  
 ADCAnADCRLL: <ADCAn\_base> + 304H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	DGON	—	CRAC	CTYP	—	—	SUSMTD[1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 29.21 ADCAnADCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DGON	Self-Diagnostic Voltage Standby Control 0: The self-diagnostic voltage circuit is turned off. 1: The self-diagnostic voltage circuit is turned on, or the reference voltage is updated.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CRAC	Alignment Control 0: The results of conversion to PWDDR and ADCAnDRj are stored right-aligned. 1: The results of conversion to PWDDR and ADCAnDRj are stored left-aligned.
4	CTYP	12/10 Bit Select Mode 0: 12-bit mode 1: 10-bit mode
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SUSMTD [1:0]	Suspend Mode Select These bits are used to select the suspend method when a higher-priority scan group interrupts a lower-priority scan group. 00: Synchronous suspend when a higher-priority SG or SVSTOP interrupt. 01: Asynchronous suspend when a higher-priority SG (SG2, SG3, SG4) and SVSTOP interrupt SG1, and synchronous suspend when a higher-priority SG (SG3, SG4) and SVSTOP interrupt SG2, or when a higher-priority SG (SG4) and SVSTOP interrupt SG3. 10: Asynchronous suspend when a higher-priority SG or SVSTOP interrupt. 11: Setting prohibited

#### CAUTION

To prevent malfunction, ADCAnADCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

---

**NOTE**

- Synchronous suspend:  
If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the A/D conversion for the higher-priority SG is performed after the on-going A/D conversion of a channel is completed. After processing for the higher-priority SG is completed, the suspended channel processing for the lower-priority SG is resumed.
- Asynchronous suspend:  
If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the on-going channel processing is suspended, and then the A/D conversion for the higher-priority SG is performed. After processing for the higher-priority SG is completed, the suspended A/D channel conversion for the lower-priority SG is resumed.

For details, see **Figure 29.21, Example of Synchronous Suspend and Resume Operation** and **Figure 29.22, Example of Asynchronous Suspend and Resume Operation**.

---

### 29.3.2.9 ADCAnMPXCURR — MPX Current Register

This register is used to store the MPX value for an external analog multiplexer.

**Access:** ADCAnMPXCURR register is a read-only register that can be read in 32-bit units.  
 ADCAnMPXCURRL register is a read-only register that can be read in 16-bit units.  
 ADCAnMPXCURRL register is a read-only register that can be read in 8-bit units.

**Address:** ADCAnMPXCURR: <ADCAn\_base> + 30CH  
 ADCAnMPXCURRL: <ADCAn\_base> + 30CH  
 ADCAnMPXCURRL: <ADCAn\_base> + 30CH

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPXCUR[2:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.22 ADCAnMPXCURR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	MPXCUR[2:0]	These bits are used to store the current MPX value. If conversion of a virtual channel starts after setting ADCAnVCRj.MPXE to 1, the setting of ADCAnVCRj.MPXV[2:0] is stored. If conversion of a virtual channel starts after setting ADCAnPWDVCR.MPXE to 1, the setting of ADCAnPWDVCR.MPXV[2:0] is stored.

#### NOTE

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.10 ADCAnTHSMPSTCR — T&H Sampling Start Control Register

This register is used to control the start of sampling for all T&H<sub>k</sub> (k = 0 to 5). The bits are always read as 0.

**Access:** ADCAnTHSMPSTCR register is a write-only register that can be written in 32-bit units.  
 ADCAnTHSMPSTCTRL register is a write-only register that can be written in 16-bit units.  
 ADCAnTHSMPSTCRL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnTHSMPSTCR: <ADCAn\_base> + 314<sub>H</sub>  
 ADCAnTHSMPSTCTRL: <ADCAn\_base> + 314<sub>H</sub>  
 ADCAnTHSMPSTCRL: <ADCAn\_base> + 314<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SMPST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 29.23 ADCAnTHSMPSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SMPST	T&H Sampling Start Control Trigger 0: No effect 1: Sampling for all T&H is started.

The conditions to place the T&H circuit in the sampling state are as follows:

- Condition to start sampling while T&H is stopped:  
1 being written to ADCAnTHSMPSTCR.SMPST while ADCAnTHER.THkE = 1 (k = 0 to 5).
- Condition to start continuous sampling in automatic sampling:  
A/D conversion of the hold value for T&H<sub>k</sub> being completed while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHCR.ASMPMSK = 1.

#### NOTE

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.11 ADCAnTHCR — T&H Control Register

This register controls the sampling transition after A/D conversion of the hold value for T&H is completed.

Automatic start of sampling on the T&H circuit after A/D conversion of the hold value for T&H is completed shortens the time required for the generation of succeeding hold completion triggers.

**Access:** ADCAnTHCR register can be read or written in 32-bit units.  
 ADCAnTHCRL register can be read or written in 16-bit units.  
 ADCAnTHCRL register can be read or written in 8-bit units.

**Address:** ADCAnTHCR: <ADCAn\_base> + 318<sub>H</sub>  
 ADCAnTHCRL: <ADCAn\_base> + 318<sub>H</sub>  
 ADCAnTHCRL: <ADCAn\_base> + 318<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASMPM SK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.24 ADCAnTHCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ASMPMSK	Automatic Sampling Mask Control 0: Automatic sampling is not performed. 1: Automatic sampling is performed.

#### CAUTION

To prevent malfunction, ADCAnTHCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

#### NOTE

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.12 ADCAnTHAHDSTCR — T&H Group A Hold Start Control Register

This register is used to control the start of the hold for T&H group A. The bits are always read as 0.

**Access:** ADCAnTHAHDSTCR register is a write-only register that can be written in 32-bit units.  
 ADCAnTHAHDSTCRL register is a write-only register that can be written in 16-bit units.  
 ADCAnTHAHDSTCRL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnTHAHDSTCR: <ADCAn\_base> + 31C<sub>H</sub>  
 ADCAnTHAHDSTCRL: <ADCAn\_base> + 31C<sub>H</sub>  
 ADCAnTHAHDSTCRL: <ADCAn\_base> + 31C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 29.25 ADCAnTHAHDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Group A Hold Start Control Trigger 0: No effect 1: Hold for T&H group A is started.

The condition to place T&H group A in the hold state is as follows:

- 1 being written to ADCAnTHAHDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 0 (k = 0 to 5).

#### NOTE

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.13 ADCAnTHBHDSTCR — T&H Group B Hold Start Control Register

This register is used to control the start of the hold for T&H group B. The bits are always read as 0.

**Access:** ADCAnTHBHDSTCR register is a write-only register that can be written in 32-bit units.  
 ADCAnTHBHDSTCRL register is a write-only register that can be written in 16-bit units.  
 ADCAnTHBHDSTCRL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnTHBHDSTCR: <ADCAn\_base> + 320<sub>H</sub>  
 ADCAnTHBHDSTCRL: <ADCAn\_base> + 320<sub>H</sub>  
 ADCAnTHBHDSTCRL: <ADCAn\_base> + 320<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 29.26 ADCAnTHBHDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Group B Hold Start Control Trigger 0: No effect 1: Hold for T&H group B is started.

The condition to place T&H group B in the hold state is as follows:

- 1 being written to ADCAnTHBHDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 1 (k = 0 to 5).

#### NOTE

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.14 ADCAnTHACR — T&H Group A Control Register

This register is used to control T&H group A.

**Access:** ADCAnTHACR register can be read or written in 32-bit units.  
 ADCAnTHACRL register can be read or written in 16-bit units.  
 ADCAnTHACRLL register can be read or written in 8-bit units.

**Address:** ADCAnTHACR: <ADCAn\_base> + 324<sub>H</sub>  
 ADCAnTHACRL: <ADCAn\_base> + 324<sub>H</sub>  
 ADCAnTHACRLL: <ADCAn\_base> + 324<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 29.27 ADCAnTHACR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	T&H Group A Hold Completion Trigger Enable This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis does not proceed. 1: Self-diagnosis proceeds. Note: The SGx_TRG (x = 1 to 3) trigger is selected for the trigger input of the scan group that is not selected in SGS[1:0] of ADCAnTHACR and SGS[1:0] of ADCAnTHBCR.
4	HLDTE	T&H Group A Hold Trigger Enable 0: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is selected for the hold start trigger of T&H group A. 1: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is not selected for the hold start trigger of T&H group A. Note: ADCAnTHAHLDSCTR.HLDST becomes a hold start trigger regardless of the ADCAnTHACR.HLDTE setting. Set this bit to 0 when self-diagnosis of the T&H circuit is to proceed.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SGS[1:0]	T&H Group A Scan Group Select 00: No scan group is selected for T&H group A. 01: SG1 is selected for T&H group A. 10: SG2 is selected for T&H group A. 11: SG3 is selected for T&H group A. Note: 1. If ADCAnTHACR.SGS[1:0] is set to 0 <sub>H</sub> , T&H does not operate. When you enable T&Hk in ADCAnTHER.THkE, make sure that a scan group is specified in SGS[1:0]. 2. Selecting the same scan group as T&H group B is prohibited.

#### CAUTION

To prevent malfunction, ADCAnTHACR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

**NOTE**

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.15 ADCAnTHBCR — T&H Group B Control Register

This register is used to control T&H group B.

**Access:** ADCAnTHBCR register can be read or written in 32-bit units.  
 ADCAnTHBCRL register can be read or written in 16-bit units.  
 ADCAnTHBCRLL register can be read or written in 8-bit units.

**Address:** ADCAnTHBCR: <ADCAn\_base> + 328<sub>H</sub>  
 ADCAnTHBCRL: <ADCAn\_base> + 328<sub>H</sub>  
 ADCAnTHBCRLL: <ADCAn\_base> + 328<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

**Table 29.28 ADCAnTHBCR Register Contents**

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	T&H Group B Hold Completion Trigger Enable This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis does not proceed. 1: Self-diagnosis proceeds.  Note: The SGx_TRG (x = 1 to 3) trigger is selected for the trigger input of the scan group that is not selected in SGS[1:0] of ADCAnTHACR and SGS[1:0] of ADCAnTHBCR.
4	HLDTE	T&H Group B Hold Trigger Enable 0: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is selected for the hold start trigger of T&H group B. 1: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is not selected for the hold start trigger of T&H group B.  Note: ADCAnTHBHDSTCR.HLDST becomes a hold start trigger regardless of the ADCAnTHBCR.HLDTE setting. Set this bit to 0 when self-diagnosis of the T&H circuit is to proceed.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

**Table 29.28 ADCAAnTHBCR Register Contents**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
1 to 0	SGS[1:0]	<p>T&amp;H Group B Scan Group Select            00: No scan group is selected for T&amp;H group B.            01: SG1 is selected for T&amp;H group B.            10: SG2 is selected for T&amp;H group B.            11: SG3 is selected for T&amp;H group B.</p> <p><b>Note:</b></p> <ol style="list-style-type: none"> <li>If ADCAAnTHBCR.SGS[1:0] is set to 0<sub>H</sub>, T&amp;H does not operate.            When you enable T&amp;Hk in ADCAAnTHER.THkE, make sure that a scan group is specified in SGS[1:0].</li> <li>Selecting the same scan group as T&amp;H group A is prohibited.</li> </ol>

**CAUTION**

To prevent malfunction, ADCAAnTHBCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

**NOTE**

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.16 ADCAnTHER — T&H Enable Register

This register controls enabling and disabling of each T&H.

**Access:** ADCAnTHER register can be read or written in 32-bit units.  
 ADCAnTHERL register can be read or written in 16-bit units.  
 ADCAnTHERLL register can be read or written in 8-bit units.

**Address:** ADCAnTHER: <ADCAn\_base> + 32C<sub>H</sub>  
 ADCAnTHERL: <ADCAn\_base> + 32C<sub>H</sub>  
 ADCAnTHERLL: <ADCAn\_base> + 32C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	TH5E	TH4E	TH3E	TH2E	TH1E	TH0E	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.29 ADCAnTHER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TH5E	T&H5 Enable 0: T&H5 is disabled. 1: T&H5 is enabled. Note: If TH5E is set to 0, T&H5 is always stopped.
4	TH4E	T&H4 Enable 0: T&H4 is disabled. 1: T&H4 is enabled. Note: If TH4E is set to 0, T&H4 is always stopped.
3	TH3E	T&H3 Enable 0: T&H3 is disabled. 1: T&H3 is enabled. Note: If TH3E is set to 0, T&H3 is always stopped.
2	TH2E	T&H2 Enable 0: T&H2 is disabled. 1: T&H2 is enabled Note: If TH2E is set to 0, T&H2 is always stopped.
1	TH1E	T&H1 Enable 0: T&H1 is disabled. 1: T&H1 is enabled. Note: If TH1E is set to 0, T&H1 is always stopped.
0	TH0E	T&H0 Enable 0: T&H0 is disabled. 1: T&H0 is enabled Note: If TH0E is set to 0, T&H0 is always stopped.

#### CAUTION

To prevent malfunction, ADCAnTHER should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

**NOTE**

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.17 ADCAnTHGSR — T&H Group Select Register

This register is used to select a T&H group for each T&H.

**Access:** ADCAnTHGSR register can be read or written in 32-bit units.  
 ADCAnTHGSRL register can be read or written in 16-bit units.  
 ADCAnTHGSRLL register can be read or written in 8-bit units.

**Address:** ADCAnTHGSR: <ADCAn\_base> + 330<sub>H</sub>  
 ADCAnTHGSRL: <ADCAn\_base> + 330<sub>H</sub>  
 ADCAnTHGSRLL: <ADCAn\_base> + 330<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TH5GS	TH4GS	TH3GS	TH2GS	TH1GS	TH0GS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

**Table 29.30 ADCAnTHGSR Register Contents**

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TH5GS	T&H5 Group Select 0: T&H5 is selected to group A. 1: T&H5 is selected to group B.
4	TH4GS	T&H4 Group Select 0: T&H4 is selected to group A. 1: T&H4 is selected to group B.
3	TH3GS	T&H3 Group Select 0: T&H3 is selected to group A. 1: T&H3 is selected to group B.
2	TH2GS	T&H2 Group Select 0: T&H2 is selected to group A. 1: T&H2 is selected to group B.
1	TH1GS	T&H1 Group Select 0: T&H1 is selected to group A. 1: T&H1 is selected to group B.
0	TH0GS	T&H0 Group Select 0: T&H0 is selected to group A. 1: T&H0 is selected to group B.

**CAUTION**

- Do not set T&H0 to T&H2 to the same group as T&H3 to T&H5.  
Example
  - Group A: 0ch, 1ch, 2ch  
Group B: 3ch, 4ch, 5ch → Setting allowed
  - Group A: 0ch  
Group B: 1ch, 2ch → Setting allowed
  - Group A: 0ch, 1ch, 3ch  
Group B: 2ch, 4ch → Setting prohibited
- To prevent malfunction, ADCAnTHGSR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

**NOTE**

In RH850/F1L, only ADCA0 supports this function.

### 29.3.2.18 ADCAnSMPCTR — Sampling Control Register

This register controls sampling.

ADCTLnSMPCTR controls the sampling time for SG4 (PWM-Diag) and SG1 to SG3.

**Access:** ADCAnSMPCTR register can be read or written in 32-bit units.  
 ADCAnSMPCRL register can be read or written in 16-bit units.  
 ADCAnSMPCRLL register can be read or written in 8-bit units.

**Address:** ADCAnSMPCTR: <ADCAn\_base> + 380<sub>H</sub>  
 ADCAnSMPCRL: <ADCAn\_base> + 380<sub>H</sub>  
 ADCAnSMPCRLL: <ADCAn\_base> + 380<sub>H</sub>

**Value after reset:** 0000 0018<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SMPT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.31 ADCAnSMPCTR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	SMPT[7:0]	These bits are used to set the sampling time (the number of cycles). $12_{H}$ : 18 cycles (PCLK = 8 MHz to 32 MHz) $18_{H}$ : 24 cycles (PCLK = 8 MHz to 40 MHz) Settings other than above are prohibited.

#### CAUTION

- To prevent malfunction, ADCTLnSMPCTR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
- When SMPT is changed, the A/D conversion wait time is also changed when MPX is used by virtual channel register j (ADCAnVCRj) or PWM-Diag virtual channel register (ADCAnPWDVCR).

### 29.3.2.19 ADCAnSFTCR — Safety Control Register

This is a register for safety control.

**Access:** ADCAnSFTCR register can be read or written in 32-bit units.  
 ADCAnSFTCRL register can be read or written in 16-bit units.  
 ADCAnSFTCRLL register can be read or written in 8-bit units.

**Address:** ADCAnSFTCR: <ADCAn\_base> + 334<sub>H</sub>  
 ADCAnSFTCRL: <ADCAn\_base> + 334<sub>H</sub>  
 ADCAnSFTCRLL: <ADCAn\_base> + 334<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RDCLR E	ULEIE	OWEIE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 29.32 ADCAnSFTCR Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	RDCLRE	Read & Clear Enable When the A/D conversion result is read, this bit selects whether the A/D conversion result is cleared by hardware. 0: ADCAnPWDTNSDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are not cleared when ADCAnPWDTNSDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read. 1: ADCAnPWDTNSDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are cleared when ADCAnPWDTNSDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read.  WFLG of ADCAnDIRj is cleared regardless of the RDCLRE setting when ADCAnDRj or ADCAnDIRj is read.
3	ULEIE	A/D Error Interrupt (INT_ADE) Enable on Upper/Lower Limit Error Detection 0: Disabled 1: Enabled
2	OWEIE	A/D Error Interrupt (INT_ADE) Enable on Overwrite Error Detection 0: Disabled 1: Enabled
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

#### CAUTION

To prevent malfunction, ADCAnSFTCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

### 29.3.2.20 ADCAnULLMTBR0 to 2 — Upper Limit/Lower Limit Table Registers 0 to 2

These registers are used to set the threshold for detection of an upper limit or lower limit error in the A/D converted value. Any of ADCAnULLMTBR0 to ADCAnULLMTBR2 is specified by setting ADCAnPWDVCR.ULS[1:0] and ADCAnVCRj.ULS[1:0] and compared with ADCAnPWDTSNDR and ADCAnDRj.

**Access:** ADCAnULLMTBR0 to 2 register can be read or written in 32-bit units.  
ADCAnULLMTBR0L to 2L and ADCAnULLMTBR0H to 2H registers can be read or written in 16-bit units.

**Address:** ADCAnULLMTBR0: <ADCAn\_base> + 338<sub>H</sub>  
ADCAnULLMTBR1: <ADCAn\_base> + 33C<sub>H</sub>  
ADCAnULLMTBR2: <ADCAn\_base> + 340<sub>H</sub>

**Address:** ADCAnULLMTBR0L: <ADCAn\_base> + 338<sub>H</sub>  
ADCAnULLMTBR1L: <ADCAn\_base> + 33C<sub>H</sub>  
ADCAnULLMTBR2L: <ADCAn\_base> + 340<sub>H</sub>

**Address:** ADCAnULLMTBR0H: <ADCAn\_base> + 338<sub>H</sub> + 2<sub>H</sub>  
ADCAnULLMTBR1H: <ADCAn\_base> + 33C<sub>H</sub> + 2<sub>H</sub>  
ADCAnULLMTBR2H: <ADCAn\_base> + 340<sub>H</sub> + 2<sub>H</sub>

**Value after reset:** FFF0 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ULMTB[11:0]																
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R
LLMTB[11:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Table 29.33 ADCAnULLMTBR0 to 2 Registers Contents

Bit Position	Bit Name	Function
31 to 20	ULMTB[11:0]	Upper Limit Table Specify the threshold for detection of an upper limit error in the A/D converted value. The upper limit error (ADCAnULER.UE) is set when the following condition is met: ULMTB[11:0] < A/D converted value
19 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 4	LLMTB[11:0]	Lower Limit Table Specify the threshold for detection of a lower limit error in the A/D converted value. The lower limit error (ADCAnULER.LE) is set when the following condition is met: LLMTB[11:0] > A/D converted value
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

#### CAUTION

- When A/D conversion is executed in 10-bit mode (ADCAnADCR.CTYP = 1), ULMTB[1:0] and LLMTB[1:0] should be set to 11<sub>B</sub> and 00<sub>B</sub>, respectively.
- To prevent malfunction, ADCAnULLMTBR0 to 2 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
- The upper-limit table (ULMTB[11:0]) must be greater than the lower-limit table (LLMTB[11:0]).

### 29.3.2.21 ADCAnECR — Error Clear Register

This register is used to control clearing of an error. The read value is always 0.

**Access:** ADCAnECR register is a write-only register that can be written in 32-bit units.  
 ADCAnECRL register is a write-only register that can be written in 16-bit units.  
 ADCAnECRLL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnECR: <ADCAn\_base> + 344<sub>H</sub>  
 ADCAnECRL: <ADCAn\_base> + 344<sub>H</sub>  
 ADCAnECRLL: <ADCAn\_base> + 344<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ULEC	OWEC	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	W	W	R	R	R

Table 29.34 ADCAnECR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	ULEC	Upper Limit Error Flag (ADCAnULER.UE) / Lower Limit Error Flag (ADCAnULER.LE) Clear, Upper/Lower Limit Error Capture (ADCAnULER.ULECAP[5:0]), Scan Group Bit (ULSG[1:0]) When Upper/Lower Limit Error Occurs, MPX Usage Bit (MPXE), and the MPX Value Storing Bit (MPXV[2:0]) Clear When Upper/Lower Limit Error Occurs 0: No effect. 1: Clears the flag.
2	OWEC	Overwrite Error Flag (ADCAnOWER.OWE) and Overwrite Error Capture (ADCAnOWER.OWECAP[5:0]) Clear 0: No effect. 1: Clears the flag.
1, 0	Reserved	When writing, write the value after reset.

### 29.3.2.22 ADCAnULER — Upper Limit/Lower Limit Error Register

This register is a read-only register that indicates information regarding the upper limit/lower limit errors.

**Access:** ADCAnULER register is a read-only register that can be read in 32-bit units.  
ADCAnULERL register is a read-only register that can be read in 16-bit units.

ADCAnULERLH register is a read-only register that can be read in 8-bit units.  
ADCAnULERLL register is a read-only register that can be read in 8-bit units.

**Address:** ADCAnULER: <ADCAn\_base> + 348<sub>H</sub>  
ADCAnULERL: <ADCAn\_base> + 348<sub>H</sub>

ADCAnULERLL: <ADCAn\_base> + 348<sub>H</sub>  
ADCAnULERLH: <ADCAn\_base> + 348<sub>H</sub> + 1<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UE	LE	ULSG[1:0]	MPXE	MPXV[2:0]	—	—	—	—	—	—	—	—	ULECAP[5:0]	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.35 ADCAnULER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	UE	<p>Upper Limit Error Flag 0: An upper limit error is not detected. 1: An upper limit error is detected.</p> <p>Setting condition: The A/D converted value exceeds the upper limit threshold specified by the upper limit/lower limit table registers 0 to 2 (ADCAnULLMTBR0 to 2). If a subsequent upper limit error is detected in A/D conversion while this bit is set to 1, the ADCAnULER register is not updated.</p> <p>Clearing condition: 1 is written to ADCAnECR.ULEC.</p>
14	LE	<p>Lower Limit Error Flag 0: A lower limit error is not detected. 1: A lower limit error is detected.</p> <p>Setting condition: The A/D converted value is lower than the lower limit threshold specified by the upper limit/lower limit table registers 0 to 2 (ADCAnULLMTBR0 to 2). If a subsequent lower limit error is detected in A/D conversion while this bit is set to 1, the ADCAnULER register is not updated.</p> <p>Clearing condition: 1 is written to ADCAnECR.ULEC.</p>
13, 12	ULSG[1:0]	<p>Scan Group where an Upper Limit/Lower Limit Error Occurs 00: No upper limit/lower limit error occurred. 01: The scan group where an upper limit/lower limit error occurred is SG1 to SG3. 10: The scan group where an upper limit/lower limit error occurred is PWM-Diag.</p> <p>Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0.</p> <p>Clearing condition: When 1 is written to ADCAnECR.ULEC.</p>

**Table 29.35 ADCA<sub>n</sub>ULER Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
11	MPXE <sup>*1</sup>	<p>MPX Usage</p> <p>0: The MPX function was not used when an upper limit/lower limit error occurred.</p> <p>1: The MPX function was used when an upper limit/lower limit error occurred.</p> <p>Capture condition:</p> <p>When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0</p> <p>Clearing condition:</p> <p>When 1 is written to ADCA<sub>n</sub>ECR.ULEC.</p>
10 to 8	MPXV[2:0] <sup>*1</sup>	<p>The value of MPX is stored when the errors of the upper and lower limit occurred</p> <p>Capture condition:</p> <p>When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0.</p> <p>Clearing condition:</p> <p>When 1 is written to ADCA<sub>n</sub>ECR.ULEC.</p>
7	Reserved	When read, an undefined value is read.
6	Reserved	When read, the value after reset is returned.
5 to 0	ULECAP[5:0]	<p>Upper Limit/Lower Limit Error Capture</p> <p>The physical channel is captured when an upper limit/lower limit error occurred.</p> <p>Capturing condition:</p> <p>When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0.</p> <p>Clearing condition:</p> <p>1 is written to ADCA<sub>n</sub>ECR.ULEC.</p>

Note 1. These bits are only supported for ADCA0.  
For ADCA1, when read, the value after reset is returned.

#### NOTE

ADCA<sub>n</sub>ULER is updated when the A/D converted value is set in ADCA<sub>n</sub>DRj or ADCA<sub>n</sub>PWDTSNDR.

### 29.3.2.23 ADCAnOWER — Overwrite Error Register

This register is a 32/16/8-bit read-only register that indicates an overwrite error. The target for overwrite errors is SG1 to SG3, and not PWM-Diag.

**Access:** ADCAnOWER register is a read-only register that can be read in 32-bit units.  
 ADCAnOWERL register is a read-only register that can be read in 16-bit units.  
 ADCAnOWERLL register is a read-only register that can be read in 8-bit units.

**Address:** ADCAnOWER: <ADCAn\_base> + 34C<sub>H</sub>  
 ADCAnOWERL: <ADCAn\_base> + 34C<sub>H</sub>  
 ADCAnOWERLL: <ADCAn\_base> + 34C<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	OWE	—	—	—	—	—	—	—	OWECAP[5:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.36 ADCAnOWER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	OWE	Overwrite Error Flag 0: An overwrite error is not detected. 1: An overwrite error is detected. Setting condition: ADCAnDIRj.WFLG = 1, and the A/D converted value is written to ADCAnDRj. If a subsequent overwrite error is detected in A/D conversion while this bit is set to 1, the ADCAnOWER register is not updated. Clearing condition: 1 is written to ADCAnECR.OWEC.
6	Reserved	When read, the value after reset is returned.
5 to 0	OWECAP[5:0]	Overwrite Error Capture The virtual channel number is captured when an overwrite error occurs. Capturing condition: OWE = 0 and ADCAnDIRj.WFLG = 1, and the A/D converted value is written to ADCAnDRj Clearing condition: 1 is written to ADCAnECR.OWEC.

#### NOTE

ADCAnOWER is updated when the A/D converted value is set in ADCAnDRj.

### 29.3.3 Scan Group (SG) Specific Registers

This section describes the registers provided for each scan group.

#### 29.3.3.1 ADCAnSGSTCRx — Scan Group x Start Control Register

This register is used to control the start of scan group x. The read value is always 0.

**Access:** ADCAnSGSTCRx is a write-only register that can be written in 32-bit units.  
 ADCAnSGSTCRxL is a write-only register that can be written in 16-bit units.  
 ADCAnSGSTCRxLL is a write-only register that can be written in 8-bit units.

**Address:** ADCAnSGSTCRx: <ADCAn\_base> + 400<sub>H</sub> + x × 40<sub>H</sub>  
 ADCAnSGSTCRxL: <ADCAn\_base> + 400<sub>H</sub> + x × 40<sub>H</sub>  
 ADCAnSGSTCRxLL: <ADCAn\_base> + 400<sub>H</sub> + x × 40<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

**Table 29.37 ADCAnSGSTCRx Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SGST	Scan Group Start Trigger Writing 1 to SGST while ADCAnSGSTR.SGACT[3:1] = 0 starts the target SGx.

### 29.3.3.2 ADCAnSGCRx — Scan Group x Control Register

This register controls scan group x.

**Access:** ADCAnSGCRx register can be read or written in 32-bit units.  
 ADCAnSGCRxL register can be read or written in 16-bit units.  
 ADCAnSGCRxLL register can be read or written in 8-bit units.

**Address:** ADCAnSGCRx: <ADCAn\_base> + x × 40H + 408H  
 ADCAnSGCRxL: <ADCAn\_base> + x × 40H + 408H  
 ADCAnSGCRxLL: <ADCAn\_base> + x × 40H + 408H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	SCANMD	ADIE	SCT[1:0]	—	TRGMD	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W

Table 29.38 ADCAnSGCRx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	SCANMD	Scan Mode 0: Multicycle scan mode 1: Continuous scan mode Write 0 to this bit for SG2 and SG3.
4	ADIE	Scan End Interrupt Enable 0: INT_SGx is not output when the scan for SGx ends. 1: INT_SGx is output when the scan for SGx ends.
3, 2	SCT[1:0]	Channel Repeat Times Select 00: The selected number of channel repeat times is one. 01: The selected number of channel repeat times is two. 10: The selected number of channel repeat times is four. 11: Setting prohibited
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TRGMD	Trigger Mode 0: Trigger input to SGx_TRG is disabled (Hardware trigger disabled). 1: SGx_TRG start trigger or hold complete trigger A/B is selected for the trigger input to SGx. <b>NOTE</b> The software trigger is valid regardless of the TRGMD bit setting.

#### CAUTION

To prevent malfunction, ADCAnSGCRx should be set (except clearing TRGMD upon completion of A/D conversion) when SGACT of all scan groups is 0 (before the scan group is started) and TRGMD of all scan groups is 0.

### 29.3.3.3 ADCAnPWDSGCR — PWM-Diag Scan Group Control Register

This register is used to control PWM-Diag.

**Access:** ADCAnPWDSGCR register can be read or written in 32-bit units.  
 ADCAnPWDSGCRL register can be read or written in 16-bit units.  
 ADCAnPWDSGCRL register can be read or written in 8-bit units.

**Address:** ADCAnPWDSGCR: <ADCAn\_base> + 508H  
 ADCAnPWDSGCRL: <ADCAn\_base> + 508H  
 ADCAnPWDSGCRL: <ADCAn\_base> + 508H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDTR GMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.39 ADCAnPWDSGCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWDTRGMD	PWM-Diag Trigger Mode Select 0: PWSA_ADTRG trigger input is disabled. 1: PWSA_ADTRG is selected for the trigger input to the PWM-Diag scan group.

#### CAUTION

To prevent malfunction, ADCAnPWDSGCR should be set when SGACT of the PWM-Diag scan group (SG4) is 0 (before the scan group is started).

### 29.3.3.4 ADCAnSGVCSPx — Scan Group x Start Virtual Channel Pointer

This register specifies the start pointer of a virtual channel.

**Access:** ADCAnSGVCSPx register can be read or written in 32-bit units.  
 ADCAnSGVCSPxL register can be read or written in 16-bit units.  
 ADCAnSGVCSPxLL register can be read or written in 8-bit units.

**Address:** ADCAnSGVCSPx: <ADCAn\_base> + x × 40H + 40CH  
 ADCAnSGVCSPxL: <ADCAn\_base> + x × 40H + 40CH  
 ADCAnSGVCSPxLL: <ADCAn\_base> + x × 40H + 40CH

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VCSP[5:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 29.40 ADCAnSGVCSPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCSP[5:0]	Start Virtual Channel Pointer These bits are used to specify the virtual channel from which the SGx scan is to be started.

#### CAUTION

- ADCAnSGVCSPx must be equal to or smaller than ADCAnSGVCEPx.
- When writing to the channel pointers, be sure to write in the following order: ADCAnSGVCSPx → ADCAnSGVCEPx. When SGx is started, the A/D conversion for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is executed.
- Though ADCAnSGVCSPx can be written during the A/D conversion, the register is updated at the time when ADCAnSGVCEPx is written. The new setting is applied when SGx is started next time.
- When the hardware trigger is used, writing to this register during operation is prohibited.

### 29.3.3.5 ADCAnSGVCEPx — Scan Group x End Virtual Channel Pointer

This register specifies the end pointer of a virtual channel.

**Access:** ADCAnSGVCEPx register can be read or written in 32-bit units.  
 ADCAnSGVCEPxL register can be read or written in 16-bit units.  
 ADCAnSGVCEPxLL register can be read or written in 8-bit units.

**Address:** ADCAnSGVCEPx: <ADCAn\_base> + x × 40H + 410H  
 ADCAnSGVCEPxL: <ADCAn\_base> + x × 40H + 410H  
 ADCAnSGVCEPxLL: <ADCAn\_base> + x × 40H + 410H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCEP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.41 ADCAnSGVCEPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCEP[5:0]	End Virtual Channel Pointer These bits are used to specify the virtual channel at which the SGx scan is to be ended.

#### CAUTION

- ADCAnSGVCSPx must be equal to or smaller than ADCAnSGVCEPx.
- When SGx is started, processing for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is executed.
- ADCAnSGVCEPx can be rewritten even when SGx is being processed. The new setting is applied when SGx is started next time.

### 29.3.3.6 ADCAnSGMCYCRx — Scan Group x Multicycle Register

This register is a 32/16/8-bit read/write register that indicates the number of scan times in multicycle scan mode.

**Access:** ADCAnSGMCYCRx register can be read or written in 32-bit units.  
 ADCAnSGMCYCRxL register can be read or written in 16-bit units.  
 ADCAnSGMCYCRxLL register can be read or written in 8-bit units.

**Address:** ADCAnSGMCYCRx: <ADCAn\_base> +  $x \times 40_H + 414_H$   
 ADCAnSGMCYCRxL: <ADCAn\_base> +  $x \times 40_H + 414_H$   
 ADCAnSGMCYCRxLL: <ADCAn\_base> +  $x \times 40_H + 414_H$

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCYC[1:0]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 29.42 ADCAnSGMCYCRx Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MCYC[1:0]	Multicycle Number Specification These bits are used to specify the number of scan times in multicycle scan mode. 00 <sub>B</sub> : Number of scans = 1 01 <sub>B</sub> : Number of scans = 2 10 <sub>B</sub> : Setting prohibited 11 <sub>B</sub> : Number of scans = 4

#### CAUTION

- To prevent malfunction, ADCAnSGMCYCRx should be set when SGACT of scan group x is 0 (before the scan group is started) and TRGMD is 0.
- When SGx is started, the scan for the virtual channels within the range specified in ADCAnSGVCS Px and ADCAnSGVCE Px is repeatedly executed as many times as specified in ADCAnSGMCYCRx.

### 29.3.3.7 ADCAnPWDGSEFCR — PWM-Diag Scan End Flag Clear Register

This register is used to control the clearing of PWM-Diag scan end flag (SEF). The bits are always read as 0.

**Access:** ADCAnPWDGSEFCR register is a write-only register that can be written in 32-bit units.  
 ADCAnPWDGSEFCRL register is a write-only register that can be written in 16-bit units.  
 ADCAnPWDGSEFCRLL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnPWDGSEFCR: <ADCAn\_base> + 518<sub>H</sub>  
 ADCAnPWDGSEFCRL: <ADCAn\_base> + 518<sub>H</sub>  
 ADCAnPWDGSEFCRLL: <ADCAn\_base> + 518<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDSEFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 29.43 ADCAnPWDGSEFCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	PWDSEFC	PWM-Diag Scan End Flag Clear Trigger 0: No effect. 1: Clears the PWM-Diag scan end flag (ADCAnSGSTR.SEF[4]).

### 29.3.3.8 ADCAnSGSEFCRx — Scan Group x Scan End Flag Clear Register

This register is a write-only register that clears the scan end flag (ADCAnSGSTR.SEFx). The read value is always 0.

**Access:** ADCAnSGSEFCRx register is a write-only register that can be written in 32-bit units.

ADCAnSGSEFCRxL register is a write-only register that can be written in 16-bit units.

ADCAnSGSEFCRxLL register is a write-only register that can be written in 8-bit units.

**Address:** ADCAnSGSEFCRx: <ADCAn\_base> + x × 40<sub>H</sub> + 418<sub>H</sub>

ADCAnSGSEFCRxL: <ADCAn\_base> + x × 40<sub>H</sub> + 418<sub>H</sub>

ADCAnSGSEFCRxLL: <ADCAn\_base> + x × 40<sub>H</sub> + 418<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 29.44 ADCAnSGSEFCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SEFC	Scan End Flag Clear Trigger 0: No effect. 1: Clears the target SG scan end flag (ADCAnSGSTR.SEF[3:1]).

### 29.3.3.9 ADCAnSGSTR — Scan Group Status Register

This register indicates the state of T&H, SVSTOP, scan group x, and PWM-Diag scan group. The SHACT and SGACT bits are cleared when HALT is executed.

<b>Access:</b>	ADCAnSGSTR register is a read-only register that can be read in 32-bit units. ADCAnSGSTRL register is a read-only register that can be read in 16-bit units.															
<b>Address:</b>	ADCAnSGSTR: <ADCAn_base> + 308 <sub>H</sub> ADCAnSGSTRL: <ADCAn_base> + 308 <sub>H</sub>															
<b>Value after reset:</b>	0000 0000 <sub>H</sub>															
<b>Bit</b> 31    30    29    28    27    26    25    24    23    22    21    20    19    18    17    16																
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
<b>Bit</b> 15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0																
	—	SHACT	SGACT[5:1]					—	—	—	—	SEF[4:1]				—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.45 ADCAnSGSTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	SHACT	T&H Status Flag 0: T&H is stopped. 1: T&H conversion or sampling is in progress.
13	SGACT[5]	SVSTOP Status Flag 0: SVSTOP is canceled. 1: SVSTOP is accepted.
12	SGACT[4]	PWM-Diag Scan Group (SG4) Status Flag 0: A/D conversion for PWM-Diag (SG4) is completed. 1: A/D conversion for PWM-Diag (SG4) is in processing or suspension.
11	SGACT[3]	Scan Group 3 (SG3) Status Flag 0: A/D conversion for SG3 is completed. 1: A/D conversion for SG3 is in processing or suspension.
10	SGACT[2]	Scan Group 2 (SG2) Status Flag 0: A/D conversion for SG2 is completed. 1: A/D conversion for SG2 is in processing or suspension.
9	SGACT[1]	Scan Group 1 (SG1) Status Flag 0: A/D conversion for SG1 is completed. 1: A/D conversion for SG1 is in processing or suspension.
8 to 5	Reserved	When read, the value after reset is returned.
4	SEF[4]	PWM-Diag Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: <ul style="list-style-type: none"><li>• ADCAnPWDTSNDR for PMW-Diag is read.</li><li>• ADCAnPWDDIR for PWM-Diag is read.</li><li>• ADCAnPWDSEFCR.PWDSEFC is written as 1.</li></ul> 1: The A/D conversion result is written to ADCAnPWDTSNDR for PWM-Diag.

**Table 29.45 ADCAAnSGSTR Register Contents (2/2)**

<b>Bit Position</b>	<b>Bit Name</b>	<b>Function</b>
3	SEF[3]	<p>SG3 Scan End Flag Indicates the status of the scan result data.</p> <p>0: The flag is cleared when any of the following operations is performed:</p> <ul style="list-style-type: none"> <li>• ADCAAnDRj for the virtual channel which ADCAAnSGVCEP3 indicates is read.</li> <li>• ADCAAnDIRj for the virtual channel which ADCAAnSGVCEP3 indicates is read.</li> <li>• ADCAAnSGSEFCRx.SEFC is written as 1.</li> </ul> <p>1: The A/D conversion result is written to ADCAAnDRj for the virtual channel which ADCAAnSGVCEP3 indicates.</p>
2	SEF[2]	<p>SG2 Scan End Flag Indicates the status of the scan result data.</p> <p>0: The flag is cleared when any of the following operations is performed:</p> <ul style="list-style-type: none"> <li>• ADCAAnDRj for the virtual channel which ADCAAnSGVCEP2 indicates is read.</li> <li>• ADCAAnDIRj for the virtual channel which ADCAAnSGVCEP2 indicates is read.</li> <li>• ADCAAnSGSEFCRx.SEFC is written as 1.</li> </ul> <p>1: The A/D conversion result is written to ADCAAnDRj for the virtual channel which ADCAAnSGVCEP2 indicates.</p>
1	SEF[1]	<p>SG1 Scan End Flag Indicates the status of the scan result data.</p> <p>0: The flag is cleared when any of the following operations is performed:</p> <ul style="list-style-type: none"> <li>• ADCAAnDRj for the virtual channel which ADCAAnSGVCEP1 indicates is read.</li> <li>• ADCAAnDIRj for the virtual channel which ADCAAnSGVCEP1 indicates is read.</li> <li>• ADCAAnSGSEFCRx.SEFC is written as 1.</li> </ul> <p>1: The A/D conversion result is written to ADCAAnDRj for the virtual channel which ADCAAnSGVCEP1 indicates.</p>
0	Reserved	When read, the value after reset is returned.

### 29.3.4 Hardware Trigger Specific Register

#### 29.3.4.1 ADCAnSGTSELx — Scan Group x Start Trigger Control Register x

This register is used to select the A/D conversion trigger (hardware trigger) for SGx.

**Access:** ADCAnSGTSELx register can be read or written in 32-bit units.  
ADCAnSGTSELxL register can be read or written in 16-bit units.

**Address:** ADCAnSGTSELx: <ADCAn\_base> +  $x \times 40_{\text{H}} + 41C_{\text{H}}$   
ADCAnSGTSELxL: <ADCAn\_base> +  $x \times 40_{\text{H}} + 41C_{\text{H}}$

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TxSEL8 *1	TxSEL7 *1	TxSEL6 *1	TxSEL5 *1	TxSEL4 *1	TxSEL3 *1	TxSEL2 *1	TxSEL1 *1	TxSEL0 *1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W								

Note 1. ADCA1 supports only TxSEL0 to TxSEL3. When writing to the other bits, write the value after reset.

Table 29.46 ADCAnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	TxSELp (p = 0 to 8)	A/D Conversion Trigger (Hardware Trigger) Select 0: Hardware trigger is disabled. 1: Hardware trigger is enabled.
<b>CAUTION</b>		
When setting TxSELp to 1, set only one of the bits to 1.		

The list below shows the hardware triggers to be selected.

Table 29.47 List of A/D Conversion Hardware Triggers (1/2)

Unit	Control Register/Bit		Trigger Input Signal	
	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCA0	ADCA0SGTSEL1	T1SEL0	ADCA0TRG0	External trigger pin
		T1SEL1	INTTAUJ0I3	TAUJ0
		T1SEL2	INTTAUD0I7	TAUD0
		T1SEL3	INTTAUD0I15	TAUD0
		T1SEL4	SEQADTRG	LPS
		T1SEL5	INTENCA0I1	ENCA0
		T1SEL6	TAPATADOUT0	Motor control (TAPA0)
		T1SEL7	TAPATADOUT1	Motor control (TAPA0)
		T1SEL8	ADOPA0ADCATTIN00	Motor control (PIC0)

Table 29.47 List of A/D Conversion Hardware Triggers (2/2)

Unit	Control Register/Bit		Trigger Input Signal	
	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCA0	ADCA0SGTSEL2	T2SEL0	ADCA0TRG1	External trigger pin
		T2SEL1	INTTAUJ0I3	TAUJ0
		T2SEL2	INTTAUD0I7	TAUD0
		T2SEL3	INTTAUD0I15	TAUD0
		T2SEL4	SEQADTRG	LPS
		T2SEL5	INTENCA0I1	ENCA0
		T2SEL6	TAPATADOUT0	Motor control (TAPA0)
		T2SEL7	TAPATADOUT1	Motor control (TAPA0)
		T2SEL8	ADOPA1ADCATTIN00	Motor control (PIC0)
	ADCA0SGTSEL3	T3SEL0	ADCA0TRG2	External trigger pin
		T3SEL1	INTTAUJ0I3	TAUJ0
		T3SEL2	INTTAUD0I7	TAUD0
		T3SEL3	INTTAUD0I15	TAUD0
		T3SEL4	SEQADTRG	LPS
		T3SEL5	INTENCA0I1	ENCA0
		T3SEL6	TAPATADOUT0	Motor control (TAPA0)
		T3SEL7	TAPATADOUT1	Motor control (TAPA0)
		T3SEL8	ADOPA2ADCATTIN00	Motor control (PIC0)
ADCA1	ADCA1SGTSEL1	T1SEL0	ADCA1TRG0	External trigger pin
		T1SEL1	INTTAUJ1I3	TAUJ1
		T1SEL2	INTTAUB0I7	TAUB0
		T1SEL3	INTTAUB0I15	TAUB0
	ADCA1SGTSEL2	T2SEL0	ADCA1TRG1	External trigger pin
		T2SEL1	INTTAUJ1I3	TAUJ1
		T2SEL2	INTTAUB0I7	TAUB0
		T2SEL3	INTTAUB0I15	TAUB0
	ADCA1SGTSEL3	T3SEL0	ADCA1TRG2	External trigger pin
		T3SEL1	INTTAUJ1I3	TAUJ1
		T3SEL2	INTTAUB0I7	TAUB0
		T3SEL3	INTTAUB0I15	TAUB0

**CAUTIONS**

- When enabling the LPS trigger factor (SEQADTRG), select and enable only one of ADCA0SGTSEL1.T1SEL4, ADCA0SGTSEL2.T2SEL4, and ADCA0SGTSEL3.T3SEL4.
- To prevent malfunction, ADCAnSGTSELx should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

### 29.3.5 Self-Diagnosis Specific Registers

#### 29.3.5.1 ADCAnDGCTL0 — Self-Diagnosis Control Register 0

This register controls the self-diagnostic voltage level.

**Access:** ADCAnDGCTL0 register can be read or written in 32-bit units.  
 ADCAnDGCTL0L register can be read or written in 16-bit units.  
 ADCAnDGCTL0LL register can be read or written in 8-bit units.

**Address:** ADCAnDGCTL0: <ADCAn\_base> + 350<sub>H</sub>  
 ADCAnDGCTL0L: <ADCAn\_base> + 350<sub>H</sub>  
 ADCAnDGCTL0LL: <ADCAn\_base> + 350<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	PSEL2	PSEL1	PSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 29.48 ADCAnDGCTL0 Register Contents

Bit Position	Bit Name	Function				
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.				
2 to 0	PSEL[2:0]	Self-Diagnostic Voltage Level Select				
ADCAnDGCTL0						
PSEL2	PSEL1	PSEL0	ADDIAGOUT	DIAGOUT2	DIAGOUT1	DIAGOUT0
0	0	0	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	AnVSS	2/3AnV <sub>REF</sub>	1/2AnV <sub>REF</sub>	1/3AnV <sub>REF</sub>
0	1	0	1/3AnV <sub>REF</sub>	1/3AnV <sub>REF</sub>	2/3AnV <sub>REF</sub>	1/2AnV <sub>REF</sub>
0	1	1	1/2AnV <sub>REF</sub>	1/2AnV <sub>REF</sub>	1/3AnV <sub>REF</sub>	2/3AnV <sub>REF</sub>
1	0	0	2/3AnV <sub>REF</sub>	Hi-z	Hi-z	Hi-z
1	0	1	AnV <sub>REF</sub>	1/3AnV <sub>REF</sub>	1/3AnV <sub>REF</sub>	1/3AnV <sub>REF</sub>
1	1	0	AnV <sub>REF</sub>	1/2AnV <sub>REF</sub>	1/2AnV <sub>REF</sub>	1/2AnV <sub>REF</sub>
1	1	1	AnV <sub>REF</sub>	2/3AnV <sub>REF</sub>	2/3AnV <sub>REF</sub>	2/3AnV <sub>REF</sub>

### 29.3.5.2 ADCAnDGCTL1 — Self-Diagnosis Control Register 1

This register controls the self-diagnostic channel.

**Access:** ADCAnDGCTL1 register can be read or written in 32-bit units.  
ADCAnDGCTL1L register can be read or written in 16-bit units.

**Address:** ADCAnDGCTL1: <ADCAn\_base> + 354H  
ADCAnDGCTL1L: <ADCAn\_base> + 354H

**Value after reset:** 0000 0000H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.49 ADCAnDGCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15, 12, 9, 6, 3, 0	CDG [15, 12, 9, 6, 3, 0]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT0 is selected.
13, 10, 7, 4, 1	CDG [13, 10, 7, 4, 1]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT1 is selected.
14, 11, 8, 5, 2	CDG [14, 11, 8, 5, 2]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT2 is selected.

#### CAUTION

To prevent malfunction, ADCAnDGCTL1 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

### 29.3.5.3 ADCAnPDCTL1 — Pull Down Control Register 1

This register specifies the channel to which the pull down resistor is connected.

For details, see **Section 29.5.3, Diagnosis of Open Pins**.

**Access:** ADCAnPDCTL1 register can be read or written in 32-bit units.  
ADCAnPDCTL1L register can be read or written in 16-bit units.

**Address:** ADCAnPDCTL1: <ADCAn\_base> + 358<sub>H</sub>  
ADCAnPDCTL1L: <ADCAn\_base> + 358<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 29.50 ADCAnPDCTL1 Register Contents**

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDNA[15:0]	Pull Down Enable Control These bits set whether an on-chip pull-down resistor is to be connected to the corresponding physical channel (ANIn[00:15]). 0: An on-chip pull-down resistor is not connected. 1: An on-chip pull-down resistor is connected.

#### CAUTION

To prevent malfunction, ADCAnPDCTL1 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

#### NOTE

For on-chip pull-down resistor values, see the Electrical Characteristics section in the Data Sheet document.

### 29.3.5.4 ADCAnPDCTL2 — Pull Down Control Register 2

This register specifies the channel to which the pull down resistor is connected.

For details, see **Section 29.5.3, Diagnosis of Open Pins**.

**Access:** ADCAnPDCTL2 register can be read or written in 32-bit units.  
 ADCAnPDCTL2H and ADCAnPDCTL2L registers can be read or written in 16-bit units.  
 ADCAnPDCTL2HL, ADCAnPDCTL2LH, and ADCAnPDCTL2LL registers can be read or written in 8-bit units.

**Address:** ADCAnPDCTL2: <ADCAn\_base> + 35C<sub>H</sub>  
 ADCAnPDCTL2L: <ADCAn\_base> + 35C<sub>H</sub>  
 ADCAnPDCTL2H: <ADCAn\_base> + 35C<sub>H</sub> + 2<sub>H</sub>  
 ADCAnPDCTL2HL: <ADCAn\_base> + 35C<sub>H</sub> + 2<sub>H</sub>  
 ADCAnPDCTL2LL: <ADCAn\_base> + 35C<sub>H</sub>  
 ADCAnPDCTL2LH: <ADCAn\_base> + 35C<sub>H</sub> + 1<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PDNB[19:16] <sup>*1</sup>
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNB[15:0] <sup>*1</sup>															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ADCA1 supports only PDNB[7:0]. When writing to the other bits, write the value after reset.

**Table 29.51 ADCAnPDCTL2 Register Contents**

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	PDNB[19:0]	Pull Down Enable Control These bits set whether the on-chip pull-down resistor is to be connected with the corresponding physical channel (ANIO[16:35], ANI1[16:23]). 0: The on-chip pull-down resistor is not connected. 1: The on-chip pull-down resistor is connected.

#### CAUTION

To prevent malfunction, ADCAnPDCTL2 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

#### NOTE

For on-chip pull-down resistor values, see the Electrical Characteristics section in the Data Sheet document.

## 29.3.6 Emulation Specific Register

### 29.3.6.1 ADCAnEMU — Emulation Control Register

This register controls the SVSTOP disable signal.

**Access:** This register can be read or written in 8-bit units.

**Address:** <ADCAn\_base> + 388<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

R/W	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

**Table 29.52 ADCAnEMU Register Contents**

Bit Position	Bit Name	Function
7	SVSDIS	SVSTOP Disable 0: SVSTOP is enabled 1: SVSTOP is disabled For the A/D conversion when SVSTOP is enabled, see <b>Section 29.4.10.3, SVSTOP Operation</b> .
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

#### CAUTION

To prevent malfunction, SVSDIS should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

## 29.4 Operation

### 29.4.1 Initial Setting

**Figure 29.6** shows an initial setting example of the A/D conversion. For trigger input, see **Figure 29.7** in the next section. For interrupt request signals, see **Section 29.4.12, Scan End Interrupt Request**.

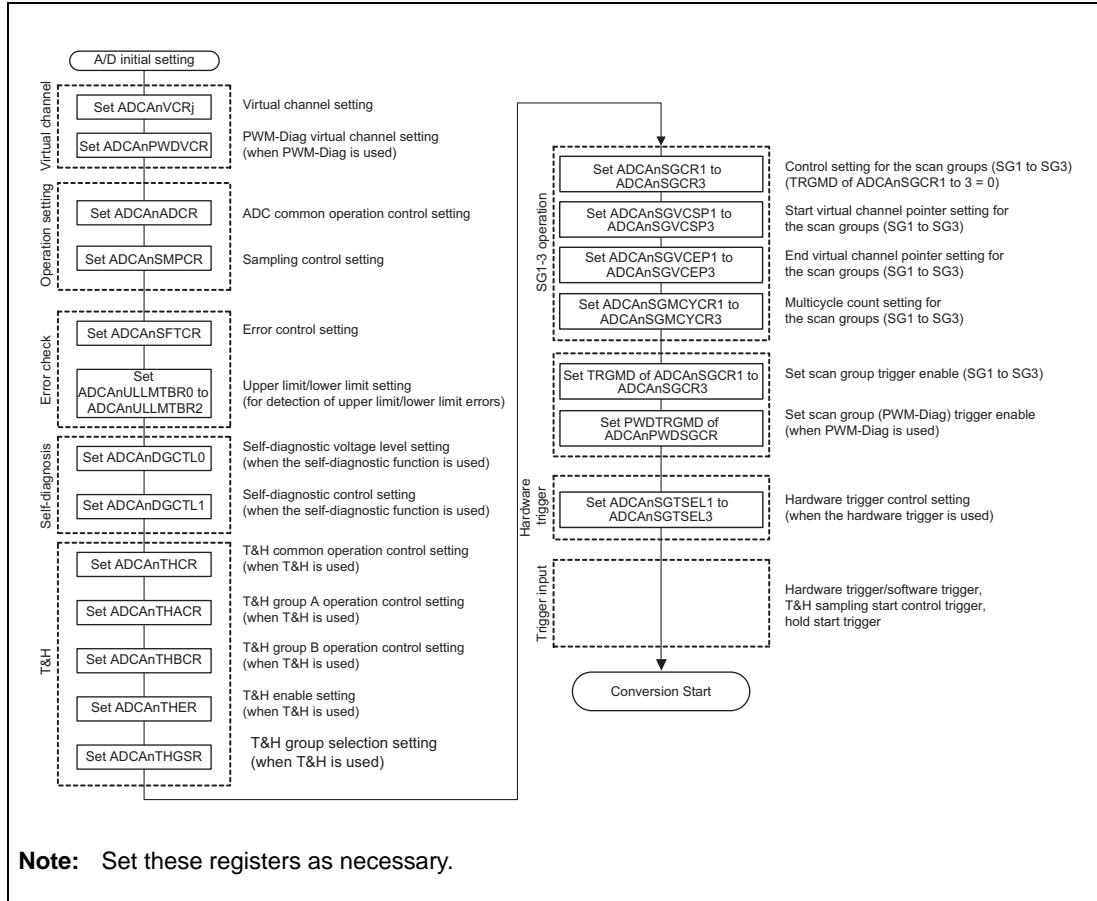


Figure 29.6 Flowchart for Initial Setting

### 29.4.2 Trigger Input

The following figure shows the flowchart for trigger input.

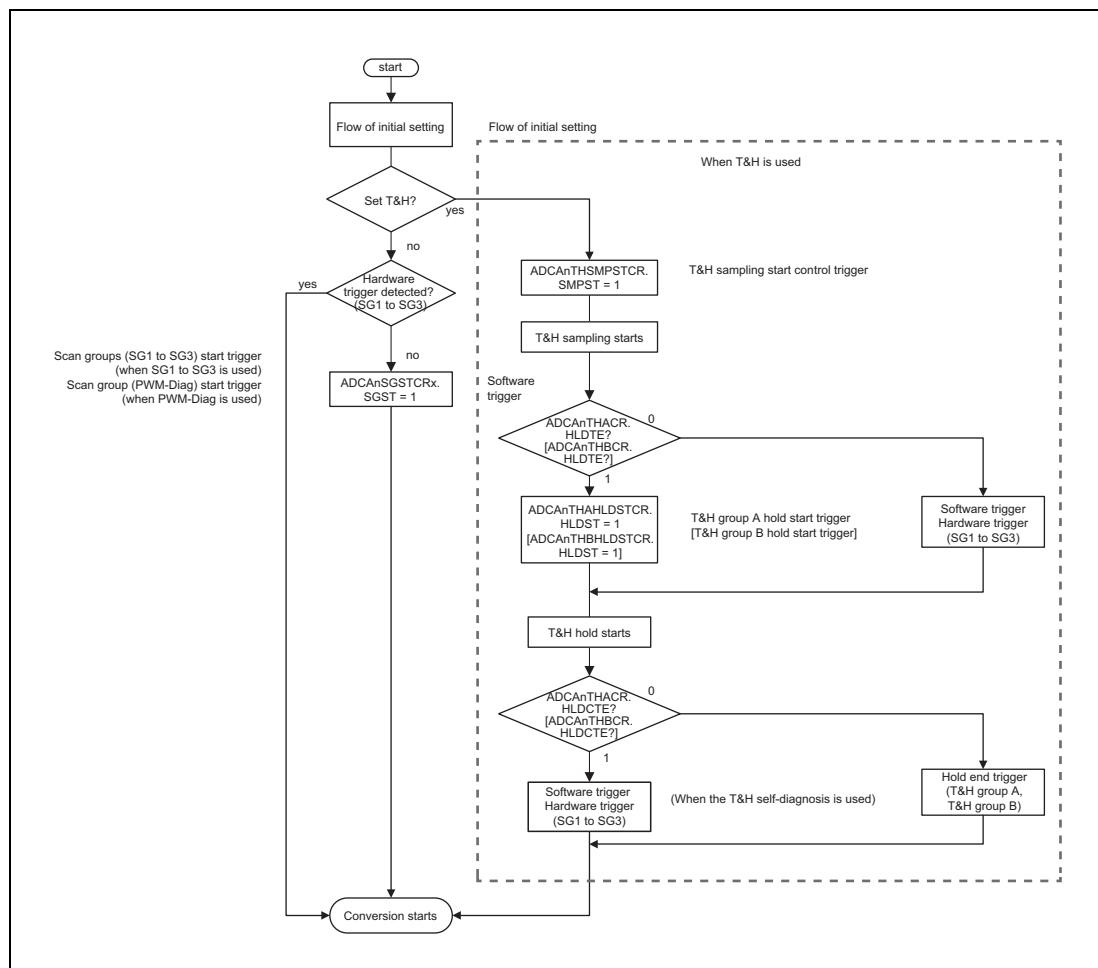


Figure 29.7 Flowchart for Trigger Input

#### NOTE

When an SG start trigger is generated during scanning, the SG start trigger is ignored.

### 29.4.3 Ending A/D Conversion

The flow for ending A/D conversion is shown below.

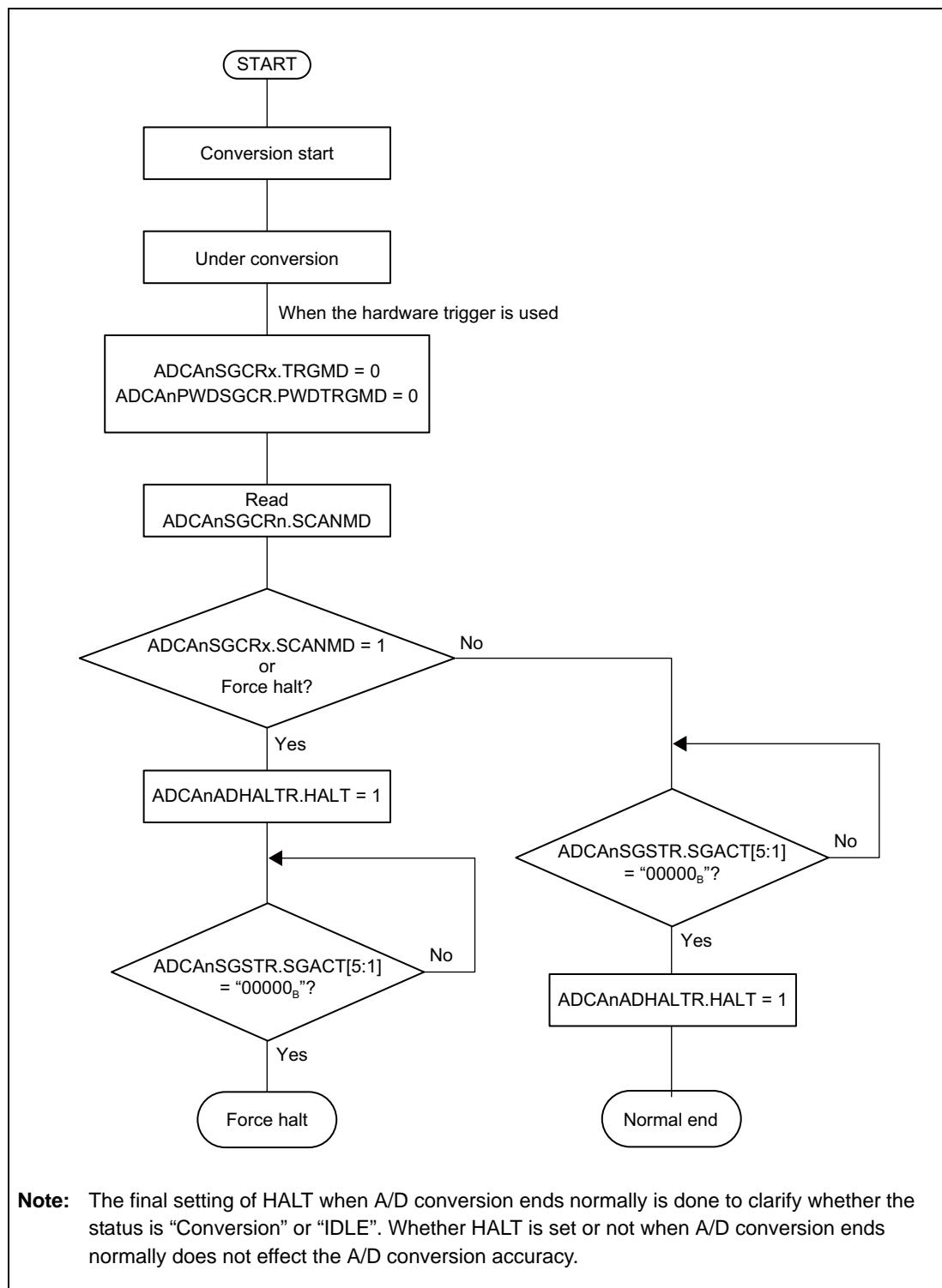
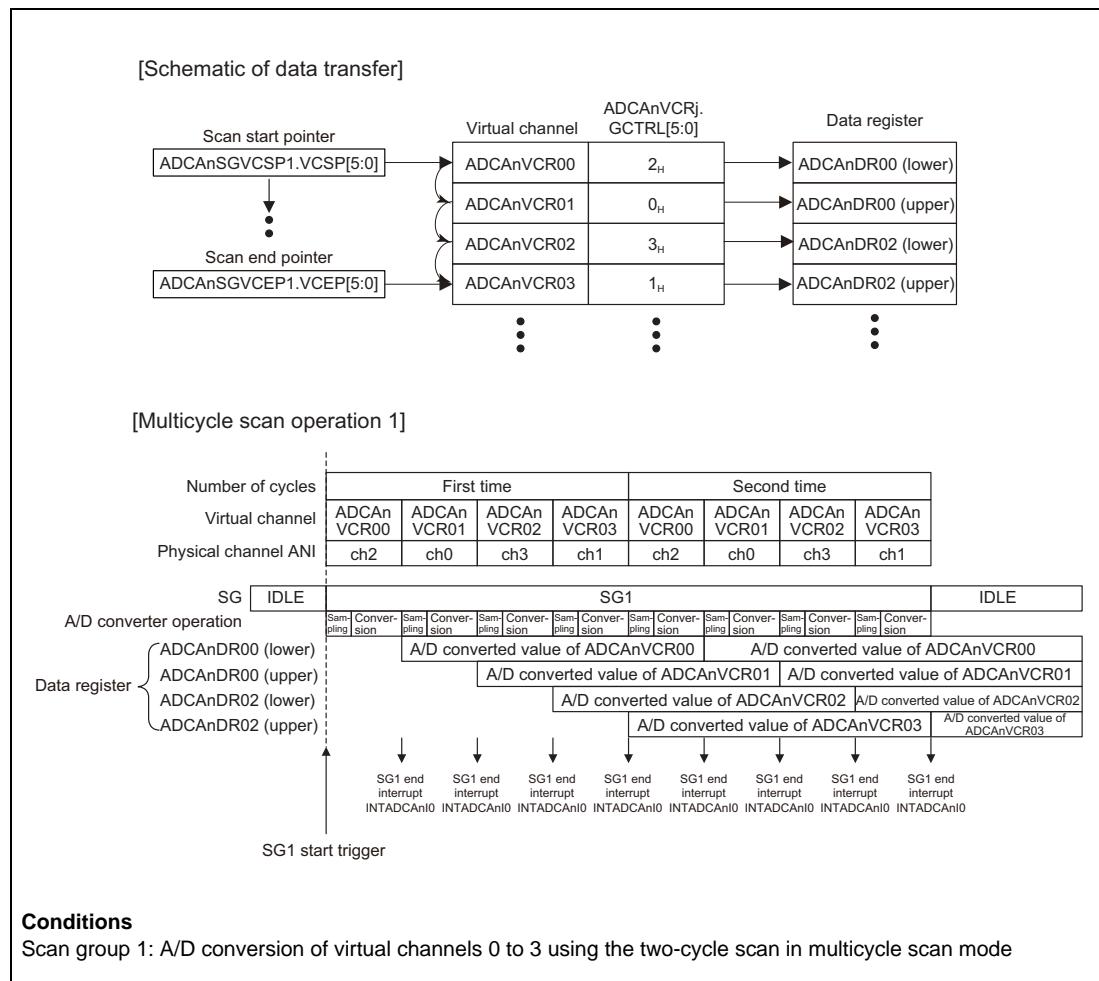


Figure 29.8 Flowchart for Ending A/D Conversion

#### **29.4.4 Example of Scan Group Operation**

### (1) Multicycle scan mode

The following figure illustrates an operation example where four virtual channels of scan group 1 are converted using the two-cycle scan in multicycle scan mode.



**Figure 29.9 Example of Multicycle Scan Operation 1**

The following figure illustrates an operation example where a pin is scanned once in multicycle scan mode.

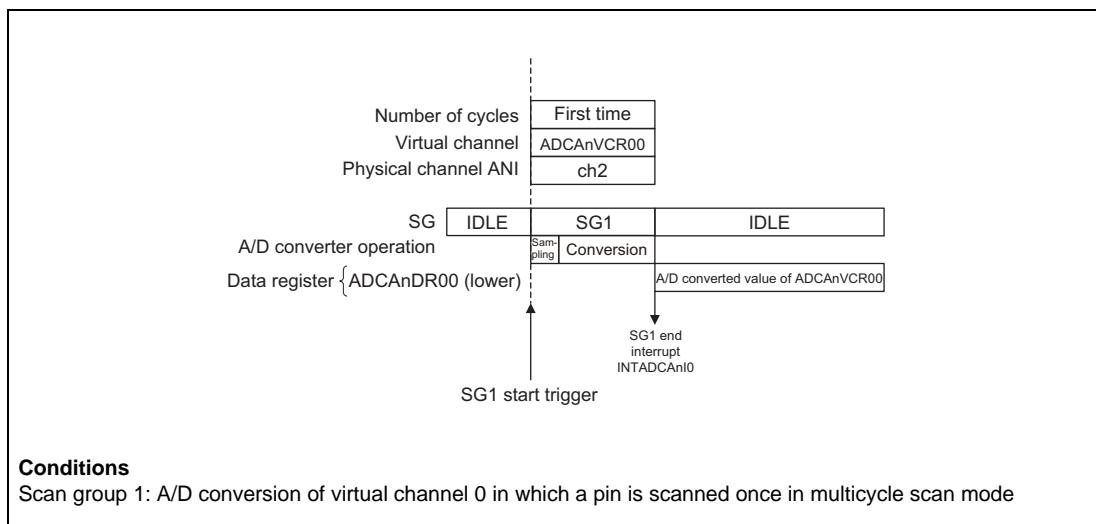


Figure 29.10 Example of Multicycle Scan Operation 2

## (2) Continuous scan mode

Continuous scan mode allows A/D conversion of the SG channels indicated by the pointers specified by ADCAnSGVCSPx.VCSP[5:0] to ADCAnSGVCEPx.VCEP[5:0] to continue until ADCAnADHALTR.HALT is asserted. This mode can be used only with SG1.

The following figure shows an example of operation in continuous scan mode.

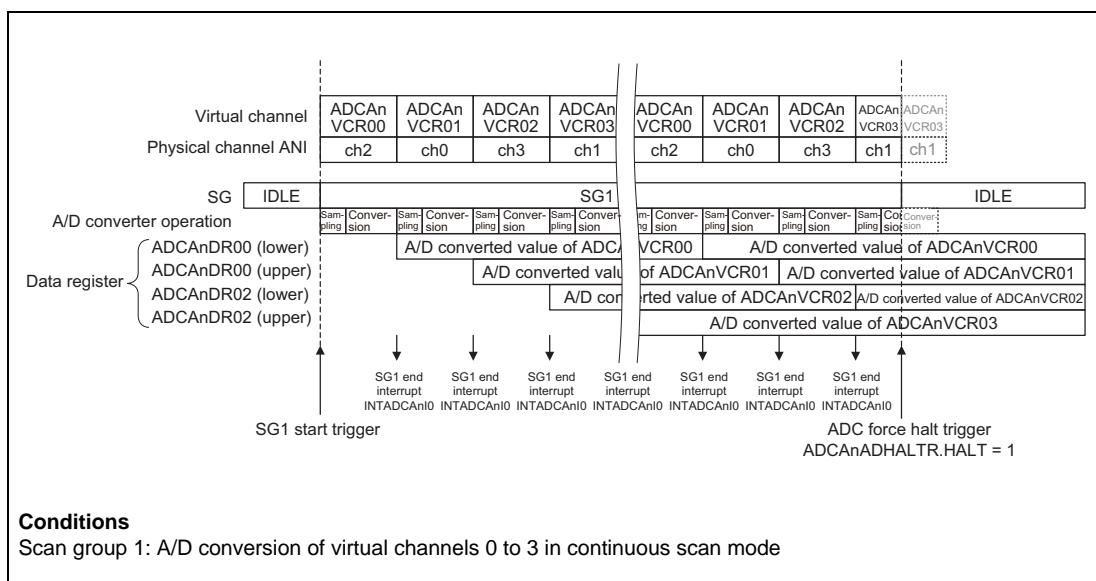
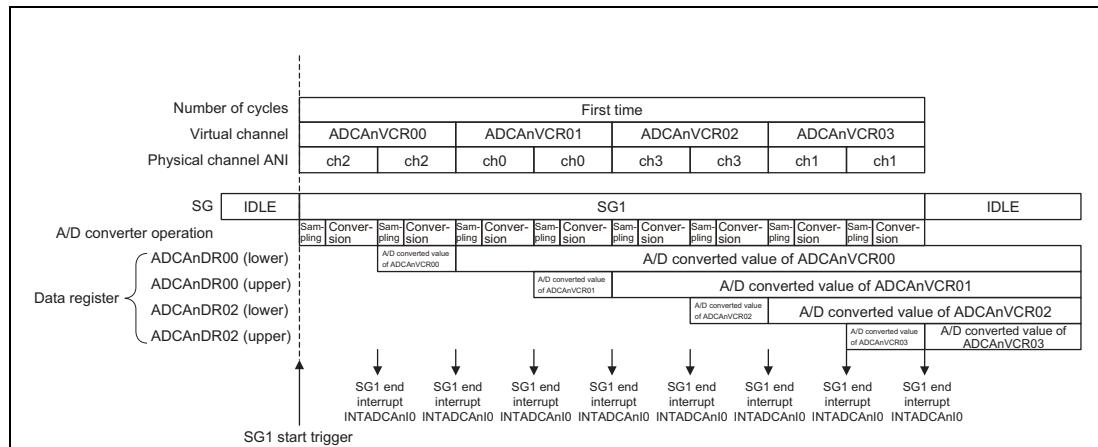


Figure 29.11 Example of Continuous Scan Operation

#### **29.4.5 Channel Repeat Mode**

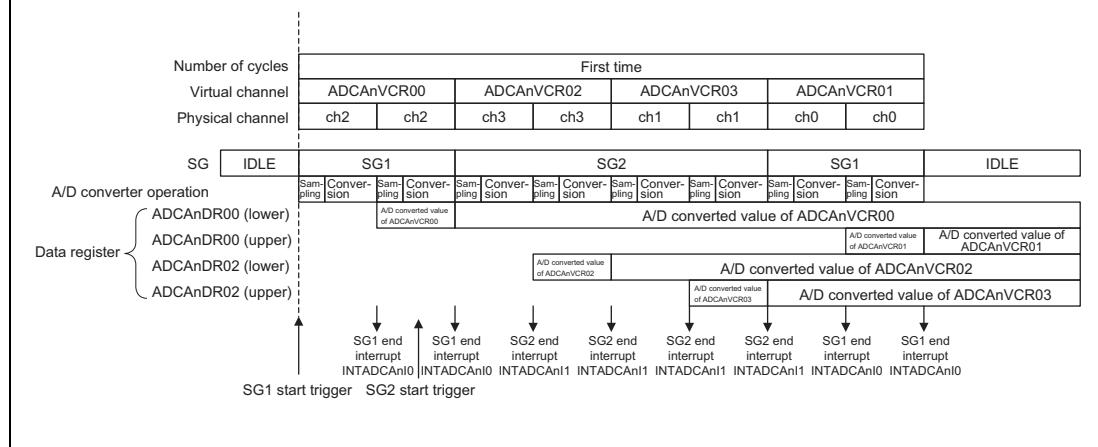
Channel repeat mode allows A/D conversion of the SG channel indicated by the pointer specified by ADCAnSGVCS Px.VCSP[5:0] to ADCAnSGVCE Px.VCEP[5:0] to repeat number of channel repeat times specified by ADCAnSGCRx.SCT[1:0]. This mode operates exclusively in each SG. The number of channel repeat times is selectable from 1, 2, and 4.

The following figures show examples of operation under respective conditions.



## Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode



## Conditions

Scan group 1: A/D conversion of virtual channels 0 and 1 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

Scan group 2: A/D conversion of virtual channels 2 and 3 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

**Figure 29.12 Example of Channel Repeat Operation 1**

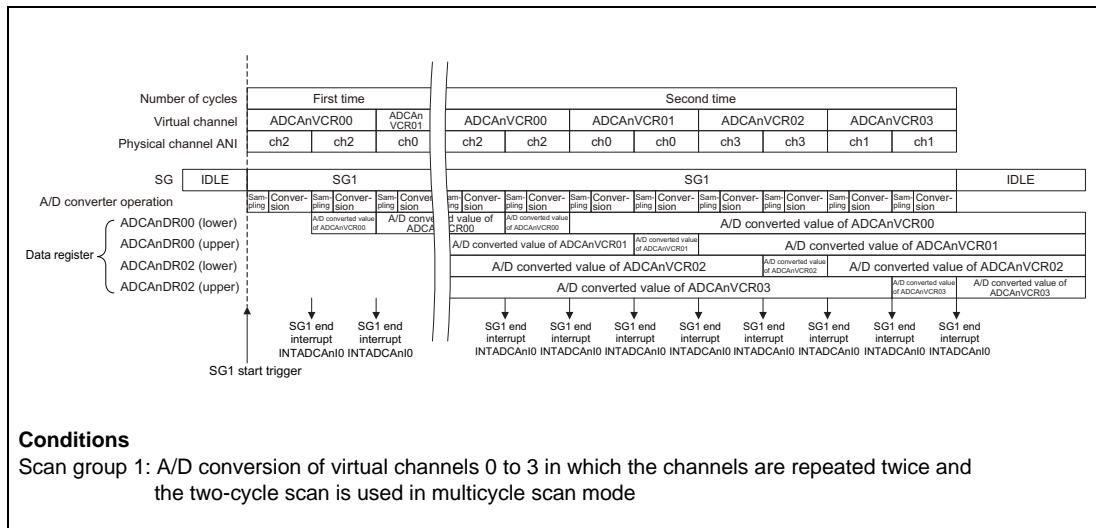


Figure 29.13 Example of Channel Repeat Operation 2

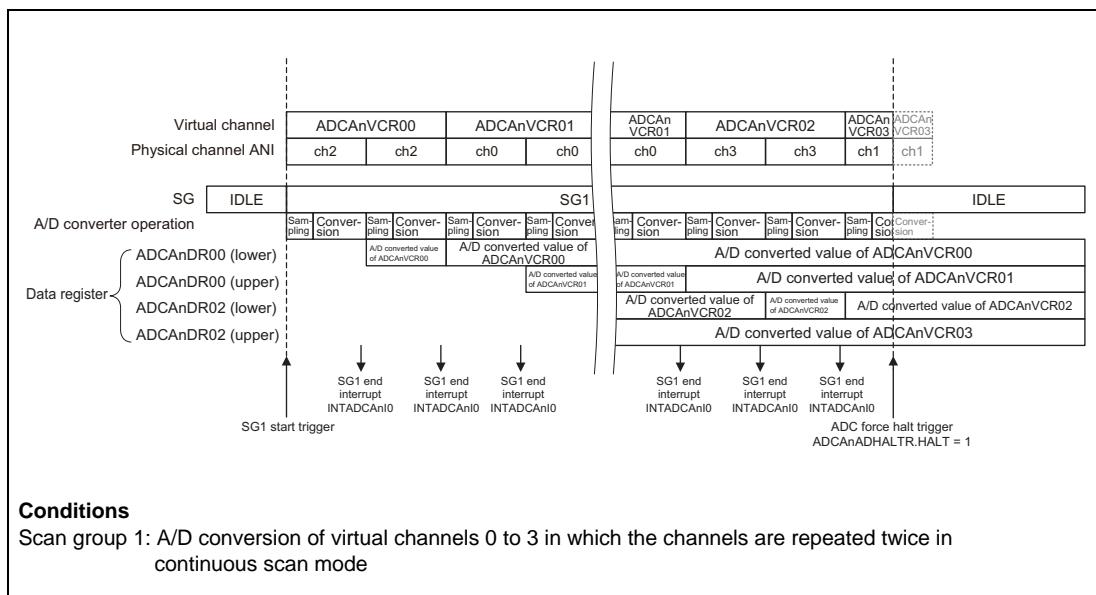


Figure 29.14 Example of Channel Repeat Operation 3

### 29.4.6 Example of Simultaneous Track and Hold Operation

Figure 29.15 shows an operation example of simultaneous track and hold.

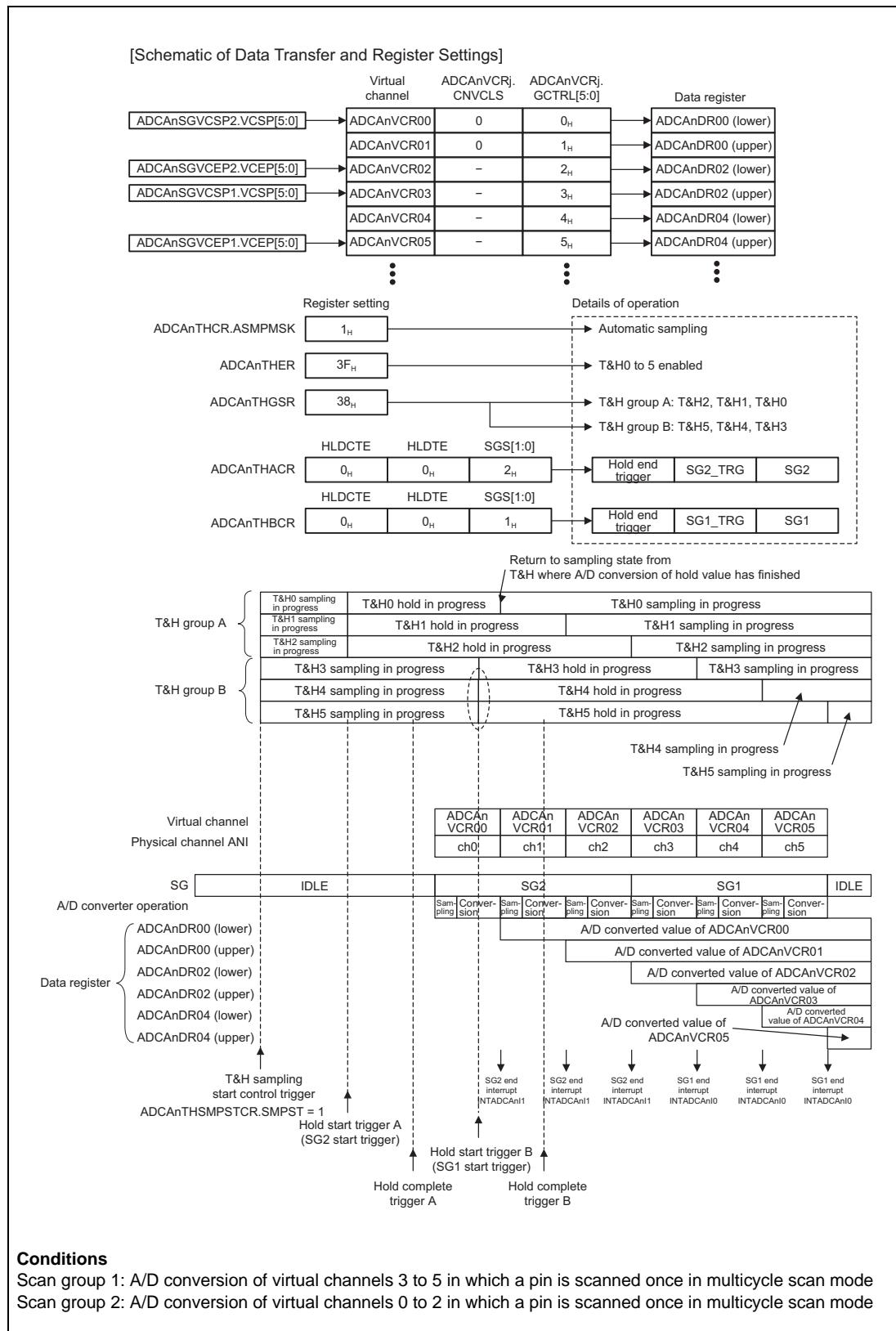


Figure 29.15 Example of Simultaneous Track & Hold Operation 1

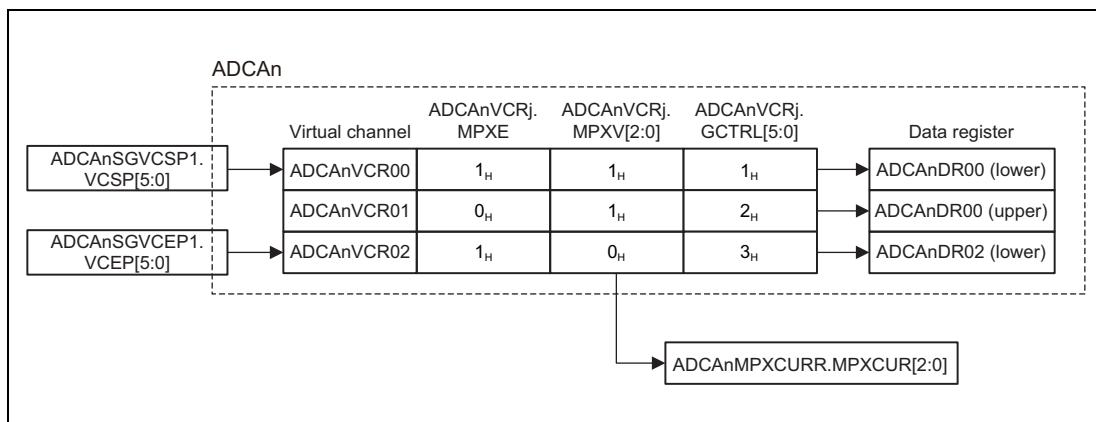
---

**CAUTION**

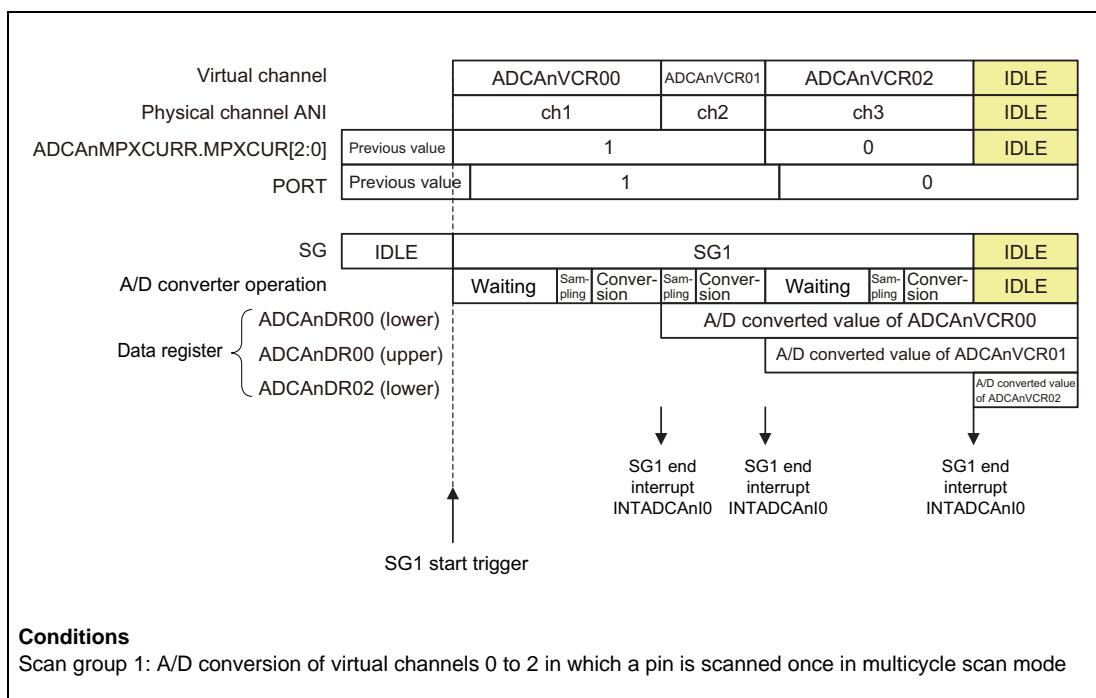
- Do not specify the same physical channel in different groups.
  - Two-cycle (or more) scan in multicycle scan mode and track & hold operation using continuous scan mode are prohibited.
  - Because ADCAnTHSMPSTCR.SMPST is common to group A and group B, set SMPST after T&H operation for both group A and group B has been completed.
  - If the hardware trigger is asserted before HLDCTE = 1 and HLDTE = 1 of ADCAnTHACR register or ADCAnTHBCR register are set and HLDST of ADCAnTHAHLDSCTR register or ADCAnTHBHLDSCTR register is written to hold T&H, scan operation starts. In that case, the channel switch opens with T&H staying in the sampling state. Therefore, all scan results are undefined. Do not assert the hardware trigger by setting HLDCTE = 1 and HLDTE = 1 before writing HLDST.
  - Setting any channel from among 0 to 2 and any channel from among 3 to 5 in the same scan group is prohibited.
  - Set the interval between T&H sampling start control trigger and hold start trigger to be 450 ns or more.
  - Set the interval between hold start trigger and completion of the group A/D conversion to be 10  $\mu$ s or less.  
In suspend mode, do not use T&H for the channel of a scan group with low priority setting, if suspend time exceeds 10  $\mu$ s.
-

### 29.4.7 A/D Conversion with External Analog Multiplexer

The following figures show examples of A/D conversion in each case.



**Figure 29.16 Schematic of Data Transfer and Register Settings**



**Figure 29.17 A/D Conversion 1 at an External Analog Multiplexer**

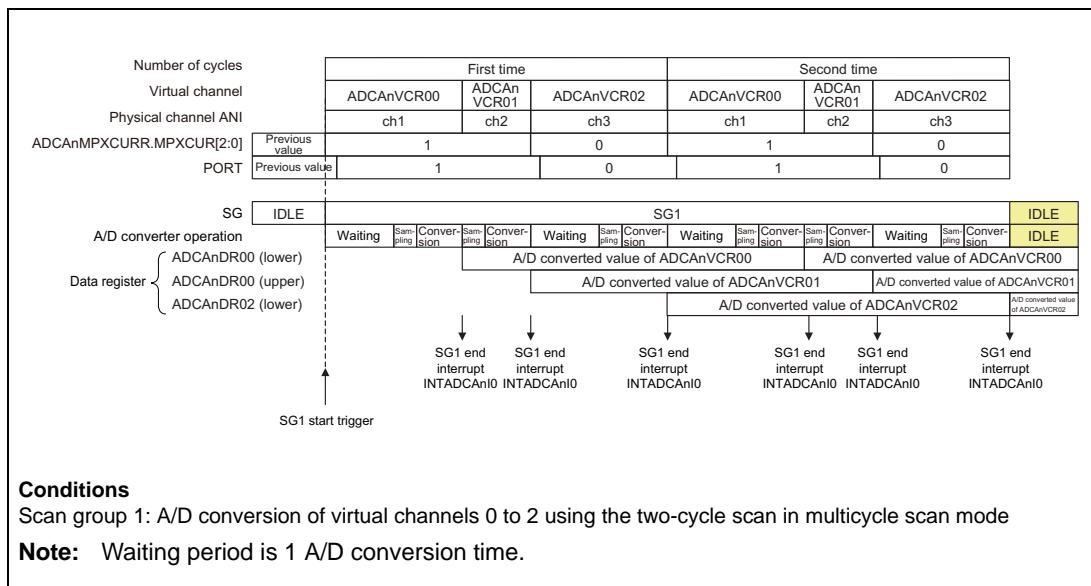


Figure 29.18 A/D Conversion 2 at an External Analog Multiplexer

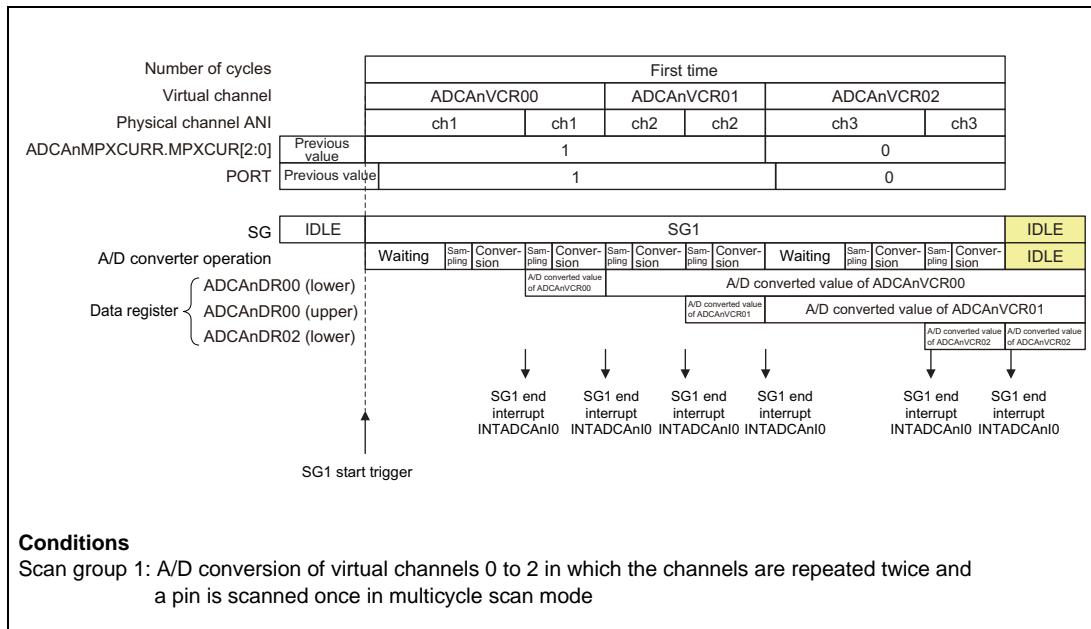


Figure 29.19 A/D Conversion 3 at an External Analog Multiplexer

### 29.4.7.1 A/D Conversion with PWM-Diag Enabled

With the PWM-Diag function enabled, A/D conversion is performed by the signal from the PWM-Diag.

For details on the PWM-Diag function, see **Section 28, PWM Output/Diagnostic (PWM-Diag)**.

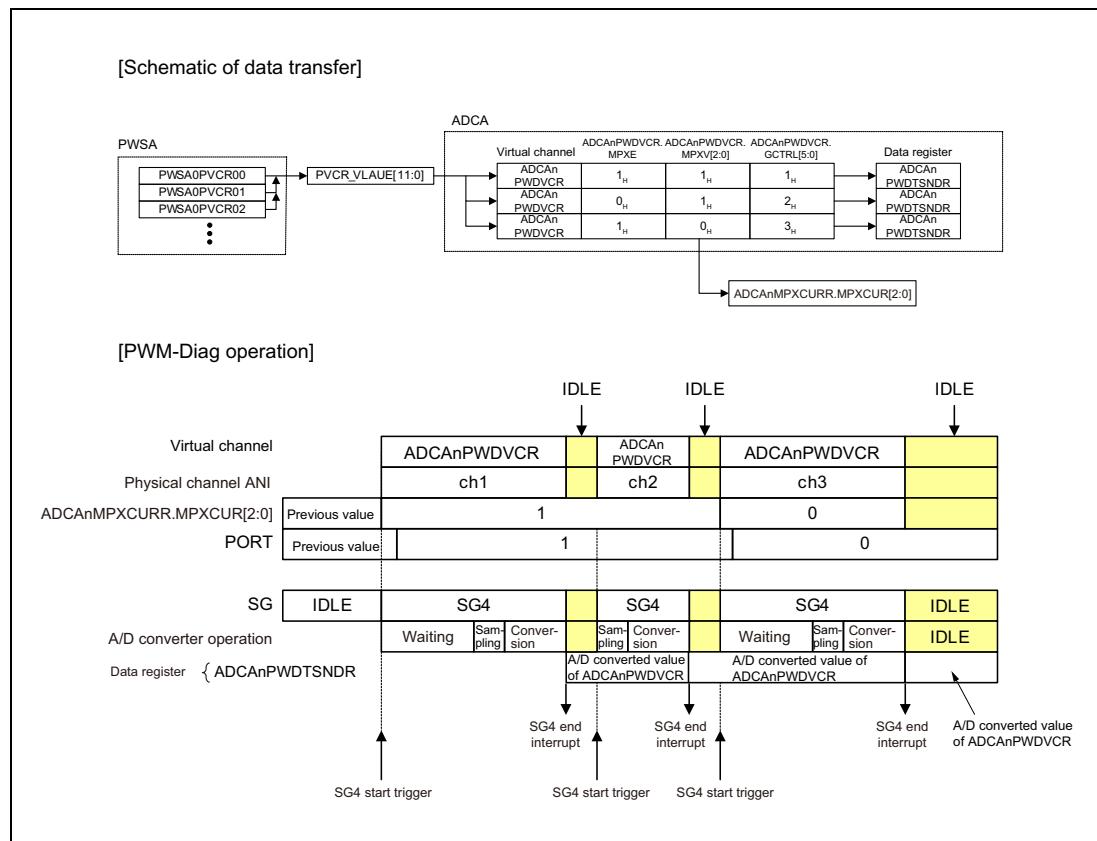
To control the A/D conversion, the A/D converter receives the setting information on the MPX by the A/D conversion trigger select (PWSA) signal. The flow of A/D conversion with PWM-Diag is as follows.

- (1) Set the channel MPX value of the MPX to ADCAnPWDVCR.MPXV[2:0]. Up to 8 channels can be specified to the MPX.
- (2) The A/D conversion is started by the trigger signal PWSA\_ADTRG from the PWM-Diag. In addition, when the MPX enable bit (ADCAnPWDVCR.MPXE) is 1, a wait of one A/D-conversion time is inserted before A/D conversion is performed.
- (3) At the end of A/D conversion, the scan end is notified to the PWM-Diag.

#### **CAUTION**

**As the trigger signal PWSA\_ADTRG of PWM-Diag function has a higher-priority than SGx\_TRG (x = 1 to 3), the operation of other scan groups may be kept waiting until the PWM-Diag function is ended.**

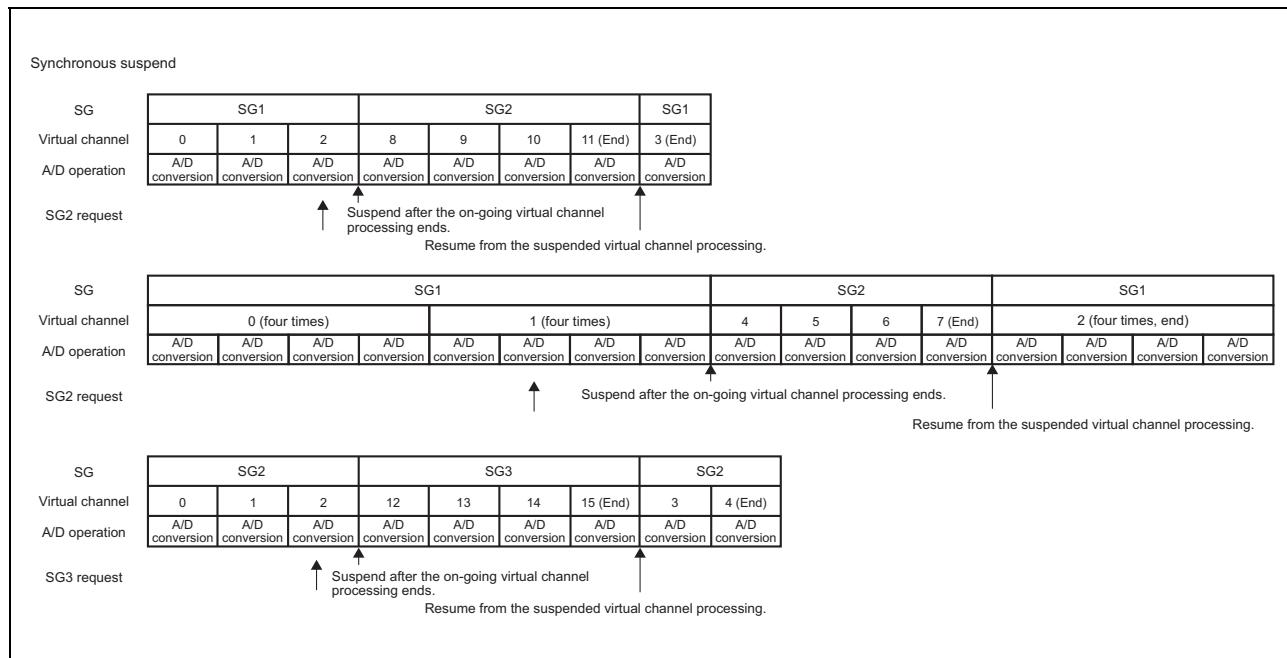
**Figure 29.20** shows an example of PWM-Diag operation using an MPX.



**Figure 29.20** PWM-Diag Operation

### 29.4.8 Example of Synchronous Suspend and Resume Operation

**Figure 29.21** shows an example of synchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.



**Figure 29.21 Example of Synchronous Suspend and Resume Operation**

#### NOTE

Priority of scan groups is as follows.

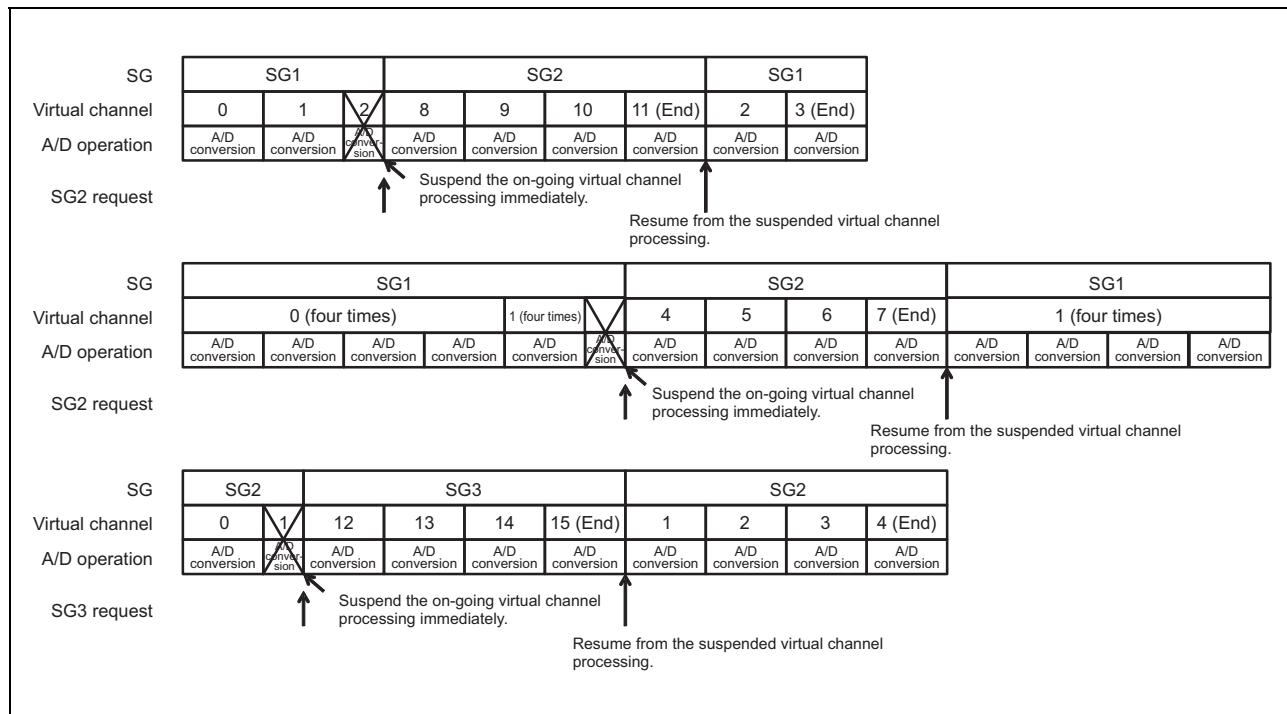
Lower

Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

#### **29.4.9 Example of Asynchronous Suspend and Resume Operation**

**Figure 29.22** shows an example of asynchronous suspend and resume operation. In this example, a higher-priority SG interrupt a lower-priority SG.



**Figure 29.22 Example of Asynchronous Suspend and Resume Operation**

## NOTE

Priority of scan groups is as follows.

SG1 < SG2 < SG3 < PWM-Diag (SG4)

## 29.4.10 Error Detecting Functions

ADCAn covers upper-limit error, lower-limit error, and overwrite error.

### 29.4.10.1 Upper-Limit/Lower-Limit Error Detecting Function

The upper-limit/lower-limit error detecting function determines whether the A/D converted data is larger than the upper-limit table ADCAnULLMTB0.ULMTB[11:0] or smaller than the lower-limit table ADCAnULLMTB0.LLMTB[11:0] at the end of A/D conversion.

### 29.4.10.2 Overwrite Error Detecting Function

If the ADCAnDIRj register or ADCAnDRj register of a virtual channel is not read while ADCAnDIRj.WFLG = 1 (A/D converted value is stored) and the next A/D converted value is written in the ADCAnDRj register, an overwrite error is detected.

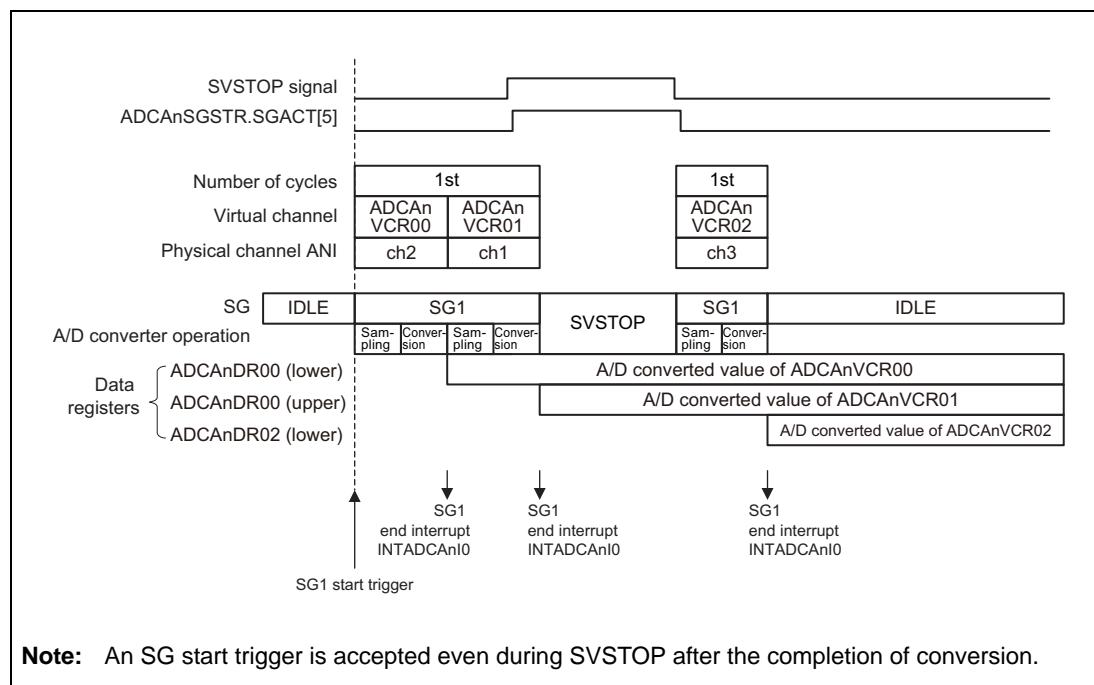
### 29.4.10.3 SVSTOP Operation

The SVSTOP function is supported by the SVSTOP signal sent from the on-chip debugger control unit. The SVSTOP function stops conversion of the A/D converter when the SVSTOP signal is input during an emulation break. While the SVSTOP signal is high, reading registers ADCAnDRj, ADCAnDIRj, ADCAnSGSTR, ADCAnULER, ADCAnOWER, ADCAnPWDTSNDR, and ADCAnPWDDIR by external access does not affect these registers.

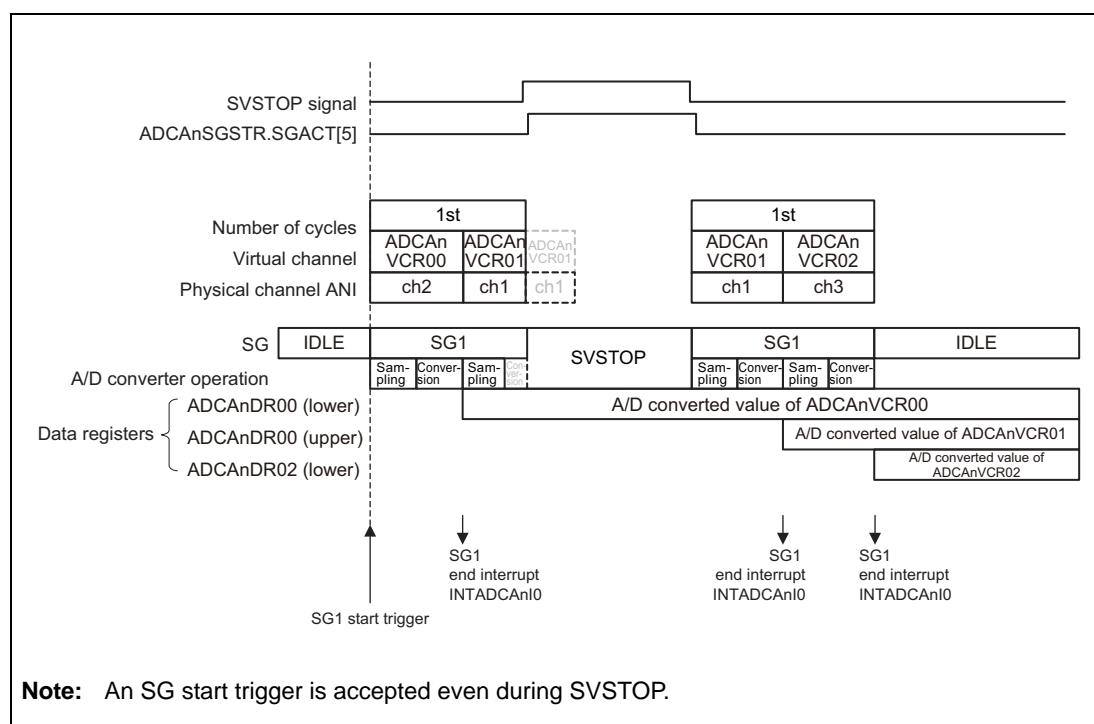
When the high level is input to SVSTOP while ADCAnEMU.SVSDIS = 0, ADCAnSGSTR.SGACT[5] is set to 1 to make a transition to the SVSTOP state. Hardware triggers and software triggers are valid in the SVSTOP state. When the high level is input to SVSTOP while ADCAnEMU.SVSDIS = 1, the ADCA does not make a transition to the SVSTOP state. ADHALT (forced termination of A/D conversion) should not be performed in the SVSTOP state.

In operations for synchronous suspension, a new start trigger cannot be accepted over the time from when the high level is input to SVSTOP to the completion of conversion on the channel where conversion is currently proceeding. This time can be up to the time taken for one A/D conversion.

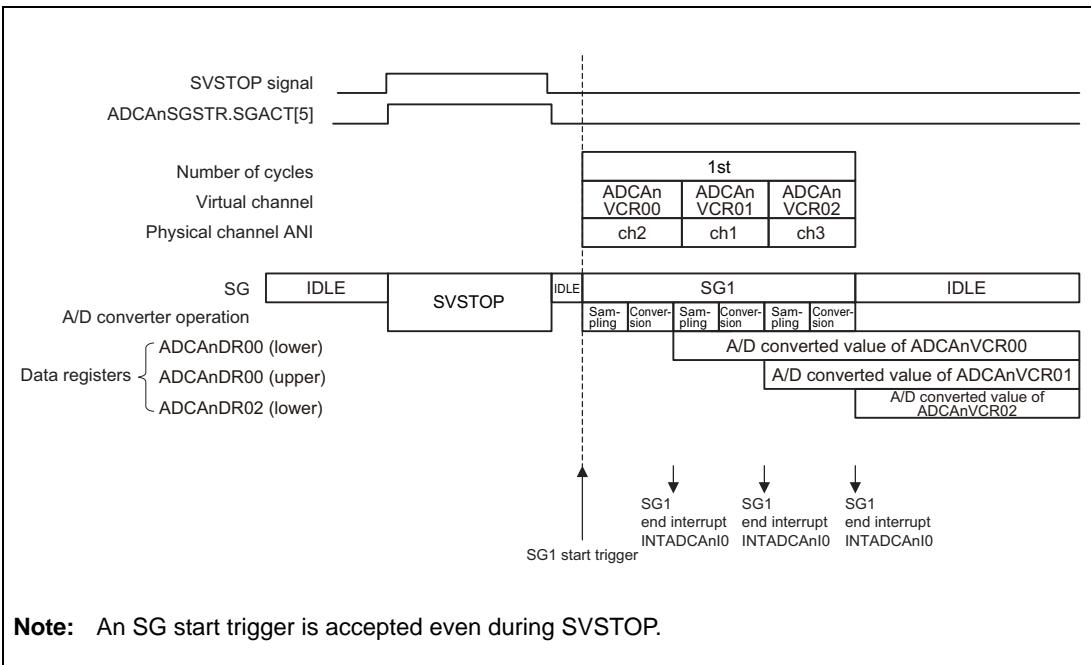
The following example illustrates a SVSTOP operation example.



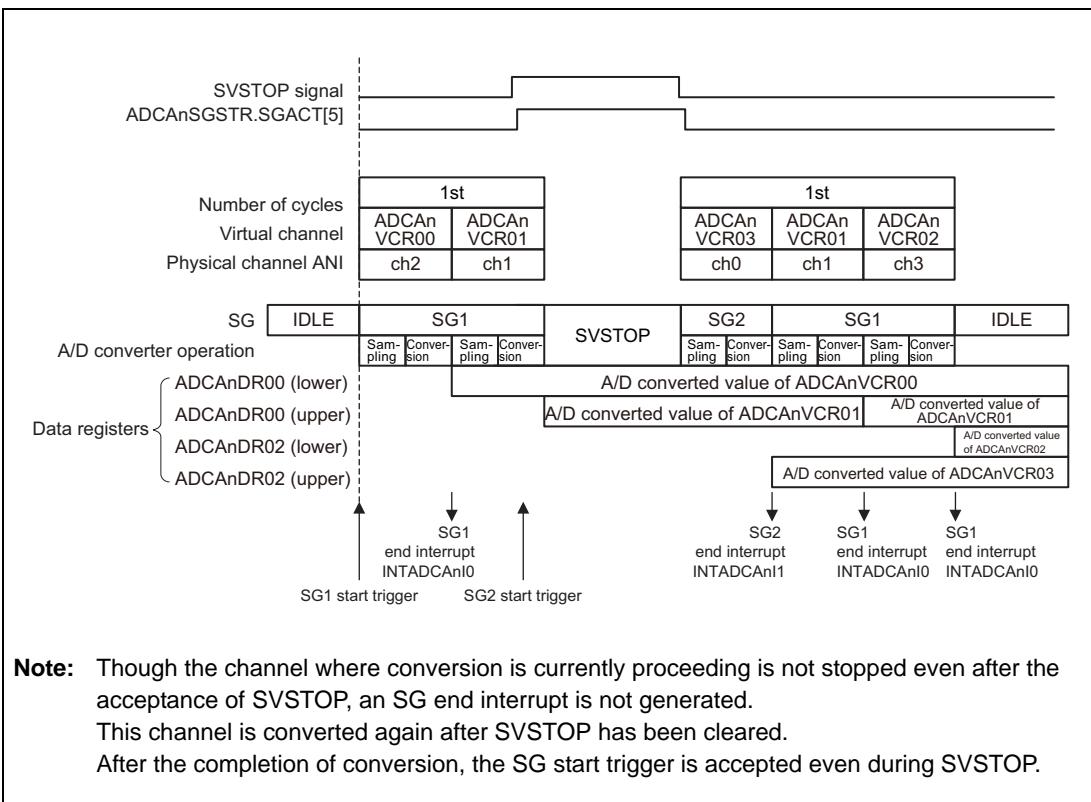
**Figure 29.23 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)**



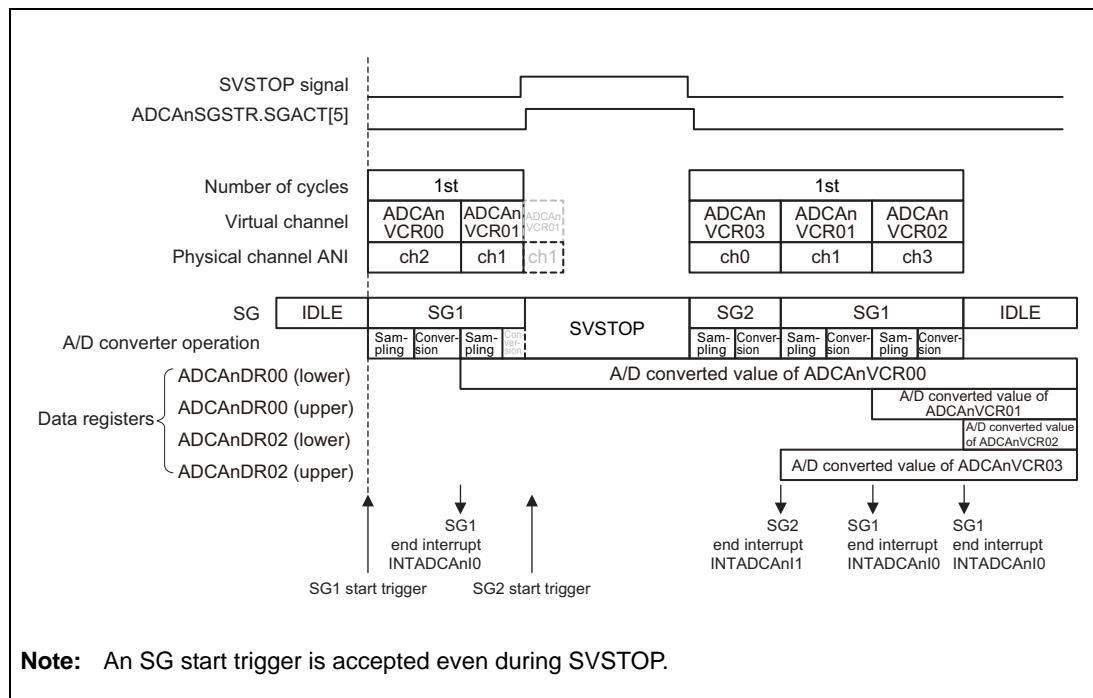
**Figure 29.24 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 10 and ADCAnEMU.SVSDIS = 0)**



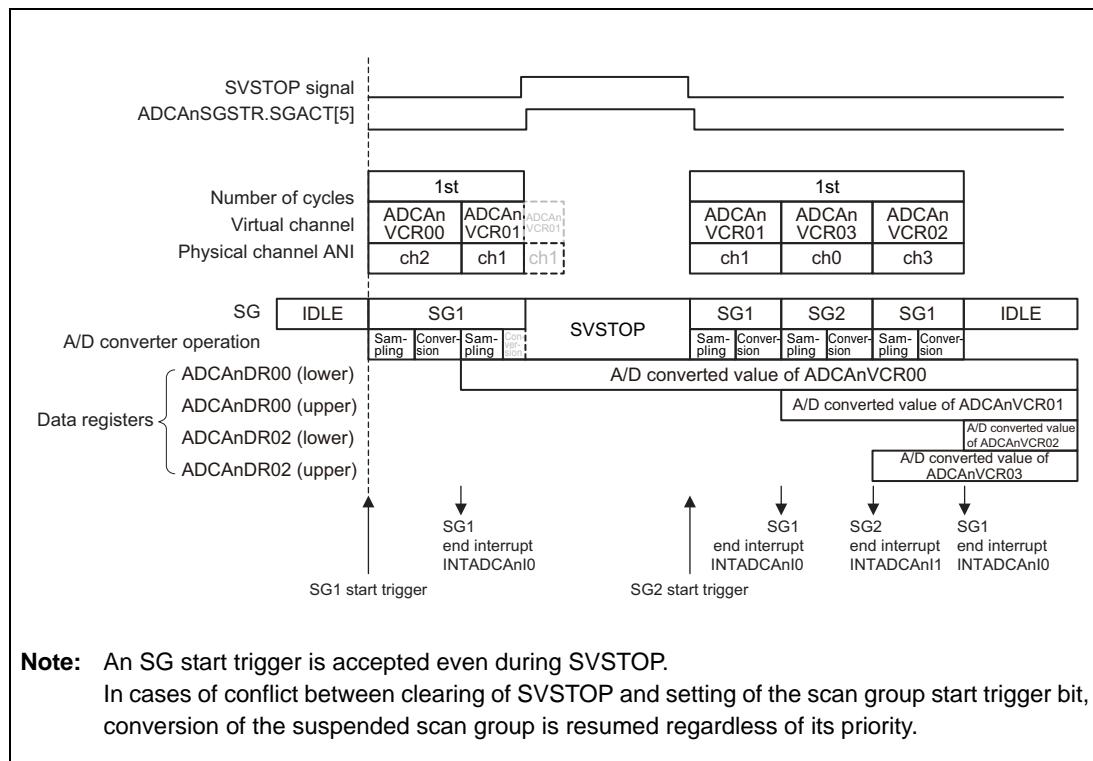
**Figure 29.25 Example of SVSTOP Operation in the IDLE State (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)**



**Figure 29.26 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 00, ADCAnEMU.SVSDIS = 0)**



**Figure 29.27 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)**



**Figure 29.28 Conflict of SVSTOP Clear and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)**

### 29.4.11 Activating Scan Group by a Hardware Trigger

Scan group x can be activated by the hardware trigger input to SGx\_TRG. As for the hardware trigger sources to be used, see **Table 29.47, List of A/D Conversion Hardware Triggers**. When activating SGx\_TRG by the hardware trigger, set the peripheral function to be used by the trigger and set the start trigger in the A/D conversion trigger select control register (ADCAnSGTSELx).

A hardware trigger from external trigger input pin requires digital filter setting. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**. More than one start trigger can be specified.

#### 29.4.11.1 Stopping Scan Group by ADHALT

Setting ADCAnADHALTR.HALT (A/D force halt trigger) to 1 forcibly halts the A/D conversion and clears the scan group status register (ADCAnSGSTR). The error flag of ADCAnULER (upper limit/lower limit error register) is not cleared. When ADCAnADHALTR.HALT is set, make sure that ADCAnSGSTR.SGACT has been cleared.

### 29.4.12 Scan End Interrupt Request

Scan group x can issue a scan end interrupt request (INT\_SGx) to INTC. If ADIE of ADCAnSGCRx is set to 1, INT\_SGx can be output after the SGx scan ends. If ADIE of ADCAnSGCRx is set to 0, the INT\_SGx output when the SGx scan ends can be disabled. If ADIE of ADCAnVCRj is set to 1, INT\_SGx can be output when A/D conversion for virtual channel j in SGx ends. If ADIE of ADCAnVCRj is set to 0, the INT\_SGx output when A/D conversion for virtual channel j in SGx ends can be disabled. Since SGx scan ending is simultaneous with A/D conversion ending for virtual channel j in SGx when ADIEs of both ADCAnSGCRx and ADCAnVCRj are set to 1, INT\_SGx occurs only once.

Example 1: A scan is executed for virtual channel 0 or 1 in SG1 when ADIE of ADCAnSGCR1 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 0.  
INT\_SG1 is output when A/D conversion ends for virtual channel 0.

Example 2: A scan is executed for virtual channel 0 or 1 in SG2 when ADIE of ADCAnSGCR2 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 1.  
INT\_SG2 is output when A/D conversion ends for virtual channel 0 and virtual channel 1.

Example 3: A scan is executed for virtual channel 0 or 1 in SG3 when ADIE of ADCAnSGCR3 is 1, ADIE of VCR0 is 0, and ADIE of VCR1 is 0.  
INT\_SG3 is output when a scan ends (when A/D conversion for virtual channel 1 ends).

Furthermore, the DMAC can be started when scan ends.

For the setting of DMAC, see **Section 7, DMA**.

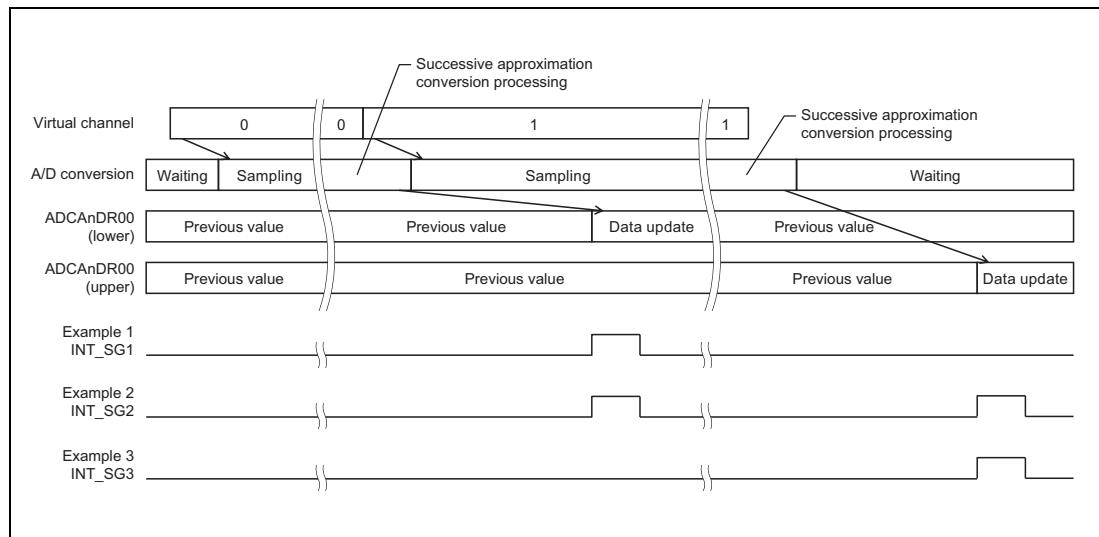


Figure 29.29 Example of a Scan Conversion End Interrupt Occurrence

### 29.4.13 A/D Error Interrupt Request

ADCA can issue an A/D error interrupt request (INT\_ADE) to INTC. For an error source for which ULEIE and OWEIE of ADCAnSFTCR are set to 1, the OR condition of the error source is issued as INT\_ADE. For an error source for which ULEIE and OWEIE of ADCAnSFTCR are set to 0, INT\_ADE does not output an interrupt.

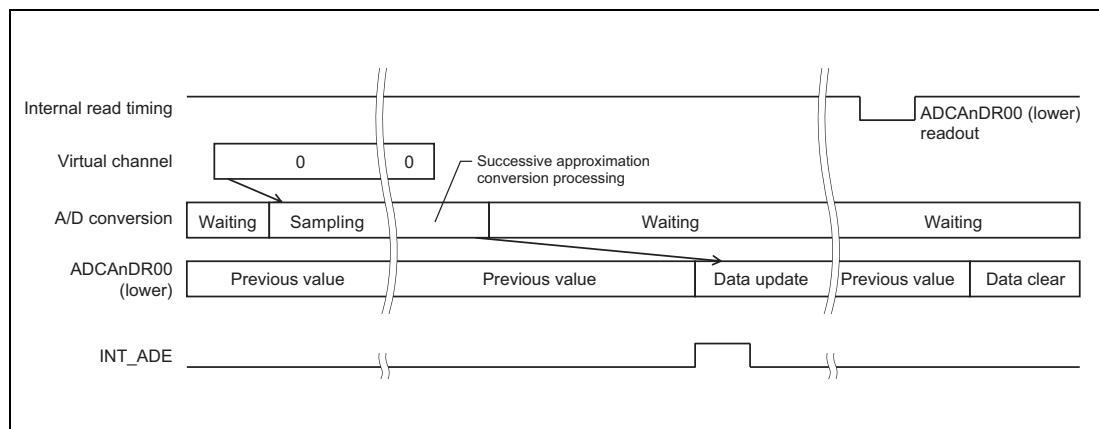


Figure 29.30 A/D Error Interrupt (Example: Overwrite Error)

## 29.5 Self-Diagnostic Function

To check the ADCAn function, the following self-diagnostic functions are available.

### Section 29.5.1, Diagnosis of A/D Conversion Circuit

### Section 29.5.2, Diagnosis of Channel Multiplexer

### Section 29.5.3, Diagnosis of Open Pins

### Section 29.5.4, Diagnosis of T&H Circuit

The overview of the self-diagnostic functions is shown in the figure below. A detailed description is given in the following sections.

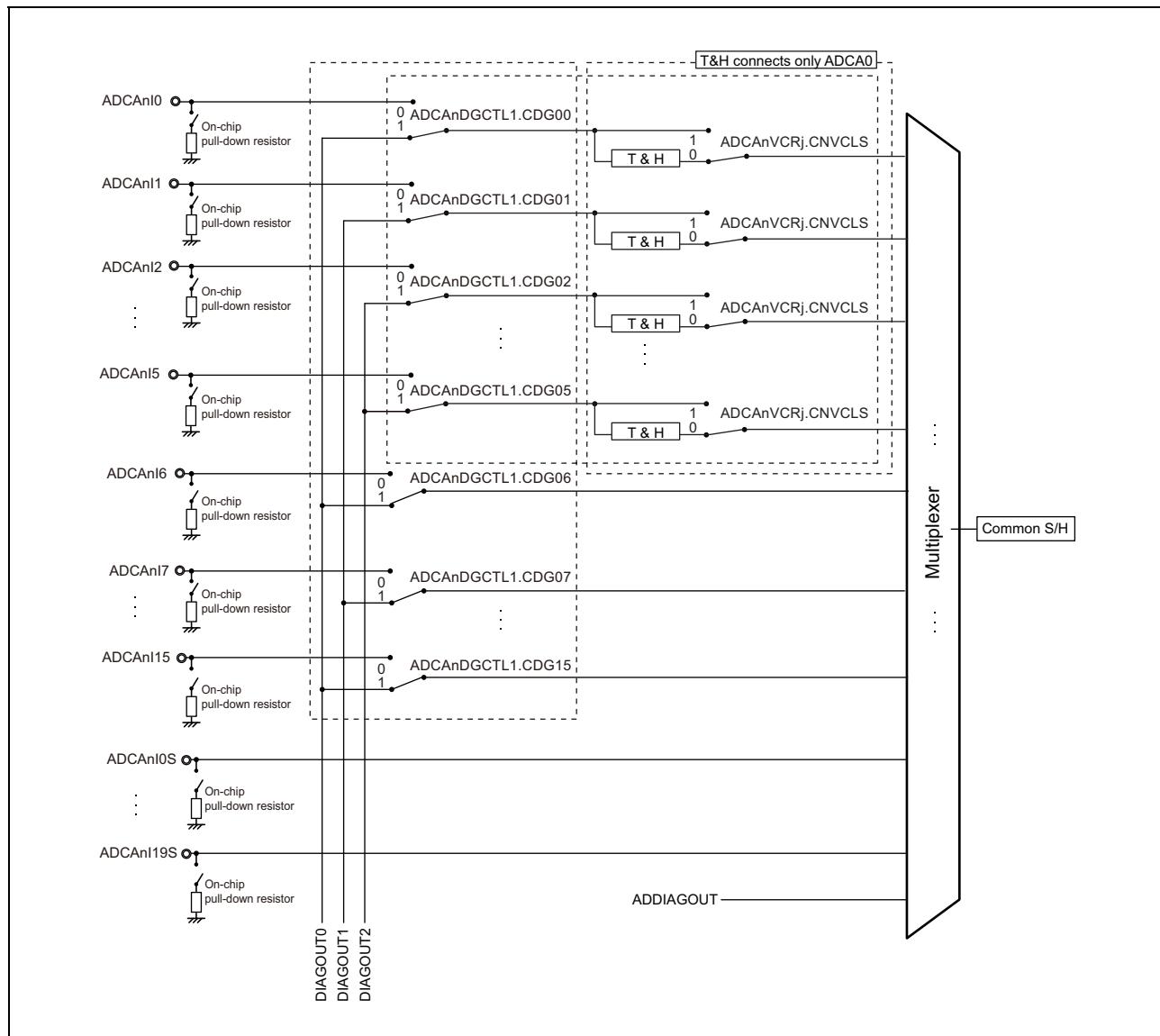


Figure 29.31 Overview of Self-Diagnostic Functions

#### NOTE

The functions in the dashed-line frames depend on the product.

## 29.5.1 Diagnosis of A/D Conversion Circuit

This function checks whether the A/D converter is operating normally by verifying the A/D conversion for self-diagnostic voltage (ADDIAGOUT) and the result of conversion. If the result of A/D conversion differs from the expected value, an internal circuit may be broken. The features of self-diagnosis of the A/D converter are as follows:

- As the self-diagnostic voltage (ADDIAGOUT) level,  $AnV_{REF}$ ,  $2/3AnV_{REF}$ ,  $1/3AnV_{REF}$ ,  $1/2AnV_{REF}$ , and  $AnVSS$  are selectable by the PSEL[2:0] bits in the ADCAnDGCTL0 register.
- Self-diagnosis of the A/D converter is enabled by performing A/D conversion on one of SG1 to SG3.

### 29.5.1.1 Diagnostic procedure

The diagnostic procedures are shown below.

Common settings for ADC should be made before self-diagnosis is to proceed.

1. Set ADCAnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCAnDGCTL0.PSEL[2:0] to select a self-diagnostic voltage level.
4. Set ADCAnADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Set an arbitrary bit of ADCAnVCRj.GCTRL[5:0] to  $100100_B$  to select the diagnosis channel.
7. Set ADCAnVCRj.ADIE = 1 to enable the A/D conversion end interrupt.
8. Set ADCAnSGVCSPx to specify the start pointer of virtual channel.
9. Set ADCAnSGVCEPx to specify the end pointer of virtual channel.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

#### NOTE

- During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCAnDGCTL0.PSEL[2:0]. However, the value of ADCAnDGCTL0.PSEL[2:0] becomes effective from the next A/D conversion.
- To clear ADCAnADCR.DGON, follow the procedure below:
  1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
  2. Clear ADCAnDGCTL0.PSEL[2:0].
  3. Clear ADCAnADCR.DGON.

### 29.5.2 Diagnosis of Channel Multiplexer

This function checks whether the path from the analog input to the A/D converter is normal.

Set the A/D conversion reference voltage (DIAGOUT0, DIAGOUT1, DIAGOUT2) by

ADCAnDGCTL0.PSEL[2:0] and the channels to be connected by the ADCAnDGCTL1 register to perform A/D conversion using multiple analog channels.

If the result of A/D conversion differs from the expected value, an internal circuit may be broken. The features of self-diagnosis of the channel multiplexer are as follows:

- Channels for diagnosis can be arbitrarily selected from among ADCA0I0 to ADCA0I15 and ADCA1I0 to ADCA1I15.
- As the self-diagnostic voltage level,  $2/3AnV_{REF}$ ,  $1/3AnV_{REF}$ , and  $1/2AnV_{REF}$  are selectable and one of the three reference voltage levels can be allocated to each channel.

**Table 29.53 Selection of Channel to be Diagnosed**

Connection	Select Channel
DIAGOUT0	Channels 0, 3, 6, 9, 12, and 15
DIAGOUT1	Channels 1, 4, 7, 10, and 13
DIAGOUT2	Channels 2, 5, 8, 11, and 14

- Self-diagnosis of the channel multiplexer is enabled by performing A/D conversion on one of SG1 to SG3 by using multiple channels.

### 29.5.2.1 Diagnostic procedure

The diagnostic procedure is shown below.

Common settings for ADC should be made before self-diagnosis is to proceed.

1. Set ADCAnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCAnDGCTL0.PSEL[2:0] to select a self-diagnostic voltage level.
4. Set ADCAnADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Use two or more ADCAnVCRj registers.  
Set ADCAnVCRj.GCTRL[5:0] bits to select physical channels.  
Set ADCAnVCRj.ADIE bit to enable the A/D conversion end interrupt.
7. Set ADCAnSGVCSPx register to specify the start pointer of virtual channel.
8. Set ADCAnSGVCEPx register to specify the end pointer of virtual channel.
9. Set ADCAnDGCTL1 register to specify the physical channel to the self-diagnostic channel.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

#### NOTE

- During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCAnDGCTL0.PSEL[2:0]. However, the value of ADCAnDGCTL0.PSEL[2:0] becomes effective from the next A/D conversion.
- To clear ADCAnADCR.DGON, follow the procedure below:
  1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
  2. Clear ADCAnDGCTL0.PSEL[2:0].
  3. Clear ADCAnADCR.DGON.

### 29.5.3 Diagnosis of Open Pins

This function detects whether the analog input pin (ADCAnIm, ADCAnImS) is open due to disconnection, etc.

An internal pull-down resistor can be connected to diagnose the analog input pin.

Connect the analog input pin (ADCAnIm, ADCAnImS) with the pull-down resistor for self-diagnosis for A/D conversion of the target channels.

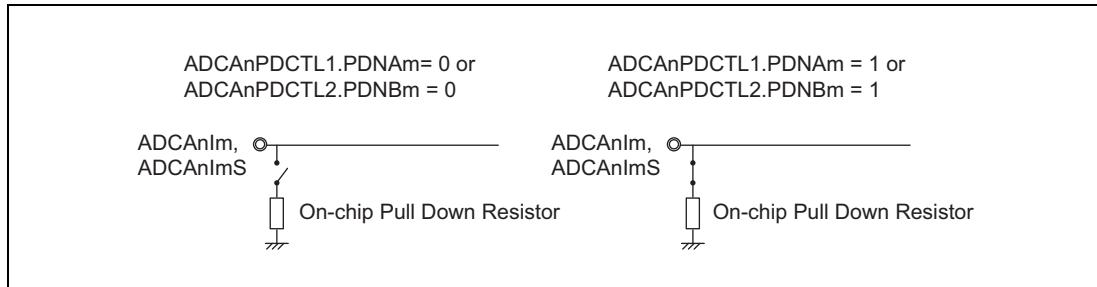


Figure 29.32 Setting of On-chip Pull Down Resistor

When there is a disconnection, the conversion result is almost 0 V and it indicates an open detection.

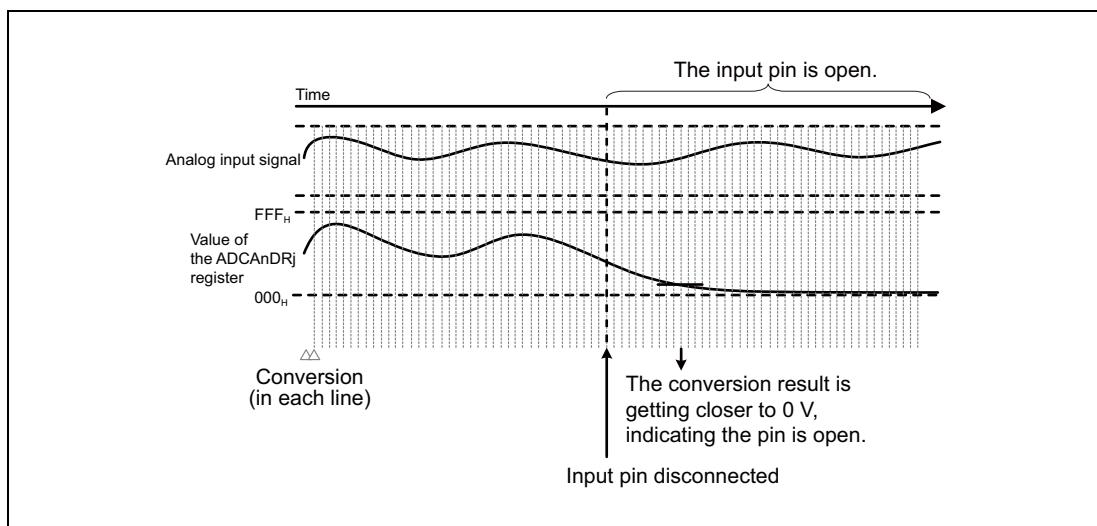


Figure 29.33 Analog Input Signal Disconnection Detection

#### CAUTIONS

1. The pull-down resistors must not be connected during normal A/D conversion operation. Connected pull-down resistors may lead to a drop in the input voltage and result in erroneous A/D conversion results.
2. When the analog input voltage is nearly equal to the voltage level which is pulled down, a disconnection cannot be detected by this function.

### 29.5.3.1 Diagnostic procedure

1. Set the ADCAnPDCTL1.PDNA or ADCAnPDCTL2.PDNB bit which correspond to analog input pins (ADCAnIm, ADCAnImS) to be diagnosed to enable the pull down resistor.
2. Generate the start trigger of scan group to perform the A/D conversion.
3. Perform the A/D conversion multiple times on the same analog input.
4. Monitor the channel's A/D conversion results and check if any result declines to 0 V.

## 29.5.4 Diagnosis of T&H Circuit

This function is used to diagnose proper operation of the T&H0 to T&H5 circuits for ADCA0I0 to ADCA0I5.

Virtual channel registers 33 to 35 (ADCA0VCR33 to 35) are used exclusively for comparison of the potential conversion result using the T&H circuit and that obtained without using the T&H circuit to detect a failure of the T&H circuit.

For this diagnosis, the ADCA0THACR.HLDCTE is set to 1 (ADCA0THBCR.HLDCTE = 1) and ADCA0THACR.HLDTE to 0 (ADCA0THBCR.HLDTE = 0) and the A/D conversion trigger is used as the hold start/end trigger. Connect the reference voltage signal (DIAGOUT0, DIAGOUT1, or DIAGOUT2) selected by ADCAnDGCTL0.PSEL[2:0] to the target channels for diagnosis by using the ADCAnDGCTL1 register.

### 29.5.4.1 Diagnostic Procedure (in case of T&H circuit ch0 diagnosis)

1. Set ADCA0ADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCA0DGCTL0.PSEL[2:0] = 001<sub>B</sub> to select 1/3AnV<sub>REF</sub> voltage level.
4. Set ADCA0ADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Set ADCA0DGCTL1.CDG0 = 1 to enable DIAGOUT0.
7. Set ADCA0VCR33.GCTRL[5:0] to ADCAVCR35.GCTRL[5:0] to 000000<sub>B</sub> to select physical ch0.
8. Set ADCA0VCR33.CNVCLS and ADCA0VCR34.CNVCLS to 1 to select normal conversion.
9. Set ADCA0VCR35.CNVCLS = 0 to select hold value conversion.
10. Set ADCA0THACR.SGS[1:0] = 01<sub>B</sub> to select SG1 to “T&H group A”.
11. Set ADCA0THER.TH0E = 1 to enable T&H circuit ch0.
12. Set ADCA0THGSR.TH0GS = 0 to select T&H circuit ch0 to “T&H group A”.
13. Set ADCA0SGVCSP1.VCSP[5:0] = 100001<sub>B</sub> to select SG1 start pointer to VCR33.
14. Set ADCA0SGVCEP1.VCEP[5:0] = 100011<sub>B</sub> to select SG1 end pointer to VCR35.
15. Set ADCA0DGCTL0.PSEL[2:0] = 011<sub>B</sub> to select 2/3AnV<sub>REF</sub> voltage level.
16. Set ADCA0THSMPSTCR.SMPST = 1 to execute T&H sampling.
17. Wait for 500 ns.
18. Set ADCA0SGSTCR1.SGST = 1 to execute SG1 A/D conversion.
19. Read ADCA0DIR33 to ADCA0DIR35, and check A/D conversion result to see if SG1 A/D conversion has finished.

#### NOTES

To clear ADCAnADCR.DGON, follow the procedure below:

1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
2. Use the ADCAnTHER register to disable the diagnosed T&Hk.

- 
3. Clear ADCAnDGCTL0.PSEL[2:0].
  4. Clear ADCAnADCR.DGON.
-

### 29.5.4.2 Diagnosis Mechanism

- (1) A reference voltage “A” is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The T&H circuit holds the voltage “A” and an A/D conversion is performed without using the T&H circuit.

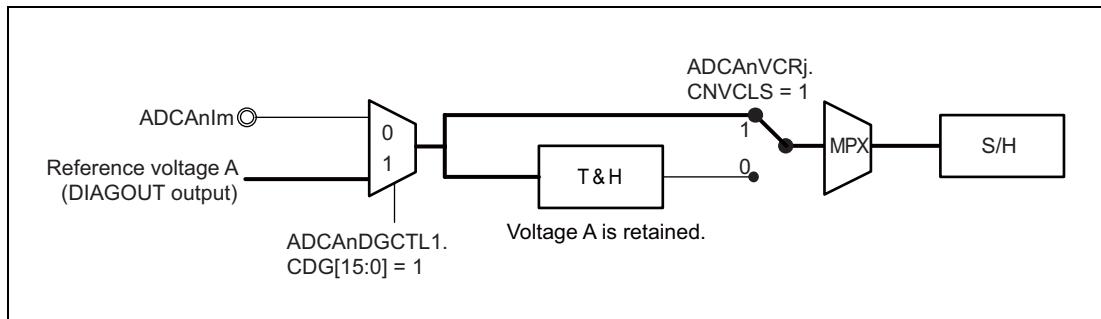


Figure 29.34 T&H Circuit Diagnostic Mechanisms (1)

- (2) A reference voltage “B” is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The T&H circuit still holds the voltage “A” and an A/D conversion is performed without using the T&H circuit.

Note that since reference voltage “A” is being held, the reference voltage “B” is not held.

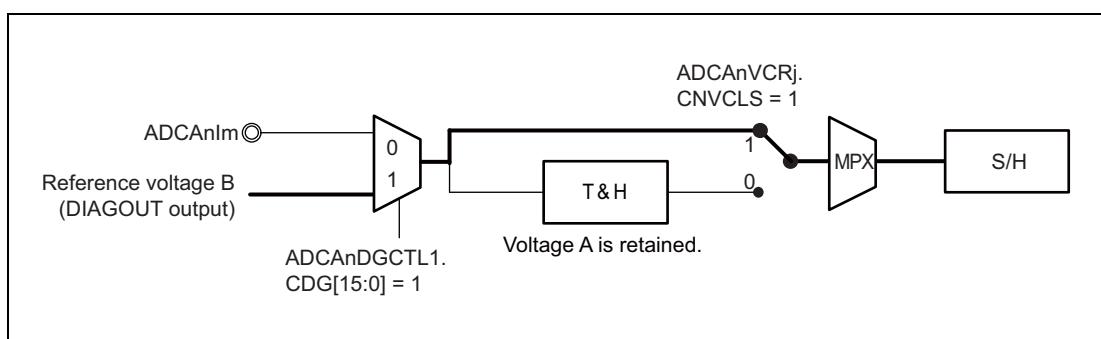


Figure 29.35 T&H Circuit Diagnostic Mechanisms (2)

- (3) An A/D conversion is performed using the T&H circuit. The T&H circuit continues to hold the voltage A.

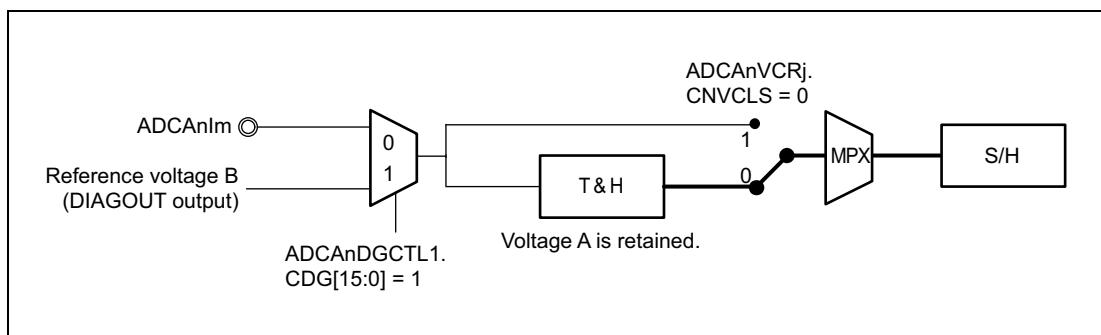


Figure 29.36 T&H Circuit Diagnostic Mechanisms (3)

- (4) The diagnosis of T&H circuit is successful if the following results are obtained:
  1. The first result (step 1) is voltage “A”.
  2. The second result (step 2) is voltage “B”.
  3. The last result (step 3) is voltage “A” again.

## 29.6 Definition of A/D Conversion Accuracy

A/D conversion accuracy is defined as follows:

- Resolution  
Digital output code value from the A/D converter
- Quantization error  
An error essentially contained in the A/D converter, which is assumed as 1/2 LSB (**Figure 29.37**).
- Offset error  
Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value  $000_H$  to  $001_H$ . However, the quantization error is not included.
- Full scale error  
Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from  $FFE_H$  to  $FFF_H$ . However, the quantization error is not included.
- DNL (Differential nonlinear error)  
Deviation between the ideal digital output code width ( $V_q$ ) and the actual digital output code width ( $V_a$ ), which is assumed as  $(V_a - V_q)/V_q$ . However, the offset error, the full scale error, and the quantization error are not included.
- INL (Integral nonlinear error)  
Deviation of the actual value from the ideal A/D conversion characteristics between the zero voltage and the full scale voltage, which is assumed as an integral of DNL from  $000_H$  to a digital output code. However, the offset error, the full scale error, and the quantization error are not included.
- Absolute accuracy  
Deviation between the digital value and the analog input value. The offset error, the full scale error, the quantization error, DNL, and INL are included.

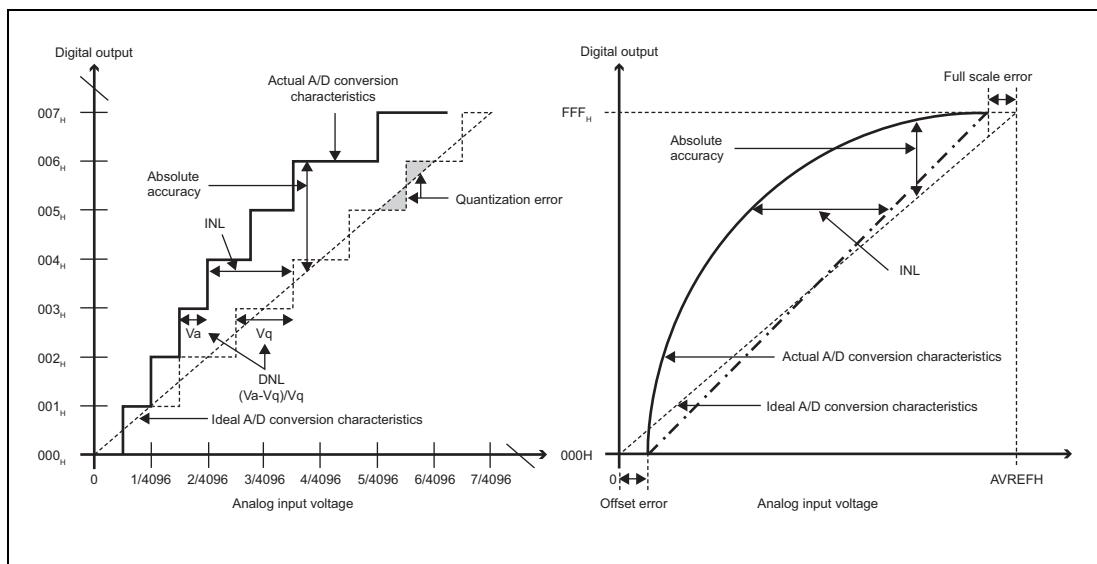


Figure 29.37 Definition of A/D Conversion Accuracy

## 29.7 Usage Notes

### 29.7.1 Range of Channel Input Voltage

#### CAUTION

ADCAnIm and ADCAnImS input voltages should be used within the specification range. If the channel input voltage exceeds AnVREF or falls below AnVSS, the converted value of the channel is saturated and may influence electric characteristics of other channels.

When ad over-voltage is applied to a pin ADCAnIm and at the same pin the ADC self-diagnosis ( diagnosis of channel multiplexer ) is executed an offset voltage to the diagnosis voltage can be measured. According to the diagnosis setting, adjacent pins are affected from the pin which have over-voltaged. There are two advices for avoiding a problem. The group which have no injected current can be measure with no influence.

**Case1 : Injected current is applied at one pin where the ADC self-diagnosis is executed**

**First ADC self-diagnosis conversion cycle:**

→ only ADCA0I0 is selected (CDG0=1, CDG1-15=0) ← injected current

**Second ADC self-diagnosis conversion cycle:**

→ other ADCA0I1-15 are selected (CDG0=0, CDG1-15=1) ← not injected current

**Case2: Injected current is applied at multiple pins where the ADC self-diagnosis is executed**

**First ADC self-diagnosis conversion cycle:**

→ only ADCA0I0 is selected (CDG0=1, CDG1-15=0) ← injected current

**Second ADC self-diagnosis conversion cycle:**

→ only ADCA0I1 is selected (CDG0=0, CDG1=1, CDG2-15=0) ← injected current

**Third ADC self-diagnosis conversion cycle:**

→ only ADCA0I2 is selected (CDG0=0, CDG1=0, CDG2=1, CDG3-15=0) ← injected current

**n ADC self-diagnosis conversion cycle:**

→ other ADCA0In-15 are selected (CDGn=0, CDGn-15=1) ← no injected current

### 29.7.2 Notes on Application Design

#### (1) Analog input pins (ADCAnIm, ADCAnImS)

- Ensure that the input voltages on the ADCAnIm and ADCAnImS pins are within the specified ranges. We recommend using diodes with  $V_F$  of 0.3V or below to form a clamp to avoid the input of voltages at or above AnVREF and at or below AnVSS. The results of conversion for input voltages at or above AnVREF and at or below AnVSS are undefined and so are not guaranteed. Input of such voltages can also affect the results of conversion on other channels.
- Reduce noise on the analog input pins (ADCAnIm and ADCAnImS) by connecting a resistor  $R_E$  between the pins and the external sources of analog input signals for conversion and capacitor  $C_E$  to the AnVSS pins.

- Avoid analog signal lines crossing digital signal lines and vice versa, since this can introduce noise and reduce performance in A/D conversion.
- We recommend avoiding the driving of large currents through input and output pins near the ADCAnIm and ADCAnImS pins and toggled signals in particular should be kept away from these pins.
- If you are using the standby functions, set the ADCAnADHALTR.HALT, ADCAnSGCRx.TRGMD, and ADCAnTHER.THkE bits, which are to be effective on standby, to 0.
- If you are using the LPS on ADCA0 (also when standby function is used), set the ADCA0SGCRx.TRGMD bit to 1 and the ADCA0SGTSELx.TxSEL bits to SEQTRG (LPS). For details, see **Section 12, Low-Power Sampler (LPS)**.
- Do not connect a channel to be used with the T&H function to an external analog multiplexer.
- Changes to physical and virtual channels during operation while the T&H function is in use is prohibited.
- Writing to PWM-Diag-related registers while PWM-Diag is not in use is prohibited.

## (2) Power Wiring

The following methods are recommended to minimize the influence of switching noise from digital circuits on A/D converter accuracy.

- Connect markedly thick wiring patterns to the mesh pattern or connect solid patterns to the power-supply lines.
  - Insert bypass capacitors between power-supply pins (EVCC, BVCC, and AnVREF) and ground pins (EVSS, BVSS, and AnVSS).
  - We recommend separating the analog power supply (AnVREF) from the digital power supplies (EVCC and BVSS) and providing the voltages from a series regulator. If the analog power supply is to come from the same source as that of the digital power supplies, wire the analog and digital power supplies to an electrolytic capacitor, and provide separate wiring patterns on the board. We also recommend inserting a chip inductor in the input for the analog power supply. Furthermore, earth the analog and digital grounds to the same point on an electrolytic capacitor, and provide separate wiring patterns for the grounds on the board.
- The analog power supply also serves as the analog reference voltage for this product.

## (3) Variation in A/D converted data

The effects of noise and variations in the power supply voltages lead to dispersal of the results of A/D conversion. Furthermore, noise on the analog input pins (ADCAnIm and ADCAnImS) or on the reference voltage input pins (AnVREF and AnVSS) can lead to the results of A/D conversion being incorrect.

Apply software processing to avoid ill effects on the system of fluctuations in or incorrectness of the results of A/D conversion.

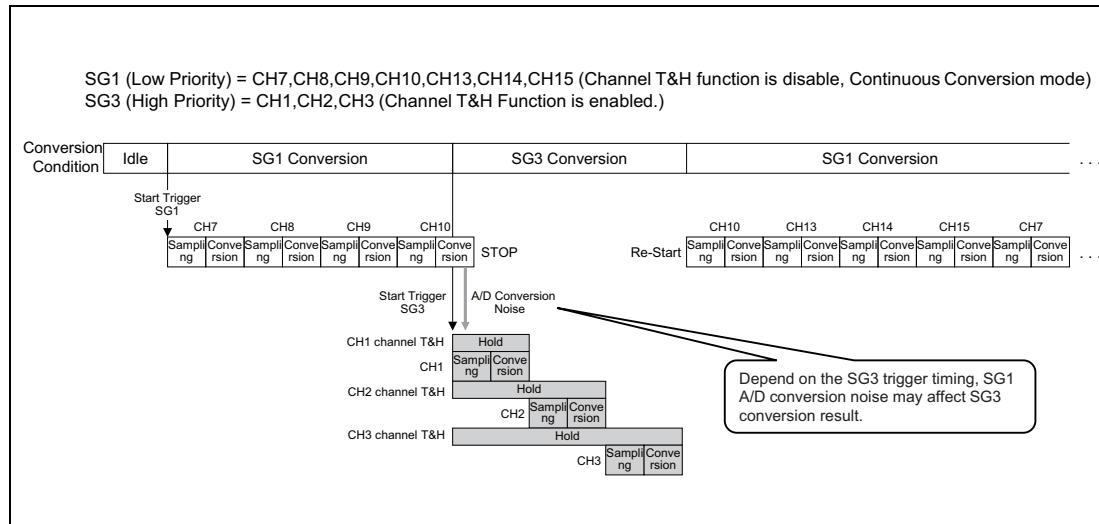
Examples of software handling are described below.

- Use averaged values from several rounds of A/D conversion
- Execute A/D conversion for several time and omit extreme results
- Repeat the processing for abnormalities to check for repeated abnormalities in the case of results of A/D conversion which will cause malfunctions of the system.

There is a possibility that A/D conversion accuracy of high priority SG become worse when following both conditions are applicable.

- (1) During the A/D conversion of low priority SG (e.g. SG1), conversion trigger of the high priority SG (e.g. SG3) occur.
- (2) The channel T&H function of high priority SG (e.g. SG3) is enabled.

The above case is one of an example of SG combination. The SG priority is SG4 > SG3 > SG2 > SG1. The fluctuation of conversion error depends on the external circuit and devices mounted on the customer board.

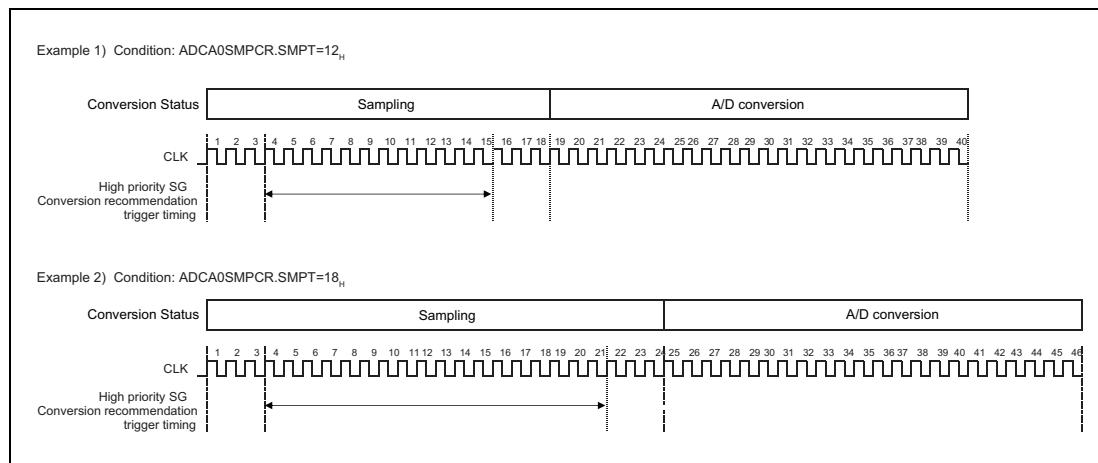


**Figure 29.38 SG priority**

Examples of software handling are described below.

- Low priority SG (e.g. SG1) A/D conversion have to be finished 3 ADCLK before the conversion trigger of high priority SG (e.g. SG3), if the SG3 contains the channels which channel T&H is available.
- If there is a case that high priority SG (e.g. SG3) conversion trigger occur during the conversion of low priority SG (e.g. SG1), disable the high priority SG channel T&H function.
- If the both conditions mentioned in previous page need to be used, adjust high priority SG (e.g. SG3) conversion trigger timing to synchronize with the following timing (the period shown by arrowed line in the following figure) during A/D conversion of low priority SG (e.g. SG1).

Even if trigger timing is adjusted above recommendation time, conversion error specified in Data Sheet cannot be removed.



**Figure 29.39 SG conversion trigger timing**

When it is not possible to use the above 3 software handlings, following process is recommended.

- Doing A/D conversion several times and using average of several A/D conversion results.
- Doing continuous A/D conversion several times, remove abnormal conversion result and use only the other results.
- When abnormal A/D conversion result is detected, not to proceed abnormal operation immediately, doing one more A/D conversion before proceeding abnormal operation.

The effect of above process is depend on the external circuit and devices mounted on the customer board. Sufficient evaluation of the system is recommended.

#### (4) Alternative Input/Output

Analog input (ADCAnIm, ADCAnImS) pins can be used as port pins.

Do not read from input port pins or write to output port pins while an ADCAnIm or ADCAnImS pin function is selected and handling A/D conversion. Doing so may reduce the accuracy of conversion.

Fluctuations in output current from output port pins due to the effects of an external circuit connected to a port pin while A/D conversion is in progress may also reduce the accuracy of conversion. If digital pulses are applied to or digital pulses are output through a pin adjacent to a pin for which A/D conversion is in progress, the A/D converted value may not be as expected due to coupling noise. Accordingly, do not apply pulses to or output pulses from a pin adjacent to a pin for which A/D conversion is in progress.

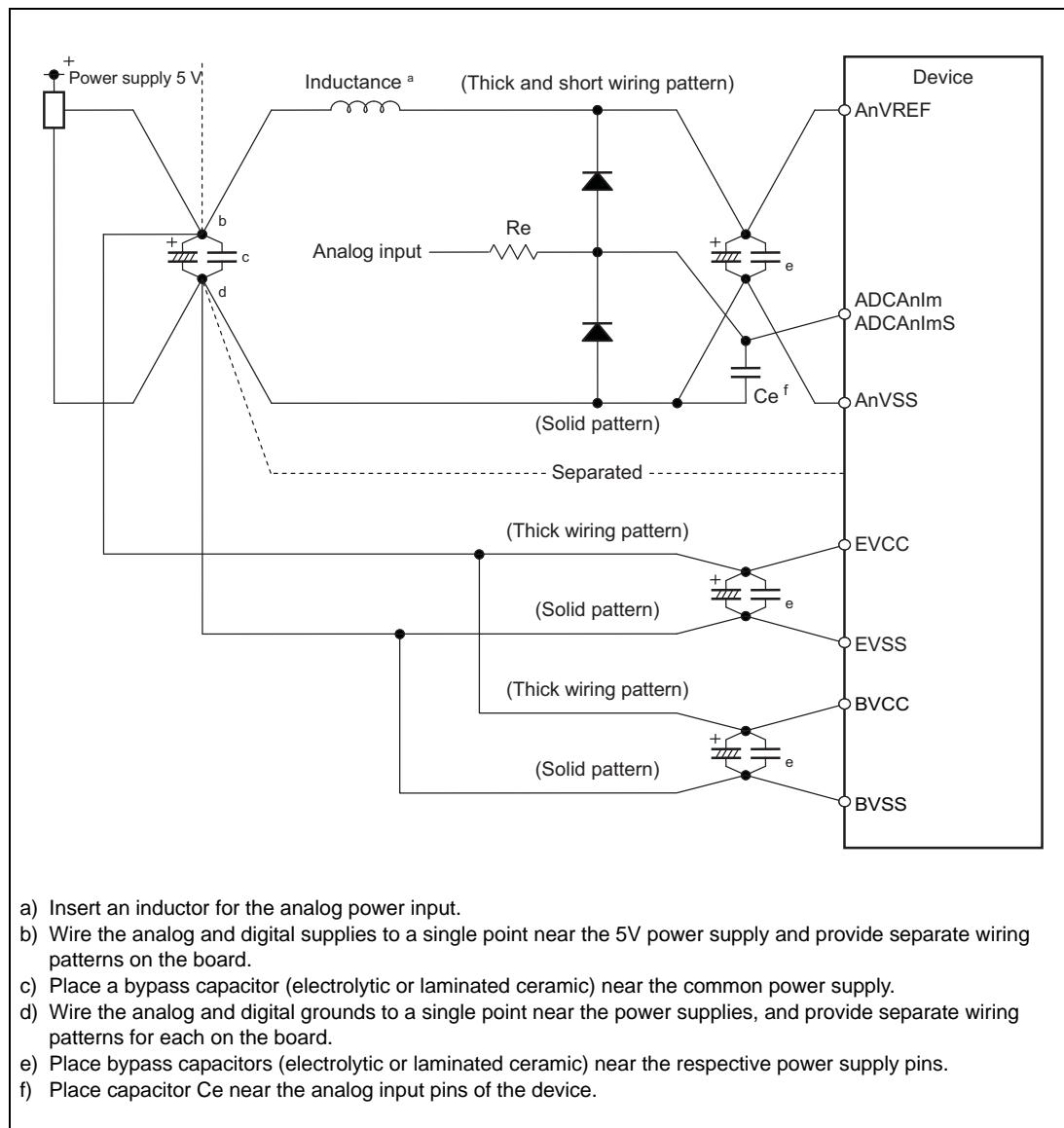


Figure 29.40 Example of Power Wiring

## Section 30 Key Return (KR)

This section contains a generic description of the Key Return (KR) function.

The first part in this section describes the RH850/F1L specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the KR.

### 30.1 Features of RH850/F1L KR

#### 30.1.1 Number of Units and Channels

This microcontroller has the following number of KR units.

**Table 30.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units				1		
Name				KRn (n = 0)		

The KR unit has the Key Return function of the following number of channels.

**Table 30.2 KRn Unit Configurations and Channels**

Unit Name (Channel Name) KRn	No. of Channels	RH850/F1L 64 pins (6 ch)	RH850/F1L 64 pins (8 ch)	RH850/F1L 80 pins (8 ch)	RH850/F1L 100 pins (8 ch)	RH850/F1L 144 pins (8 ch)	RH850/F1L 176 pins (8 ch)
KR0	8	—	√	√	√	√	√
	6	√	—	—	—	—	—

**Table 30.3 Indices**

Index	Description
n	Throughout this section, individual KR units are identified by the index "n" (n = 0); for example, KRnKRM indicates the key return mode register.
m	Throughout this section, individual KR channels are identified by the index "m" (m = 0 to 7); for example, KRnKRMm indicates the key input enable bit of KRnKRM (key return mode register).

#### 30.1.2 Register Base Address

KRn base address is listed in the following table.

KRn register addresses are given as offsets from the base address.

**Table 30.4 Register Base Address**

Base Address Name	Base Address
<KR0_base>	FFF7 8000 <sub>H</sub>

### 30.1.3 Clock Supply

The KRn clock supply is shown in the following table.

**Table 30.5 Clock Supply**

Unit Name	Unit Clock Name	Supply Clock Name
KR0	PCLK	CPUCLK2
	Register access clock	CPUCLK2

### 30.1.4 Interrupt Requests

KRn interrupt requests are listed in the following table:

**Table 30.6 Interrupt Requests**

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
KR0			
INTKRn	Key interrupt	82	—

### 30.1.5 Reset Sources

KRn reset sources are listed in the following table. KRn is initialized by these reset sources.

**Table 30.7 Reset Sources**

Unit Name	Reset Source
KR0	All reset sources (ISORES)

### 30.1.6 External Input/Output Signals

External input/output signals of KRn are listed below.

**Table 30.8 External Input/Output Signals**

Unit Signal Name	Outline	Alternative Port Pin Signal
KR0		
KRnTPKR7 to KRnTPKR0	Key input signal	KR0I7 to KR0I0

## 30.2 Overview

### 30.2.1 Functional Overview

The Key Return function has the following features:

A key interrupt request signal (INTKRn) can be generated by inputting a falling signal, that goes from high to low, to any of the eight key input pins (KRnTPKR7 to KRnTPKR0).

### 30.2.2 Block Diagram

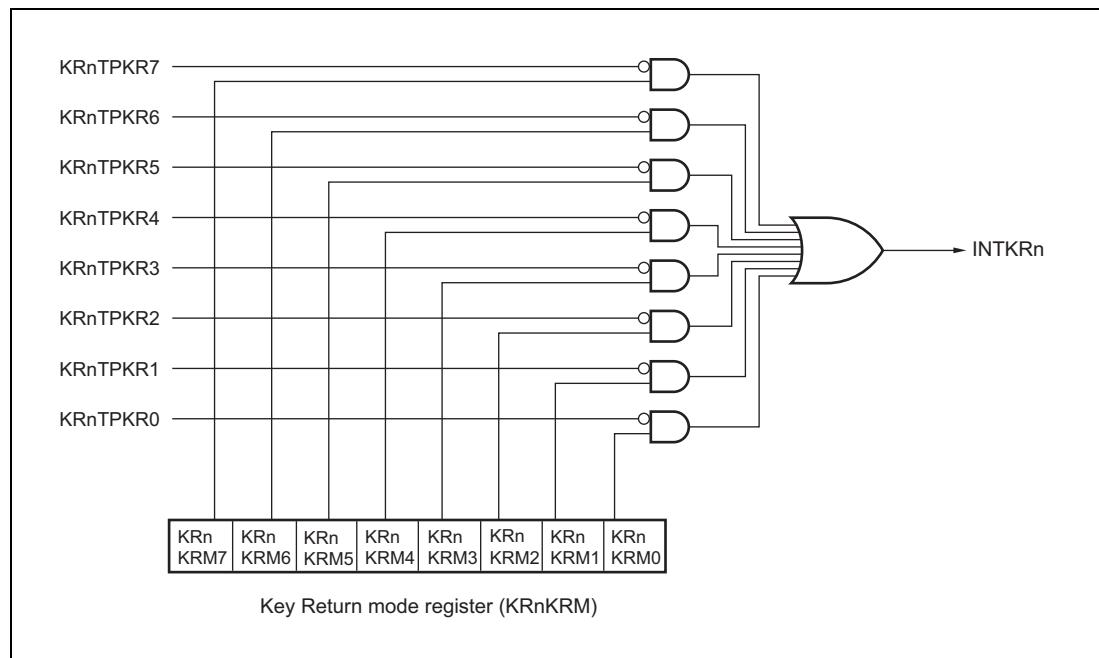


Figure 30.1 Block Diagram of the Key Return Function

## 30.3 Registers

### 30.3.1 List of Registers

KR registers are listed in the following table.

For details about <KRn\_base>, see **Section 30.1.2, Register Base Address**.

Table 30.9 List of Registers

Module Name	Register Name	Symbol	Address
KRn	Key return mode register	KRnKRM	<KRn_base>

### 30.3.2 KRnKRM — Key Return Mode Register

This register enables/disables the key input signal detection.

**Access:** This register can be read or written in 8-bit or 1-bit units.

**Address:** <KRn\_base>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	KRnKRM7	KRnKRM6	KRnKRM5	KRnKRM4	KRnKRM3	KRnKRM2	KRnKRM1	KRnKRM0
Value after reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Value after reset	0	0	0	0	0	0	0	0

Table 30.10 KRnKRM - Key Return Mode Register Contents

Bit position	Bit name	Function
7 to 0	KRnKRMm	Enables/disables the key input signal detection. 0: Disabled 1: Enabled

## 30.4 Operation

### 30.4.1 Interrupt Request INTKRn

The interrupt request INTKRn is generated when the level of the corresponding key input pin KRnTPKRM is changed from high to low while input to the key input pin KRnTPKRM is enabled (KRnKRM.KRnKRMm = 1).

**Figure 30.2** shows how the interrupt request is generated:

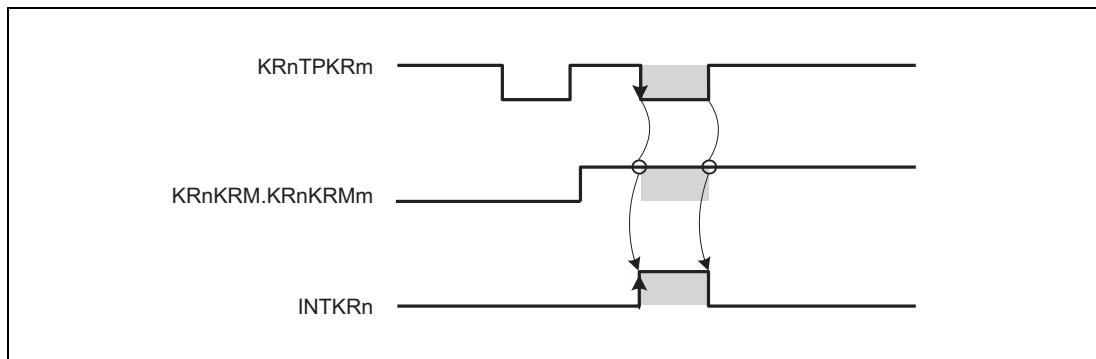


Figure 30.2 Interrupt Request Generation

#### CAUTIONS

1. The change of a key input pin (KRnTPKRM) level from high to low does not trigger another INTKRn if any of the key return input pins are already low. The next INTKRn is only triggered by a key input pin level changing from high to low if all other key input pins are high.
2. If the key input value changes at the same time the setting of KRnKRM.KRnKRMm is changed, an unintended key interrupt request INTKRn might be generated. Therefore, mask (disable) INTKRn of the interrupt controller before changing KRnKRM.KRnKRMm from 0 to 1, or from 1 to 0.

## Section 31 Functional Safety

This section provides an overview of the safety mechanisms included in the RH850/F1L Series.

This microcontroller has been developed as a Safety Element out of Context (SEooC) in accordance with ISO26262.

For more information about the development process and safety mechanisms, please contact our sales office.

The following are the failure detection functions provided by this microcontroller.

### 31.1 Overview

#### ECC and EDC

Detect failures of memories and data transfer paths and correct some types of failures.

For details, see the following sections.

- CSIIn RAM: **Section 15.7, Detection and Correction of Errors in CSIIn RAM**
- RS-CAN RAM: **Section 19.11, Detection and Correction of Errors in RS-CAN RAM**
- Code flash: **Section 35.9, ECC Error Detection and Correction for Code Flash Memory**
- Data flash: **Section 35.10, ECC Error Detection and Correction for Data Flash Memory**
- Local RAM: **Section 36.3, ECC Error Detection and Correction for Local RAM**

#### Memory Protection

Detects erroneous access to memories and peripheral circuits to protect the data in these elements against erroneous access.

For details, see **Section 3.1.1.4, Memory Management** and **Section 3.3.4, MPU Function Registers**.

#### Supply Voltage Monitor

Monitors the external and internal power supply voltages to detect an abnormal voltage.

For details, see **Section 9, Supply Voltage Monitor**.

#### Clock Monitor

Monitors the clock operation to detect an abnormal operation.

For details, see **Section 10.7, Clock Monitor A (CLMA)**.

#### Data CRC

Generates CRC to detect data errors.

For details, see **Section 32, Data CRC (DCRA)**.

### **Write-Protected Registers**

The write-protected registers are protected from inadvertent write access due to erroneous program execution.

For details, see **Section 4, Write-Protected Registers**.

## Section 32 Data CRC (DCRA)

This section contains a generic description of the data CRC function A (DCRA).

The first part in this section describes the RH850/F1L specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the DCRA.

### 32.1 Features of RH850/F1L DCRA

#### 32.1.1 Number of Units

This microcontroller has the following number of DCRA units.

**Table 32.1 Number of Units**

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	4	4	4	4
Name	DCRAn (n = 0)	DCRAn (n = 0)	DCRAn (n = 0 to 3)			

**Table 32.2 Index**

Index	Description
n	Throughout this section, the individual data CRC function A units are identified by the index "n" (n = 0 to 3); for example, DCRAnCTL indicates the DCRAn control register.

#### 32.1.2 Register Base Address

DCRAn base addresses are listed in the following table.

DCRAn register addresses are given as offsets from the base addresses.

**Table 32.3 Register Base Addresses**

Base Address Name	Base Address
<DCRA0_base>	FFF7 0000 <sub>H</sub>
<DCRA1_base>	FFF7 1000 <sub>H</sub>
<DCRA2_base>	FFF7 2000 <sub>H</sub>
<DCRA3_base>	FFF7 3000 <sub>H</sub>

### 32.1.3 Clock Supply

The DCRA<sub>n</sub> clock supply is shown in the following table.

**Table 32.4 Clock Supply**

Unit Name	Unit Clock Name	Clock Supply Name
DCRA <sub>n</sub>	PCLK	CPUCLK2
	Register access clock	CPUCLK2

### 32.1.4 Reset Sources

DCRA<sub>n</sub> reset sources are listed in the following table. DCRA<sub>n</sub> is initialized by these reset sources.

**Table 32.5 Reset Sources**

Unit Name	Reset Source
DCRA <sub>n</sub>	All reset sources (ISORES)

## 32.2 Overview

### 32.2.1 Functional Overview

The data CRC function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC  
 $(X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X^1+1)$
- 16-bit CCITT CRC  
 $(X^{16}+X^{12}+X^5+1)$
- CRC of an arbitrary data block length can be generated.
- After initialization of the CRC data register, every write access to the CRC input register generates a new CRC according to the selected polynomial, and the result is stored in the CRC data register.

### 32.2.2 Block Diagram

The following picture shows the block diagram of the data CRC function A.

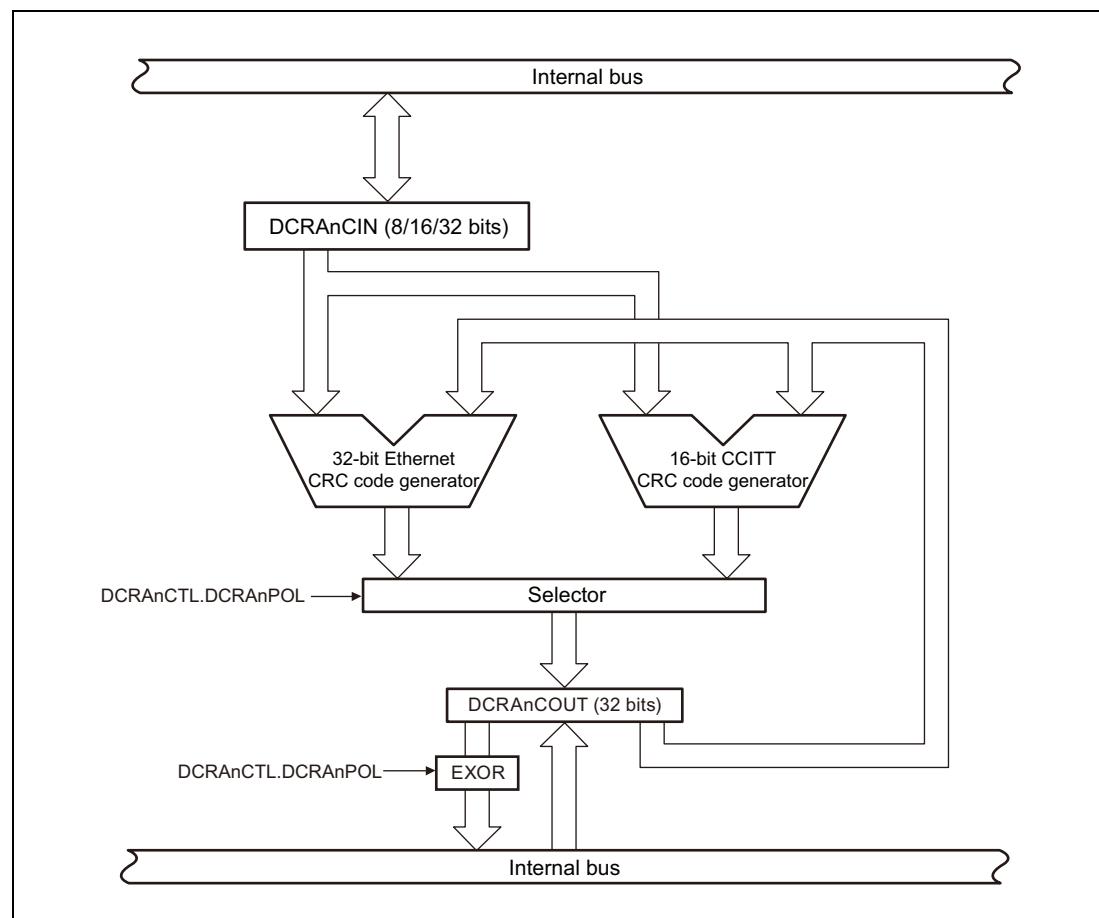
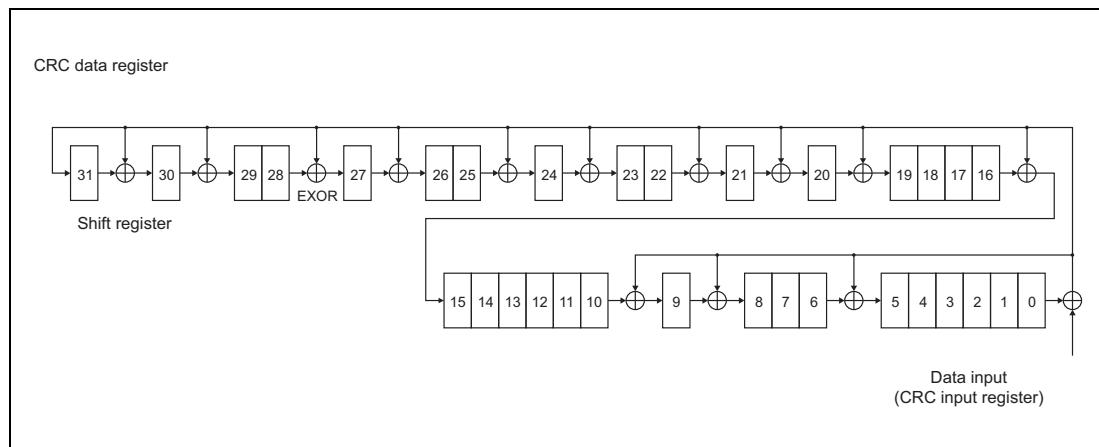


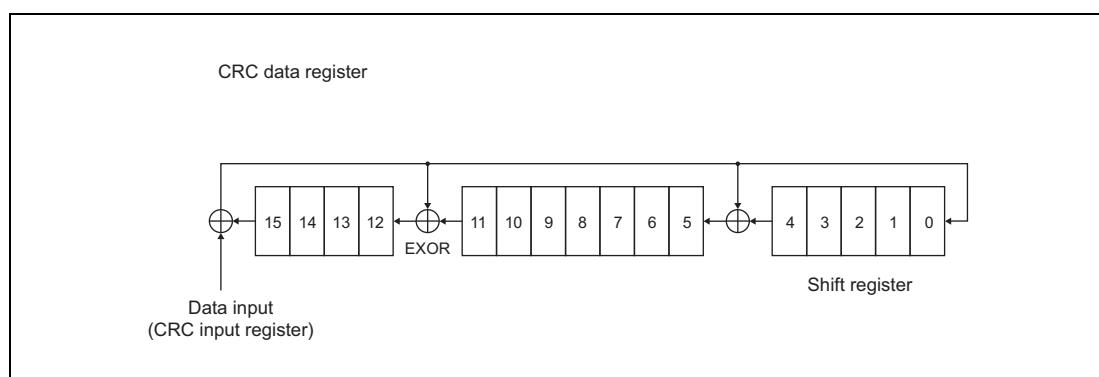
Figure 32.1 Block Diagram of Data CRC Function A

### 32.2.3 Operational Circuit

- 32-bit Ethernet



- 16-bit CCITT



## 32.3 Registers

### 32.3.1 List of Registers

DCRA registers are listed in the following table.

For details about <DCRAn\_base>, see **Section 32.1.2, Register Base Address**.

Table 32.6 List of Registers

Module Name	Register Name	Symbol	Address
DCRAn	CRC input register	DCRAnCIN	<DCRAn_base> + 00 <sub>H</sub>
DCRAn	CRC data register	DCRAnCOUT	<DCRAn_base> + 04 <sub>H</sub>
DCRAn	CRC control register	DCRAnCTL	<DCRAn_base> + 20 <sub>H</sub>

### 32.3.2 DCRAAnCIN — CRC Input Register

This register holds the input data for CRC calculation. The effective bit width used for CRC calculation must be set by DCRAAnCTL.DCRAAnISZ[1:0].

When data is written to this register, the CRC code is generated.

The CRC calculation is immediately started after the DCRAAnCIN register is written. The DCRAAnCOUT register must be initialized with the initial starting value, before the first data of the data block is written to DCRAAnCIN register.

**Access:** This register can be read or written in 32-bit units.

**Address:** <DCRAAn\_base>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCRAAnCIN[31:16]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DCRAAnCIN[15:0]																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 32.7 DCRAAnCIN Register Contents**

Bit Position	Bit Name	Function
31 to 0	DCRAAnCIN [31:0]	Input Data for CRC Calculation The valid bits are: <ul style="list-style-type: none"> <li>• For 32 bit effective bit width: DCRAAnCIN[31:0]</li> <li>• For 16 bit effective bit width: DCRAAnCIN[15:0]</li> <li>• For 8 bit effective bit width: DCRAAnCIN[7:0]</li> </ul>

### 32.3.3 DCRA<sub>n</sub>COUT — CRC Data Register

This register stores the result of the CRC code generated by the 32-bit Ethernet polynomial or the 16-bit CCITT polynomial.

**Access:** This register can be read or written in 32-bit units.

**Address:** <DCRA<sub>n</sub>\_base> + 4<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCRA <sub>n</sub> COUT[31:16]																
Value after reset <sup>*1</sup>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCRA <sub>n</sub> COUT[15:0]																
Value after reset <sup>*1</sup>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The read value after reset is 0000 0000<sub>H</sub> since the 32-bit Ethernet CRC polynomial is selected as the CRC generating function after reset.

**Table 32.8 DCRA<sub>n</sub>COUT Register Contents**

Bit Position	Bit Name	Function
31 to 0	DCRA <sub>n</sub> COUT [31:0]	<p>Result of the CRC Code Generation</p> <p>When the 16-bit CCITT polynomial is enabled, the bits 15 to 0 show the CRC result. The bits 31 to 16 are undefined.</p> <p>The read value of this register is a value obtained by performing EXOR calculation for the following value:</p> <ul style="list-style-type: none"> <li>For 32-bit Ethernet polynomial: FFFF FFFF<sub>H</sub></li> <li>For 16-bit CCITT polynomial: 0000<sub>H</sub></li> </ul> <p>For example, when DCRA<sub>n</sub>COUT = 5555 5555<sub>H</sub> for the 32-bit Ethernet polynomial, AAAA AAAA<sub>H</sub> is read.</p>

#### CAUTION

This register must initialized by setting the initial start value before the first data of the data block is written to DCRA<sub>n</sub>CIN register.

### 32.3.4 DCRAAnCTL — CRC Control Register

This register controls the CRC generation process.

**Access:** This register can be read or written in 8-bit units.

**Address:** <DCRAAn\_base> + 20H

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	DCRAAnISZ[1:0]	DCRAAnPOL	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 32.9 DCRAAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2, 1	DCRAAnISZ[1:0]	Specify the CRC input bit width. 00: 32 bits (DCRAAnCIN[31:0]) 01: 16 bits (DCRAAnCIN[15:0]) 10: 8 bits (DCRAAnCIN[7:0]) 11: Setting prohibited
0	DCRAAnPOL	Specifies the CRC generation method. 0: 32-bit Ethernet CRC polynomial generation. The byte order of the DCRAAnCIN register is LSB (least significant bit) first. This means that, if the input bit width is 8 bits (DCRAAnISZ[1:0] = 10B), bit positions 7 to 0 of the DCRAAnCIN register contain the input data and bit position 0 (LSB) is the start bit of the input data. 1: 16-bit CCITT CRC polynomial generation. The byte order of the DCRAAnCIN register is MSB (most significant bit) first. This means that, if the input bit width is 8 bits (DCRAAnISZ[1:0] = 10B), bit positions 7 to 0 of the DCRAAnCIN register contain the input data and bit position 7 (MSB) is the start bit of the input data.

#### CAUTION

- If the CRC generation method (DCRAAnCTL.DCRAAnPOL) is changed, the DCRAAnCOUT register must be initialized by setting the initial start value.
- The CRC input bit width (DCRAAnCTL.DCRAAnISZ[1:0]) must be set according to the data block bit width. Changing the CRC input bit width is not allowed during processing of a data block (a data block consists of N bytes, half-words or one word). After the final CRC result is read from DCRAAnCOUT register, the bit width can be changed. In that case, the DCRAAnCOUT register must be initialized with the initial start value.

## 32.4 Operation

The data CRC function A generates a CRC (cyclic redundancy check) of an arbitrary data block length. The data is forwarded to the data CRC function in 8-, 16- or 32-bit units. The CRC polynomial can either be selected for 32-bit Ethernet or 16-bit CCITT. The initial starting value must be set at the DCRA<sub>n</sub>COUT register before the first write access to the CRC input register (DCRA<sub>n</sub>CIN) is performed.

The flowchart below shows the CRC generating procedure.

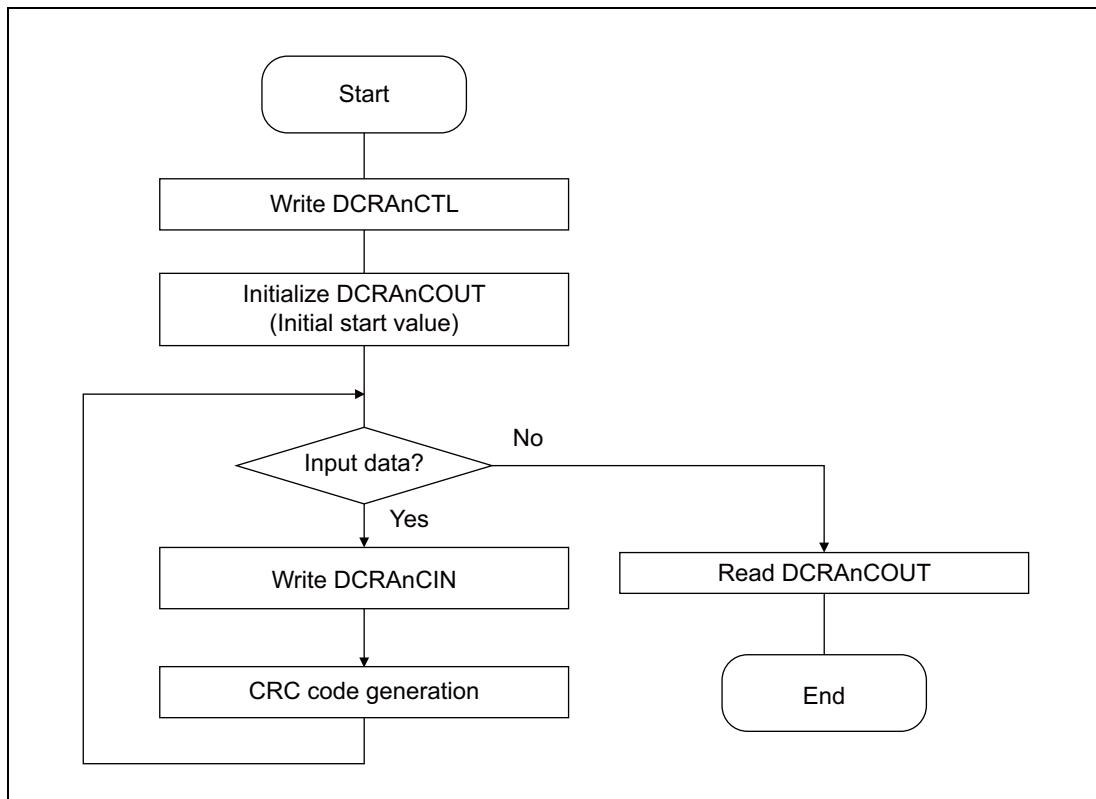


Figure 32.2 Flowchart of Data CRC Function A

### NOTES

1. Before writing the first data to DCRA<sub>n</sub>CIN, the CRC output register DCRA<sub>n</sub>COUT must be initialized with the initial start value.
2. DCRA<sub>n</sub>COUT must be re-initialized by setting the initial start value when the polynomial is changed by changing DCRA<sub>n</sub>CTL.DCRA<sub>n</sub>POL.
3. Setting example of the initial start values of the respective polynomials  
The following is the example of setting values.

Table 32.10 Setting Example of Initial Start Values (When Read at a Reset)

	Initial Start Value	EXOR Value	DCRA <sub>n</sub> COUT Read Value
16-bit CCITT	XXXX FFFF <sub>H</sub>	XXXX 0000 <sub>H</sub>	XXXX FFFF <sub>H</sub>
32-bit Ethernet	FFFF FFFF <sub>H</sub>	FFFF FFFF <sub>H</sub>	0000 0000 <sub>H</sub>

Note: X: undefined

## Section 33 Intelligent Cryptographic Unit (ICUSB)

Details of the intelligent cryptographic unit slave B (ICUSB) are provided in a separate document.

In addition, since ICUSB is mounted on a limited number of products, please contact our sales office for more information on ICUSB mounted products.

## Section 34 On-Chip Debug Unit (OCD)

This microcontroller has an on-chip debug function. By using the on-chip debug emulator, programs can be debugged with the microcontroller mounted in the target system.

The debug functions incorporated in this microcontroller conform to IEEE-ISTO 5001<sup>TM</sup>-2003 Class 3\*<sup>1</sup>, a Nexus debug interface standard.

**Note 1.** This function is supported only by products with an  $\overline{\text{EVTO}}$  pin.

### CAUTION

**The debug functions described in this section are supported by the microcontroller but whether they are usable depends on the debugger. For details on debugging, see the user's manual of the debugger.**

## 34.1 Overview of RH850/F1L OCD

### 34.1.1 Functional Overview

The on-chip debug functions described below are supported by the microcontroller.

#### (1) Debug interface

This microcontroller supports the following as debug interfaces: Nexus Interface, Low Pin Debug Interface (1-pin) - hereinafter called “LPD (1-pin)”, and Low Pin Debug Interface (4-pin) - hereinafter called “LPD (4-pin)”.

On-chip debug can be performed using these debug interfaces.

#### (2) Debug monitoring function

Debug-dedicated monitor program space is mounted and is used during debugging.

The basic debug functions below can be used by running a monitoring program.

- Downloading the user-created program
- Reading and writing the memory and registers
- Running the user-created program starting at any address

#### (3) On-chip break

A maximum of 12 breakpoints can be specified at any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified for any access (access address, access data).

#### (4) Software break

Software break points can be specified at any execution address.

#### (5) Peripheral break

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

**(6) Forced break**

Execution of the user-created program can be interrupted forcibly.

**(7) Forced reset**

This device (microcontroller) can be forcibly reset.

**(8) Real time RAM monitoring (RRM)**

The memory can be read during program execution. Because this read access uses debug-dedicated DMA, it has minimal effect on program execution.

**(9) Dynamic memory modification (DMM)**

The memory can be written during program execution. Because this write access uses debug-dedicated DMA, it has minimal effect on program execution.

**(10) Timer function**

Using a 32-bit counter, the time for running the user-created program can be measured based on the clock for debug.

For the measurement accuracy, see the user's manual of the debugger.

**(11) Mask function**

Masking the following factors is possible.

- All reset sources except for a POC reset and a wakeup reset
- MEMC0WAIT

**(12) Hot plug-in function**

Debugging can be started in normal operating mode without external reset input.

**NOTE**

When the hot plug-in function is used in power save mode, the INTDCUTDI interrupt is required to return from power save mode as the wake-up process.

**(13) Security function**

To prevent the contents of the flash memory from being read by an unauthorized person, a 128-bit ID code can be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code written to the microcontroller, the flash memory cannot be accessed.

For details on how to set the ID code, see the user's manual of the debugger.

**(14) Trace function**

Execution history, data changes, etc. of the user-created program can be obtained.

**NOTE**

The trace function is only available in devices with 2-MB memory.

## 34.2 Peripheral Break Control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

During peripheral break, the peripheral modules operate as follows.

### a. Modules that stop unconditionally regardless of the EPC.SVSTOP setting

**Table 34.1 Modules that Stop Unconditionally Regardless of the EPC.SVSTOP Setting**

Module
Window watchdog timer (WDTA)

### b. Modules that continue to operate by the setting of emulation registers even when EPC.SVSTOP = 1

**Table 34.2 Modules that Continue to Operate by the Setting of Emulation Registers even when EPC.SVSTOP = 1**

Module	Emulation Register	n
OS timer (OSTM)	OSTMnEMU.OSTMnSVSDIS 0: Stops during break 1: Continues during break	0
Timer array unit D (TAUD)	TAUDnEMU.TAUDnSVSDIS 0: Stops during break 1: Continues during break	0
Timer Array Unit B (TAUB)	TAUBnEMU.TAUBnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Timer array unit J (TAUJ)	TAUJnEMU.TAUJnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Real-Time Clock (RTCA)	RTCAnEMU.RTCAnSVSDIS 0: Stops during break 1: Continues during break	0
Clocked serial interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Clocked serial interface H (CSIH)	CSIInEMU.CSIInSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Timer Motor Control Function (TAPA)	TAPAnEMU.TAPAnSVSDIS 0: Stops during break 1: Continues during break	0
Encoder Timer (ENCA)	ENCANEMU.ENCANSVSDIS 0: Stops during break 1: Continues during break	0
PWM Output/Diagnostic (PWM-Diag)	PWBAnEMU.PWBAnSVSDIS 0: Stops during break 1: Continues during break	0
	PWSAnEMU.PWSAnSVSDIS 0: Stops during break 1: Continues during break	0
A/D converter (ADCA)	ADCAnEMU.ADCAnSVSDIS 0: Stops during break 1: Continues during break	0, 1

**CAUTION**

For details on the registers, see the register description of the corresponding section.

**c. Modules that stop when EPC.SVSTOP = 1****Table 34.3 Modules that Stop when EPC.SVSTOP = 1**

Module
LIN/UART interface (RLIN3)
Low-Power Sampler (LPS)

## 34.3 Registers

### 34.3.1 EPC — Emulation Peripheral Control Register

This register stops operation of peripheral functions (timer, serial interface, and A/D converter) in debug mode (SVSTOP).

<b>Access:</b>	Accessing from the user program is prohibited.							
<b>Address:</b>	—							
<b>Value after reset:</b>	$00_{\text{H}}$							
Bit	7	6	5	4	3	2	1	0
	—	SVSTOP	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—

**Table 34.4 EPC Register Contents**

Bit Position	Bit Name	Function
7	Reserved	—
6	SVSTOP	Stops operation of peripheral functions (timer, serial interface, and A/D converter) during debugging. 0: Does not stop operation 1: Stops operation
5 to 0	Reserved	—

#### NOTE

EPC is set by the debugger. Setting by the user program is prohibited. For the setting of the debugger, see the user's manual of the debugger.

## 34.4 Cautions on Using On-Chip Debugging

### 34.4.1 Treatment of Devices Used for Debugging

Do not install a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during system debugging and thus the write/erase endurance of the flash memory cannot be guaranteed.

### 34.4.2 Reset Assertion When a Debugger is connected

If a program in which a reset is asserted at the start of program execution is executed when a debugger is being used, the microcontroller is reset before preparation for communications between the OCD emulator and microcontroller is complete. For this reason, communications may not proceed correctly.

The period of preparation for communications depends on the host PC environment of the OCD emulator and the operating frequency of the microcontroller. To ensure that the debugger operates properly when debugging a program in which a reset is asserted at the start of program execution, insert a wait between the start of program and reset assertion.

### 34.4.3 Transition to DeepSTOP Mode When a Debugger is connected

If a program in which a transition to DeepSTOP mode occurs at the start of program execution is executed when a debugger is being used, the microcontroller stops supplying power to the isolated area before preparation for communications between the OCD emulator and microcontroller is complete. For this reason, communications may not proceed correctly.

The period of preparation for communications depends on the host PC environment of the OCD emulator and the operating frequency of the microcontroller. To ensure that the debugger operates properly when debugging a program in which a reset is asserted at the start of program execution, insert a wait between reset deassertion and transition to DeepSTOP mode.

## Section 35 Flash Memory

This section describes the flash memory mounted on RH850/F1L.

The first part in this section describes the characteristics of the mounted flash memory and the characteristics specific to RH850/F1L, such as the memory map, flash memory programming, and ECC.

### 35.1 Features

- Includes code flash memory and data flash memory
  - The code flash memory can store program codes and data and has the user area and the extended user area.
  - The data flash memory is used for storing data.
- Method of flash memory programming
  - Flash memory programming via a serial interface and programming of flash memory by a user program (self-programming) are supported.
- Support for BGO (Back Ground Operation)
  - The BGO function allows programs to be executed in the code flash memory while the data flash memory is being programmed/erased.
- Flash memory data security
  - Support for security functions to protect against illicit tampering with or reading out of data in the flash memory
  - Support for protection functions to protect against erroneous overwriting of the flash memory
- Option byte function
  - Sets the operation after releasing reset for ports, WDTA, CVM, and external bus clocks.
- Support for the error detection/correction function (ECC) in the code flash memory and data flash memory
  - Built-in ECC function can detect 2-bit errors and detect/correct 1-bit errors.
- Interrupts can be acknowledged in self-programming mode.

For code flash sizes and data flash sizes of each product, see the following sections.

- **Section 3.2.4.1, Code Flash Area**
- **Section 3.2.4.2, Data Flash Area**

## 35.2 Structure of Memory

### 35.2.1 Mapping of Code Flash Memory

**Figure 35.1** illustrates the mapping of the code flash memory for the 2-MB device. The user area of the code flash memory of the RH850/F1L is divided into 8- and 32-Kbyte blocks, which serve as the units of erasure. A single block of 32-Kbyte extended user area is also incorporated. The user area and extended user area are available as areas for storing the user program.

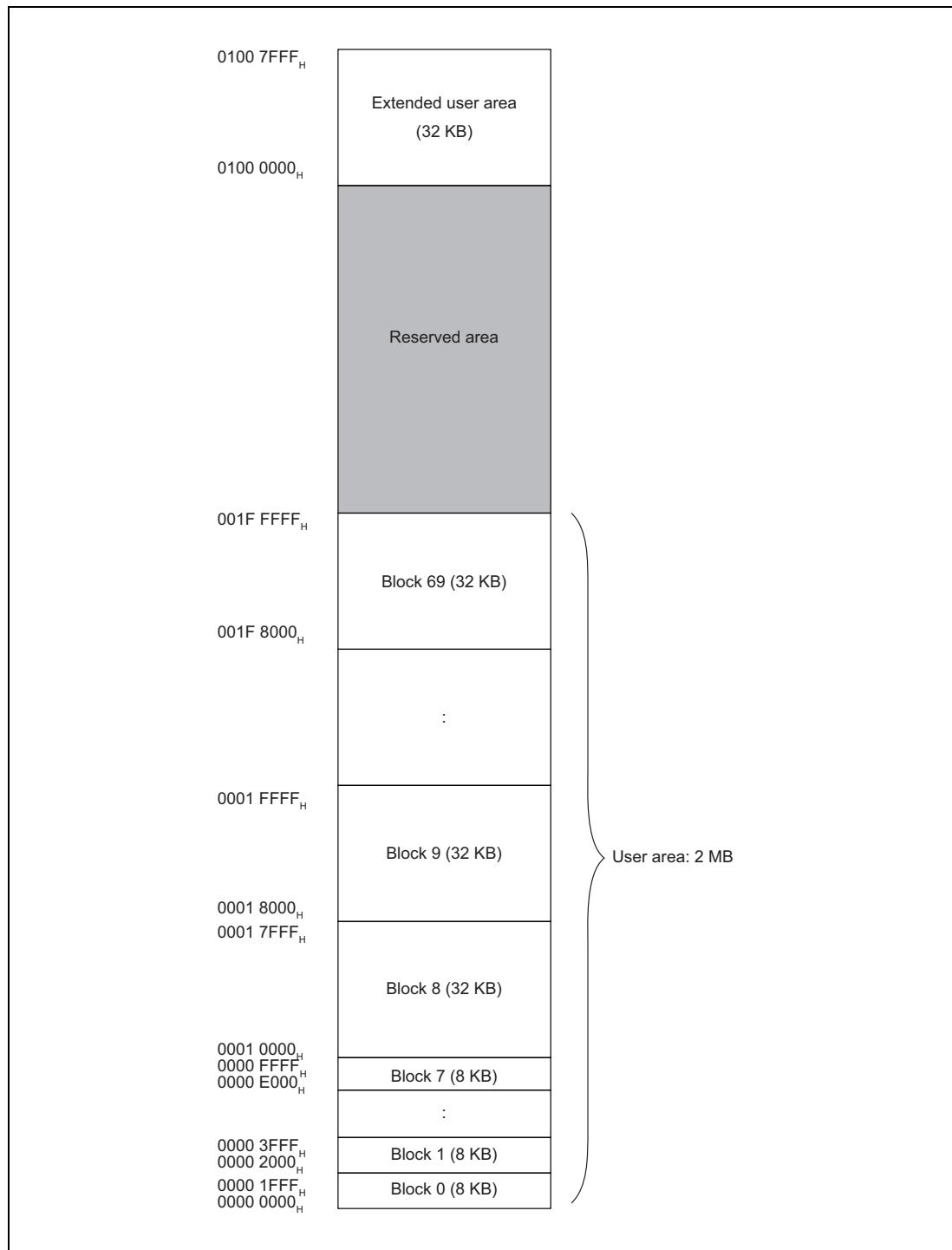


Figure 35.1 Mapping of the F1L-176 Pins/2 MB Code Flash Memory

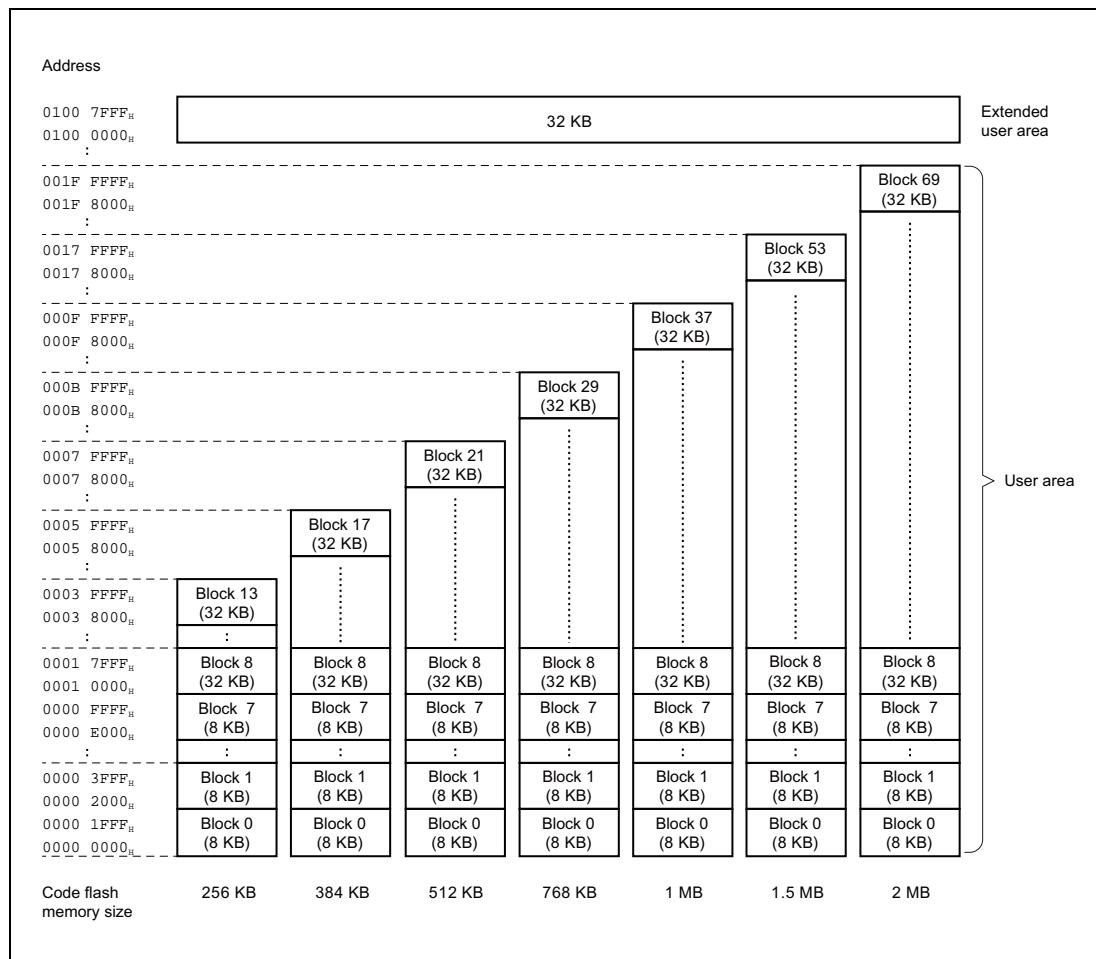


Figure 35.2 Mapping of Code Flash Memory

### 35.2.2 Mapping of Data Flash Memory

The data area of the data flash memory in the RH850/F1L is divided into 64-byte blocks, with each being a unit for erasure. **Figure 35.3** shows the mapping of the data flash memory.

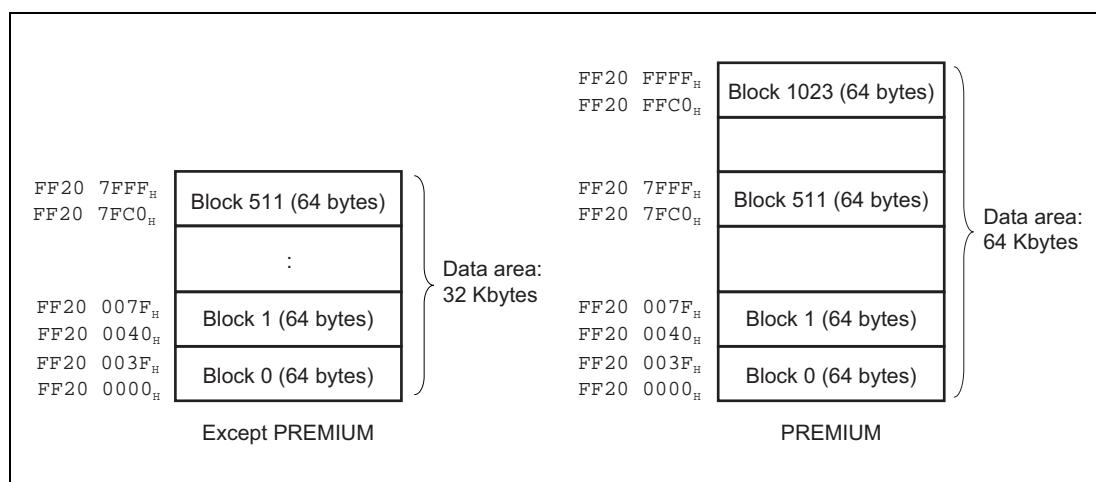
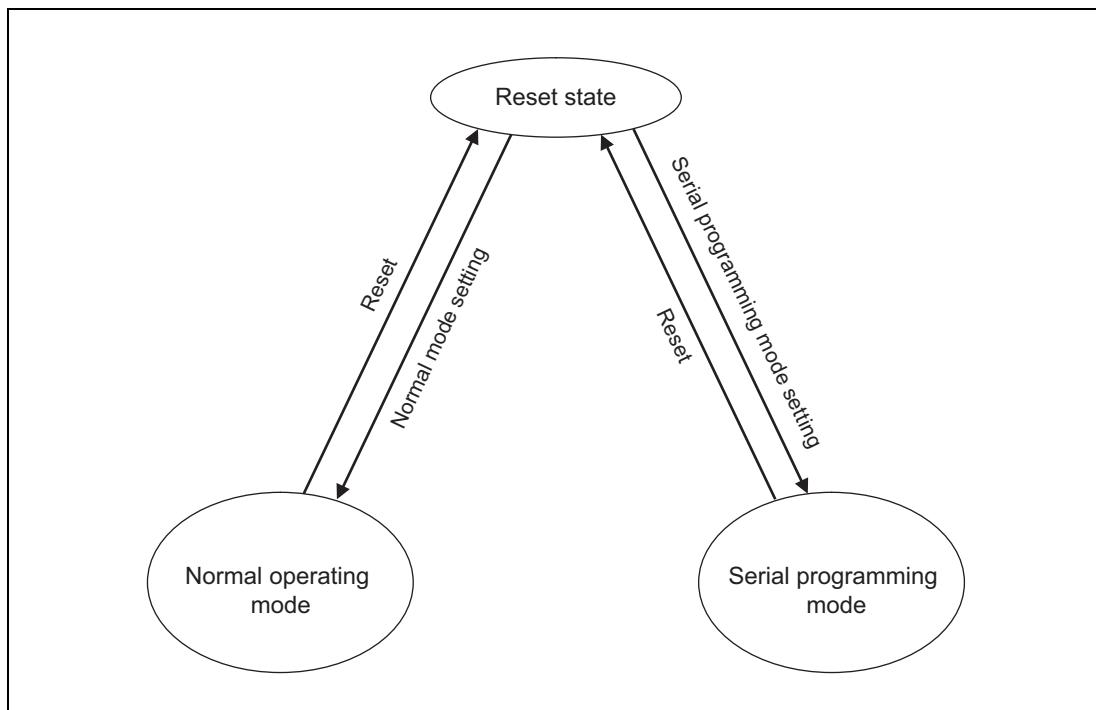


Figure 35.3 Mapping of the Data Flash Memory

### 35.3 Operating Modes Associated with Flash Memory

**Figure 35.4** is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see **Section 5, Operating Mode**.



**Figure 35.4 Mode Transition Associated with Flash Memory**

**Table 35.1** shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

**Table 35.1 Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release**

Item	Normal Operating Mode	Serial Programming Mode
Programmable and erasable area	<ul style="list-style-type: none"> <li>• User area</li> <li>• Extended user area</li> <li>• Data area</li> </ul>	<ul style="list-style-type: none"> <li>• User area</li> <li>• Extended user area</li> <li>• Data area</li> </ul>
Boot program after reset release	Program in user area or extended user area (Changeable by using the variable reset vector)	Firmware program for serial programming

## 35.4 Functions

### 35.4.1 Functional Overview

The flash memory of the RH850/F1L can be updated via a serial interface by a dedicated flash memory programmer (serial programming), before being mounted on the target system or on a flash adapter system.

Furthermore, security functions to prohibit updating of the user program written in the flash memory are incorporated, and this can prevent tampering by third parties.

Programming by the user program (self-programming) is suited for applications where the target system program may require updating after deployed to the end user. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as external communications, etc., and this allows programming under various conditions. **Table 35.2** gives an overview of the methods of programming and the corresponding operating modes.

**Table 35.2 Methods of Programming**

Method of Programming	Overview of Functionality	Operating Mode
Serial programming	A dedicated flash memory programmer allows on-board programming of the flash memory after the device is mounted on the target system.  A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	Serial programming mode
Self-programming	The user program that is written to code flash memory in advance by serial programming also allows updating the flash memory.  The background operation capability makes it possible to fetch instructions or otherwise read data in code flash memory while the data flash memory is self-programming.  For this reason, it is possible to update the data flash memory by executing a program written to the code flash memory.  Instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being updated by self-programming. In such cases, a program for updating must be transferred to the local RAM or external memory in advance and executed.	Normal operating mode

Renesas provides a library for self-programming. For details on this library, see the user's manuals for the code flash library and data flash library of this device.

**Table 35.3** lists the functions of the flash memory. Dedicated flash memory programmer commands enable serial programming, while reading of the flash memory by a library function or the user program enables self-programming.

**Table 35.3 Basic Functions at a Glance**

Function	Overview	Level of Support (√: Supported, Δ: Conditionally Supported, —: Not Supported)	
		Serial programming	Self-programming
Blank checking	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from code flash memory and data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	√	Δ (Only data flash is supported)
Block erasure	This is for erasing the contents of a specified block of memory.	√	√
Programming	This is for writing to a specified address.	√	√
Verification and checksum	Data that are read out from flash memory are compared with data transferred from the flash memory programmer.	√	— (Reading of data by the user program is possible)
Reading	Data that have been written to the flash memory are read out.	√	√
Setting for OTP (one-time programming)	A specified block of code flash memory is set for OTP (OTP can only be set, that is, it is not possible to release a block's OTP setting).	√	√
Setting an ID	An ID setting is made for use in controlling the connection of a dedicated flash memory programmer for serial programming, controlling of the on-chip debugger, and programming of the code flash memory by self-programming.	√	√
Security settings	Security settings are for use in serial programming.	√	Δ (Only when setting is prohibited after being permitted)
Protection settings	Settings for block protection of code flash memory and variable reset vector are provided.	Δ (Setting of the reset vector values for variable reset vector function is not supported.)	√
Setting of option bytes	Option bytes are set to change them from the initial values for the RH850/F1L.	√	√
Clearing the configuration	ID setting, security settings, protection settings, and option byte settings are initialized.	√	—

For details on serial programming, see the user's manual of the flash programmer.

For details on self-programming, see the user's manuals for the code flash library and data flash library of this device.

The flash memory supports various security functions.

The OTP setting and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, prohibiting connection of a dedicated flash memory programmer, and prohibition of commands (for block erasure, programming, and reading) are available for use as security functions.

The security functions supported by the flash memory are listed in **Table 35.4** and **Table 35.5**.

**Table 35.4 Summary of Security Functions**

<b>Function</b>	<b>Description</b>
OTP	OTP can be individually set for each block of the user area and the extended user area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self-programming is prohibited. Once set, the OTP setting cannot be released. Furthermore, since execution of the configuration clearing command is prohibited for any area for which OTP has been set, changing a security setting from "prohibited" to "permitted" is not possible.
ID authentication	The result of ID authentication can be used to control the connection of a dedicated flash memory programmer for serial programming. The result of ID authentication can also be used to control enabling of self-programming.
Prohibition of connection of a dedicated flash memory programmer	The connection of a dedicated flash memory programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a dedicated flash memory programmer is prohibited, changing a security setting from "prohibited" to "permitted" is not possible.
Prohibition of block erasure commands	Block erasure commands at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erasure commands are prohibited, changing a security setting from "prohibited" to "permitted" is not possible.
Prohibition of programming commands	Block erasure commands and programming commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.
Prohibition of read commands	Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.

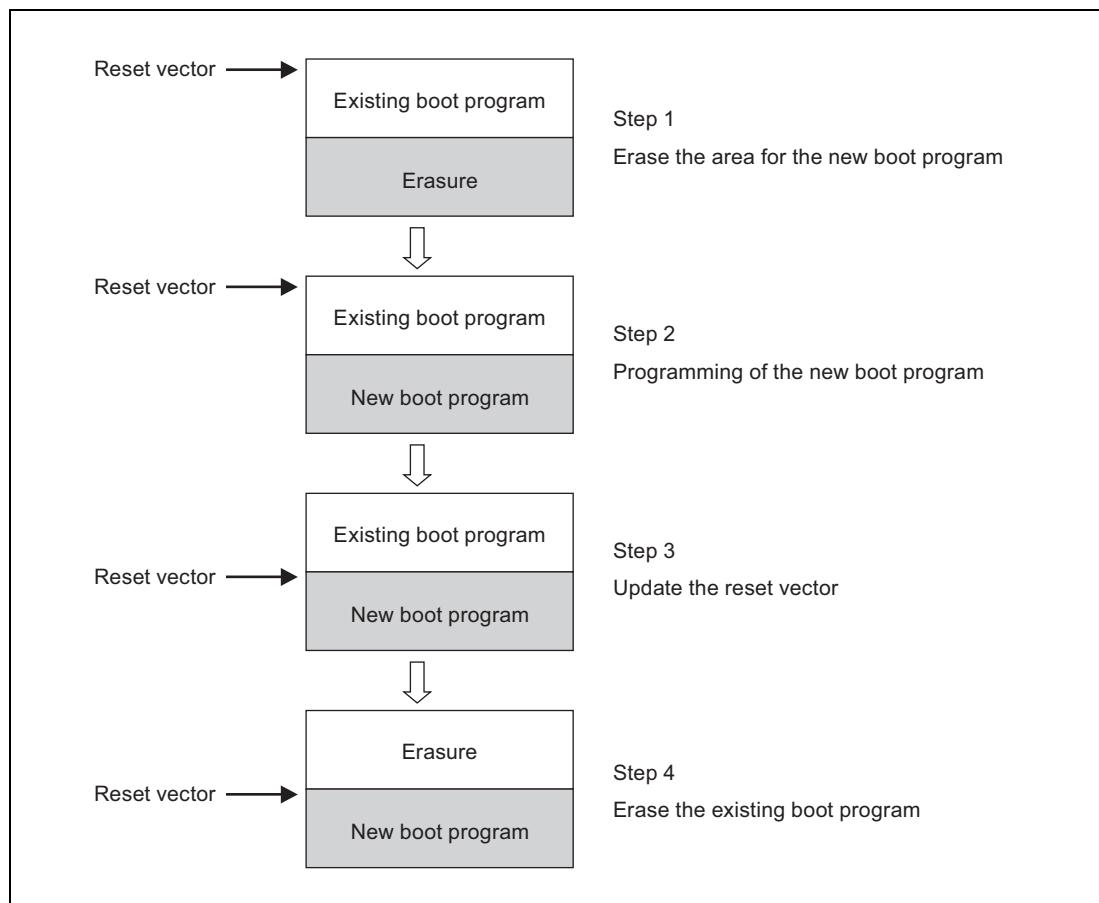
**Table 35.5 Available Operations and Security Settings**

Function	All Security Settings and Erasure, Programming, and Read Operations (√: Executable, —: Not Executable)		Point for Caution Regarding the Security Setting	
	Serial programming	Self-programming	Serial programming	Self-programming
OTP	<ul style="list-style-type: none"> <li>Areas for which OTP is set           <ul style="list-style-type: none"> <li>Block erasure commands: —</li> <li>Programming commands: —</li> <li>Read commands: √</li> </ul> </li> <li>Areas for which OTP is not set           <ul style="list-style-type: none"> <li>Block erasure commands: √</li> <li>Programming commands: √</li> <li>Read commands: √</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Areas for which OTP is set           <ul style="list-style-type: none"> <li>Block erasure: —</li> <li>Programming: —</li> <li>Reading: √</li> </ul> </li> <li>Areas for which OTP is not set           <ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>The OTP setting cannot be released.</li> <li>Execution of the configuration clearing command is not possible.</li> </ul>	The OTP setting cannot be released.
ID authentication	<ul style="list-style-type: none"> <li>When the ID codes do not match           <ul style="list-style-type: none"> <li>Block erasure commands: —</li> <li>Programming commands: —</li> <li>Read commands: —</li> </ul> </li> <li>When the ID codes match           <ul style="list-style-type: none"> <li>Block erasure commands: √</li> <li>Programming commands: √</li> <li>Read commands: √</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>When the ID codes do not match           <ul style="list-style-type: none"> <li>Code flash memory               <ul style="list-style-type: none"> <li>Block erasure: —</li> <li>Programming: —</li> <li>Reading: √</li> </ul> </li> <li>Data flash memory               <ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul> </li> </ul> </li> <li>When the ID codes match           <ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>The configuration clearing command can initialize the setting for prohibition.</li> <li>The setting for prohibition of block erasure commands is not available.</li> <li>The setting for prohibition of programming commands is not available.</li> <li>The setting for prohibition of read commands is not available.</li> </ul>	ID authentication is always in effect.
Prohibition of the connection of a dedicated flash memory programmer	<ul style="list-style-type: none"> <li>Block erasure commands: —</li> <li>Programming commands: —</li> <li>Read commands: —</li> </ul>	<ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul>	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of block erasure commands	<ul style="list-style-type: none"> <li>Block erasure commands: —</li> <li>Programming commands: √</li> <li>Read commands: √</li> </ul>	<ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul>	<ul style="list-style-type: none"> <li>Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible.</li> <li>The setting for ID authentication to be effective for serial programming is not available.</li> </ul>	<ul style="list-style-type: none"> <li>The configuration clearing command can initialize the setting for prohibition.</li> <li>The setting for ID authentication to be effective for serial programming is not available.</li> </ul>
Prohibition of programming commands	<ul style="list-style-type: none"> <li>Block erasure commands: —</li> <li>Programming commands: —</li> <li>Read commands: √</li> </ul>	<ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul>		
Prohibition of read commands	<ul style="list-style-type: none"> <li>Block erasure commands: √</li> <li>Programming commands: √</li> <li>Read commands: —</li> </ul>	<ul style="list-style-type: none"> <li>Block erasure: √</li> <li>Programming: √</li> <li>Reading: √</li> </ul>		

The flash memory supports various protection functions. The protection functions supported by the flash memory are listed in **Table 35.6**.

**Table 35.6 Summary of Protection Functions**

Function	Description
Block protection	Lock bit settings can be individually made to enable or disable programming and erasure of each block of the user area and the extended user area of code flash memory. Programming and erasure by self-programming of an area for which the lock bit is set and the lock bit function is enabled are prohibited. Programming or erasure can proceed again when the lock bit function is disabled after having been enabled. When a block of code flash memory is erased, the lock bit for that block is also erased.
Hardware protection	The level on the FLMD0 pin can be set to prohibit programming and erasure of the code flash memory. - FLMD0 = 0: Programming prohibited - FLMD0 = 1: Programming permitted
Variable reset vector	The protection settings include control of the reset vector. As shown in <b>Figure 35.5</b> , after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program. The areas that can be specified by using the reset vector are the user area and extended user area.



**Figure 35.5 Utilizing the Variable Reset Vector Function to Update the Boot Program**

#### NOTE

After step 4, a reset leads to updating of the reset vectors for which RBASE is changed.

## 35.5 Serial Programming

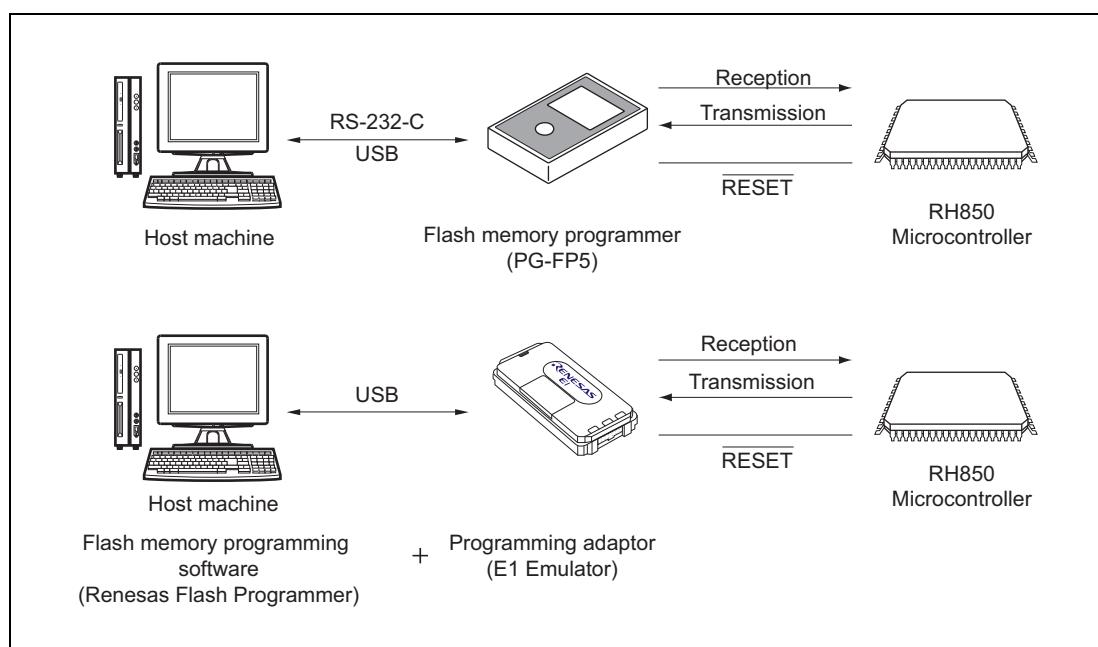
A dedicated flash memory programmer can be used to handle flash memory in serial programming mode.

### Serial programming

The microcontroller is mounted on the system board at the time of serial programming. Providing a connector to the board enables handling of the microcontroller by the flash memory programmer to proceed.

#### 35.5.1 Environments for Programming

The recommended environments for handling the flash memory of the microcontroller with data are described below.



**Figure 35.6 Environments for Handling Programs of the Flash Memory**

By using the PG-FP5 flash memory programmer or the combination of the Renesas Flash Programmer (software for writing to flash memory) running on the host machine and the E1 emulator as an programming adaptor, the user is easily able to erase, program, and verify the contents of the on-chip memory of flash-memory-equipped microcontrollers from Renesas Electronics.

The PG-FP5 flash memory programmer handles programming from a host machine or programming in stand-alone mode while the Renesas Flash Programmer only handles programming from a host machine.

#### NOTE

For details on the PG-FP5, see the *PG-FP5 Flash Memory Programmer User's Manual*. For details on the Renesas Flash Programmer of flash programming software, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

## 35.6 Communication Modes

### 35.6.1 Asynchronous Flash Programming Interface - 1-Wire UART

The single-wire asynchronous serial programming interface, 1-wire UART is connected to the flash memory programmer with the following port.

- FPDR (JP0\_0): Receive data input/transmit data output

### 35.6.2 Asynchronous Flash Programming Interface - 2-Wire UART

The double-wire asynchronous serial programming interface, 2-wire UART is connected to the flash memory programmer with the following ports.

- FPDR (JP0\_0): Receive data input
- FPDT (JP0\_1): Transmit data output

### 35.6.3 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

- FPDR (JP0\_0): Receive data input
- FPDT (JP0\_1): Transmit data output
- FPCK (JP0\_2): Serial clock input

The flash memory programmer outputs the serial data clock SCK, and the microcontroller operates as a slave.

---

#### NOTE

---

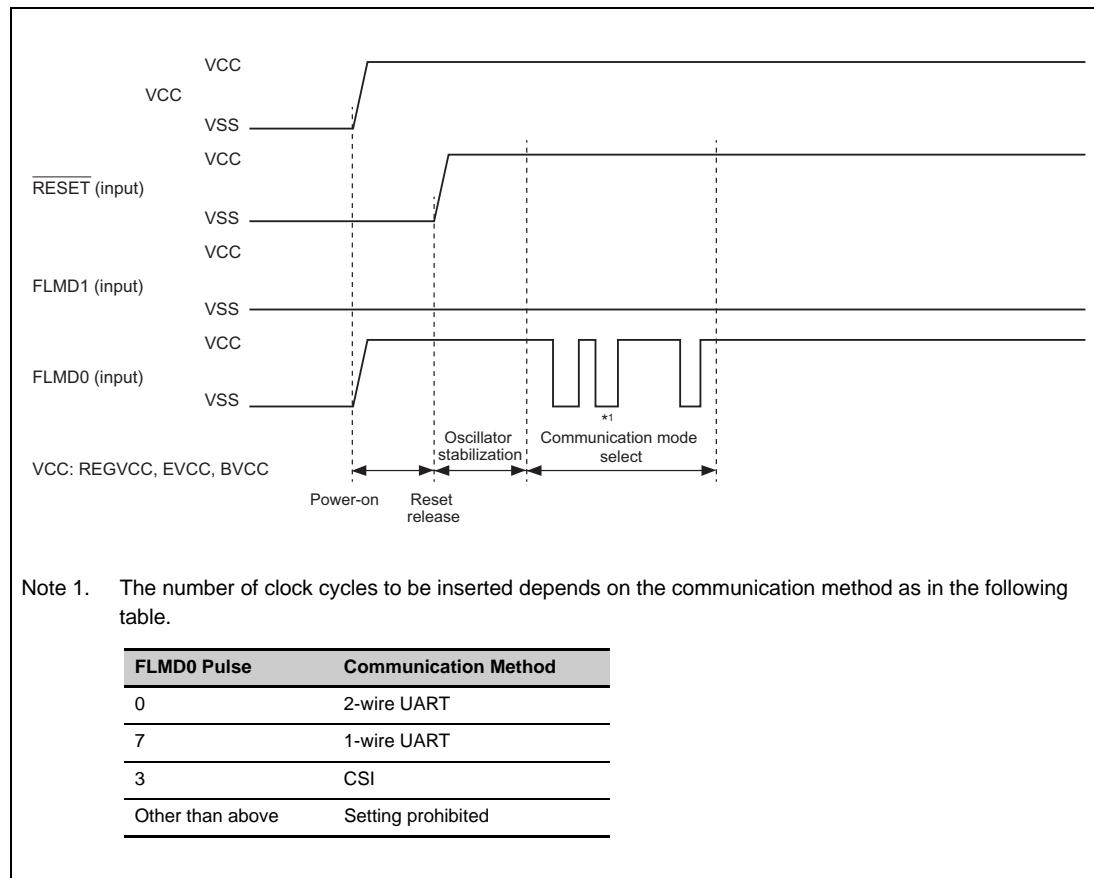
For details on Renesas Flash Programmer, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

---

### 35.6.4 Selection of Communication Method

In RH850/F1L, communication method can be selected by pulse input to the FLMD0 pin (up to 7 pulses) after transition to the flash memory programming mode. The FLMD0 pulse is generated by a dedicated flash memory programmer.

**Figure 35.7** shows the relation between the number of pulses and communication method.



**Figure 35.7 Selection of Communication Method**

## 35.7 Self-Programming

### 35.7.1 Outline

The RH850/F1L supports programming of the flash memory by the user program itself. Renesas Electronics provides a code flash library and a data flash library for use with user programs. These libraries can be used for writing to the code flash memory and to the data flash memory.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to local RAM or external memory in advance of the programming operation, and executed from the given destination to perform the programming.

The programming program can be copied to the local RAM or external memory in advance and executed to program the code flash memory.

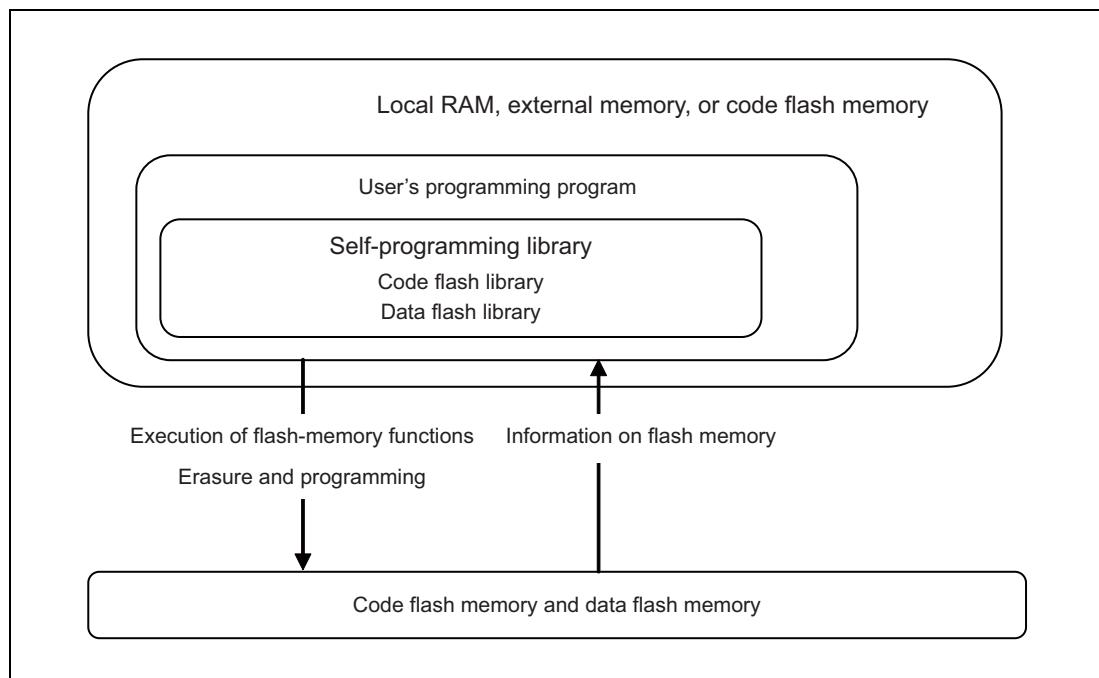


Figure 35.8 Schematic View of Self-Programming

For details on the self-programming of flash memory, see the user's manuals for the code flash and data flash libraries for this device.

### 35.7.2 Background Operation

Background operations can be used when the combination of the flash memory for writing and the flash memory for reading is any of those listed below.

**Table 35.7 Conditions under which Background Operation is Usable**

Range for Writing	Range for Reading
Data flash memory	Code flash memory

### 35.7.3 Enabling Self-Programming

The self-programming function can be activated in normal operating mode.

Erasure and programming of the code flash memory by the self-programming function is enabled by making the FLMD0 pin high level.

This prevents unnecessary overwriting of the program if the device operates incorrectly.

The FLMD0 pin is made high level by using one of the following methods.

- The FLMD0 pin is externally pulled up.
- The FLMD0 pin is pulled up by the FLMDCNT register.

The outline of the FLMDCNT register is described in **Section 35.7.3.1, FLMDCNT Register**.

#### 35.7.3.1 FLMDCNT Register

This register specifies the internal pull-up or pull-down of the FLMD0 pin.

The correct write sequence using the FLMDPCMD register is required in order to update this register. For details, refer to **Section 4, Write-Protected Registers**.

**Access:** This register can be read or written in 32-bit units.

**Address:** FFA0 0000<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

**Table 35.8 FLMDCNT Register Contents**

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FLMDPUP	FLMD0 Pin Software Control 0: Pull-down selected 1: Pull-up selected

## 35.8 Reading Flash Memory

### 35.8.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in normal mode. Data can simply be read out through access to addresses in the code flash memory.

Reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception.

### 35.8.2 Reading Data Flash Memory

Configure the number of read cycles in the EEP'RDCYCL register prior to reading data from data flash memory in normal mode. Once this register is properly configured, data can be read by simply accessing addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programmed again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

### 35.8.2.1 EEPRDCYCL — Data Flash Wait Cycle Control Register

This register is used to specify the number of wait cycles to be inserted when reading the data in the data flash.

Set the number of wait cycles to be inserted in the clock cycle when reading the data flash according to the operating clock frequency of the CPU (CPUCLK).

**Access:** This register can be read or written in 8-bit units.

**Address:** FFC5 A010<sub>H</sub>

**Value after reset:** 0F<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—				WAIT[3:0]
Value after reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

**Table 35.9 EEPRDCYCL Register Contents**

Bit Position	Bit Name	Function	CPU Operating Frequency				
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.	FCPUCLK ≤ 20 MHz	20 MHz < FCPUCLK ≤ 40 MHz	40 MHz < FCPUCLK ≤ 60 MHz	60 MHz < FCPUCLK ≤ 80 MHz	80 MHz < FCPUCLK ≤ 96 MHz
3 to 0	WAIT[3:0]	Number of Wait Cycles					
	WAIT [3:0]	Number of Wait Cycles	FCPUCLK ≤ 20 MHz	20 MHz < FCPUCLK ≤ 40 MHz	40 MHz < FCPUCLK ≤ 60 MHz	60 MHz < FCPUCLK ≤ 80 MHz	80 MHz < FCPUCLK ≤ 96 MHz
	0000	1	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	0001	2	√	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	0010	3	√	√	Setting prohibited	Setting prohibited	Setting prohibited
	0011	4	√	√	√	Setting prohibited	Setting prohibited
	0100	5	√	√	√	√	√
	0101	6	√	√	√	√	√
	0110	7	√	√	√	√	√
	0111	8	√	√	√	√	√
	1000	9	√	√	√	√	√
	Other than above	10	√	√	√	√	√

#### NOTES

- The read access time to the data flash is calculated by the number of wait cycles.

Read access time to the data flash = {12 + (Number of wait cycles) × 2} / CPU operating frequency

However, the time may be changed depending on the combination of instructions before and after the execution.

- √ indicates the number of wait cycles that can be set.

### 35.8.2.2 PRDNAME<sub>n</sub> — Product Name Storage Register ( $n = 1$ to $3$ )

This register stores the product name. The product part name is stored in 16-byte ASCII code, and PRDNAME1, PRDNAME2, and PRDNAME3 correspond to the fourth to first bytes, eighth to fifth bytes, and twelfth to ninth bytes of the product part name respectively.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** PRDNAME1: FFCD 00D0<sub>H</sub>  
PRDNAME2: FFCD 00D4<sub>H</sub>  
PRDNAME3: FFCD 00D8<sub>H</sub>

**Value after reset:** See Table 35.11 to Table 35.14.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRDNAME <sub>n</sub> [31:24] <sup>*1</sup>										PRDNAME <sub>n</sub> [23:16] <sup>*1</sup>					
	Value after reset										Value after reset					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRDNAME <sub>n</sub> [15:8] <sup>*1</sup>										PRDNAME <sub>n</sub> [7:0] <sup>*1</sup>					
	Value after reset										Value after reset					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1.  $n = 1$  to  $3$ .

Table 35.10 PRDNAME<sub>n</sub> Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Product name fourth byte (PRDNAME1), eighth byte (PRDNAME2) twelfth byte (PRDNAME3)
23 to 16	—	Product name third byte (PRDNAME1), seventh byte (PRDNAME2) eleventh byte (PRDNAME3)
15 to 8	—	Product name second byte (PRDNAME1), sixth byte (PRDNAME2) tenth byte (PRDNAME3)
7 to 0	—	Product name first byte (PRDNAME1), fifth byte (PRDNAME2) ninth byte (PRDNAME3)

Table 35.11 to Table 35.14 list registers related to product information.

Table 35.11 List of Registers Related to Product Information (ECO)

Product Part Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701006	3746 3752	3030 3130	2020 2036
R7F701007	3746 3752	3030 3130	2020 2037
R7F701008	3746 3752	3030 3130	2020 2038
R7F701009	3746 3752	3030 3130	2020 2039
R7F701010	3746 3752	3130 3130	2020 2030
R7F701011	3746 3752	3130 3130	2020 2031
R7F701012	3746 3752	3130 3130	2020 2032
R7F701013	3746 3752	3130 3130	2020 2033
R7F701014	3746 3752	3130 3130	2020 2034
R7F701015	3746 3752	3130 3130	2020 2035
R7F701016	3746 3752	3130 3130	2020 2036

**Table 35.11 List of Registers Related to Product Information (ECO)**

<b>Product Part Name</b>	<b>PRDNAME1</b>	<b>PRDNAME2</b>	<b>PRDNAME3</b>
R7F701017	3746 3752	3130 3130	2020 2037
R7F701018	3746 3752	3130 3130	2020 2038
R7F701019	3746 3752	3130 3130	2020 2039
R7F701020	3746 3752	3230 3130	2020 2030
R7F701021	3746 3752	3230 3130	2020 2031
R7F701022	3746 3752	3230 3130	2020 2032
R7F701023	3746 3752	3230 3130	2020 2033
R7F701024	3746 3752	3230 3130	2020 2034
R7F701025	3746 3752	3230 3130	2020 2035
R7F701028	3746 3752	3230 3130	2020 2038
R7F701029	3746 3752	3230 3130	2020 2039
R7F701030	3746 3752	3330 3130	2020 2030
R7F701032	3746 3752	3330 3130	2020 2032
R7F701033	3746 3752	3330 3130	2020 2033
R7F701034	3746 3752	3330 3130	2020 2034

**Table 35.12 List of Registers Related to Product Information (ADVANCED)**

<b>Product Part Name</b>	<b>PRDNAME1</b>	<b>PRDNAME2</b>	<b>PRDNAME3</b>
R7F701040	3746 3752	3430 3130	2020 2030
R7F701041	3746 3752	3430 3130	2020 2031
R7F701042	3746 3752	3430 3130	2020 2032
R7F701043	3746 3752	3430 3130	2020 2033
R7F701044	3746 3752	3430 3130	2020 2034
R7F701045	3746 3752	3430 3130	2020 2035
R7F701046	3746 3752	3430 3130	2020 2036
R7F701047	3746 3752	3430 3130	2020 2037
R7F701048	3746 3752	3430 3130	2020 2038
R7F701049	3746 3752	3430 3130	2020 2039
R7F701050	3746 3752	3530 3130	2020 2030
R7F701051	3746 3752	3530 3130	2020 2031
R7F701052	3746 3752	3530 3130	2020 2032
R7F701053	3746 3752	3530 3130	2020 2033

**Table 35.13 List of Registers Related to Product Information (PREMIUM)**

<b>Product Part Name</b>	<b>PRDNAME1</b>	<b>PRDNAME2</b>	<b>PRDNAME3</b>
R7F701054	3746 3752	3530 3130	2020 2034
R7F701055	3746 3752	3530 3130	2020 2035
R7F701056	3746 3752	3530 3130	2020 2036
R7F701057	3746 3752	3530 3130	2020 2037

**Table 35.14 List of Registers Related to Product Information (Gateway)**

<b>Product Part Name</b>	<b>PRDNAME1</b>	<b>PRDNAME2</b>	<b>PRDNAME3</b>
R7F701002	3746 3752	3030 3130	2020 2032
R7F701003	3746 3752	3030 3130	2020 2033

## 35.9 ECC Error Detection and Correction for Code Flash Memory

### 35.9.1 Outline of ECC Functions for Code Flash Memory

**Table 35.15** gives an outline of the ECC functions for the code flash memory.

**Table 35.15 Outline of the ECC Functions for the Code Flash Memory**

Item	Outline of Functions
ECC error detection and correction	<p>ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected.</p> <ul style="list-style-type: none"> <li>• 2-bit error detection and 1-bit error detection/correction</li> <li>• 2-bit error detection and 1-bit error detection</li> </ul> <p>When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.</p>
Error notification	<p>A notification is sent when an ECC error occurs. ECC error:</p> <ul style="list-style-type: none"> <li>• Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit errors is selectable. Selectable by setting the SEG_CONT.SEG_CONTROME (code flash error notification enable) bit (initial value: notification is disabled).</li> <li>• Enabling or disabling of error notification (INTECCSDFLI0 interrupt) in the case of detection of ECC 1-bit errors. This can be selected by setting the FEINTFMSK.ECCSDFLI0FEIFMSK (INTECCSDFLI0 interrupt mask) bit (initial value: interrupt is masked) and CFERRINT.SEDIE (1-bit ECC error notification control) bit (initial value: notification is disabled).</li> </ul> <p>In the initial state, error notification is disabled upon detection of a 2-bit ECC error, and error notification is disabled upon detection of a 1-bit ECC error.</p>
Error status	<p>The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The ECC 1-bit error status is set only when no error status has been set. The ECC 2-bit error status is set even when the ECC 1-bit error status is set. A register for clearing the error status is provided.</p>
Address capture	<p>When no error status has been set, the address at which the first error occurred is captured. In addition, when the retained address source is a 1-bit ECC error, the address of the 2-bit ECC error is also captured.</p>
Instruction execution suppression	<p>Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.</p>

### 35.9.2 Interrupt Request

The following table shows interrupt requests of code flash ECC.

**Table 35.16 Interrupt Requests of Code Flash ECC**

Unit Interrupt Signal	Outline	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of Code Flash	INTECCSDFLI0	—
—	ECC 2-bit error interrupt of Code Flash	SYSERR	—

### 35.9.3 Registers

#### 35.9.3.1 List of Registers

This table below lists the registers for the code flash ECC.

**Table 35.17 List of Code Flash ECC Registers**

Register Name	Symbol	Address
Code flash ECC control register	CFECCCTL	FFC6 2000 <sub>H</sub>
Code flash first error status register	CFFSTERSTR	FFC6 2004 <sub>H</sub>
Code flash error status clear register	CFFSTSTC	FFC6 2024 <sub>H</sub>
Code flash error overflow status register	CFOVFSTR	FFC6 2028 <sub>H</sub>
Code flash error overflow status clear register	CFOVFSTC	FFC6 202C <sub>H</sub>
Code flash error notification control register	CFERRINT	FFC6 2030 <sub>H</sub>
Code flash first error address register	CFFSTEADR	FFC6 2034 <sub>H</sub>
Code flash test control register	CFTSTCTL	FFC6 2054 <sub>H</sub>

### 35.9.3.2 CFECCCTL — Code Flash ECC Control Register

This register enables or disables the ECC error detection/correction and 1-bit error correction. Set the PROT1 and PROTO bits to 01<sub>B</sub> when writing to this register.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC6 2000<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROTO	—	—	—	—	—	—	—	—	—	—	—	—	SEDDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 35.18 CFECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the ECCDIS and SEDDIS bits. The written data is not retained.
14	PROTO	These bits are always read as 0. Set PROT1 and PROTO to 01 when writing to this register.
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SEDDIS	1-Bit Error Correction Disable This bit enables or disables 1-bit error correction when the ECC error detection/correction is enabled. When writing, write 01 to PROT1 and PROTO at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable This bit enables or disables ECC error detection/correction. When writing, write 01 to PROT1 and PROTO at the same time. ECC error detection/correction is enabled in the initial state. 0: ECC error detection/correction is enabled. 1: ECC error detection/correction is disabled.

### 35.9.3.3 CFFSTERSTR — Code Flash First Error Status Register

This register monitors occurrence of the first error.

Detecting an ECC 1-bit error sets the SEDF bit and detecting an ECC 2-bit error sets the DEDF bit while ECC error detection/correction is enabled.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFC6 2004<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.19 CFFSTERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	<p>ECC 2-Bit Error Monitor Flag</p> <p>This bit is set to 1 when an ECC 2-bit error occurs, regardless of the SEDF or DEDF state.</p> <p>0: An internal reset or a pin reset has been generated. The FSTERRCLR bit in code flash error status clear register has been set.</p> <p>1: An ECC 2-bit error has occurred.</p>
0	SEDF	<p>ECC 1-Bit Error Monitor Flag</p> <p>This bit is set to 1 when an ECC 1-bit error occurs while both SEDF and DEDF are 0.</p> <p>0: An internal reset or a pin reset has been generated. The FSTERRCLR bit in code flash error status clear register has been set.</p> <p>1: An ECC 1-bit error has occurred while both SEDF and DEDF are 0.</p>

### 35.9.3.4 CFFSTSTC — Code Flash Error Status Clear Register

This register clears error flags in the code flash error status register.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** FFC6 2024<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	FSTERRCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.20 CFFSTSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	FSTERRCLR	Clear SEDF and DEDF Flags in CFFSTERSTR Writing 1 to this bit clears the SEDF and DEDF flags in the CFFSTERSTR register.

### 35.9.3.5 CFOVFSTR — Code Flash Error Overflow Status Register

This register monitors occurrence of a code flash error overflow.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC6 2028<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.21 CFOVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF	Error Overflow Flag This bit is set when an ECC error occurs (an overflow occurs) while the error address registers is valid. <ul style="list-style-type: none"> <li>• If the retained address source is SEDF: When a 1-bit error occurs in the same address, an overflow does not occur. When a 1-bit error occurs in a different address, an overflow occurs. When a 2-bit error occurs, an overflow occurs.</li> <li>• If the retained address source is DEDF: When a 1-bit error occurs, an overflow occurs. When a 2-bit error occurs in the same address, an overflow does not occur. When a 2-bit error occurs in a different address, an overflow occurs.</li> </ul>

### 35.9.3.6 CFOVFSTC — Code Flash Error Overflow Status Clear Register

This register is used to clear the code flash error overflow flag. The flag is cleared by writing 1 to the ERROVFCLR bit.

**Access:** This register a write-only register that can be written in 8-bit units.

**Address:** FFC6 202CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVFCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.22 CFOVFSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear Writing 1 to this bit clears the ERROVF flag in the CFOVFSTR register. This bit is always read as 0.

### 35.9.3.7 CFERRINT — Code Flash Error Notification Control Register

This register enables or disables an error notification signal (FE level maskable interrupt request) when an ECC error is detected.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFC6 2030H

**Value after reset:** 02H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R/W

Table 35.23 CFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	1-Bit ECC Error Notification Control This bit controls error notification (FE level maskable interrupt request) on 1-bit error detection when ECC error detection/correction is enabled. 0: 1-bit ECC error notification is disabled. 1: 1-bit ECC error notification is enabled.

#### NOTE

When the SEG\_CONTROME flag in the SEG\_CONT register is 1, the SYSERR interrupt is generated when a 2-bit ECC error is detected.

### 35.9.3.8 CFFSTEADR — Code Flash First Error Address Register

This register holds the address where an ECC error has occurred.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFC6 2034<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
	—	—	—	—	—	—	—	CFFSTEADR[24:16]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
	CFFSTEADR[15:4]															—	—	—	—			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R					

**Table 35.24 CFFSTEADR Register Contents**

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	CFFSTEADR [24:4]	<p>First Error Occurrence Address</p> <p>These bits are read-only bits to monitor the address where the first error has occurred.</p> <p>The address is updated as follows:</p> <ul style="list-style-type: none"> <li>If the retained address source is SEDF: When a 1-bit error occurs in the same address, the address is not overwritten. When a 1-bit error occurs in a different address, the address is not overwritten. When a 2-bit error occurs, the address is overwritten.</li> <li>If the retained address source is DEDF: When a 1-bit error occurs, the address is not overwritten. When a 2-bit error occurs in the same address, the address is not overwritten. When a 2-bit error occurs in a different address, the address is not overwritten.</li> </ul>
3 to 0	Reserved	When read, the value after reset is returned.

### 35.9.3.9 CFTSTCTL — Code Flash Test Control Register

This register is used for the ECC test (self-diagnosis). The ECC bits can be read after ECC test mode is set by setting ECCTST to 1.

During this mode, the ECC check bits are read out instead of flash data.

Set the PROT1 and PROT0 bits to 01<sub>B</sub> when writing to this register.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC6 2054<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 35.25 CFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the ECCTST bit.
14	PROT0	The written data is not retained. These bits are always read as 0. Set PROT1 and PROT0 to 01 when writing to this register.
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test Mode Writing 1 to this bit sets ECC test mode. 0: Normal mode 1: ECC test mode

## 35.10 ECC Error Detection and Correction for Data Flash Memory

### 35.10.1 Outline of ECC Functions for Data Flash Memory

**Table 35.26** gives an outline of the ECC functions for the data flash memory.

**Table 35.26 Outline of the ECC Functions for the Data Flash Memory**

Item	Outline of Functions
ECC error detection and correction	<p>ECC errors detection and correction can be enabled or disabled. When enabled, either of the following settings can be also selected.</p> <ul style="list-style-type: none"> <li>• 2-bit error detection and 1-bit error detection/correction</li> <li>• 2-bit error detection and 1-bit error detection</li> </ul> <p>When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.</p>
Error notification	<p>A notification is sent when an ECC error occurs.</p> <ul style="list-style-type: none"> <li>• Enabling or disabling of error notification (INTECCDEEP0 interrupt) in the case of detection of ECC 2-bit errors is selectable. Selectable by setting the FEINTFMSK.ECCDEEP0FEIFMSK (INTECCDEEP0 interrupt mask) bit (initial value: interrupt is masked) and DFERRINT.DEDIE (2-bit ECC error notification control) bit (initial value: notification is enabled).</li> <li>• An error is not notified when a 1-bit ECC error is detected.</li> </ul> <p>In the initial state, error notification is disabled upon detection of a 2-bit ECC error.</p>
Error status	<p>The detection of ECC 2-bit errors and ECC 1-bit errors can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.</p>

### 35.10.2 Interrupt Request

The following table shows interrupt requests of data flash ECC.

**Table 35.27 Interrupt Requests of Data Flash ECC (Reading by the CPU)**

Unit Interrupt Signal	Outline	Name	DMA Trigger Number
—	ECC 2-bit error interrupt of data flash	INTECCDEEP0	—

### 35.10.3 Registers

#### 35.10.3.1 List of Registers

The following table lists the registers for ECC for the data flash memory.

**Table 35.28 List of Data Flash ECC Registers**

Register Name	Symbol	Address
Data flash ECC control register	DFECCCTL	FFC6 6000 <sub>H</sub>
Data flash error status register	DFERSTR	FFC6 6004 <sub>H</sub>
Data flash error status clear register	DFERSTC	FFC6 6008 <sub>H</sub>
Data flash error notification control register	DFERRINT	FFC6 6014 <sub>H</sub>
Data flash test control register	DFTSTCTL	FFC6 601C <sub>H</sub>

#### 35.10.3.2 DFECCCTL — Data Flash ECC Control Register

This register enables or disables the ECC error detection/correction and 1-bit error correction. Set the PROT1 and PROTO bits to 01<sub>B</sub> when writing to this register.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC6 6000<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SEDDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 35.29 DFECCCTL Register Contents**

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the SEDDIS and ECCDIS bits. The written data is not retained.
14	PROT0	These bits are always read as 0. Set PROT1 and PROT0 to 01 when writing to this register.
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SEDDIS	1-Bit Error Correction Disable This bit enables or disables 1-bit error correction when the ECC error detection/correction is enabled. When writing, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable This bit enables or disables ECC error detection/correction. When writing, write 01 to PROT1 and PROT0 at the same time. ECC error detection/correction is enabled in the initial state. 0: ECC error detection/correction is enabled. 1: ECC error detection/correction is disabled.

### 35.10.3.3 DFERSTR — Data Flash Error Status Register

This register monitors an error occurrence.

Detecting an ECC 1-bit error sets the SEDF bit and detecting an ECC 2-bit error sets the DEDF bit while ECC error detection/correction is enabled.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFC6 6004<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.30 DFERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	<p>ECC 2-Bit Error Monitor Flag</p> <p>This bit is set to 1 when an ECC 2-bit error occurs while both SEDF and DEDF are 0.</p> <p>0: An internal reset or a pin reset has been generated. The ERRCLR bit in data flash error status clear register is set. 1: An ECC 2-bit error has occurred while both SEDF and DEDF are 0.</p>
0	SEDF	<p>ECC 1-Bit Error Monitor Flag</p> <p>This bit is set to 1 when an ECC 1-bit error occurs while both SEDF and DEDF are 0.</p> <p>0: An internal reset or a pin reset has been generated. The ERRCLR bit in data flash error status clear register is set. 1: An ECC 1-bit error has occurred while both SEDF and DEDF are 0.</p>

### 35.10.3.4 DFERSTC — Data Flash Error Status Clear Register

This register clears error flags in the data flash error status register.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** FFC6 6008<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERRCLR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.31 DFERSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERRCLR	Clear SEDF and DEDF Flags in DFERSTR Writing 1 to this bit clears the SEDF and DEDF flags in the DFERSTR register.

### 35.10.3.5 DFERRINT — Data Flash Error Notification Control Register

This register enables or disables error notification signal generation when an ECC 2-bit error is detected.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFC6 6014<sub>H</sub>

**Value after reset:** 02<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDIE	—
Value after reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R

Table 35.32 DFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Control This bit controls error notification on 2-bit error detection when ECC error detection/correction is enabled. 0: ECC 2-bit error notification is disabled. 1: ECC 2-bit error notification is enabled.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

### 35.10.3.6 DFTSTCTL — Data Flash Test Control Register

This register is used for the ECC test (self-diagnosis). The ECC bits can be read after ECC test mode is set by setting ECCTST to 1.

Set the PROT1 and PROT0 bits to 01<sub>B</sub> when writing to this register.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC6 601C<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

**Table 35.33 DFTSTCTL Register Contents**

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the ECCTST bit. The written data is not retained.
14	PROT0	These bits are always read as 0. Set PROT1 and PROT0 to 01 when writing to this register.
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test Mode Writing 1 to this bit sets ECC test mode. 0: Normal mode 1: ECC test mode

## 35.11 Option Bytes

The option bytes of the flash memory are an expansion area and hold data specified by the user for a variety of purposes. Initial settings for peripheral modules and so on as specified by the option bytes become effective on release from the reset state.

### 35.11.1 Option Byte Setting

Be sure to set the option byte area that corresponds to the optional functions listed below, before writing a program to the flash memory.

The optional functions specified by the option bytes are as follows.

- Function of port group JP0
- Activation code method of WDTA1
- Start mode of WDTA1
- Enabling or disabling WDTA1
- Activation code method of WDTA0
- Start mode of WDTA0
- Enabling or disabling WDTA0
- Initial value of the overflow interval time for WDTA1 and WDTA0
- Frequency division for the external memory controller
- Enabling the high voltage monitor
- Enabling the low voltage monitor

### 35.11.2 OPBT0 — Option Byte 0

The settings and bit positions of the option bytes are listed below.

For details on how to set an option byte, refer to the flash programming user's manual for serial programming as well as the code flash library for self programming.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	OPJTAG[1:0]	—	—	WDT 1_3	—	WDT 1_1	WDT 1_0	WDT 0_3	—	WDT 0_1	WDT 0_0	WDT_2	WDT_1	WDT_0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MEMC DIV	—	—	CVM_H D_EN	CVM_L D_EN	—	—	—	—	—

Table 35.34 Option Byte Setting (1/2)

Bit Position	Bit Name	Function
31	Reserved	When writing, write “1”.
30, 29	OPJTAG[1:0]	These bits control the function of port group JP0. 00: JP0 is used for general purpose/alternative function port. 01: JP0 is used for LPD (4-pin). 10: JP0 is used for LPD (1-pin). 11: JP0 is used for Nexus I/F.
28, 27	Reserved	When writing, write “1”.
26	WDT1_3	Specifies the activation code method of WDTA1. 0: Fixed activation code 1: Variable activation code
25	Reserved	When writing, write “1”.
24	WDT1_1	Specifies the start mode of WDTA1. 0: Software trigger start mode 1: Default start mode
23	WDT1_0	Enables or disables WDTA1. 0: WDTA1 is disabled 1: WDTA1 is enabled
22	WDT0_3	Specifies the activation code method of WDTA0. 0: Fixed activation code 1: Variable activation code
21	Reserved	When writing, write “1”.
20	WDT0_1	Specifies the start mode of WDTA0. 0: Software trigger start mode 1: Default start mode
19	WDT0_0	Enables or disables WDTA0. 0: WDTA0 is disabled 1: WDTA0 is enabled
18 to 16	WDT_[2:0]	Control of the overflow interval time for WDTA0 and WDTA1 These bits specify the reset value of WDTAnMD.WDTAnOVF[2:0].
15 to 9	Reserved	When writing, write “1”.
8	MEMCDIV	Specifies the frequency division for the external memory controller <sup>*1</sup> 0: CPUCLK/2 1: CPUCLK/4
7, 6	Reserved	When writing, write “1”.
5	CVM_HD_EN	High Voltage Monitor Enable 0: Disable high voltage detection 1: Enable high voltage detection

**Table 35.34 Option Byte Setting (2/2)**

Bit Position	Bit Name	Function
4	CVM_LD_EN	Low Voltage Monitor Enable 0: Disable low voltage detection 1: Enable low voltage detection
3 to 0	Reserved	When writing, write "1".

Note 1. Set this bit to 1 (CPUCLK/4) when CPUCLK is to run at 96 MHz.

## 35.12 Usage Notes

### (1) Reading areas where programming or erasure was interrupted

When programming or erasure of an area of flash memory is interrupted, the data stored in the area become undefined. To avoid the reading out of undefined data, which might cause a malfunction, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.

### (2) Reading the code flash memory that has been erased but not yet been programmed again

Note that reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state.

### (3) Prohibition of additional writing

Writing to a given area two or more times is not possible. When overwriting data in an area of flash memory after writing to the area has been completed, erase the area first.

### (4) Resets during programming and erasure

In the case of an external reset during programming and erasure, wait for at least the minimum value of RESET input low level width once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.

### (5) Allocation of vectors for interrupts and other exceptions during programming and erasure

Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.

### (6) Abnormal termination of programming and erasure

Even if programming/erasure ends abnormally due to the assertion of a reset by the RESET pin, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming/erasure ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again to ensure that the corresponding area is completely erased before using.

If programming and erasure of code flash memory are not completed normally, the lock bit for the target area may be enabled (locked). In such cases, erase the block to erase the lock bit while the lock bit is in the disabled state (the area is not locked).

### (7) Items prohibited during programming and erasure

Do not perform the following operations during programming and erasure.

- Set the operating voltage from the power supply outside the allowed range.
- Change the frequency of the peripheral clock.

## Section 36 RAM

This section describes the local RAM mounted on RH850/F1L and the error correction function (ECC) of the local RAM.

### 36.1 Features

- RH850/F1L includes the following RAMs:

- Primary local RAM

The primary local RAM is accessible at high speed. Values in the primary local RAM are not retained in DeepSTOP mode.

- Secondary local RAM

Values in the secondary local RAM are not retained in DeepSTOP mode.

- Retention RAM

Values in the retention RAM are retained in DeepSTOP mode.

In addition, even if the power-supply voltage (REGVCC) falls below the POC voltage, data in the retention RAM are retained as long as the voltage does not fall below the RAM retention voltage ( $V_{VLVI}$ ).

Access time for each RAM is shown in the table below.

**Table 36.1 RAM Access Time**

Type of Access	RAM	1st Access (CPUCLK)	Continuous Access (CPUCLK)
Instruction fetch	Primary local RAM	2	2 or 1 <sup>*1</sup>
	Secondary local RAM	3	3 or 1 <sup>*1</sup>
	Retention RAM	3	3 or 1 <sup>*1</sup>
Read access	Primary local RAM	1	1
	Secondary local RAM	2	2
	Retention RAM	2	2
Write access	Primary local RAM	1	1
	Secondary local RAM	1	1
	Retention RAM	1	1

Note 1. Depending on the executing instruction length, fetch is possible by 1 cycle time.

#### NOTE

There is possibility that number of access clock of above table is changed depending on the combination before and after instructions.

When RAM access is misaligned, these number is increased.

- Error detection/correction function (ECC) in the local RAM

The ECC function is included, which can detect 2-bit errors and detect/correct 1-bit errors.

The size of the RAM mounted on each product is shown below.

- **Section 3.2.4.3, Primary Local RAM Area**
- **Section 3.2.4.4, Secondary Local RAM Area**
- **Section 3.2.4.5, Retention RAM Area**

## 36.2 Memory Configuration

Figure 36.1 shows the memory map of the local RAM.

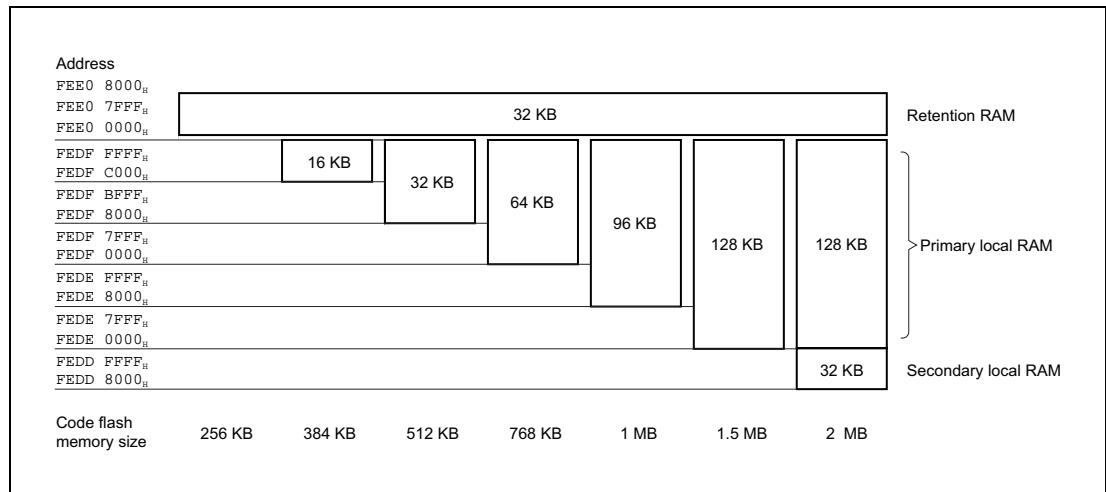


Figure 36.1 Memory Map of the Local RAM

## 36.3 ECC Error Detection and Correction for Local RAM

### 36.3.1 Outline of ECC Functions for Local RAM

**Table 36.2** gives an outline of the ECC functions for the local RAM.

**Table 36.2 List of the ECC Functions for the Local RAM**

Item	Outline of Functions
ECC error detection/correction	<p>The ECC error detection/correction can be enabled or disabled. When enabled, either of the following two settings can be selected.</p> <ul style="list-style-type: none"> <li>• ECC error detection/correction (2-bit error detection and 1-bit error detection/correction)</li> <li>• ECC error detection (2-bit error detection and 1-bit error detection)</li> </ul> <p>When disabled, errors are not detected or corrected. In the initial state, the ECC function is enabled, and 1-bit error detection/correction and 2-bit error detection/notification are selected.</p>
Error notification	<p>Notifies an ECC error upon occurrence.</p> <ul style="list-style-type: none"> <li>• Error notification can be enabled or disabled when a 2-bit ECC error (SYSERR exception) is detected. Selectable by setting the SEG_CONT.SEG_CONTRAME (primary/secondary local RAM/retention RAM area error notification enable) bit (value after a reset: notification is disabled).</li> <li>• Error notification can be enabled or disabled when a 1-bit ECC error (INTECCRAM interrupt) is detected. Selectable by setting the FEINTFMSK.ECCRAMFEIFMSK (INTECCRAM interrupt mask) bit (value after a reset: interrupt is masked) and LRERRINT.SEDIE (1-bit ECC error notification enable) bit (value after a reset: notification is disabled).</li> </ul>
Error status	<p>Monitoring for the detection of 2-bit ECC errors and for the detection of 1-bit ECC errors is available. The 1-bit ECC error status is set only in a situation where no error status setting has been made. The 2-bit ECC error status is set even when the 1-bit ECC error status has been set. A register for clearing the error status is provided.</p>
Address capture	<p>When no error status has been set, the address at which the first ECC error occurred is captured. In addition, when the retained address source is a 1-bit ECC error, the address of the 2-bit ECC error is also captured. The error signal for a 2-bit ECC error and the error signal for a 1-bit ECC error serve as triggers for address capture. These enable the bits for the error status.</p>
Others	<p>Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching avoids the execution of incorrect instructions.</p>

#### CAUTION

**When ECC error detection/correction for the embedded RAM is enabled, initialize the RAM with the 32-bit length access size before the RAM is used.**

**If the RAM before initialization is read, an FE-level maskable interrupt or SYSERR exceptional processing may be generated.**

**Moreover, if the RAM is not initialized with the 32-bit length (and initialized with 8- or 16-bit length), an FE-level maskable interrupt or SYSERR exceptional processing may be generated.**

### 36.3.2 Interrupt Request

The following table shows Local RAM ECC interrupt requests.

**Table 36.3 Local RAM ECC interrupt requests**

Unit Interrupt number	Description	Name	DMA Trigger Number
—	1-bit ECC error interrupt of local RAM	INTECCRAM	—
—	2-bit ECC error interrupt of local RAM	SYSERR	—

## 36.4 Registers

### 36.4.1 List of Registers

The following table lists the ECC-related registers for the local RAM.

**Table 36.4 List of RAM ECC Registers**

Register Name	Symbol	Address
Local RAM ECC control register	LRECCCTL	FFC6 3000 <sub>H</sub>
Local RAM first error status register	LRFSTERSTR	FFC6 3004 <sub>H</sub>
Local RAM error status clear register	LRSTCLR	FFC6 3024 <sub>H</sub>
Local RAM error overflow status register	LROVFSTR	FFC6 3028 <sub>H</sub>
Local RAM error overflow status clear register	LROVFSTC	FFC6 302C <sub>H</sub>
Local RAM first error address register 0	LRFSTEADR0	FFC6 3030 <sub>H</sub>
Local RAM error notification control register	LRERRINT	FFC6 30B0 <sub>H</sub>
Local RAM test control register	LRTSTCTL	FFC6 30B4 <sub>H</sub>
Local RAM test data read buffer 0	LRTDATBF0	FFC6 30B8 <sub>H</sub>

### 36.4.2 LRECCCTL — Local RAM ECC Control Register

The LRECCCTL register specifies whether to enable or disable ECC error detection/correction and whether to enable or disable 1-bit error correction. When writing to the LRECCCTL register, PROT1 and PROTO must be 01<sub>B</sub>.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC6 3000<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 36.5 LRECCCTL Register Contents**

Bit Position	Bit Name	Function
15	PROT1	These two bits specify whether updating the ECCDIS and SECDIS bits is disabled or enabled.
14	PROT0	The value written to these bits is not retained. When these bits are read, the read value is always 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. Write a value to this bit simultaneously with setting PROT1 to 0 and PROT0 to 1. 1-bit error correction is enabled in the initial state. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
0	ECCDIS	ECC Disable This bit specifies whether to enable or disable the ECC error detection/correction function. Write a value to this bit simultaneously with setting PROT1 to 0 and PROT0 to 1. ECC error detection/correction is enabled in the initial state. 0: ECC error detection/correction function is enabled. 1: ECC error detection/correction function is disabled. Even when the error detection/correction function is disabled, ECC data is still written.

### 36.4.3 LRFSTERSTR — Local RAM First Error Status Register

LRFSTERSTR is a register for monitoring the first error.

If the ECC error detection/correction is enabled, the SEDF0 bit is set when a 1-bit ECC error is detected, and the DEDF0 bit is set when a 2-bit ECC error is detected.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFC6 3004<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDFO	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Table 36.6 LRFSTERSTR Register Contents**

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDFO	<p>Local RAM 2-Bit ECC Error Monitor Flag This bit is set when a 2-bit ECC error occurs regardless of the SEDF0 and DEDF0 settings.</p> <ul style="list-style-type: none"> <li>• Clearing condition: Internal reset or pin reset is asserted. The FSTERRCLR0 bit in the local RAM error status clear register is set.</li> <li>• Setting condition: A 2-bit ECC error occurs.</li> </ul>
0	SEDF0	<p>Local RAM 1-Bit ECC Error Monitor Flag This bit is set when a 1-bit ECC error occurs while both SEDF0 and DEDF0 are 0.</p> <ul style="list-style-type: none"> <li>• Clearing condition: Internal reset or pin reset is asserted. The FSTERRCLR0 bit in the local RAM error status clear register is set.</li> <li>• Setting condition: A 1-bit ECC error occurs while both SEDF0 and DEDF0 are 0.</li> </ul>

### 36.4.4 LRSTCLR — Local RAM Error Status Clear Register

LRSTCLR is a register used for clearing the error flag in the local RAM error status register.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** FFC6 3024<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	FSTERR CLR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

**Table 36.7 LRSTCLR Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	FSTERRCLR0	First Error Status Register Clear Write 1 to this bit to clear the error flag in the first error status register.

### 36.4.5 LROVFSTR — Local RAM Error Overflow Status Register

LROVFSTR is a register used for monitoring the occurrence of local RAM error overflow. The ERROVF0 flag is cleared by setting 1 to the ERROVFCLR0 bit in the LROVFSTC register.

**Access:** This register is a read-only register that can be read in 8-bit units.

**Address:** FFC6 3028<sub>H</sub>

**Value after reset:** 00<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 36.8 LROVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	<p>Error Overflow Flag</p> <p>This bit is set when an ECC error occurs while error address registers are enabled.</p> <ul style="list-style-type: none"> <li>• When the retained address source is SEDF           <ul style="list-style-type: none"> <li>1-bit error occurs at the same address → Overflow does not occur</li> <li>1-bit error occurs at a different address → Overflow occurs</li> <li>2-bit error occurs → Overflow occurs</li> </ul> </li> <li>• When the retained address source is DEDF           <ul style="list-style-type: none"> <li>1-bit error occurs → Overflow occurs</li> <li>2-bit error occurs at the same address → Overflow does not occur</li> <li>2-bit error occurs at a different address → Overflow occurs</li> </ul> </li> </ul>

### 36.4.6 LROVFSTC — Local RAM Error Overflow Status Clear Register

LROVFSTC is a register used for clearing the local RAM error overflow flag.

The error overflow flag is cleared when the ERROVFCLR0 bit is set to 1.

**Access:** This register is a write-only register that can be written in 8-bit units.

**Address:** FFC6 302CH

**Value after reset:** 00H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF CLR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

**Table 36.9 LROVFSTC Register Contents**

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERROVFCLR0	Error Overflow Flag Clear Write 1 to this bit to clear the ERROVF0 flag. This bit is always read as 0.

### 36.4.7 LRFSTEADRO — Local RAM First Error Address Register 0

LRFSTEADRO retains the address of an ECC error occurrence. This register retains an internal address. Add the base address FEC0 0000<sub>H</sub> to convert it to the actual address.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFC6 3030<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LRFSTEADRO[21:16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LRFSTEADRO[15:2]															—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.10 LRFSTEADRO Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 2	LRFSTEADRO [21:2]	Address of the First Error Occurrence These bits are read only and used to monitor the address of the first error occurrence. <ul style="list-style-type: none"> <li>• When the retained address source is SEDF 1-bit error occurs at the same address → The address is not overwritten. 1-bit error occurs at a different address → The address is not overwritten. 2-bit error occurs → The address is overwritten</li> <li>• When the retained address source is DEDF 1-bit error occurs → The address is not overwritten. 2-bit error occurs at the same address → The address is not overwritten. 2-bit error occurs at a different address → The address is not overwritten.</li> </ul>
1, 0	Reserved	When read, the value after reset is returned.

### 36.4.8 LRERRINT — Local RAM Error Notification Control Register

The LRERRINT register enables or disables an error notification signal when an 1 bit ECC error is detected.

**Access:** This register can be read or written in 8-bit units.

**Address:** FFC6 30B0<sub>H</sub>

**Value after reset:** 02<sub>H</sub>

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R/W

Table 36.11 LRERRINT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	<p>1-Bit ECC Error Notification Enable</p> <p>This bit controls whether to notify an error when a 1-bit error is detected while ECC error detection/correction is enabled.</p> <p>0: 1-bit ECC error notification is disabled.</p> <p>1: 1-bit ECC error notification is enabled.</p>

### 36.4.9 LRTSTCTL — Local RAM Test Control Register

This register is used for an ECC test (self diagnosis). After the ECC test mode is enabled by setting ECCTST = 1, arbitrary data can be written to the ECC bits. The DATSEL bit is used to select the RAM data or the ECC bits.

When writing to the LRTSTCTL register, PROT1 and PROT0 must be 01<sub>B</sub>.

**Access:** This register can be read or written in 16-bit units.

**Address:** FFC6 30B4<sub>H</sub>

**Value after reset:** 0000<sub>H</sub>

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DATSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

**Table 36.12 LRTSTCTL Register Contents**

Bit Position	Bit Name	Function
15	PROT1	These two bits specify whether updating the ECCTST and DATSEL bits is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0.
14	PROT0	Set PROT1 to 0 and PROTO to 1 when writing to this register.
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	ECC Test Mode This bit specifies the ECC test mode. 0: Normal mode 1: ECC test mode
0	DATSEL	Data Selection This bit is enabled when ECCTST = 1. This bit specifies the RAM bit which can be accessed when writing. 0: RAM data is selected. 1: ECC bits are selected.

### 36.4.10 LRTDATBF0 — Local RAM Test Data Read Buffer 0

In test mode (ECCTST = 1), the ECC bits can be read. While ECCTST = 1 in the local RAM test control register, reading the RAM reads out the data of the ECC bits stored in the RAM, and the data is stored in this buffer.

**Access:** This register is a read-only register that can be read in 32-bit units.

**Address:** FFC6 30B8<sub>H</sub>

**Value after reset:** 0000 0000<sub>H</sub>

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
LRTDATBF0[6:0]																

Table 36.13 LRTDATBF0 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	LRTDATBF0 [6:0]	These bits are enabled when ECCTST = 1 (test mode) in the local RAM test control register. When reading the RAM, the data of the ECC bits are stored in LRTDATBF06 to LRTDATBF00.

## Section 37 Boundary Scan

This section contains a generic description of boundary scan.

The RH850/F1L has a JTAG interface and provides a boundary scan function.

### 37.1 Overview

Boundary scan is a test method defined in the IEEE standard 1149.1 that is used to test the connection between the devices mounted on the printed-circuit board. The boundary scan of the RH850/F1L conforms to IEEE Std 1149.1-2001.

### 37.2 Features

- Five control signals (DCUTCK, DCUTDI, DCUTDO, DCUTMS, and DCUTRST)
- TAP controller
- Instruction register
- Bypass register
- Boundary scan register

The JTAG interface has four instruction modes.

- BYPASS  
Test mode conforming to the IEEE 1149.1
- EXTEST  
Test mode conforming to the IEEE 1149.1
- SAMPLE/PRELOAD  
Test mode conforming to the IEEE 1149.1
- IDCODE  
Test mode conforming to the IEEE 1149.1

Figure 37.1 shows a block diagram of the JTAG interface.

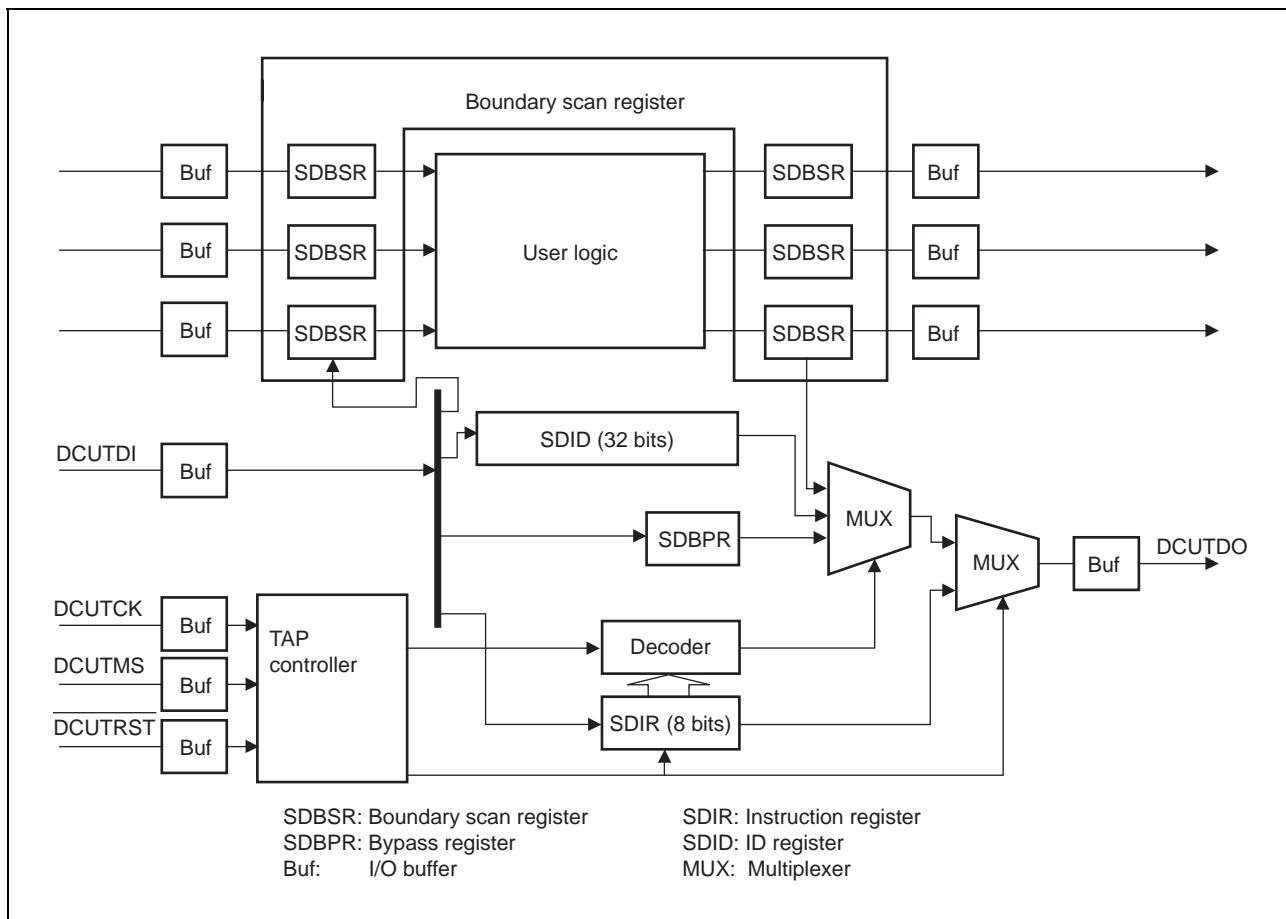


Figure 37.1 Block Diagram of JTAG Interface

### 37.3 External Input/Output Pins

There are five JTAG control signals: DCUTCK, DCUTDI, DCUTMS, DCUTDO, and  $\overline{\text{DCUTRST}}$ .

**Table 37.1** shows the pin configuration.

**Table 37.1 Pin Configuration**

Pin Name	Description
DCUTCK	Serial data input/output clock pin Data is input to DCUTDI and is output from DCUTDO in synchronization with this clock signal.
DCUTMS	Mode select input pin Changing the level of this signal in synchronization with DCUTCK changes the state of the TAP controller. For the protocol, see <b>Figure 37.2, TAP Controller State Transition Diagram</b> .
DCUTRST	Reset input pin A low-level input of this signal resets the JTAG interface. This signal is accepted asynchronously with DCUTCK.
DCUTDI	Serial data input pin Data is input in synchronization with DCUTCK and sent to the JTAG interface.
DCUTDO	Serial data output pin Data to be read from the JTAG interface is output in synchronization with DCUTCK.

## 37.4 Register Descriptions

The JTAG interface has the following registers. None of the registers can be accessed by the CPU.

- SDIR: Instruction register
- SDID: ID register
- SDBPR: Bypass register
- SDBSR: Boundary scan register

**Table 37.2 Register Configuration**

Register Name	Symbol	Access Size	Initial Value <sup>*1</sup>
Instruction register	SDIR	8	55H
ID register	SDID	32	<sup>*2</sup>
Bypass register	SDBPR	1	Undefined
Boundary scan register	SDBSR	—	Undefined

Note 1. Registers are initialized when DCUTRST pin is 0 or when TAP is in the Test-Logic-Reset state.

Note 2. The initial value differs depending on the device. Please contact our sales representative for details.

Instructions can be serially transferred from the serial data input pin (DCUTDI) and input to the instruction register (SDIR). The bypass register (SDBPR) is a 1-bit register, to which DCUTDI and DCUTDO are connected in BYPASS mode. The boundary scan register (SDBSR) is connected to DCUTDI and DCUTDO in SAMPLE/PRELOAD mode and EXTEST mode. The ID code register (SDID) is a 32-bit register, from which the ID code is output via DCUTDO in IDCODE mode.

**Table 37.3** shows the serial transfer types possible with the JTAG interface registers.

**Table 37.3 Serial Transfer Types**

Register	Serial Input	Serial Output
SDIR	Possible	Impossible <sup>*1</sup>
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDID	Impossible	Possible

Note 1. A fixed value is read out.

### 37.4.1 Instruction Register (SDIR)

SDIR is an 8-bit register that holds a boundary scan instruction. SDIR is initialized by a low-level input of DCUTRST or in the TAP Test-Logic-Reset state. Operation is not guaranteed if a reserved instruction is set in this register.

**Table 37.4 Boundary Scan Instructions**

Instruction Code								Description
0	0	0	0	0	0	0	0	JTAG EXTEST
0	1	0	0	0	0	0	0	JTAG SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	JTAG IDCODE (initial value)
1	1	1	1	1	1	1	1	JTAG BYPASS
Other than above								Reserved

### 37.4.2 ID Register (SDID)

SDID is a 32-bit register with a device specific ID.

SDID can be read from the JTAG interface when the IDCODE instruction is set, but cannot be accessed from the CPU.

For the read values, see **Table 37.2, Register Configuration**.

### 37.4.3 Bypass Register (SDBPR)

SDBPR is a 1-bit register. When SDIR is set to BYPASS mode, SDBPR is connected to the position between DCUTDI and DCUTDO. The initial value is undefined. SDBPR is not initialized by a power-on reset or by a low-level input of DCUTRST.

### 37.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register for controlling the external Input/Output pins. When SDIR is set to SAMPLE/PRELOAD or EXTEST mode, SDBSR is connected to the position between DCUTDI and DCUTDO. The initial value is undefined. SDBSR is not initialized by a power-on reset or a low-level input of DCUTRST.

## 37.5 Operation

### 37.5.1 TAP Controller

**Table 37.2** shows the state transition of the TAP controller. Transition is triggered by the DCUTMS value at the rising edge of DCUTCK.

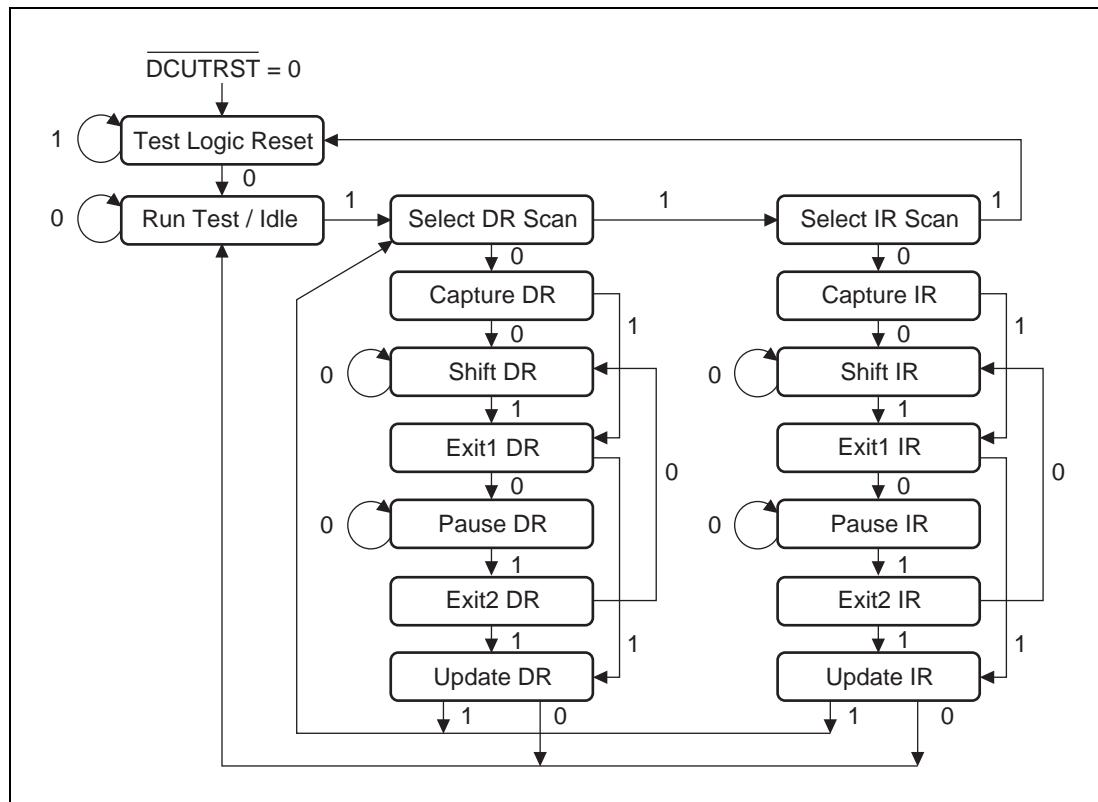


Figure 37.2 TAP Controller State Transition Diagram

#### NOTE

The DCUTDI value is sampled at the rising edge of DCUTCK and is shifted at the falling edge. DCUTDO is in the high-impedance state in the states other than Shift-DR and Shift-IR. A low-level input of DCUTRST causes transition to Test-Logic-Reset state asynchronously with DCUTCK.

### 37.5.2 Supported Instructions

#### 37.5.2.1 BYPASS

The BYPASS instruction is a standard instruction indispensable to bypass register operation. This instruction shortens the shift path to achieve high-speed serial data transfer of other devices on the printed-circuit board. During execution of this instruction, the test circuit has no effect on the system circuit.

#### 37.5.2.2 SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input the value to the boundary scan register from the internal circuits of this device; to output the value from the scan path; and to load data onto the scan path. During execution of this instruction, the level of the input pin of this device is sent to the internal circuits as is, and the value of the internal circuits is output to the outside via the output pin as is. Executing this instruction has no effect on the system circuit of this device.

The SAMPLE operation allows taking in the snapshots of the value to be transferred to the internal circuits from the input pin or the value to be transferred to the output pin from the internal circuits to the boundary scan register and allows reading the snapshots from the scan path. Snapshots can be taken in without preventing the normal operation of this device.

The PRELOAD operation allows setting the initial value to the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. If the EXTEST instruction is executed without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transfer to the output latch) because the parallel output latch value is always output to the output pin by the EXTEST instruction.

#### 37.5.2.3 EXTEST

The EXTEST instruction is used to test the external circuits when this device is mounted on the printed-circuit board. When this instruction is executed, the output pin is used to output the test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed-circuit board; whereas the input pin is used to take in the test result from the printed-circuit board to the boundary scan register. When the EXTEST instruction is executed N times for testing, the test data for the Nth execution is scanned in at the (N - 1)th scan-out.

If the data is loaded onto the boundary scan register of the output pin in the Capture-DR state of this instruction, it is not used for testing the external circuits (replaced through shift operation).

#### 37.5.2.4 IDCODE

The IDCODE instruction sets the JTAG interface pins to IDCODE mode, which is defined by the JTAG standard. When the JTAG interface is initialized (by a low-level input of DCUTRST or placing TAP in the Test-Logic-Reset state), IDCODE mode is set.

### 37.5.3 Pins Subject to Boundary Scan

All pins, excluding pins such as external clock input pins or power supply pins, are subject to boundary scan.

The pins which are not subject to boundary scan are listed in **Table 37.5**.

**Table 37.5 Pins not Subject to Boundary Scan**

Type	Pins
JTAG interface	DCUTCK, DCUTDI, DCUTDO, DCUTMS, <u>DCUTRST</u>
Power supply pins	REGVCC, AWOVCL, AWOVSS, ISOVCL, ISOVSS EVCC, BVCC <sup>*1</sup> , EVSS, BVSS <sup>*1</sup>
Power supply pins (A/D converter)	A0VREF, A1VREF <sup>*1</sup> , A0VSS, A1VSS <sup>*1</sup>
Clock signals	X1, X2, XT1 <sup>*1</sup> , IP0_0/XT2 <sup>*1</sup>
MODE	P10_8/FLMD1, P10_1/MODE0, P10_2/MODE1

Note 1. Only available for 144 and 176 pin devices.

The following signals are only sampled in boundary scan mode.

**Table 37.6 Pins Subject to Boundary Scan (Sampling Only)**

Function	Pin Name
Reset	<u>RESET</u>
MODE	FLMD0

The following pins are shared by the analog buffer. Accordingly, boundary scan only applies to general I/O pins.

**Table 37.7 Pins Subject to Boundary Scan (Only General I/O Pins)**

Function	Pin Name
ADCA0 input	P8_0-12, P9_0-6, AP0_0-15
ADCA1 input	P18_0-7, AP1_0-15

#### NOTE

In boundary scan mode, the level of the following pins must be fixed:  
P10\_1: Low, P10\_2: High, P10\_8: High

## 37.6 Usage Notes

1. Once an instruction is set, it is not modified until another instruction is issued. To issue the same instruction twice in a row, insert an instruction that has no effect on chip operation (such as BYPASS) between the instructions.
2. To start the system in boundary scan mode, de-assert  $\overline{\text{DCUTRST}}$  while  $\overline{\text{RESET}}$  is high. Also be sure to set DCUTMS to high before de-asserting  $\overline{\text{DCUTRST}}$  and ensure that DCUTMS remains high for 600 ns + five DCUTCK clock cycles after negating  $\overline{\text{DCUTRST}}$ .
3. For the maximum clock frequency that can be input to DCUTCK, see the Electrical Characteristics section in the Data Sheet document.
4. If serial transfer is performed exceeding the number of bits of the register connected between DCUTDI and DCUTDO, the data that is input from DCUTDI is output from DCUTDO as is.
5. If the serial transfer sequence is corrupted, be sure to assert  $\overline{\text{DCUTRST}}$ . In this case, transfer starts again from the beginning regardless of the point of transfer corruption.
6. Data is output via DCUTDO at the falling edge of DCUTCK.
7. To facilitate debugging, route  $\overline{\text{DCUTRST}}$  on the board in such a way that patterns can be easily cut.

## Section 38 Power Supply and Power Domains

This section describes the power supply and power domains of the RH850/F1L.

### 38.1 Function

The internal circuits are separated into two independent power domains, the Always-On area (AWO) and the Isolated area (ISO).

The power supply of the Always-On area is always powered in all operating modes and stand-by modes.

The power supply of the Isolated area can be turned off to reduce the overall power consumption depending on the types of stand-by mode.

For each power domain, a dedicated on-chip voltage regulator generates the internal supply voltage.

For operation of the device, the following voltages are required:

- Power supply voltage REGVCC for the on-chip voltage regulators. The output voltage of the voltage regulators is supplied to the digital circuits in each power domain.
- Power supply voltages EVCC and BVCC<sup>\*1</sup> for I/O port.
- Power supply voltages A0VREF and A1VREF<sup>\*1</sup> for the A/D converters and the dedicated I/O ports.

**Note 1.** Only available for 144 pin and 176 pin devices.

### 38.1.1 Power Supply Pins

The table below lists all power supply pins and what they are used for.

**Table 38.1 Power Supply Pins**

Power Supply	Power Supply Pins	Power Supply for
Power supply for internal circuits	REGVCC	• On-chip voltage regulators for the Always-On area and Isolated area
	AWOVCL <sup>*1</sup>	• Port group IP0 <sup>*2</sup>
	AWOVSS	• MainOSC
	ISOVCL <sup>*1</sup>	• SubOSC <sup>*2</sup>
	ISOVSS	• POC / LVI
Power supply for I/O port	EVCC	RESET, FLMD0 (176 pin device)
	EVSS	• Port groups JP0, P0, P1, P2, P8, P9, P20 (144 pin device) • Port groups JP0, P0, P1, P8, P9, P20 (80 pin and 100 pin devices) • Port groups JP0, P0, P8, P9, P10, P11 (48 pin and 64 pin devices) • Port groups JP0, P0, P8, P9, P10
	BVCC	(144 pin and 176 pin devices)
	BVSS	• Port groups P10, P11, P12, P18
	A0VREF	• Analog circuits of ADCA0, port group AP0
Power supply for A/D converters	A0VSS	
	A1VREF	(144 pin and 176 pin devices)
	A1VSS	• Analog circuits of ADCA1, port group AP1

**Note:** Refer to the Electrical Characteristics in the data sheet for the voltage range of each power supply.

Note 1. Pin to connect a stabilization capacitor for on-chip voltage regulator.

Note 2. Supported only for the 144 pin and 176 pin products.

### 38.1.2 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

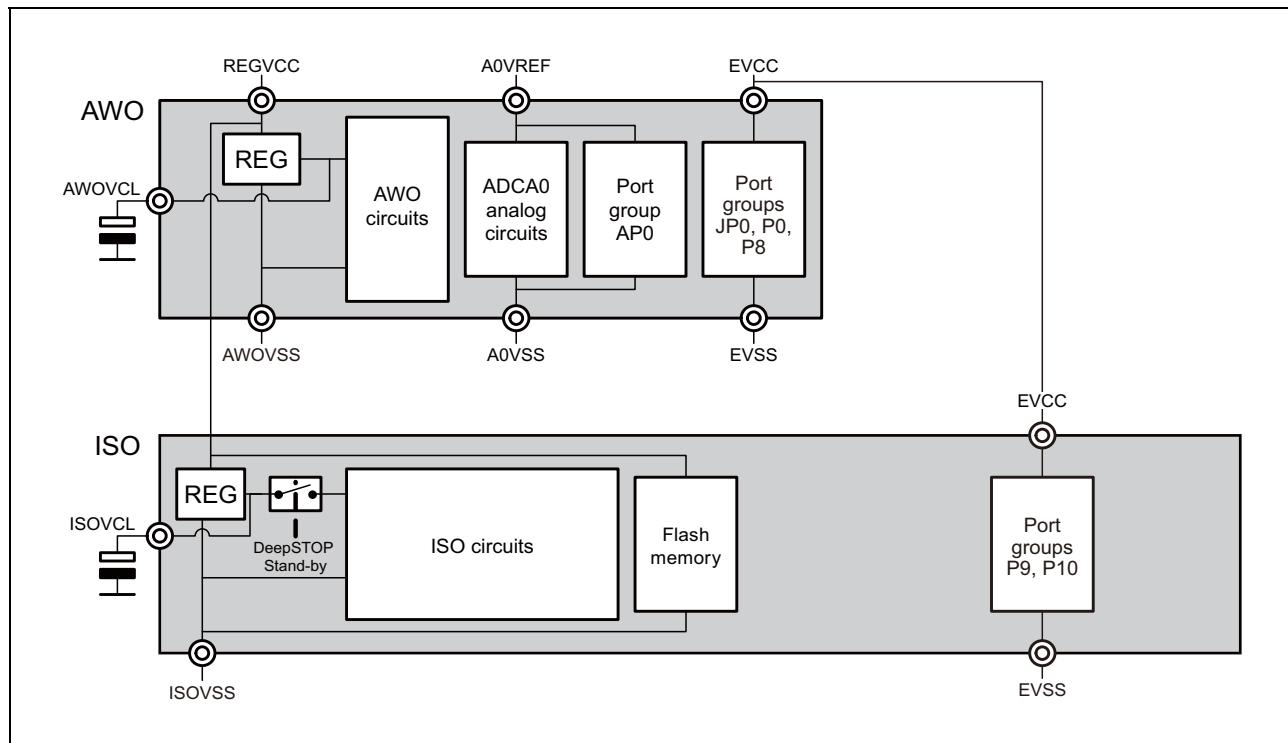


Figure 38.1 Overview of Power Supply Circuit (48 pin and 64 pin)

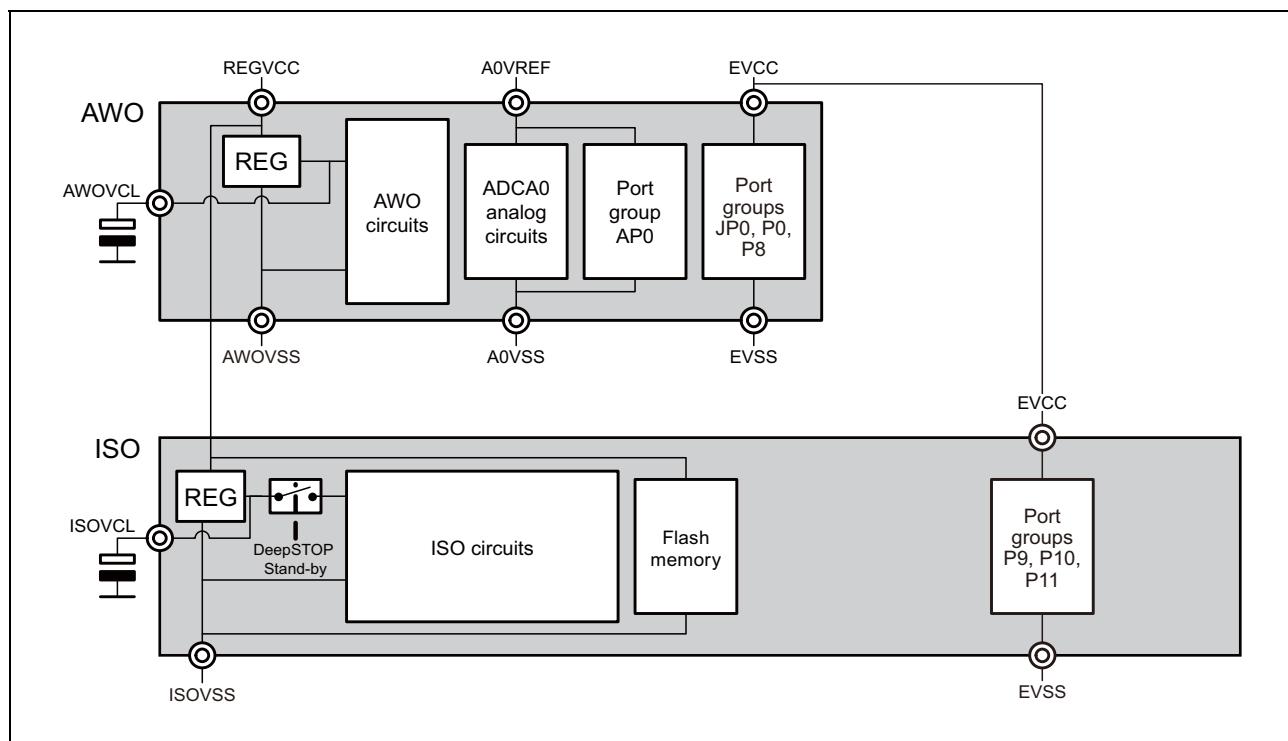


Figure 38.2 Overview of Power Supply Circuit (80 pin and 100 pin)

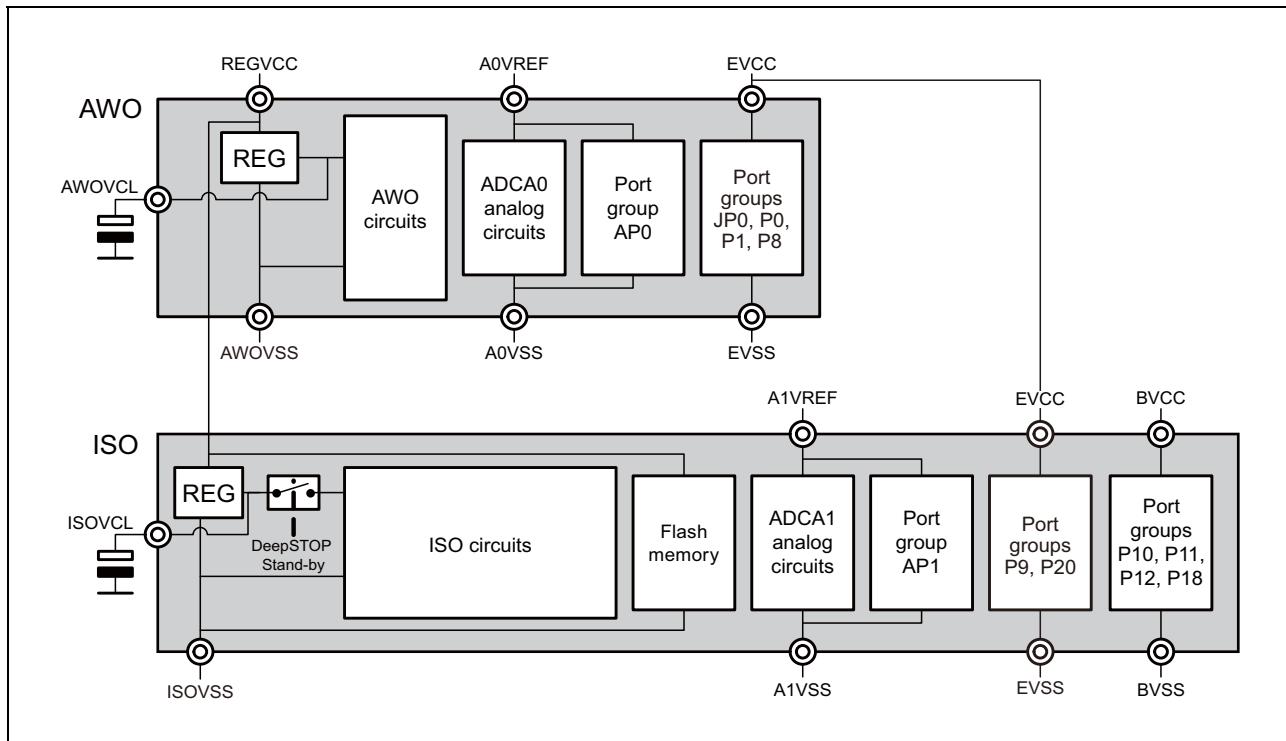


Figure 38.3 Overview of Power Supply Circuit (144 pin)

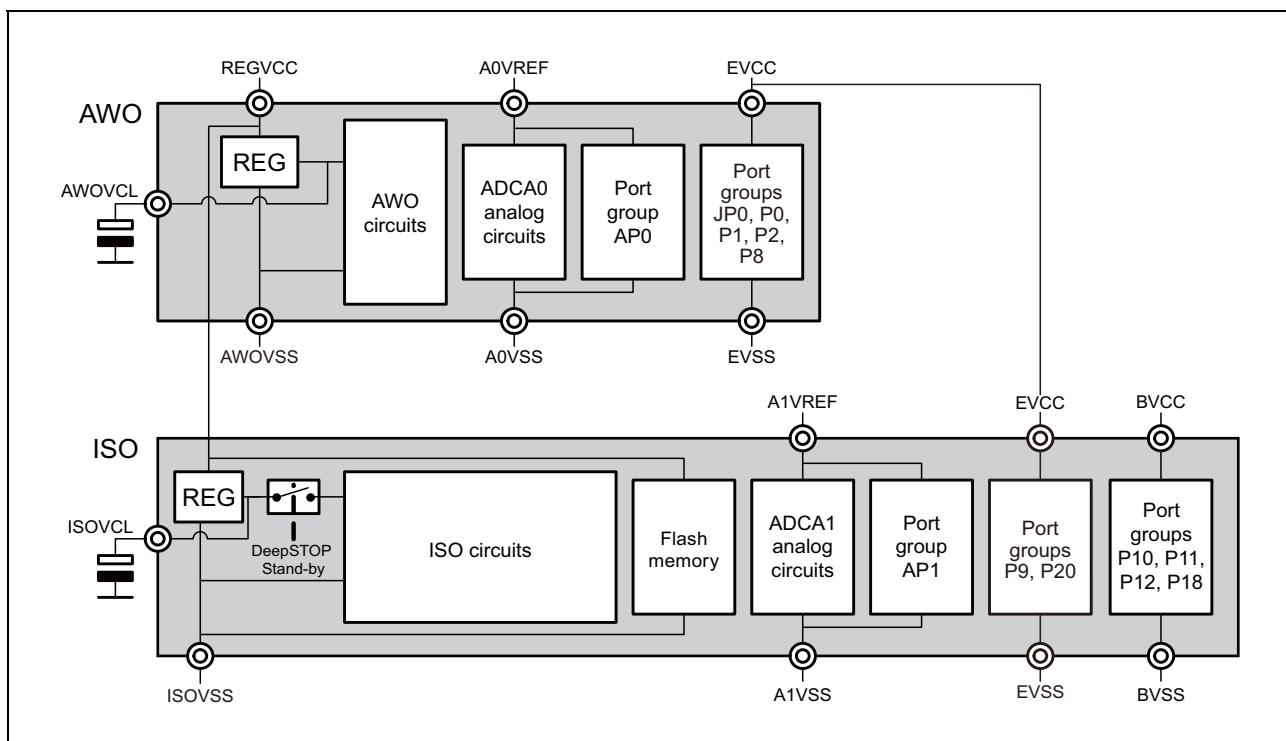


Figure 38.4 Overview of Power Supply Circuit (176 pin)

### 38.1.3 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

**Table 38.2 Functional Modules and Power Domain**

Power Domain	Functions
AWO area	<ul style="list-style-type: none"><li>• STBC, Reset controller</li><li>• Retention RAM</li><li>• MainOSC, SubOSC, LS IntOSC, HS IntOSC, CLMA0, CLMA1</li><li>• WDTA0, RTCAn, TAUJ0, ADCA0, LPS</li><li>• Port groups JP0, P0, P1, P2, P8, AP0, IP0</li></ul>
ISO area	<ul style="list-style-type: none"><li>• CPU subsystem</li><li>• Code flash, Data flash, Primary local RAM, Secondary local RAM</li><li>• PLL, CLMA2</li><li>• WDAT1, DCRAn, TAUDn, TAUBn, TAUJ1, OSTMn, PWM-Diag, CSIGn, CSIhn, RSCANN, RLIN2m, RLIN3n, RIICn, ADCA1, Motor Control, ENCAN, KRn</li><li>• Port groups P9, P10, P11, P12, P18, P20, AP1</li></ul>

## Appendix A. Pin List

Table A.1 Pin List (1/4)

Pin Name	IO	Function	Input					Output	Other			Device							
			CMOS	SHMT1	SHMT2	SHMT4	TTL		output driver strength	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
RESET	I	System control		✓										✓	✓	✓	✓	✓	✓
A0VREF	—	Power for ADCA0												✓	✓	✓	✓	✓	✓
A0VSS	—	GND for ADCA0												✓	✓	✓	✓	✓	✓
A1VREF	—	Power for ADCA1																✓	✓
A1VSS	—	GND for ADCA1																✓	✓
AP0_0	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_1	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_2	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_3	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_4	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_5	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_6	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_7	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_8	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_9	IO	Analog port (AP0)	✓					✓	Low	*4	✓			✓	✓	✓	✓	✓	✓
AP0_10	IO	Analog port (AP0)	✓					✓	Low	*4	✓					✓	✓	✓	✓
AP0_11	IO	Analog port (AP0)	✓					✓	Low	*4	✓					✓	✓	✓	✓
AP0_12	IO	Analog port (AP0)	✓					✓	Low	*4	✓					✓	✓	✓	✓
AP0_13	IO	Analog port (AP0)	✓					✓	Low	*4	✓					✓	✓	✓	✓
AP0_14	IO	Analog port (AP0)	✓					✓	Low	*4	✓					✓	✓	✓	✓
AP0_15	IO	Analog port (AP0)	✓					✓	Low	*4	✓					✓	✓	✓	✓
AP1_0	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_1	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_2	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_3	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_4	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_5	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_6	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_7	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓					✓	✓	✓
AP1_8	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_9	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_10	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_11	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_12	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_13	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_14	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AP1_15	IO	Analog port (AP1)	✓					✓	Low	*4	✓	✓							✓
AW0VCL	—	Power for AWO area												✓	✓	✓	✓	✓	✓
AW0VSS	—	GND for AWO area												✓	✓	✓	✓	✓	✓
BVCC	—	Power for IO																✓	✓
BVSS	—	GND for IO																✓	✓
EVCC	—	Power for IO												✓	✓	✓	✓	✓	✓

Table A.1 Pin List (2/4)

Pin Name	IO	Function	Input					Output	Other			Device						
			CMOS	SIM/T1	SIM/T2	SIM/T4	TTL		output driver strength	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins
EVSS	—	GND for IO												✓	✓	✓	✓	✓
FLMD0	I	System control		✓						✓*5	✓*5			✓	✓	✓	✓	✓
IP0_0	I	IP0/SOSC																✓
ISOVCL	—	Power for ISO area												✓	✓	✓	✓	✓
ISOVSS	—	GND for ISO area												✓	✓	✓	✓	✓
JP0_0	IO	JTAG port (JP0)			✓	✓		Low	✓	✓	✓*1		✓	✓	✓	✓	✓	✓
JP0_1	IO	JTAG port (JP0)			✓			Low	✓	✓	✓*1		✓	✓	✓	✓	✓	✓
JP0_2	IO	JTAG port (JP0)			✓	✓		Low	✓	✓	✓*1		✓	✓	✓	✓	✓	✓
JP0_3	IO	JTAG port (JP0)			✓	✓		Low	✓	✓	✓*1		✓	✓	✓	✓	✓	✓
JP0_4	IO	JTAG port (JP0)			✓			Low	✓	✓	✓*1		✓	✓	✓	✓	✓	✓
JP0_5	IO	JTAG port (JP0)			✓			Low	✓	✓	✓*1		✓	✓	✓	✓	✓	✓
JP0_6	IO	JTAG port (JP0)			✓			Low	✓	✓	✓*1						✓	✓
P0_0	IO	Digital port (P0)	✓		✓			Low	✓	✓	*2		✓	✓	✓	✓	✓	✓
P0_1	IO	Digital port (P0)	✓		✓			Low	✓	✓	✓		✓	✓	✓	✓	✓	✓
P0_2	IO	Digital port (P0)	✓		✓			Low / Fast *7	✓	✓	✓		✓*10	✓	✓	✓	✓	✓
P0_3	IO	Digital port (P0)	✓		✓			Low / Fast *7	✓	✓	✓		✓	✓	✓	✓	✓	✓
P0_4	IO	Digital port (P0)	✓		✓			Low	✓	✓	✓		✓	✓	✓	✓	✓	✓
P0_5	IO	Digital port (P0)	✓		✓			Low / Fast*9	✓	✓	✓		✓	✓	✓	✓	✓	✓
P0_6	IO	Digital port (P0)	✓		✓			Low / Fast*9	✓	✓	✓		✓	✓	✓	✓	✓	✓
P0_7	IO	Digital port (P0)	✓		✓			Low / Fast	✓	✓	✓				✓	✓	✓	✓
P0_8	IO	Digital port (P0)			✓			Low	✓	✓	✓			✓	✓	✓	✓	✓
P0_9	IO	Digital port (P0)	✓		✓			Low	✓	✓	✓		✓*10	✓	✓	✓	✓	✓
P0_10	IO	Digital port (P0)			✓			Low	✓	✓	✓			✓	✓	✓	✓	✓
P0_11	IO	Digital port (P0)	✓		✓			Low	✓	✓	✓			✓	✓	✓	✓	✓
P0_12	IO	Digital port (P0)	✓		✓			Low	✓	✓	✓			✓	✓	✓	✓	✓
P0_13	IO	Digital port (P0)	✓		✓			Low / Fast	✓		✓				✓	✓	✓	✓
P0_14	IO	Digital port (P0)			✓			Low / Fast	✓		✓				✓	✓	✓	✓
P1_0	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_1	IO	Digital port (P1)			✓			Low	✓		✓					✓	✓	✓
P1_2	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_3	IO	Digital port (P1)			✓			Low	✓		✓					✓	✓	✓
P1_4	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_5	IO	Digital port (P1)			✓			Low	✓		✓					✓	✓	✓
P1_6	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_7	IO	Digital port (P1)			✓			Low	✓		✓					✓	✓	✓
P1_8	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_9	IO	Digital port (P1)			✓			Low	✓		✓					✓	✓	✓
P1_10	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_11	IO	Digital port (P1)			✓			Low	✓		✓					✓	✓	✓
P1_12	IO	Digital port (P1)	✓		✓			Low	✓		✓					✓	✓	✓
P1_13	IO	Digital port (P1)			✓			Low	✓		✓						✓	✓
P1_14	IO	Digital port (P1)	✓		✓			Low	✓		✓							✓
P1_15	IO	Digital port (P1)			✓			Low	✓		✓							✓
P2_0	IO	Digital port (P2)	✓		✓			Low	✓		✓							✓
P2_1	IO	Digital port (P2)			✓			Low	✓		✓							✓

Table A.1 Pin List (3/4)

Pin Name	IO	Function	Input					Output	Other			Device						
			CMOS	SHMT1	SHMT2	SHMT4	TTL		output driver strength	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins
P2_2	IO	Digital port (P2)		✓		✓		Low	✓		✓							✓
P2_3	IO	Digital port (P2)				✓		Low	✓		✓							✓
P2_4	IO	Digital port (P2)	✓		✓			Low	✓		✓							✓
P2_5	IO	Digital port (P2)			✓			Low	✓		✓							✓
P2_6	IO	Digital port (P2)			✓			Low	✓		✓							✓
P8_0	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_1	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_2	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_3	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_4	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_5	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_6	IO	Digital port (P8)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓		✓	✓	✓	✓	✓	✓
P8_7	IO	Digital port (P8)			✓		✓	Low	✓	*4	✓			✓	✓	✓	✓	✓
P8_8	IO	Digital port (P8)			✓		✓	Low	✓	*4	✓			✓	✓	✓	✓	✓
P8_9	IO	Digital port (P8)			✓		✓	Low	✓	*4	✓			✓	✓	✓	✓	✓
P8_10	IO	Digital port (P8)			✓		✓	Low	✓	*4	✓				✓	✓	✓	✓
P8_11	IO	Digital port (P8)			✓		✓	Low	✓	*4	✓				✓	✓	✓	✓
P8_12	IO	Digital port (P8)			✓		✓	Low	✓	*4	✓				✓	✓	✓	✓
P9_0	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P9_1	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P9_2	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P9_3	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P9_4	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P9_5	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P9_6	IO	Digital port (P9)			✓		✓	Low	✓	✓ <sup>*4</sup>	✓	✓	✓	✓	✓	✓	✓	✓
P10_0	IO	Digital port (P10)		✓		✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_1	IO	Digital port (P10)				✓		Low / Fast <sup>*9</sup>	✓	✓	*6	✓	✓	✓	✓	✓	✓	✓
P10_2	IO	Digital port (P10)	✓		✓			Low / Fast <sup>*9</sup>	✓	✓	*6	✓	✓	✓	✓	✓	✓	✓
P10_3	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_4	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_5	IO	Digital port (P10)				✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_6	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓ <sup>*10</sup>	✓	✓	✓
P10_7	IO	Digital port (P10)				✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_8	IO	Digital port (P10)				✓		Low / Fast	✓	✓	*3	✓	✓	✓	✓	✓	✓	✓
P10_9	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_10	IO	Digital port (P10)				✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_11	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_12	IO	Digital port (P10)				✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_13	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓ <sup>*10</sup>	✓	✓	✓
P10_14	IO	Digital port (P10)				✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P10_15	IO	Digital port (P10)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓ <sup>*10</sup>	✓	✓	✓
P11_0	IO	Digital port (P11)				✓		Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P11_1	IO	Digital port (P11)	✓		✓			Low / Fast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P11_2	IO	Digital port (P11)				✓		Low / Fast <sup>*9</sup>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P11_3	IO	Digital port (P11)	✓		✓			Low / Fast <sup>*9</sup>	✓	✓	✓	✓	✓	✓	✓ <sup>*10</sup>	✓ <sup>*11</sup>	✓	✓

Table A.1 Pin List (4/4)

Pin Name	IO	Function	Input						Output	Other				Device				
			CMOS	SIM1	SIM2	SIM4	TTL	Analog		Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins
P11_4	IO	Digital port (P11)				✓			Low / Fast	✓	✓	✓	✓		✓	✓	✓	✓
P11_5	IO	Digital port (P11)		✓		✓			Low / Fast	✓		✓	✓			✓*11	✓*12	✓
P11_6	IO	Digital port (P11)		✓		✓			Low / Fast*9	✓		✓	✓			✓	✓	✓
P11_7	IO	Digital port (P11)				✓			Low / Fast*9	✓		✓	✓			✓	✓	✓
P11_8	IO	Digital port (P11)				✓			Low / Fast	✓		✓	✓			✓	✓	✓
P11_9	IO	Digital port (P11)		✓		✓			Low / Fast	✓		✓	✓			✓	✓	✓
P11_10	IO	Digital port (P11)				✓			Low / Fast	✓		✓	✓			✓	✓	✓
P11_11	IO	Digital port (P11)				✓			Low / Fast	✓		✓	✓			✓	✓	✓
P11_12	IO	Digital port (P11)		✓		✓			Low	✓		✓	✓			✓	✓	✓
P11_13	IO	Digital port (P11)		✓		✓			Low / Fast	✓		✓	✓			✓	✓	✓
P11_14	IO	Digital port (P11)				✓			Low / Fast	✓		✓	✓			✓	✓	✓
P11_15	IO	Digital port (P11)		✓		✓			Low / Fast	✓		✓	✓			✓	✓	✓
P12_0	IO	Digital port (P12)				✓			Low / Fast	✓		✓	✓			✓	✓	✓
P12_1	IO	Digital port (P12)		✓		✓			Low / Fast	✓		✓	✓			✓	✓	✓
P12_2	IO	Digital port (P12)				✓			Low / Fast	✓		✓	✓			✓	✓	✓
P12_3	IO	Digital port (P12)		✓		✓			Low	✓		✓	✓					✓
P12_4	IO	Digital port (P12)				✓			Low	✓		✓	✓					✓
P12_5	IO	Digital port (P12)				✓			Low	✓		✓	✓					✓
P18_0	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓			✓	✓	✓
P18_1	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓			✓	✓	✓
P18_2	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓			✓	✓	✓
P18_3	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓			✓	✓	✓
P18_4	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓					✓
P18_5	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓					✓
P18_6	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓					✓
P18_7	IO	Digital port (P18)				✓		✓	Low	✓	*4	✓	✓					✓
P20_0	IO	Digital port (P20)		✓		✓			Low	✓		✓	✓					✓
P20_1	IO	Digital port (P20)				✓			Low	✓		✓	✓					✓
P20_2	IO	Digital port (P20)		✓		✓			Low	✓		✓	✓					✓
P20_3	IO	Digital port (P20)				✓			Low	✓		✓	✓					✓
P20_4	IO	Digital port (P20)		✓		✓			Low	✓		✓	✓			✓	✓	✓
P20_5	IO	Digital port (P20)				✓			Low	✓		✓	✓			✓	✓	✓
REGVCC	—	Power for VREG												✓	✓	✓	✓	✓
X1	—	MOSC												✓	✓	✓	✓	✓
X2	—	MOSC												✓	✓	✓	✓	✓
XT1	—	SOSC														✓	✓	✓

Note 1. During on-chip debug, this pin does not become Hi-Z.

During LPD mode has a different state in the 4 pin mode or 1 pin mode. For details, see **Table 2.68**.

Note 2. Low level output. This pin does not become Hi-Z.

Note 3. During serial programming mode and boundary scan mode, this pin is used as FLMD1 mode pin.

Note 4. Pull-down resistor for ADC diagnostic purpose. Control via ADCAnPDCTL1, ADCAnPDCTL2 registers.

Note 5. Pull-up/Pull-down resistor for flash memory control. Control via FLMDCNT register.

Note 6. Hi-Z state is not used in boundary scan mode.

Note 7. Set fast mode if the load capacitance of CSIH is 100 pF.

Note 8. See data sheet for input characteristics of IP0\_0.

Note 9. Set fast mode if the load capacitance of CSIH is 50 pF.

Note 10. SHMT1 is not supported.

Note 11. SHMT1 is supported in the Geteway products only.

Note 12. SHMT1 is supported in the devices with 1.5- and 2-MB code flash memories only.

---

RH850/F1L User's Manual: Hardware

Publication Date:    Rev.0.11    Oct 05, 2012  
                            Rev.1.32    Dec 18, 2015

Published by:        Renesas Electronics Corporation

---



## SALES OFFICES

## Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

### California Eastern Laboratories, Inc.

4590 Patrick Henry Drive, Santa Clara, California 95054-1817, U.S.A.  
Tel: +1-408-919-2500, Fax: +1-408-988-0279

### Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

### Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

### Renesas Electronics (China) Co., Ltd.

Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

### Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

### Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F, Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2886-9022

### Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

### Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300

### Renesas Electronics Malaysia Sdn.Bhd.

Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

### Renesas Electronics India Pvt. Ltd.

No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India  
Tel: +91-80-67208700, Fax: +91-80-67208777

### Renesas Electronics Korea Co., Ltd.

12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RH850/F1L



Renesas Electronics Corporation