Document number: MC68HC16Y5TSJ / D

MC68HC (9) 16Y5 / 916Y6_TSJ / PDF

Rev: 1.1

Aug. 2000

Technical summary
16-BIT MICROCONTROLLER UNIT

M68HC16 Family

MC68HC (9) 16Y5
MC68HC916Y6

Page 5

Chapter 1
First

The MC68HC (9) 16Y5 / 916Y6 has a built-in CPU16 that is upwardly compatible with the M68HC11 core at the code level 16 A bit of a single-chip microcontroller unit (MCU). This MCU is M68300 / 68HC16 A member of the modular microcontroller unit family.

The MCU is a 16-bit CPU (CPU16), streamline low power integration module. Time Processing Unit (TPU2) with SLIM, 4K Byte TPU Flash EEPROM (TPUFLASH) Enhanced version of, Configurable Timer Module (CTM3), Queued Analog-De Gital converter (QADC), RAM buffered SPI (RSPI), multi-channel communication interface (MCCI), Auxiliary Chip Select (ACS), 164K Byte Flash EEPROM (FLASH EEPROM Version) / 160K Byte ROM (MASK) ROM version) and 7K bytes standby RAM (SRAM / RSPI-RAM) are built-in.

All external bus cycles are handled by SLIM. SLIM is a 512-byte boot ROM, period It has an interrupt timer, a watchdog timer, and three chip select. ACS Mod Will provide three more chip select signals for external devices.

The TPU2 and CTM3 timer modules provide a total of 34 timer channels. TPU2 is a microphone An intelligent timer that allows you to program functions with local code. CTM3 channel is the CPU operation You can handle double events on each channel without interruption.

Two serial peripheral interfaces (SPIs) and two asynchronous communication inputs to RSPI and MCCI Surface (SCI) is implemented. RSPI synchronizes 3K bytes of SRAM data at high speed. Transfer while maintaining high data reliability through the al interface. External device is RSPI You can update the internal pointer of to specify the data block to transfer.

The QADC has 44 queue pointers and 44 channel control words. There is a result register. The data converted from 16 channels is from 44 result registers. You can transfer to either. With this method, the CPU in real-time applications Interrupts are also significantly reduced.

Due to the use of flash EEPROM and TPU flash EEPROM, electrical erasure and programming operation You can execute the work as many times as you like. Users can use these ROMs to specify during system development and manufacturing. Update the contents of the program at any time, as long as the maximum number of erases and the program cycle is not exceeded. Can be

The M68HC16 device has a built-in PLL with a reference frequency of 4-8MHz (fast reference mode) or 32.768kHz. It is possible to operate at the reference frequency (low speed reference mode). For system hardware and software Therefore, the clock speed can be changed during operation.

The information contained in this document is for new products. Specifications and information change without notice You may.

In this document, the Microcontroller Unit is abbreviated as MCU below. And the Serial Peripheral Interface is abbreviated as SPI.

To do. IMB Is a registered trademark of Motorola Corporation.

Page 9

Table 1 Order information

product name	Package type	Frequency (MHz)	Temperature	order number
MC68HC16Y5	Plastic surface mount	20.00	-40 ~ + 105 °C	MC68HC16Y5VFT20
		25.00	-40 ~ + 105 °C	MC68HC16Y5VFT25
MC68HC916Y5	Plastic surface mount	20.00	-40 ~ + 105 °C	MC68HC916Y5VFT20
		25.00	-40 ~ + 105 °C	MC68HC916Y5VFT25
MC68HC916Y6	Plastic surface mount	20.00	-40 ~ + 105 °C	MC68HC916Y6VFT20
		25.00	-40 ~ + 105 °C	MC68HC916Y6VFT25

□ Reference materials

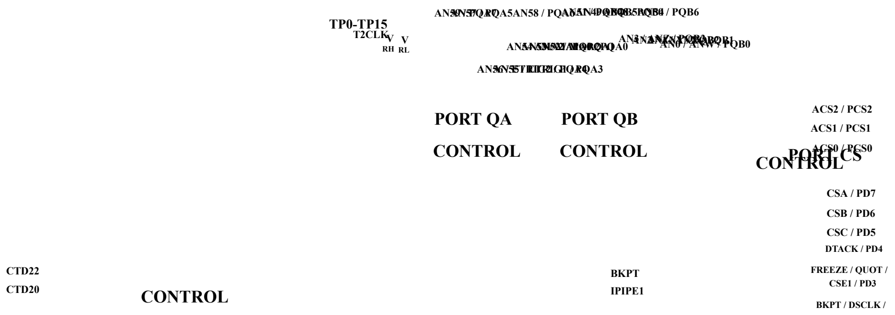
CPU16 Central Processor Unit Reference Manual (CPU16RMJ / AD)
TPU Reference Manual (TPURMJ / AD)
CTM User's Manual (CTMRMJ / D)
QADC Reference Manual (QADCRMJ / AD)
MCCI Reference Manual (MCCIRM / AD)

Page 10

□ Features

· Central processing unit (CPU16)
--16-bit architecture
--Full set of 16-bit instructions
--Three 16-bit index registers

--Two 16-bit accumulators
--Digital signal processing function mainly for control
--High-level language support
--High-speed interrupt response time
--Background debug mode
-Streamline Low Power Integration Module (SLIM)
--Single chip or multiplexed external bus support
--Three programmable chip select outputs
--512 byte boot ROM
--System protection logic (improved oscillator loss protection, watchdog timer, and bus mode) Including Nita)
--Automatic configuration from shadow register
--High-speed reference mode with on-chip phase-locked loop (PLL) for system clock support
・Auxiliary chip select (ACS)
--Three additional chip select for 16-bit non-multiplexed bus
• Time Processor Unit 2 (TPU2) with 4K Byte Flash EEPROM (TPUFLASH)
—— 16 channels
--Each channel can perform a time function stored in 4K bytes of ∝χοδε ROM.
--TPUFLASH stores the ∝ code for debugging the timer function at the time of development.
・Configurable timer module 3 (CTM3)
--Four Modular Counter Submodules (MCSM)
--Five PWM submodules (PWMSM)
--13 Double Action Submodules (DASM)
--External clock input
• Queued analog-to-digital converter (QADC)
--16 analog channels with 2 external triggers
--44 result registers controlled by two conversion command queues
• Serial Peripheral Interface with RAM Buffer (RSPI)
--3K bytes RAM buffer
--Start / End / Current data pointers for sending and receiving
--CRC or horizontal parity check for block data transfer
・Serial I / O module (MCCI)
--2-channel high performance SCI (UART)
--1-channel SPI
• 160K bytes ・ Mask ROM (64K bytes x 2 / 32K bytes configuration): MC68HC16Y5 only
• 160K bytes ・ Flash EEPROM (32K bytes x 5 configuration): MC68HC916Y5 only
• 160K bytes ・ Flash EEPROM (64K bytes x 2 / 32K bytes configuration): MC68HC916Y6 only
・4K bytes TPU Flash EEPROM (TPU FLASH): MC68HC916Y5 / MC68HC916Y6
--Erasable for each 1K byte block
--Can be used for TPU2 ∝ code ROM emulation or normal operation
• 7K bytes · Standby RAM (4K bytes for SRAM / 3K bytes for RSPI RAM)
--Each can be mapped to any 4K byte boundary
--External standby voltage supply and power down status flag
--3K bytes of RSPI SRAM that can be accessed directly from RSPI



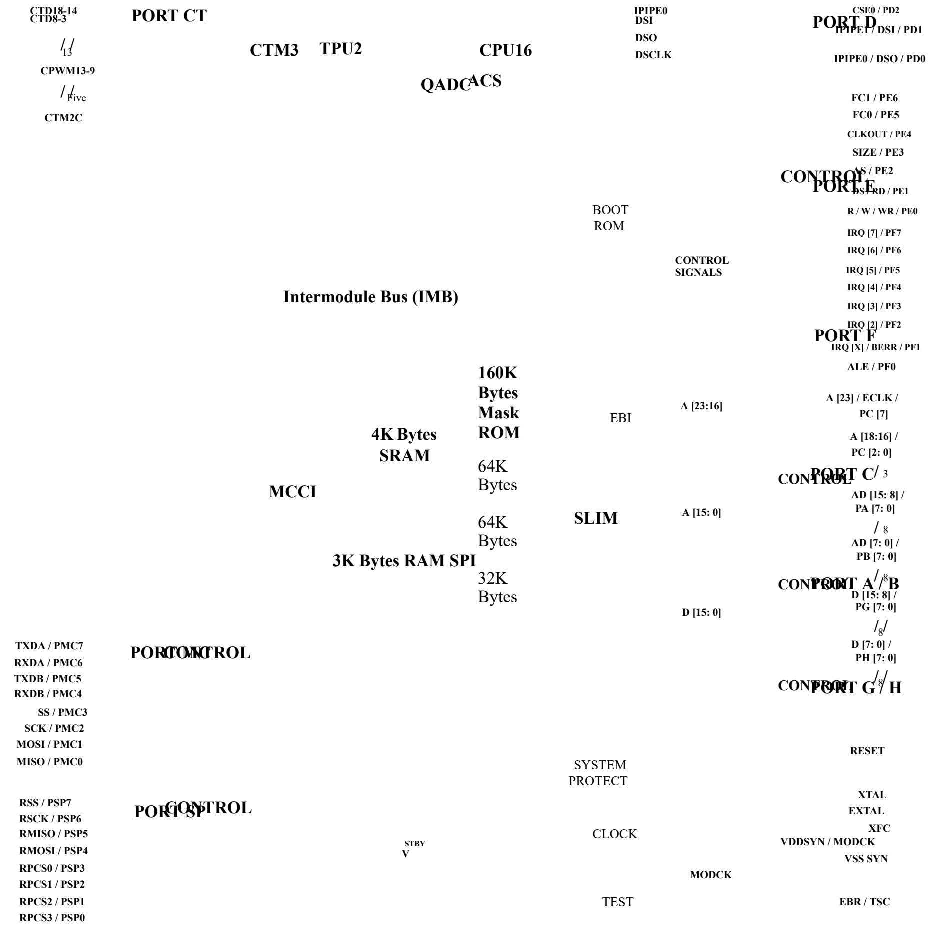
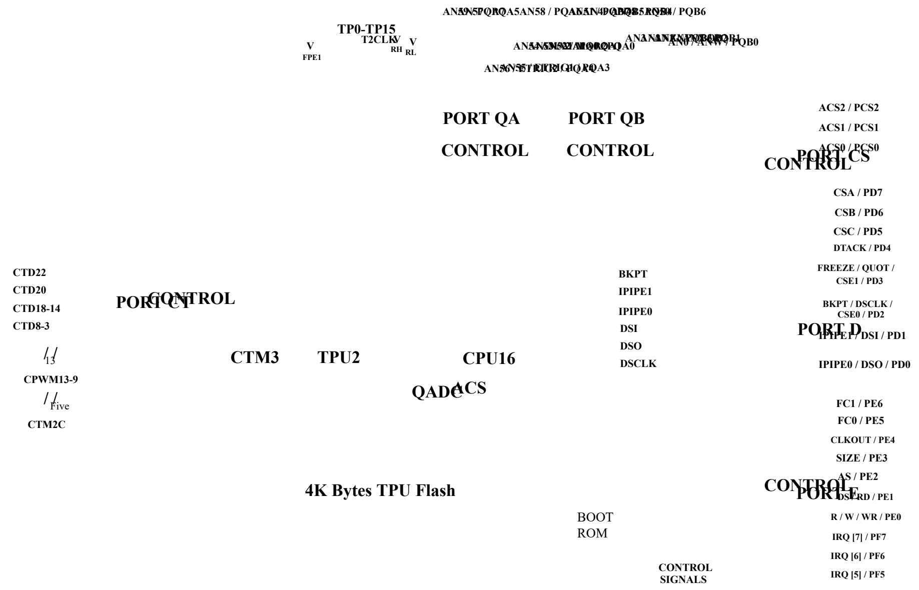


Figure 1 (a) MC68HC16Y5



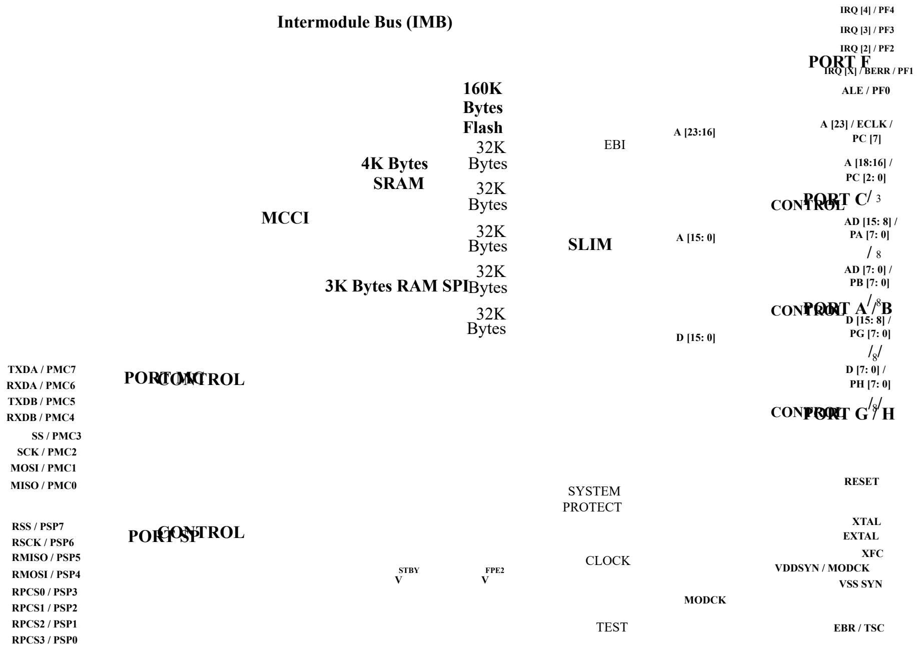
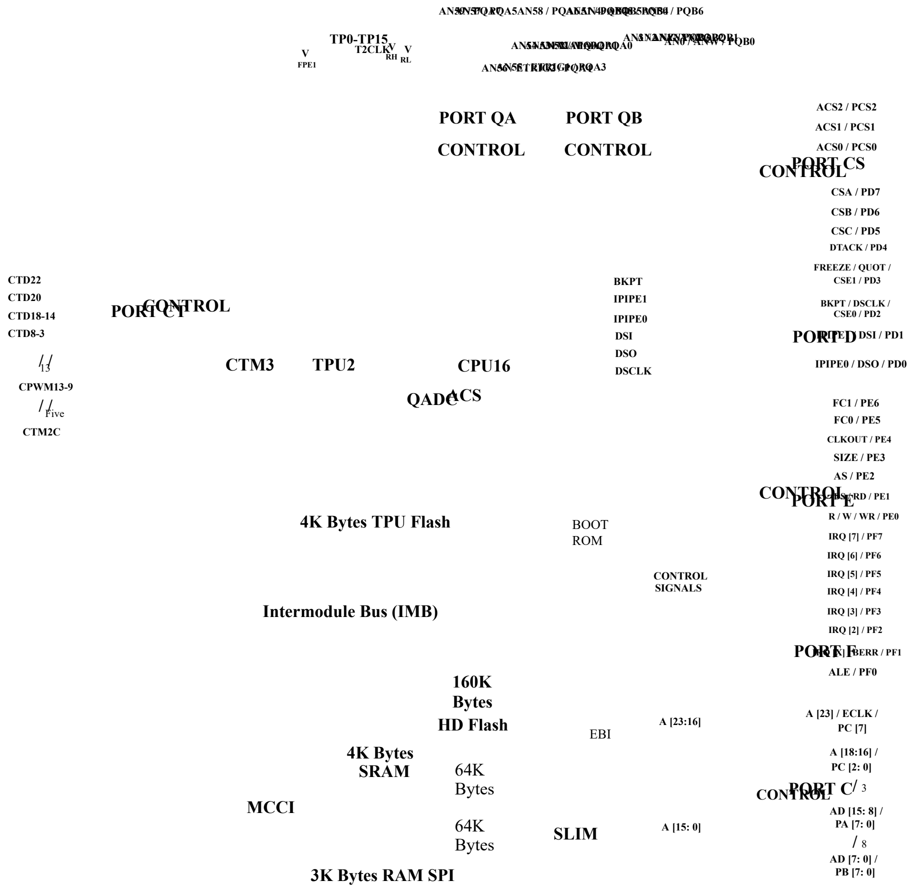


Figure 1 (b) MC68HC916Y5



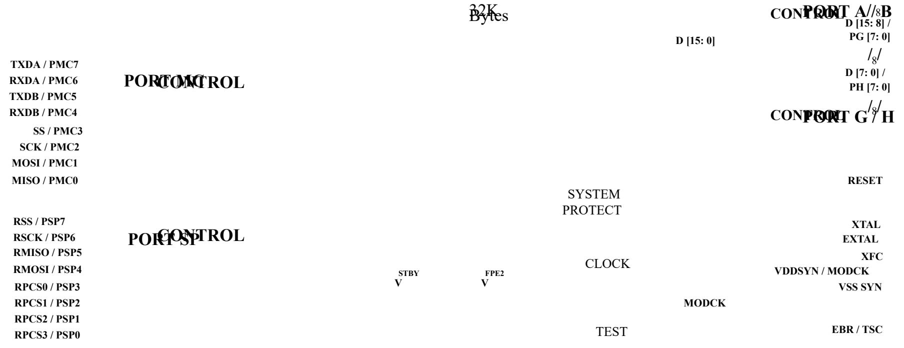
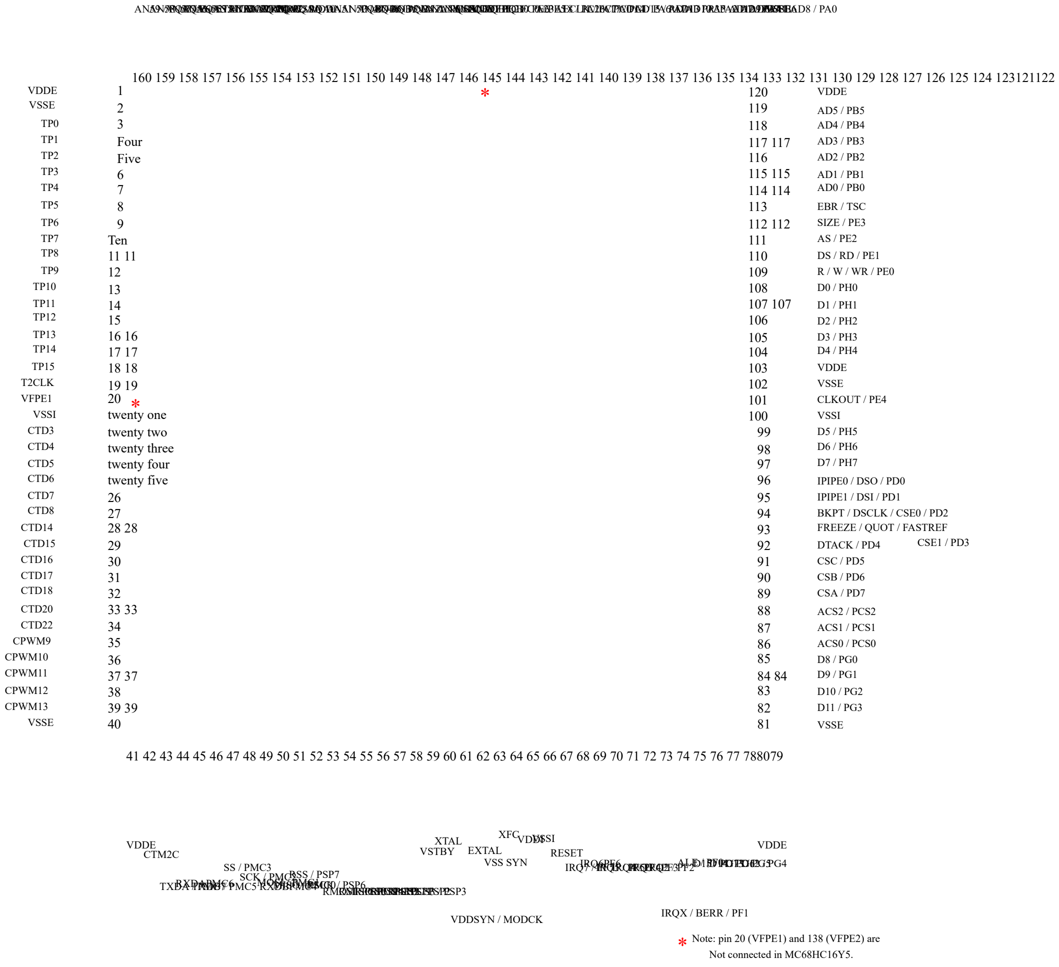


Figure 1 (c) MC68HC916Y6

Page 14



* Note: pin 20 (VFPE1) and 138 (VFPE2) are
Not connected in MC68HC16Y5.

Figure 2 MC68HC (9) 16Y5 / 916Y6 160-pin QFP pinout

1.1 Pin description

The following table shows the characteristics of the MCU pins. All inputs use the CMOS logic level. also,
All outputs can be placed in a high impedance state, but the way to do this is with a pin machine.
It depends on the ability. See the Driver Types table for a description of the output drivers. Pin special
The discrete I / O column of the sex table indicates if the pin has another function. Port name is applicable
Only the ones are listed. See the MCU block diagram for port configurations.

Table 2 Pin characteristics

Pin mnemonics	Output driver	Input synchronization	Input Hysteresis	Discrete I / O	Port name
A23 / ECLK	Aw	Note 1	Y	I / O	PC7
A [18:16]	Aw	Note 1	Y	I / O	PC [2: 0]
ACS [2: 0]	A	————	————	O	PCS [2: 0]
AD [15: 8]	Aw	Note 1	Y	I / O	PA [7: 0]
AD [7: 0]	Aw	Note 1	Y	I / O	PB [7: 0]
ALE	A	Note 1	N	I / O	PF0
AN [3: 0] / AN [Z, Y, X, W]	————	Y	Y	I	PQB [3: 0]
AN [51:48]	————	Y	Y	I	PQB [7: 4]
AN [54:52]	A	Y	Y	I / O	PQA [2: 0]
AN [59:55]	Ba	Y	Y	I / O	PQA [7: 3]
AS	Bw	Note 1	Y	I / O	PE2
BKPT / DSCLK / CSE0	Aw	Note 1	Y	I / O	PD2
CLKOUT	A	Note 1	N	I / O	PE4
CSA	Aw	Note 1	N	I / O	PD7
CSB	B	Note 1	N	I / O	PD6
CSC	Aw	Note 1	N	I / O	PD5
CPWM [13: 9]	A	————	————	O	————
CTD [22] / [20] / [18:14] / [8: 3]	A	Y	Y	I / O	————
CTM2C	————	Y	Y	I	————
D [15: 8]	A	Note 1	Y	I / O	PG [7: 0]
D [7: 0]	A	Note 1	Y	I / O	PH [7: 0]
DS / RD	Bw	Note 1	Y	I / O	PE1
DTACK	B	Note 1	Y	I / O	PD4
EBR / TSC	————	Y 3	Y	————	————
EXTAL 2	————	————	Note 2	————	————
FC [1: 0]	Aw	Note 1	Y	I / O	PE [6: 5]
FREEZE / QUOT / CSE1	Aw	Note 1	N	I / O	PD3

Page 16

Table 2 Pin characteristics (continued)

Pin mnemonics	Output driver	Input synchronization	Input Hysteresis	Discrete I / O	Port name
IPIPE1 / DSI	Aw	Y	Y	I / O	PD1
IPIPE0 / DSO	A	————	Y	I / O	PD0
IRQX / BERR	Bw	Y	Y	I / O	PF1
IRQ [7: 2]	Bw	Y	Y	I / O	PF [7: 2]
MISO	Bo	Note 1	Y	I / O	PMC0
MOSI	Bo	Note 1	Y	I / O	PMC1
RESET	Bo	Y	Y	————	————
RMISO	Bo	Note 1	Y	I / O	PSP5
RMOSI	Bo	Note 1	Y	I / O	PSP4
RPCS [3: 0]	Bo	Y	Y	I / O	PSP [3: 0]
RSCK	Bo	Note 1	Y	I / O	PSP6
RSS	Bo	Note 1	Y	I / O	PSP7
R / W / WR	Bw	Note 1	Y	I / O	PE0
RXDA	Bo	Y	Y	I / O	PMC6
RXDB	Bo	Y	Y	I / O	PMC4
SCK	Bo	Note 1	Y	I / O	PMC2
SIZE	Bw	————	Y	I / O	PE3
SS	Bo	Note 1	Y	I / O	PMC3
T2CLK	A	Y	Y	————	————
TP [15: 0]	A	Y	Y	————	————
TXDA	Bo	Y	Y	I / O	PMC7
TXDB	Bo	Y	Y	I / O	PMC5
XFC	————	————	————	————	————
XTAL	————	————	————	————	————

Note: 1. Synchronous inputs are used only when configured as digital I / O pins.
2. Hysteresis is given when the PLL is enabled.
3. 3. Use asynchronous input for the External Bus Request (EBR) feature.

Page 17

Table 3 pin conditions

Table 3 (a) Single chip mode pin conditions

pin	Pin name	Single chip mode (DSI = 1, MCRM [11: 9] =% 111)	
		Condition during reset	Condition after reset
1	VDDE	(Power)	
2	VSSE	(GND)	
3	TP0	Hi Z (disable)	Hi Z (for input)
Four	TP1	Hi Z (disable)	Hi Z (for input)
Five	TP2	Hi Z (disable)	Hi Z (for input)
6	TP3	Hi Z (disable)	Hi Z (for input)
7	TP4	Hi Z (disable)	Hi Z (for input)
8	TP5	Hi Z (disable)	Hi Z (for input)
9	TP6	Hi Z (disable)	Hi Z (for input)
Ten	TP7	Hi Z (disable)	Hi Z (for input)
11 11	TP8	Hi Z (disable)	Hi Z (for input)
12	TP9	Hi Z (disable)	Hi Z (for input)
13	TP10	Hi Z (disable)	Hi Z (for input)
14	TP11	Hi Z (disable)	Hi Z (for input)
15	TP12	Hi Z (disable)	Hi Z (for input)
16 16	TP13	Hi Z (disable)	Hi Z (for input)
17 17	TP14	Hi Z (disable)	Hi Z (for input)
18 18	TP15	Hi Z (disable)	Hi Z (for input)
19 19	T2CLK	Hi Z (disable)	Hi Z (for input)
20	VFPE1	(Power)	
twenty one	VSSI	(GND)	
twenty two	CTD3	Hi Z (disable)	Hi Z (for input)
twenty three	CTD4	Hi Z (disable)	Hi Z (for input)
twenty four	CTD5	Hi Z (disable)	Hi Z (for input)
twenty five	CTD6	Hi Z (disable)	Hi Z (for input)
26	CTD7	Hi Z (disable)	Hi Z (for input)
27	CTD8	Hi Z (disable)	Hi Z (for input)
28 28	CTD14	Hi Z (disable)	Hi Z (for input)
29	CTD15	Hi Z (disable)	Hi Z (for input)
30	CTD16	Hi Z (disable)	Hi Z (for input)
31	CTD17	Hi Z (disable)	Hi Z (for input)
32	CTD18	Hi Z (disable)	Hi Z (for input)
33 33	CTD20	Hi Z (disable)	Hi Z (for input)
34	CTD22	Hi Z (disable)	Hi Z (for input)
35	CPWM9	Hi Z (output only)	Low drive
36	CPWM10	Hi Z (output only)	Low drive
37 37	CPWM11	Hi Z (output only)	Low drive
38	CPWM12	Hi Z (output only)	Low drive
39 39	CPWM13	Hi Z (output only)	Low drive
40	VSSE	(GND)	
41 41	VDDE	(Power)	
42	CTM2C	Hi Z (disable)	Hi Z (for input)
43	TXDA / PMC7	Hi Z (disable)	Hi Z (for input port)
44	RXDA / PMC6	Hi Z (disable)	Hi Z (for input port)
45 45	TXDB / PMC5	Hi Z (disable)	Hi Z (for input port)
46	RDXB / PMC4	Hi Z (disable)	Hi Z (for input port)

Table 3 (a) Single chip mode pin conditions (continued)

pin	Pin name	Single chip mode (DSI = 1, MCRM [11: 9] =% 111)	
		Condition during reset	Condition after reset
47 47	SS / PMC3	Hi Z (disable)	Hi Z (for input port)
48	SCK / PMC2	Hi Z (disable)	Hi Z (for input port)
49	MOSI / PMC1	Hi Z (disable)	Hi Z (for input port)
50	MISO / PMC0	Hi Z (disable)	Hi Z (for input port)
51	RSS / PSP7	Hi Z (disable)	Hi Z (for input port)
52 52	RSCK / PSP6	Hi Z (disable)	Hi Z (for input port)
53	RMOSI / PSP5	Hi Z (disable)	Hi Z (for input port)
54	RMISO / PSP4	Hi Z (disable)	Hi Z (for input port)
55 55	RPCS0 / PSP0	Hi Z (disable)	Hi Z (for input port)
56	RPCS1 / PSP1	Hi Z (disable)	Hi Z (for input port)
57 57	RPCS2 / PSP2	Hi Z (disable)	Hi Z (for input port)

58	RPCS3 / PSP3	Hi Z (disable)	Hi Z (for input port)
59	VSTBY	(Power)	
60	XTAL	(Clock)	
61	VDDSYN / MODCK	(Power)	
62	EXTAL	(Clock)	
63 63	VSS SYN	(GND)	
64	XFC	(Clock filter)	
65 65	VDDI	(Power)	
66 66	VSSI	(GND)	
67 67	RESET	(RESET)	
68 68	IRQ7 / PF7	Pull up	Hi Z (for IRQ)
69	IRQ6 / PF6	Pull up	Hi Z (for IRQ)
70	IRQ5 / PF5	Pull up	Hi Z (for IRQ)
71 71	IRQ4 / PF4	Pull up	Hi Z (for IRQ)
72	IRQ3 / PF3	Pull up	Hi Z (for IRQ)
73	IRQ2 / PF2	Pull up	Hi Z (for IRQ)
74 74	IRQX / BERR / PF1	Pull up	Hi Z (for IRQ)
75	ALE / PF0	Hi Z (for input)	Hi Z (for input port)
76 76	D15 / PG7	Hi Z (for input)	Hi Z (for input port)
77 77	D14 / PG6	Hi Z (for input)	Hi Z (for input port)
78 78	D13 / PG5	Hi Z (for input)	Hi Z (for input port)
79 79	D12 / PG4	Hi Z (for input)	Hi Z (for input port)
80	VDDE	(Power)	
81	VSSE	(GND)	
82	D11 / PG3	Hi Z (for input)	Hi Z (for input port)
83	D10 / PG2	Hi Z (for input)	Hi Z (for input port)
84 84	D9 / PG1	Hi Z (for input)	Hi Z (for input port)
85	D8 / PG0	Hi Z (for input)	Hi Z (for input port)
86	ACS0 / PCS0	High drive	Drive for CS
87	ACS1 / PCS1	High drive	Drive for CS
88	ACS2 / PCS2	High drive	Drive for CS
89	CSA / PD7	Pull up	Hi Z (for input port)
90	CSB / PD6	Hi Z (for input)	Hi Z (for input port)
91	CSC / PD5	Pull up	Hi Z (for input port)
92	DTACK / PD4	Hi Z (for input)	Hi Z (for input port)
93	FREEZE / QUOT / FASTREF / CSE1 / PD3	Pull up (if BKPT = 1) Low drive (if BKPT = 0)	Hi Z (for input port) (if BKPT = 1) Drive for FREEZE (if BKPT = 0)

Page 19

Table 3 (a) Single chip mode pin conditions (continued)

pin	Pin name	Single chip mode (DSI = 1, MCRM [11: 9] =% 111)	
		Condition during reset	Condition after reset
94	BKPT / DSCLK / CSE0 / PD2	Pull up	Hi Z (for input port) (if BKPT = 1) Hi Z (for DSCLK) (if BKPT = 0)
95	IPIPE1 / DSI / PD1	Pull up	Hi Z (for input port) (if BKPT = 1) Hi Z (for DSI) (if BKPT = 0)
96	IPIPE0 / DSO / PD0	Low drive	High drive (output port) (if BKPT = 1) Drive for DSO (if BKPT = 0)
97	D7 / PH7	Hi Z (for input)	Hi Z (for input port)
98	D6 / PH6	Hi Z (for input)	Hi Z (for input port)
99	D5 / PH5	Hi Z (for input)	Hi Z (for input port)
100	VSSI	(GND)	
101	CLKOUT / PE4	Hi Z (for input) (if AD0 = 0)	Hi Z (for input port) (if AD0 = 0)
102	VSSE	(GND)	
103	VDDE	(Power)	
104	D4 / PH4	Hi Z (for input)	Hi Z (for input port)
105	D3 / PH3	Hi Z (for input)	Hi Z (for input port)
106	D2 / PH2	Hi Z (for input)	Hi Z (for input port)
107 107	D1 / PH1	Hi Z (for input)	Hi Z (for input port)
108	D0 / PH0	Hi Z (for input)	Hi Z (for input port)
109	R / W / WR / PE0	Pull up	Hi Z (for input port)
110	DS / RD / PE1	Pull up	Hi Z (for input port)
111	AS / PE2	Pull up	Hi Z (for input port)
112 112	SIZE / PE3	Pull up	Hi Z (for input port)

113	EBR / TSC	Hi Z (for input)	HiZ (pull up reg. Must be needed)
114 114	A0 / AD0 / PB0	Pull up	Hi Z (for input port)
115 115	A1 / AD1 / PB1	Pull up	Hi Z (for input port)
116	A2 / AD2 / PB2	Pull up	Hi Z (for input port)
117 117	A3 / AD3 / PB3	Pull up	Hi Z (for input port)
118	A4 / AD4 / PB4	Pull up	Hi Z (for input port)
119	A5 / AD5 / PB5	Pull up	Hi Z (for input port)
120	VDDE	(Power)	
121	VSSE	(GND)	
122	A6 / AD6 / PB6	Pull up	Hi Z (for input port)
one two Three	A7 / AD7 / PB7	Pull up	Hi Z (for input port)
124	A8 / AD8 / PA0	Pull up	Hi Z (for input port)
125	A9 / AD9 / PA1	Pull up	Hi Z (for input port)
126	A10 / AD10 / PA2	Pull up	Hi Z (for input port)
127 127	A11 / AD11 / PA3	Pull up	Hi Z (for input port)
128 128	A12 / AD12 / PA4	Pull up	Hi Z (for input port)
129	A13 / AD13 / PA5	Pull up	Hi Z (for input port)
130	A14 / AD14 / PA6	Pull up	Hi Z (for input port)
131 131	A15 / AD15 / PA7	Pull up	Hi Z (for input port)
132 132	A16 / PC0	Pull up	Hi Z (for input port)
133 133	A17 / PC1	Pull up	Hi Z (for input port)
134 134	A18 / PC2	Pull up	Hi Z (for input port)
135	A23 / ECLK / PC7	Pull up	Hi Z (for input port)
136 136	FC0 / PE5	Pull up	Hi Z (for input port)
137	FC1 / PE6	Pull up	Hi Z (for input port)
138	VFPE2	(Power)	
139	VDDI	(Power)	

Page 20

Table 3 (a) Single chip mode pin conditions (continued)

pin	Pin name	Single chip mode (DSI = 1, MCRM [11: 9] =% 111)	
		Condition during reset	Condition after reset
140	VSSI	(GND)	
141	AN0 / ANW / PQB0	Hi Z (disable)	Hi Z (for input port)
142	AN1 / ANX / PQB1	Hi Z (disable)	Hi Z (for input port)
143	AN2 / ANY / PQB2	Hi Z (disable)	Hi Z (for input port)
144	AN3 / ANZ / PQB3	Hi Z (disable)	Hi Z (for input port)
145	AN48 / PQB4	Hi Z (disable)	Hi Z (for input port)
146	AN49 / PQB5	Hi Z (disable)	Hi Z (for input port)
147	AN50 / PQB6	Hi Z (disable)	Hi Z (for input port)
148	AN51 / PQB7	Hi Z (disable)	Hi Z (for input port)
149	VDDA	(Power)	
150	VSSA	(GND)	
151 151	VRL	(A / D ref. Volt)	
152	VRH	(A / D ref. Volt)	
153	AN52 / MA0 / PQA0	Hi Z (disable)	Hi Z (for input port)
154	AN53 / MA1 / PQA1	Hi Z (disable)	Hi Z (for input port)
155	AN54 / MA2 / PQA2	Hi Z (disable)	Hi Z (for input port)
156	AN55 / ETRIG1 / PQA3	Hi Z (disable)	Hi Z (for input port)
157	AN56 / ETRIG2 / PQA4	Hi Z (disable)	Hi Z (for input port)
158	AN57 / PQA5	Hi Z (disable)	Hi Z (for input port)
159	AN58 / PQA6	Hi Z (disable)	Hi Z (for input port)
160	AN59 / PQA7	Hi Z (disable)	Hi Z (for input port)

Table 3 (b) Single chip mode pin conditions

pin	Pin name	Single chip mode (DSI = 0, DS = 1, SIZE = 1, RW = X)	
		Condition during reset	Condition after reset
1	VDDE	(Power)	
2	VSSE	(GND)	
3	TP0	Hi Z (disable)	Hi Z (for input)
Four	TP1	Hi Z (disable)	Hi Z (for input)

Five	TP2	Hi Z (disable)	Hi Z (for input)
6	TP3	Hi Z (disable)	Hi Z (for input)
7	TP4	Hi Z (disable)	Hi Z (for input)
8	TP5	Hi Z (disable)	Hi Z (for input)
9	TP6	Hi Z (disable)	Hi Z (for input)
Ten	TP7	Hi Z (disable)	Hi Z (for input)
11 11	TP8	Hi Z (disable)	Hi Z (for input)
12	TP9	Hi Z (disable)	Hi Z (for input)
13	TP10	Hi Z (disable)	Hi Z (for input)
14	TP11	Hi Z (disable)	Hi Z (for input)
15	TP12	Hi Z (disable)	Hi Z (for input)
16 16	TP13	Hi Z (disable)	Hi Z (for input)
17 17	TP14	Hi Z (disable)	Hi Z (for input)
18 18	TP15	Hi Z (disable)	Hi Z (for input)
19 19	T2CLK	Hi Z (disable)	Hi Z (for input)
20	VFPE1	(Power)	
twenty one	VSSI	(GND)	

Page 21

Table 3 (b) Single chip mode pin conditions (continued)

pin	Pin name	Single chip mode (DSI = 0, DS = 1, SIZE = 1, RW = X)	
		Condition during reset	Condition after reset
twenty two	CTD3	Hi Z (disable)	Hi Z (for input)
twenty three	CTD4	Hi Z (disable)	Hi Z (for input)
twenty four	CTD5	Hi Z (disable)	Hi Z (for input)
twenty five	CTD6	Hi Z (disable)	Hi Z (for input)
26	CTD7	Hi Z (disable)	Hi Z (for input)
27	CTD8	Hi Z (disable)	Hi Z (for input)
28 28	CTD14	Hi Z (disable)	Hi Z (for input)
29	CTD15	Hi Z (disable)	Hi Z (for input)
30	CTD16	Hi Z (disable)	Hi Z (for input)
31	CTD17	Hi Z (disable)	Hi Z (for input)
32	CTD18	Hi Z (disable)	Hi Z (for input)
33 33	CTD20	Hi Z (disable)	Hi Z (for input)
34	CTD22	Hi Z (disable)	Hi Z (for input)
35	CPWM9	Hi Z (output only)	Low drive
36	CPWM10	Hi Z (output only)	Low drive
37 37	CPWM11	Hi Z (output only)	Low drive
38	CPWM12	Hi Z (output only)	Low drive
39 39	CPWM13	Hi Z (output only)	Low drive
40	VSSE	(GND)	
41 41	VDDE	(Power)	
42	CTM2C	Hi Z (disable)	Hi Z (for input)
43	TXDA / PMC7	Hi Z (disable)	Hi Z (for input port)
44	RXDA / PMC6	Hi Z (disable)	Hi Z (for input port)
45 45	TXDB / PMC5	Hi Z (disable)	Hi Z (for input port)
46	RDXB / PMC4	Hi Z (disable)	Hi Z (for input port)
47 47	SS / PMC3	Hi Z (disable)	Hi Z (for input port)
48	SCK / PMC2	Hi Z (disable)	Hi Z (for input port)
49	MOSI / PMC1	Hi Z (disable)	Hi Z (for input port)
50	MISO / PMC0	Hi Z (disable)	Hi Z (for input port)
51	RSS / PSP7	Hi Z (disable)	Hi Z (for input port)
52 52	RSCK / PSP6	Hi Z (disable)	Hi Z (for input port)
53	RMOSI / PSP5	Hi Z (disable)	Hi Z (for input port)
54	RMISO / PSP4	Hi Z (disable)	Hi Z (for input port)
55 55	RPCS0 / PSP0	Hi Z (disable)	Hi Z (for input port)
56	RPCS1 / PSP1	Hi Z (disable)	Hi Z (for input port)
57 57	RPCS2 / PSP2	Hi Z (disable)	Hi Z (for input port)
58	RPCS3 / PSP3	Hi Z (disable)	Hi Z (for input port)
59	VSTBY	(Power)	
60	XTAL	(Clock)	
61	VDDSYN / MODCK	(Power)	
62	EXTAL	(Clock)	
63 63	VSS SYN	(GND)	
64	XFC	(Clock filter)	
65 65	VDDI	(Power)	

66 66	VSSI	(GND)	
67 67	RESET	(RESET)	
68 68	IRQ7 / PF7	Pull up	Hi Z (for IRQ)
69	IRQ6 / PF6	Pull up	Hi Z (for IRQ)
70	IRQ5 / PF5	Pull up	Hi Z (for IRQ)
71 71	IRQ4 / PF4	Pull up	Hi Z (for IRQ)

Table 3 (b) Single chip mode pin conditions (continued)

pin	Pin name	Single chip mode (DSI = 0, DS = 1, SIZE = 1, RW = X)	
		Condition during reset	Condition after reset
72	IRQ3 / PF3	Pull up	Hi Z (for IRQ)
73	IRQ2 / PF2	Pull up	Hi Z (for IRQ)
74 74	IRQX / BERR / PF1	Pull up	Hi Z (for IRQ)
75	ALE / PF0	Hi Z (for input)	Hi Z (for input port)
76 76	D15 / PG7	Hi Z (for input)	Hi Z (for input port)
77 77	D14 / PG6	Hi Z (for input)	Hi Z (for input port)
78 78	D13 / PG5	Hi Z (for input)	Hi Z (for input port)
79 79	D12 / PG4	Hi Z (for input)	Hi Z (for input port)
80	VDDE	(Power)	
81	VSSE	(GND)	
82	D11 / PG3	Hi Z (for input)	Hi Z (for input port)
83	D10 / PG2	Hi Z (for input)	Hi Z (for input port)
84 84	D9 / PG1	Hi Z (for input)	Hi Z (for input port)
85	D8 / PG0	Hi Z (for input)	Hi Z (for input port)
86	ACS0 / PCS0	High drive	Drive for CS
87	ACS1 / PCS1	High drive	Drive for CS
88	ACS2 / PCS2	High drive	Drive for CS
89	CSA / PD7	Pull up	Hi Z (for input port)
90	CSB / PD6	Hi Z (for input)	Hi Z (for input port)
91	CSC / PD5	Pull up	Hi Z (for input port)
92	DTACK / PD4	Hi Z (for input)	Hi Z (for input port)
93	FREEZE / QUOT / FASTREF / CSE1 / PD3	Pull up (if BKPT = 1) Low drive (if BKPT = 0)	Hi Z (for input port) (if BKPT = 1) Drive for FREEZE (if BKPT = 0)
94	BKPT / DSCLK / CSE0 / PD2	Pull up	Hi Z (for input port) (if BKPT = 1) Hi Z (for DSCLK) (if BKPT = 0)
95	IPIPE1 / DSI / PD1	Pull up	Hi Z (for input port) (if BKPT = 1) Hi Z (for DSI) (if BKPT = 0)
96	IPIPE0 / DSO / PD0	Low drive	High drive (output port) (if BKPT = 1) Drive for DSO (if BKPT = 0)
97	D7 / PH7	Hi Z (for input)	Hi Z (for input port)
98	D6 / PH6	Hi Z (for input)	Hi Z (for input port)
99	D5 / PH5	Hi Z (for input)	Hi Z (for input port)
100	VSSI	(GND)	

101	CLKOUT / PE4	Drive for CLKOUT (if AD0 = 1)
		CLKOUT = XTALx2 (if BERR = 1)
		CLKOUT = XTALx1 (if BERR = 0)
		Hi Z (for input) (if AD0 = 0)
			Drive for CLKOUT (if AD0 = 1)
			CLKOUT = XTALx2 (if BERR = 1)
			CLKOUT = XTALx1 (if BERR = 0)
			Hi Z (for input port) (if AD0 = 0)

102	VSSE	(GND)	
103	VDDE	(Power)	
104	D4 / PH4	Hi Z (for input)	Hi Z (for input port)
105	D3 / PH3	Hi Z (for input)	Hi Z (for input port)
106	D2 / PH2	Hi Z (for input)	Hi Z (for input port)
107 107	D1 / PH1	Hi Z (for input)	Hi Z (for input port)
108	D0 / PH0	Hi Z (for input)	Hi Z (for input port)
109	R / W / WR / PE0	Pull up	Hi Z (for input port)
110	DS / RD / PE1	Pull up	Hi Z (for input port)
111	AS / PE2	Pull up	Hi Z (for input port)
112 112	SIZE / PE3	Pull up	Hi Z (for input port)
113	EBR / TSC	Hi Z (for input)	HiZ (pull up reg. Must be needed)
114 114	A0 / AD0 / PB0	Pull up	Hi Z (for input port)
115 115	A1 / AD1 / PB1	Pull up	Hi Z (for input port)

Table 3 (b) Single chip mode pin conditions (continued)

pin	Pin name	Single chip mode (DSI = 0, DS = 1, SIZE = 1, RW = X)	
		Condition during reset	Condition after reset
116	A2 / AD2 / PB2	Pull up	Hi Z (for input port)
117 117	A3 / AD3 / PB3	Pull up	Hi Z (for input port)
118	A4 / AD4 / PB4	Pull up	Hi Z (for input port)
119	A5 / AD5 / PB5	Pull up	Hi Z (for input port)
120	VDDE	(Power)	
121	VSSE	(GND)	
122	A6 / AD6 / PB6	Pull up	Hi Z (for input port)
one two Three	A7 / AD7 / PB7	Pull up	Hi Z (for input port)
124	A8 / AD8 / PA0	Pull up	Hi Z (for input port)
125	A9 / AD9 / PA1	Pull up	Hi Z (for input port)
126	A10 / AD10 / PA2	Pull up	Hi Z (for input port)
127 127	A11 / AD11 / PA3	Pull up	Hi Z (for input port)
128 128	A12 / AD12 / PA4	Pull up	Hi Z (for input port)
129	A13 / AD13 / PA5	Pull up	Hi Z (for input port)
130	A14 / AD14 / PA6	Pull up	Hi Z (for input port)
131 131	A15 / AD15 / PA7	Pull up	Hi Z (for input port)
132 132	A16 / PC0	Pull up	Hi Z (for input port)
133 133	A17 / PC1	Pull up	Hi Z (for input port)
134 134	A18 / PC2	Pull up	Hi Z (for input port)
135	A23 / ECLK / PC7	Pull up	Hi Z (for input port)
136 136	FC0 / PE5	Pull up	Hi Z (for input port)
137	FC1 / PE6	Pull up	Hi Z (for input port)
138	VFPE2	(Power)	
139	VDDI	(Power)	
140	VSSI	(GND)	
141	AN0 / ANW / PQB0	Hi Z (disable)	Hi Z (for input port)
142	AN1 / ANX / PQB1	Hi Z (disable)	Hi Z (for input port)
143	AN2 / ANY / PQB2	Hi Z (disable)	Hi Z (for input port)
144	AN3 / ANZ / PQB3	Hi Z (disable)	Hi Z (for input port)
145	AN48 / PQB4	Hi Z (disable)	Hi Z (for input port)
146	AN49 / PQB5	Hi Z (disable)	Hi Z (for input port)
147	AN50 / PQB6	Hi Z (disable)	Hi Z (for input port)
148	AN51 / PQB7	Hi Z (disable)	Hi Z (for input port)
149	VDDA	(Power)	
150	VSSA	(GND)	
151 151	VRL	(A / D ref. Volt)	
152	VRH	(A / D ref. Volt)	
153	AN52 / MA0 / PQA0	Hi Z (disable)	Hi Z (for input port)
154	AN53 / MA1 / PQA1	Hi Z (disable)	Hi Z (for input port)
155	AN54 / MA2 / PQA2	Hi Z (disable)	Hi Z (for input port)
156	AN55 / ETRIG1 / PQA3	Hi Z (disable)	Hi Z (for input port)
157	AN56 / ETRIG2 / PQA4	Hi Z (disable)	Hi Z (for input port)
158	AN57 / PQA5	Hi Z (disable)	Hi Z (for input port)
159	AN58 / PQA6	Hi Z (disable)	Hi Z (for input port)
160	AN59 / PQA7	Hi Z (disable)	Hi Z (for input port)

Table 3 (c) Non-multiplexed master mode pin conditions

pin	Pin name	Non-multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
1	VDDE	(Power)	
2	VSSE	(GND)	
3	TP0	Hi Z (disable)	Hi Z (for input)
Four	TP1	Hi Z (disable)	Hi Z (for input)
Five	TP2	Hi Z (disable)	Hi Z (for input)
6	TP3	Hi Z (disable)	Hi Z (for input)
7	TP4	Hi Z (disable)	Hi Z (for input)
8	TP5	Hi Z (disable)	Hi Z (for input)
9	TP6	Hi Z (disable)	Hi Z (for input)
Ten	TP7	Hi Z (disable)	Hi Z (for input)
11 11	TP8	Hi Z (disable)	Hi Z (for input)
12	TP9	Hi Z (disable)	Hi Z (for input)
13	TP10	Hi Z (disable)	Hi Z (for input)
14	TP11	Hi Z (disable)	Hi Z (for input)
15	TP12	Hi Z (disable)	Hi Z (for input)
16 16	TP13	Hi Z (disable)	Hi Z (for input)
17 17	TP14	Hi Z (disable)	Hi Z (for input)
18 18	TP15	Hi Z (disable)	Hi Z (for input)
19 19	T2CLK	Hi Z (disable)	Hi Z (for input)
20	VFPE1	(Power)	
twenty one	VSSI	(GND)	
twenty two	CTD3	Hi Z (disable)	Hi Z (for input)
twenty three	CTD4	Hi Z (disable)	Hi Z (for input)
twenty four	CTD5	Hi Z (disable)	Hi Z (for input)
twenty five	CTD6	Hi Z (disable)	Hi Z (for input)
26	CTD7	Hi Z (disable)	Hi Z (for input)
27	CTD8	Hi Z (disable)	Hi Z (for input)
28 28	CTD14	Hi Z (disable)	Hi Z (for input)
29	CTD15	Hi Z (disable)	Hi Z (for input)
30	CTD16	Hi Z (disable)	Hi Z (for input)
31	CTD17	Hi Z (disable)	Hi Z (for input)
32	CTD18	Hi Z (disable)	Hi Z (for input)
33 33	CTD20	Hi Z (disable)	Hi Z (for input)
34	CTD22	Hi Z (disable)	Hi Z (for input)
35	CPWM9	Hi Z (output only)	Low drive
36	CPWM10	Hi Z (output only)	Low drive
37 37	CPWM11	Hi Z (output only)	Low drive
38	CPWM12	Hi Z (output only)	Low drive
39 39	CPWM13	Hi Z (output only)	Low drive
40	VSSE	(GND)	
41 41	VDDE	(Power)	
42	CTM2C	Hi Z (disable)	Hi Z (for input)
43	TXDA / PMC7	Hi Z (disable)	Hi Z (for input port)
44	RXDA / PMC6	Hi Z (disable)	Hi Z (for input port)
45 45	TXDB / PMC5	Hi Z (disable)	Hi Z (for input port)
46	RDXB / PMC4	Hi Z (disable)	Hi Z (for input port)
47 47	SS / PMC3	Hi Z (disable)	Hi Z (for input port)
48	SCK / PMC2	Hi Z (disable)	Hi Z (for input port)

Table 3 (c) Non-multiplexed master mode pin conditions (continued)

pin	Pin name	Non-multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
49	MOSI / PMC1	Hi Z (disable)	Hi Z (for input port)
50	MISO / PMC0	Hi Z (disable)	Hi Z (for input port)
51	RSS / PSP7	Hi Z (disable)	Hi Z (for input port)
52 52	RSCK / PSP6	Hi Z (disable)	Hi Z (for input port)
53	RMOSI / PSP5	Hi Z (disable)	Hi Z (for input port)
54	RMISO / PSP4	Hi Z (disable)	Hi Z (for input port)
55 55	RPCS0 / PSP0	Hi Z (disable)	Hi Z (for input port)
56	RPCS1 / PSP1	Hi Z (disable)	Hi Z (for input port)
57 57	RPCS2 / PSP2	Hi Z (disable)	Hi Z (for input port)
58	RPCS3 / PSP3	Hi Z (disable)	Hi Z (for input port)
59	VSTBY	(Power)	

60	XTAL	(Clock)	
61	VDDSYN / MODCK	(Power)	
62	EXTAL	(Clock)	
63 63	VSS SYN	(GND)	
64	XFC	(Clock filter)	
65 65	VDDI	(GND)	
66 66	VSSI	(Clock)	
67 67	RESET	(RESET)	
68 68	IRQ7 / PF7	Pull up	Hi Z (for IRQ)
69	IRQ6 / PF6	Pull up	Hi Z (for IRQ)
70	IRQ5 / PF5	Pull up	Hi Z (for IRQ)
71 71	IRQ4 / PF4	Pull up	Hi Z (for IRQ)
72	IRQ3 / PF3	Pull up	Hi Z (for IRQ)
73	IRQ2 / PF2	Pull up	Hi Z (for IRQ)
74 74	IRQX / BERR / PF1	Pull up	Hi Z (for IRQ)
75	ALE / PF0	Hi Z (for input)	Hi Z (for ALE)
76 76	D15 / PG7	Hi Z (for input)	Hi Z (for data)
77 77	D14 / PG6	Hi Z (for input)	Hi Z (for data)
78 78	D13 / PG5	Hi Z (for input)	Hi Z (for data)
79 79	D12 / PG4	Hi Z (for input)	Hi Z (for data)
80	VDDE	(Power)	
81	VSSE	(GND)	
82	D11 / PG3	Hi Z (for input)	Hi Z (for data)
83	D10 / PG2	Hi Z (for input)	Hi Z (for data)
84 84	D9 / PG1	Hi Z (for input)	Hi Z (for data)
85	D8 / PG0	Hi Z (for input)	Hi Z (for data)
86	ACS0 / PCS0	High drive	Drive for CS
87	ACS1 / PCS1	High drive	Drive for CS
88	ACS2 / PCS2	High drive	Drive for CS
89	CSA / PD7	High drive (if PCON10 = 1)	Drive for CSA (if PCON10 = 1)
		Pull up (if PCON10 = 0)	Hi Z (for input port) (if PCON10 = 0)
90	CSB / PD6	High drive (if AD7 = 1)	Drive for CSB (if AD7 = 1)
		Hi Z (for input) (if AD7 = 0)	Hi Z (for input port) (if AD7 = 0)
91	CSC / PD5	High drive (if PCON6 = 1)	Drive for CSC (if PCON6 = 1)
		Pull up (if PCON6 = 0)	Hi Z (for input port) (if PCON6 = 0)
92	DTACK / PD4	Hi Z (for input)	Hi Z for DTACK (if AD8 = 1)
			Hi Z (for input port) (if AD8 = 0)
93	FREEZE / QUOT / FASTREF / CSE1 / PD3	Pull up (if BKPT = 1)	Hi Z (for input port) (if BKPT = 1)
		Low drive (if BKPT = 0)	Drive for FREEZE (if BKPT = 0)

Table 3 (c) Non-multiplexed master mode pin conditions (continued)

pin	Pin name	Non-multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
94	BKPT / DSCLK / CSE0 / PD2	Pull up	Hi Z (for input port) (if BKPT = 1)
			Hi Z (for DSCLK) (if BKPT = 0)
95	IPIPE1 / DSI / PD1	Pull up	Hi Z (for input port) (if BKPT = 1)
			Hi Z (for DSI) (if BKPT = 0)
96	IPIPE0 / DSO / PD0	Low drive	High drive (output port) (if BKPT = 1)'
			Drive for DSO (if BKPT = 0)
97	D7 / PH7	Hi Z (for input)	Hi Z (for data)
98	D6 / PH6	Hi Z (for input)	Hi Z (for data)
99	D5 / PH5	Hi Z (for input)	Hi Z (for data)
100	VSSI	(GND)	

101	CLKOUT / PE4	Drive for CLKOUT (if AD0 = 1)
		CLKOUT = XTALx2 (if BERR = 1)
		CLKOUT = XTALx1 (if BERR = 0)
			Drive for CLKOUT (if AD0 = 1)
			CLKOUT = XTALx2 (if BERR = 1)
			CLKOUT = XTALx1 (if BERR = 0)

		Hi Z (for input) (if AD0 = 0)	Hi Z (for input port) (if AD0 = 0)
102	VSSE	(GND)	
103	VDDE	(Power)	
104	D4 / PH4	Hi Z (for input)	Hi Z (for data)
105	D3 / PH3	Hi Z (for input)	Hi Z (for data)
106	D2 / PH2	Hi Z (for input)	Hi Z (for data)
107 107	D1 / PH1	Hi Z (for input)	Hi Z (for data)
108	D0 / PH0	Hi Z (for input)	Hi Z (for data)
109	R / W / WR / PE0	Pull up	Drive for R / W

110	DS / RD / PE1	Pull up	Drive for DS
111	AS / PE2	High drive	Drive for AS
112 112	SIZE / PE3	Pull up	Drive for SIZE
113	EBR / TSC	Hi Z (for input)	HiZ (pull up reg. Must be needed)
114 114	A0 / AD0 / PB0	Pull up	Drive for address
115 115	A1 / AD1 / PB1	Pull up	Drive for address
116	A2 / AD2 / PB2	Pull up	Drive for address
117 117	A3 / AD3 / PB3	Pull up	Drive for address
118	A4 / AD4 / PB4	Pull up	Drive for address
119	A5 / AD5 / PB5	Pull up	Drive for address
120	VDDE	(Power)	
121	VSSE	(GND)	
122	A6 / AD6 / PB6	Pull up	Drive for address
one two Three	A7 / AD7 / PB7	Pull up	Drive for address
124	A8 / AD8 / PA0	Pull up	Drive for address
125	A9 / AD9 / PA1	Pull up	Drive for address
126	A10 / AD10 / PA2	Pull up	Drive for address
127 127	A11 / AD11 / PA3	Pull up	Drive for address
128 128	A12 / AD12 / PA4	Pull up	Drive for address
129	A13 / AD13 / PA5	Pull up	Drive for address
130	A14 / AD14 / PA6	Pull up	Drive for address
131 131	A15 / AD15 / PA7	Pull up	Drive for address
132 132	A16 / PC0	Pull up	Drive for address (if AD [3: 1] =>% 001)
			Hi Z (for input port) (if AD [3: 1] =% 000)
133 133	A17 / PC1	Pull up	Drive for address (if AD [3: 1] =>% 010)
			Hi Z (for input port) (if AD [3: 1] =% 010)
134 134	A18 / PC2	Pull up	Drive for address (if AD [3: 1] =>% 011)
			Hi Z (for input port) (if AD [3: 1] =% 000)

Table 3 (c) Non-multiplexed master mode pin conditions (continued)

pin	Pin name	Non-multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
135	A23 / ECLK / PC7	Pull up (if AD [5: 4] =% 1x)	Drive for address (if AD [5: 4] =% 1x)
		ECLK (if AD [5: 4] =% 01)	ECLK (if AD [5: 4] =% 01)
		Pull up (if AD [5: 4] =% 00)	Hi Z (for input port) (if AD [5: 4] =% 00)
136 136	FC0 / PE5	Pull up	Hi Z (for input port) (if AD9 = 0)
			Drive for FC0 (if AD9 = 1)
137	FC1 / PE6	Pull up	Hi Z (for input port) (if AD9 = 0)
			Drive for FC1 (if AD9 = 1)
138	VFPE2	(Power)	
139	VDDI	(Power)	
140	VSSI	(GND)	
141	AN0 / ANW / PQB0	Hi Z (disable)	Hi Z (for input port)
142	AN1 / ANX / PQB1	Hi Z (disable)	Hi Z (for input port)
143	AN2 / ANY / PQB2	Hi Z (disable)	Hi Z (for input port)
144	AN3 / ANZ / PQB3	Hi Z (disable)	Hi Z (for input port)
145	AN48 / PQB4	Hi Z (disable)	Hi Z (for input port)
146	AN49 / PQB5	Hi Z (disable)	Hi Z (for input port)
147	AN50 / PQB6	Hi Z (disable)	Hi Z (for input port)
148	AN51 / PQB7	Hi Z (disable)	Hi Z (for input port)
149	VDDA	(Power)	
150	VSSA	(GND)	
151 151	VRL	(A / D ref. Volt)	
152	VRH	(A / D ref. Volt)	
153	AN52 / MA0 / PQA0	Hi Z (disable)	Hi Z (for input port)
154	AN53 / MA1 / PQA1	Hi Z (disable)	Hi Z (for input port)
155	AN54 / MA2 / PQA2	Hi Z (disable)	Hi Z (for input port)
156	AN55 / ETRIG1 / PQA3	Hi Z (disable)	Hi Z (for input port)
157	AN56 / ETRIG2 / PQA4	Hi Z (disable)	Hi Z (for input port)
158	AN57 / PQA5	Hi Z (disable)	Hi Z (for input port)
159	AN58 / PQA6	Hi Z (disable)	Hi Z (for input port)
160	AN59 / PQA7	Hi Z (disable)	Hi Z (for input port)

Page 28

Table 3 (d) Multiplexed master mode pin conditions

pin	Pin name	Multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
1	VDDE	(Power)	
2	VSSE	(GND)	
3	TP0	Hi Z (disable)	Hi Z (for input)
Four	TP1	Hi Z (disable)	Hi Z (for input)
Five	TP2	Hi Z (disable)	Hi Z (for input)
6	TP3	Hi Z (disable)	Hi Z (for input)
7	TP4	Hi Z (disable)	Hi Z (for input)
8	TP5	Hi Z (disable)	Hi Z (for input)
9	TP6	Hi Z (disable)	Hi Z (for input)
Ten	TP7	Hi Z (disable)	Hi Z (for input)
11 11	TP8	Hi Z (disable)	Hi Z (for input)
12	TP9	Hi Z (disable)	Hi Z (for input)
13	TP10	Hi Z (disable)	Hi Z (for input)
14	TP11	Hi Z (disable)	Hi Z (for input)
15	TP12	Hi Z (disable)	Hi Z (for input)
16 16	TP13	Hi Z (disable)	Hi Z (for input)
17 17	TP14	Hi Z (disable)	Hi Z (for input)
18 18	TP15	Hi Z (disable)	Hi Z (for input)
19 19	T2CLK	Hi Z (disable)	Hi Z (for input)
20	VFPE1	(Power)	
twenty one	VSSI	(GND)	
twenty two	CTD3	Hi Z (disable)	Hi Z (for input)
twenty three	CTD4	Hi Z (disable)	Hi Z (for input)
twenty four	CTD5	Hi Z (disable)	Hi Z (for input)
twenty five	CTD6	Hi Z (disable)	Hi Z (for input)
26	CTD7	Hi Z (disable)	Hi Z (for input)
27	CTD8	Hi Z (disable)	Hi Z (for input)
28 28	CTD14	Hi Z (disable)	Hi Z (for input)
29	CTD15	Hi Z (disable)	Hi Z (for input)
30	CTD16	Hi Z (disable)	Hi Z (for input)
31	CTD17	Hi Z (disable)	Hi Z (for input)
32	CTD18	Hi Z (disable)	Hi Z (for input)
33 33	CTD20	Hi Z (disable)	Hi Z (for input)
34	CTD22	Hi Z (disable)	Hi Z (for input)
35	CPWM9	Hi Z (output only)	Low drive
36	CPWM10	Hi Z (output only)	Low drive
37 37	CPWM11	Hi Z (output only)	Low drive
38	CPWM12	Hi Z (output only)	Low drive
39 39	CPWM13	Hi Z (output only)	Low drive
40	VSSE	(GND)	
41 41	VDDE	(Power)	

42	CTM2C	Hi Z (disable)	Hi Z (for input)
43	TXDA / PMC7	Hi Z (disable)	Hi Z (for input port)
44	RXDA / PMC6	Hi Z (disable)	Hi Z (for input port)
45 45	TXDB / PMC5	Hi Z (disable)	Hi Z (for input port)
46	RDXB / PMC4	Hi Z (disable)	Hi Z (for input port)
47 47	SS / PMC3	Hi Z (disable)	Hi Z (for input port)
48	SCK / PMC2	Hi Z (disable)	Hi Z (for input port)

Table 3 (d) Multiplexed master mode pin conditions (continued)

pin	Pin name	Multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
49	MOSI / PMC1	Hi Z (disable)	Hi Z (for input port)
50	MISO / PMC0	Hi Z (disable)	Hi Z (for input port)
51	RSS / PSP7	Hi Z (disable)	Hi Z (for input port)
52 52	RSCK / PSP6	Hi Z (disable)	Hi Z (for input port)
53	RMOSI / PSP5	Hi Z (disable)	Hi Z (for input port)
54	RMISO / PSP4	Hi Z (disable)	Hi Z (for input port)
55 55	RPCS0 / PSP0	Hi Z (disable)	Hi Z (for input port)
56	RPCS1 / PSP1	Hi Z (disable)	Hi Z (for input port)
57 57	RPCS2 / PSP2	Hi Z (disable)	Hi Z (for input port)
58	RPCS3 / PSP3	Hi Z (disable)	Hi Z (for input port)
59	VSTBY	(Power)	
60	XTAL	(Clock)	
61	VDDSYN / MODCK	(Power)	
62	EXTAL	(Clock)	
63 63	VSS SYN	(GND)	
64	XFC	(Clock filter)	
65 65	VDDI	(GND)	
66 66	VSSI	(Clock)	
67 67	RESET	(RESET)	
68 68	IRQ7 / PF7	Pull up	Hi Z (for IRQ)
69	IRQ6 / PF6	Pull up	Hi Z (for IRQ)
70	IRQ5 / PF5	Pull up	Hi Z (for IRQ)
71 71	IRQ4 / PF4	Pull up	Hi Z (for IRQ)
72	IRQ3 / PF3	Pull up	Hi Z (for IRQ)
73	IRQ2 / PF2	Pull up	Hi Z (for IRQ)
74 74	IRQX / BERR / PF1	Pull up	Hi Z (for IRQ)
75	ALE / PF0	Hi Z (for input)	Drive for ALE
76 76	D15 / PG7	Hi Z (for input)	Hi Z (for data)
77 77	D14 / PG6	Hi Z (for input)	Hi Z (for data)
78 78	D13 / PG5	Hi Z (for input)	Hi Z (for data)
79 79	D12 / PG4	Hi Z (for input)	Hi Z (for data)
80	VDDE	(Power)	
81	VSSE	(GND)	
82	D11 / PG3	Hi Z (for input)	Hi Z (for data)
83	D10 / PG2	Hi Z (for input)	Hi Z (for data)
84 84	D9 / PG1	Hi Z (for input)	Hi Z (for data)
85	D8 / PG0	Hi Z (for input)	Hi Z (for data)
86	ACS0 / PCS0	High drive	Drive for CS
87	ACS1 / PCS1	High drive	Drive for CS
88	ACS2 / PCS2	High drive	Drive for CS
89	CSA / PD7	High drive (if PCON10 = 1)	Drive for CSA (if PCON10 = 1)
		Pull up (if PCON10 = 0)	Hi Z (for input port) (if PCON10 = 0)
90	CSB / PD6	High drive (if AD7 = 1)	Drive for CSB (if AD7 = 1)
		Hi Z (for input) (if AD7 = 0)	Hi Z (for input port) (if AD7 = 0)
91	CSC / PD5	High drive (if PCON6 = 1)	Drive for CSC (if PCON6 = 1)
		Pull up (if PCON6 = 0)	Hi Z (for input port) (if PCON6 = 0)
92	DTACK / PD4	Hi Z (for input)	Hi Z for DTACK (if AD8 = 1)
			Hi Z (for input port) (if AD8 = 0)
93	FREEZE / QUOT / FASTREF / CSE1 / PD3	Pull up (if BKPT = 1)	Hi Z (for input port) (if BKPT = 1)
		Low drive (if BKPT = 0)	Drive for FREEZE (if BKPT = 0)

Table 3 (d) Multiplexed master mode pin conditions (continued)

pin	Pin name	Multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)	
		Condition during reset	Condition after reset
94	BKPT / DSCLK / CSE0 / PD2	Pull up	Hi Z (for input port) (if BKPT = 1) Hi Z (for DSCLK) (if BKPT = 0)
95	IPIPE1 / DSI / PD1	Pull up	Hi Z (for input port) (if BKPT = 1) Hi Z (for DSI) (if BKPT = 0)
96	IPIPE0 / DSO / PD0	Low drive	High drive (output port) (if BKPT = 1) Drive for DSO (if BKPT = 0)
97	D7 / PH7	Hi Z (for input)	Hi Z (for data)
98	D6 / PH6	Hi Z (for input)	Hi Z (for data)
99	D5 / PH5	Hi Z (for input)	Hi Z (for data)
100	VSSI	(GND)	

101	CLKOUT / PE4	Drive for CLKOUT (if AD0 = 1)
		CLKOUT = XTALx2 (if BERR = 1)
		CLKOUT = XTALx1 (if BERR = 0)
		Hi Z (for input) (if AD0 = 0)
			Drive for CLKOUT (if AD0 = 1)
			CLKOUT = XTALx2 (if BERR = 1)
			CLKOUT = XTALx1 (if BERR = 0)
			Hi Z (for input port) (if AD0 = 0)

102	VSSE	(GND)	
103	VDDE	(Power)	
104	D4 / PH4	Hi Z (for input)	Hi Z (for data)
105	D3 / PH3	Hi Z (for input)	Hi Z (for data)
106	D2 / PH2	Hi Z (for input)	Hi Z (for data)
107 107	D1 / PH1	Hi Z (for input)	Hi Z (for data)
108	D0 / PH0	Hi Z (for input)	Hi Z (for data)
109	R / W / WR / PE0	Pull up	Drive for R / W
110	DS / RD / PE1	Pull up	Drive for DS
111	AS / PE2	High drive	Drive for AS
112 112	SIZE / PE3	Pull up	Drive for SIZE
113	EBR / TSC	Hi Z (for input)	HiZ (pull up reg. Must be needed)
114 114	A0 / AD0 / PB0	Pull up	Drive for address / data
115 115	A1 / AD1 / PB1	Pull up	Drive for address / data
116	A2 / AD2 / PB2	Pull up	Drive for address / data
117 117	A3 / AD3 / PB3	Pull up	Drive for address / data
118	A4 / AD4 / PB4	Pull up	Drive for address / data
119	A5 / AD5 / PB5	Pull up	Drive for address / data
120	VDDE	(Power)	
121	VSSE	(GND)	
122	A6 / AD6 / PB6	Pull up	Drive for address / data
one two Three	A7 / AD7 / PB7	Pull up	Drive for address / data
124	A8 / AD8 / PA0	Pull up	Drive for address / data
125	A9 / AD9 / PA1	Pull up	Drive for address / data
126	A10 / AD10 / PA2	Pull up	Drive for address / data
127 127	A11 / AD11 / PA3	Pull up	Drive for address / data
128 128	A12 / AD12 / PA4	Pull up	Drive for address / data
129	A13 / AD13 / PA5	Pull up	Drive for address / data
130	A14 / AD14 / PA6	Pull up	Drive for address / data
131 131	A15 / AD15 / PA7	Pull up	Drive for address / data
132 132	A16 / PC0	Pull up	Drive for address (if AD [3: 1] =>% 001) Hi Z (for input port) (if AD [3: 1] =% 000)
133 133	A17 / PC1	Pull up	Drive for address (if AD [3: 1] =>% 010) Hi Z (for input port) (if AD [3: 1] =% 010)
134 134	A18 / PC2	Pull up	Drive for address (if AD [3: 1] =>% 011) Hi Z (for input port) (if AD [3: 1] =% 000)

Table 3 (d) Multiplexed master mode pin conditions (continued)

pin	Pin name	Multiplexed master mode (DSI = 0, DS = 1, SIZE = 0, RW = 0)

		Condition during reset	Condition after reset
135	A23 / ECLK / PC7	Pull up (if AD [5: 4] =% 1x) ECLK (if AD [5: 4] =% 01) Pull up (if AD [5: 4] =% 00)	Drive for address (if AD [5: 4] =% 1x) ECLK (if AD [5: 4] =% 01) Hi Z (for input port) (if AD [5: 4] =% 00)
136 136	FC0 / PE5	Pull up	Hi Z (for input port) (if AD9 = 0) Drive for FC0 (if AD9 = 1)
137	FC1 / PE6	Pull up	Hi Z (for input port) (if AD9 = 0) Drive for FC1 (if AD9 = 1)
138	VFPE2	(Power)	
139	VDDI	(Power)	
140	VSSI	(GND)	
141	AN0 / ANW / PQB0	Hi Z (disable)	Hi Z (for input port)
142	AN1 / ANX / PQB1	Hi Z (disable)	Hi Z (for input port)
143	AN2 / ANY / PQB2	Hi Z (disable)	Hi Z (for input port)
144	AN3 / ANZ / PQB3	Hi Z (disable)	Hi Z (for input port)
145	AN48 / PQB4	Hi Z (disable)	Hi Z (for input port)
146	AN49 / PQB5	Hi Z (disable)	Hi Z (for input port)
147	AN50 / PQB6	Hi Z (disable)	Hi Z (for input port)
148	AN51 / PQB7	Hi Z (disable)	Hi Z (for input port)
149	VDDA	(Power)	
150	VSSA	(GND)	
151 151	VRL	(A / D ref. Volt)	
152	VRH	(A / D ref. Volt)	
153	AN52 / MA0 / PQA0	Hi Z (disable)	Hi Z (for input port)
154	AN53 / MA1 / PQA1	Hi Z (disable)	Hi Z (for input port)
155	AN54 / MA2 / PQA2	Hi Z (disable)	Hi Z (for input port)
156	AN55 / ETRIG1 / PQA3	Hi Z (disable)	Hi Z (for input port)
157	AN56 / ETRIG2 / PQA4	Hi Z (disable)	Hi Z (for input port)
158	AN57 / PQA5	Hi Z (disable)	Hi Z (for input port)
159	AN58 / PQA6	Hi Z (disable)	Hi Z (for input port)
160	AN59 / PQA7	Hi Z (disable)	Hi Z (for input port)

Note: "Hi-Z": Pin is in high impedance state
"Low Drive": Pin outputs low
"High Drive": Pin outputs high
"Pull up": High output by internal pull-up function (about 50K ohms)
"Drive for signal name": Output the specified signal
"disable": Since the input side / output side processes internally, no external processing is required.
"for input": function as an input pin
"for input port": function as a general-purpose input pin
"output only": function as an output-only pin
"by output port": Function as a general-purpose output pin
"Clock, Clock filter": Pins used to generate the system clock
"Power": Power supply pin

Page 32

The following table shows the types of I / O drivers used with this MCU. All input types are CMOS theory Please note that you are using the physics level. In addition, the bus controller directly stays in three stays. There are some output types (type A) that cannot be made, but include external bus signals and module pins. All outputs can be three-stated with the three-state control pin.

Table 4 Driver types

type	I / O	Explanation	Equivalent circuit
A	O	Output-only signal that is always driven (no external pull-up required) Note: Hi-Z only in test mode (by TSC signal) It is possible to. It will be Hi-Z in normal operation	Internal signal

		There is no.	
Aw	O	Type A output. Weak P-channel pull-up on reset Become active.	Internal reset Internal signal Control Signal
B 1	O	Three-state output. High impedance is stable To pull up the output before (to reduce the rise time) Built-in circuit. When the pin is in high impedance state Requires an external hold resistor to maintain the logic level.	Internal signal Control Signal
Bo	O	With open / drain output or output that always drives Can be operated. Note: The RESET signal is fixed to the open / drain output. I am.	Control Signal Internal signal
Ba	O	Type B output. Can only operate in open drain mode.	Internal signal
Bw	O	Type B output. Weak P-channel pull-up on reset Become active.	Internal reset Internal signal Control Signal

Note: 1. Pins with this type of driver will only have high impedance under certain conditions.
vinegar. All of these can be put into a high impedance state with a TSC signal.

Page 33

Table 5 Power connection

pin	Explanation
V_{DDA} , V_{SSA} , V_{RH} , V_{RL}	A / D converter power supply and reference power supply
V_{DDSYN} / MODCK, V_{SSSYN}	Clock synthesizer power / mode selection
V_{DDE}, V_{SSE}	External peripheral power supply (source and drain)
V_{DDI}, V_{SSI}	Internal module power supply (source and drain)
V_{STBY}	Standby RAM power supply
V_{FPE1}	Write / Erase Power to 4K TPU Flash EEPROM
V_{FPE2}	Write / Erase Power to 160K Flash / HD Flash EEPROM

1.2 Signal description

The following table shows the types and functions of MCU signals.

Table 6 Signal characteristics

Signal name	MCU module	Signal type	Active state

A [23:16]	SLIM	Bus	————
ACS [2: 0]	ACS2	Output Output	0
AD [15: 0]	SLIM	Bus	————
ALE	SLIM	Output Output	1
AN [3: 0] / AN [Z, Y, X, W]	QADC	Input	————
AN [59:48]	QADC	Input	————
AS	SLIM	Output Output	0
BERR	SLIM	Input	0
BKPT	CPU16	Input	0
CLKOUT	SLIM	Output Output	————
CPWM [13: 9]	CTM3	Output Output	————
CSA, CSB, CSC	SLIM	Output Output	0
CSE [1: 0]	SLIM	Output Output	————
CTD [22] / [20] / [18:14] / [8: 3]	CTM3	Input / Output	————
CTM2C	CTM3	Input	————
D [15: 0]	SLIM	Bus	————
DS	SLIM	Output Output	0
DSCLK	CPU16	Input	————
DSI	CPU16	Input	————
DSO	CPU16	Output Output	————
DTACK	SLIM	Input	0

Page 34

Table 6 Signal characteristics (continued)

Signal name	MCU module	Signal type	Active state
EBR	SLIM	————	0
EXTAL	SLIM	Input	————
FC [1: 0]	SLIM	Output Output	————
FREEZE	SLIM	Output Output	1
IPIPE [1: 0]	CPU16	Output Output	————
IRQ [7: 2]	SLIM	Input	0
IRQX	SLIM	Input	0
MISO	MCCI	Input / Output	————
MOSI	MCCI	Input / Output	————
MODCK	SLIM	Input	————
QUOT	SLIM	Output Output	————
RD	SLIM	Output Output	0
R / W	SLIM	Output Output	————
RESET	SLIM	Input / Output	0
RMISO	RSPI	Input / Output	————
RMOSI	RSPI	Input / Output	————
RPCS [3: 0]	RSPI	Input / Output	————
RSCK	RSPI	Input / Output	————
RSS	RSPI	Input / Output	0
RXDA	MCCI	Input	————
RXDB	MCCI	Input	————
SCK	MCCI	Input / Output	————
SIZE	SLIM	Output Output	————

SS	MCCI	Input	0
TP [15: 0]	TPU2	Input / Output	————
T2CLK	TPU2	Intput	————
TSC	SLIM	Input	0
TXDA	MCCI	Output Output	————
TXDB	MCCI	Output Output	————
WR	SLIM	Output Output	0
XFC	SLIM	Input	————
XTAL	SLIM	Output Output	————

Page 35

Table 7 Signal function

Signal name	Mnemonic	function
Address Bus	A [23:20]	The most significant 4-bit address is equivalent to address 19.
Address Bus	A [19:16]	Address of CPU16 [19:16]
Address / Data Bus	AD [15: 0]	16-bit address / data bus of CPU16
Auxiliary Chip Select	ACS [2: 0]	Auxiliary chip select output
Address Latch Enable	ALE	Establish address / data in multiplexed master mode
QADC Analog Input	AN [59:48] / [3: 0]	QADC input
QADC Analog Input	AN [Z, Y, X, W]	Analog input switching signal in multiplex mode
Address Strobe	AS	Address signal is valid on the address bus
Bus Error	BERR	If you enter when there is a problem with the external bus cycle, the bus Stop the cycle and handle the bus error exception.
Breakpoint / Debug Support Clock	BKPT / DSCLK	BKPT is a CPU hardware break DSCLK is the serial clock for debugging
System Clockout	CLKOUT	Output system clock
CTM3 Pulse-width Modulation	CPWM [13: 9]	CTM3 5 PWM outputs
CTM3 Double-Acton-Channel	CTD [22] / [20] / [18:14] / [8: 3]	CTM3's 13 bidirectional double-action timer channels
CTM3 Modulus Clock	CTM2C	Input for CTM3 modulus counter clock
General-purpose Chip Selects	CSA, CSB, CSC	External peripheral select signal, CSB is a select signal for boot ROM
Emulation Chip Select	CSE0, CSE1	Tracking signal for emulation
Data Bus	D [15: 0]	16-bit data bus
Data Strobe	DS	In the read cycle, DS asserts and sends data to an external device. Instruct the drive. DS is the data bar in the write cycle Notify the external device that the data on the device is valid.
Development Serial Out	DSO	Output for debug support
Development Serial In	DSI	Input for debug support
Data Transfer Acknowledge	DTACK	Notifies the CPU of the end of the bus cycle.
External Bus Interrupt	EBR	Signal to set output pin to Hi-Z for multi-master etc.
External Clock	ECLK	M6800 bus clock output
QADC External Trigger	ETRIG [2: 1]	External trigger input for A / D conversion
Crystal Oscillator	EXTAL, XTAL	Crystal connection pin, Use EXTAL to input the external clock directly.

Function Codes	FC [1: 0]	Indicates the address space.
Freeze	FREEZE	Tells the CPU that it is in background mode.
Instruction Pipeline	IPIPE0, IPIPE1	For instruction fetch
Interrupt Request	IRQ [7: 2], IRQX	External interrupt input to CPU
QADC Multipexed Address	MA [2: 0]	Address to switch external multiplex
Master In Slave Out	MISO	Serial data input from SPI in master mode Serial data output from SPI in slave mode

Page 36

Table 7 Signal function (continued)

Signal name	Mnemonic	function
Master Out Slave In	MOSI	Serial data output from SPI in master mode Serial data input from SPI in slave mode
Clock Mode Select	MODCLK	System clock selection
Port A	PA [7: 0]	SLIM general purpose I / O port
Port B	PB [7: 0]	SLIM general purpose I / O port
Port C	PC [7: 0]	SLIM general purpose I / O port
Port CS	PCS [2: 0]	ACS general purpose output port
Port D	PD [7: 0]	SLIM general purpose I / O port
Port E	PE [6: 0]	SLIM general purpose I / O port
Port F	PF [7: 0]	SLIM general purpose I / O port
Port G	PG [7: 0]	SLIM general purpose I / O port
Port H	PH [7: 0]	SLIM general purpose I / O port
Port MC	PMC [7: 0]	MCCI general purpose I / O port
Port QA	PQA [7: 0]	QADC general purpose I / O port
Port QB	PQB [7: 0]	QADC general purpose input port
Port SP	PSP [7: 0]	RSPI general purpose I / O port
QUOT	QUOT	Motorola test output
Read / Write	R / W	Indicates the transfer direction of data during the bus cycle.
Read	RD	Shows the read of data during the bus cycle.
Reset	RESET	MCU system reset
RSPI Master In Slave Out	RMISO	Serial data input from RSPI in master mode Serial data output from RSPI in slave mode
RSPI Master Out Slave In	RMOSI	Serial data output from RSPI in master mode Serial data input from RSPI in slave mode
RSPI Chip Select	RPCS [3: 0]	Chip select output from RSPI
RSPI Slave Clock	RSS	Select output in master mode Select input in slave mode
SCI A Receive Data	RXDA	SCIA serial data entry
SCI B Receive Data	RXDB	SCIB serial data entry
SPI Serial Clock	SCK	Serial sync clock output from SPI in master mode Serial sync clock input from SPI in slave mode
Size Size	SIZE	Tells you the size of the data during the bus cycle.
Slave Select	SS	Input for mode fault in master mode Input for selection from SPI in slave mode
TPU2 Clock	T2CLK	TPU2 clock input
TPU2	TP [15: 0]	Each channel of TPU2
Three-State Control	TSC	Motorola test input Set all pins to Hi-Z.
SCI A Transmit Data	TXDA	SCIA serial data output
SCI B Transmit Data	TXDB	SCIB serial data output

Table 7 Signal function (continued)

Signal name	Mnemonic	function
Synthesizer Filter Cap	XFC	For PLL filters
Write Write	WR	Shows the light of the data during the bus cycle.

1.3 Unused signal

Signals that are not pulled to the pin are internally pulled up or pulled down.

Table 8 Unused signal connections

Unused signal	Explanation
FC2	Pull up inside
A [22:19] / PC [6: 3]	Pull down internally

1.4 Address map

The following figure shows the MCU's internal address map.
Depending on the system configuration at reset, the reset may or may not enable boot ROM.
It may be. The SRAM array is placed by the base address register of the SRAM CTRL block.
The position to be used is determined. The reset disables the SRAM array. The FLASH ROM array is
The base address register of the FLASH array CTRL block determines where it is placed.
The MASK ROM array is located by the base address register of the MASK ROM array CTRL block.
The position is determined. Unimplemented blocks are mapped externally.
There are 24 address lines, but CPU16 uses only A [19: 0]. ADDR [23:20] is A19
Addresses in the range $ 080000 to $ F7FFFF cannot be accessed because they are driven in the same state
In the address map, Y = M111. However, M is a single-chip integration module.
Table shows the state of the module mapping bit (MM) of the configuration register (SLIM) of the file.
I will. On this MCU, Y must be $ F. If MM is cleared, CPU16 will
The address range from $ 080000 to $ F7FFFF is inaccessible, so IMB mode until a reset occurs.
You cannot access the joules. The MM can only be written once after a reset.

$ YFF000

$ 000000

Boot ROM
512 Bytes

$ 0001FF

$ YFF200

QADC
(512 bytes)

$ YFF3FF
$ YFF400

RAM RSPI Array
(3K)

$ YFF5FF
$ YFF600

3K RSPI
(64 bytes)

Rom Array 1
(64K)

$ YFF63F

$ YFF800
$ YFF81F **ROM 1 CTL**

Rom Array 2
(64K)

$ YFF800

Mask ROM
(160K bytes)

$ YFF820
$ YFF83F **ROM 2 CTL**
$ YFF840
$ YFF85F **ROM 3 CTL**

$ YFF85F

Rom Array 3
(32K)

$ YFF900

CTM3
(256 bytes)

$ YFF9FF
$ YFFA00
$ YFFA7F **SLIM** (128 bytes)
$ YFFA80
$ YFFABF **ACS** (64 bytes)

$ YFFB00
$ YFFB07 **SRAM CTL** (8 bytes)

RAM Array
(4K)

$ YFFC00
$ YFFC3F **MCCI** (64 bytes)

$ YFFE00

TPU2
(512 bytes)

$ YFFFFF

Figure 3

Figure 3 (a) MC68HC16Y5 system address map

$ 000000

$ YFF000

Boot ROM
512 Bytes

$ 0001FF

$ YFF200

RAM RSPI Array
(3K)

QADC

(512 bytes)

$ YFF3FF
$ YFF400

Flash Array 1
(32K)

$ YFF5FF
$ YFF600

Flash Array 2
(32K)

3K RSPI
(64 bytes)

Flash Array 3
(32K)

$ YFF63F

$ YFF7C0
$ YFF7DF **Flash 1 CTL**
$ YFF7E0
$ YFF7FF **Flash 2 CTL**

$ YFF7C0

Flash Array 4
(32K)

$ YFF7FF
Flash 1-5
$ YFF800
(192 bytes)

$ YFF800
$ YFF81F **Flash 3 CTL**
$ YFF820
$ YFF83F **Flash 4 CTL**

$ YFF85F
$ YFF860

Flash Array 5
(32K)

TPUFlash (32 bytes)

$ YFF840
$ YFF85F **Flash 5 CTL**

$ YFF87F

$ YFF860
$ YFF87F **TPUFlash CTL**

TPUFlash Array
(4K)

$ YFF900

CTM3
(256 bytes)

$ YFF9FF
$ YFFA00
SLIM (128 bytes)
$ YFFA7F
$ YFFA80
ACS (64 bytes)
$ YFFABF

RAM Array
(4K)

$ YFFB00
SRAM CTL (8 bytes)
$ YFFB07

$ YFFC00
MCCI (64 bytes)
$ YFFC3F

$ YFFE00

TPU2
(512 bytes)

$ YFFFFF

Figure 3 (b) MC68HC916Y5 system address map

Page 40

$ YFF000

$ 000000

Boot ROM
512 Bytes

$ 0001FF

$ YFF200

QADC
(512 bytes)

$ YFF3FF
$ YFF400

RAM RSPI Array
(3K)

$ YFF5FF
$ YFF600

3K RSPI
(64 bytes)

Flash Array 1
(64K)

$ YFF63F

$ YFF780
$ YFF7BF **Flash 1 CTL**
$ YFF7C0

$ YFF780

Flash Array 2
(64K)

$ YFF7FF
Flash 1-3

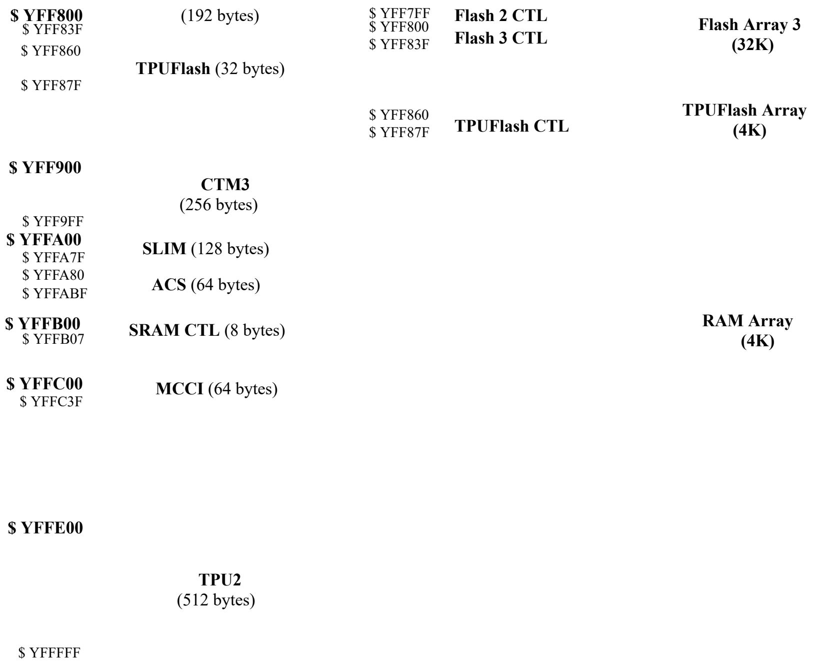


Figure 3 (c) MC68HC916Y6 system address map

Page 41

1.5 Intermodule bus

The Intermodule Bus (IMB) is open to simplify the design of modular microcontrollers.
It is a standard bus that departed. Exception handling, address space partitioning, multiple interrupt levels, and vector interrupts
It has a built-in circuit to support it. Standard modules in the CPU are mutually or externally via the IMB.
Interface with is possible. IMB supports 24 address lines and 16 data lines
However, CPU16 does not use 20 address lines, so A [23:20] has the same logic as A19.
It will be driven to the state.

Chapter 2
Central processing unit

CPU16 is a true 16-bit high speed device. Users of M68HC11 can use this device to use existing devices. You can improve performance while maintaining compatibility with the stem.

2.1 Overview

The key to using a microcontroller is that it is easy to program. CPU16 The instruction set is optimized for high performance. CPU16 is two 16-bit general purpose accums It contains a lator and three 16-bit index registers. 8 bits (bytes), 16 bits Supports (word) and 32-bit (long word) load and store operations, and 16 It also supports bit and 32-bit signed decimal operations. Background debug mode If you use it, you can proceed with program development efficiently.

The memory space of CPU16 has 1 Mbytes of data space and 1 Mbytes of program space. 20 bit Extended memory is realized by using the addressing function and the transparent bank switching function. Moreover, Most instructions handle bank boundaries automatically.

CPU16 is the instruction and hardware that realizes the digital signal processing function mainly for control with the minimum interface. Equipped with clothing. The Multiply / Cumulative (MAC) unit multiplies a 16-bit signed decimal and the result The resulting 32-bit fixed-point product can be stored in a 36-bit accumulator. Also, Juro addressing supports finite impulse response filters.

The CPU16 instruction set supports high-level languages, complicating controller applications. It supports the use of high-level languages, which increase as the control program becomes larger. High-level language With, you can develop software that is easy to port with few errors in a short period of time.

2.2 Compatibility with M68HC11

The CPU16 architecture is a superset of the M68HC11 architecture. All of M68HC11 The source is available on CPU16. M68HC11 instructions are implemented directly on CPU16 or have an equivalent format. It has been replaced by an instruction with. The instruction set is compatible at the source code level. rice field

However, some instructions behave differently on the M68HC16. These instructions are mainly interrupt and exception handling
Is related to. Therefore, interrupt processing, stack frame processing, or condition code
The code for M68HC11 that manipulates the registers needs to be rewritten.

Cycle-related delays and timed control routines due to the different execution times and number of cycles for all instructions
May also affect.

CPU16 introduces or enhances some addressing modes.
The M68HC11's direct mode addressing is special, using a new IZ register and reset vector.
It has been replaced with an index addressing format for greater flexibility.

Page 43

2.3 Programmer's model

20 16	15 8	7 0	Bit position
	A	B	Accumulators A and B
	D		Accumulator D (A: B)
	E		Accumulator E
XK	IX		Index register X
YK	IY		Index register Y
ZK	IZ		Index register Z
SK	SP		Stack pointer
PK	PC		Program counter
	CCR	PK	Condition code register
	EK XK	YK ZK	Address extension (K) register
		SK	Stack extension register
	H		MAC multiplier register
	I		MAC multiplicand register
	AM (MSB)		MAC accumulator [35:16]
	AM (LSB)		MAC accumulator [15: 0]
	XMSK	YMSK	MAC XY mask register

--Accumulator A: 8-bit general-purpose register
--Accumulator B: 8-bit general-purpose register
--Accumulator D: 16-bit register formed by concatenating accumulators A and B
--Accumulator E: 16-bit general-purpose register
--Index register X: 16-bit index register, in the XK field of the K register Expand addressing
--Index register Y: 16-bit index register, in the YK field of the K register Expand addressing
--Index register Z: 16-bit index register, in the ZK field of the K register Expand addressing
--Stack point: 16-bit dedicated register, SK register to expand addressing

Page 44

--Program counter: 16-bit dedicated register, extended addressing in PK field of CCR
--Condition code register: Condition flag, interrupt priority mask,
And also the address extension field of the program counter
16-bit register
--K register: 16-bit register consisting of four 4-bit address extension fields
--SK register: 4-bit register with stack pointer address extension field
--H register: 16-bit multiplication / cumulative input (multiplier) register
--I register: 16-bit multiplication / cumulative input (multiplicand) register
--MAC accumulator: 36-bit multiplication / cumulative result register
--XMSK, YMSK: Specify which bit to change when offset addition

2.3.1 Condition code register

Condition code register

15 14 13 12 11 10 9 8 7 6 Five Four 3 2 1 0
S MV HE V N Z V C IP SM PK

The Condition Code Register (CCR) can be thought of as two functional blocks.
The most significant bit (MSB) corresponds to the MC68HC11's CCR and is a low power stop control bit.
There is a processor status flag. The least significant bit (LSB) is the interrupt priority.
Field, digital signal processing (DSP) saturation mode control bits, and program cow
There is an address extension field.

S --- STOP enabled
0 = Stop the clock when the LPSTOP instruction is executed.
1 = NOP is executed when the LPSTOP instruction is executed.
MV --- Accumulator M Overflow Flag
When an overflow to the sign bit (AM35) of the accumulator M occurs, it is set.
Will be
H--- Half carry flag
Carry emanates from bit 3 of accumulator A or B while performing BCD addition
It is set when it is born.
EV --- Extended Bit Overflow Flag
Set when accumulator M overflows to bit 31.
N --- Negative flag
Set when the result register MSB is set.
Z --- Zero flag
Set when all bits in the result register are zero.
V --- Overflow flag
Set when 2's complement overflow occurs as a result of the operation.
C --- Carry flag
Set when a carry or borrow occurs during an arithmetic operation. Also, multiple
It is also used during shifts and rotations to facilitate word operations.
IP [2: 0] --- Interrupt priority mask
The value in this field ($ 0 to $ 7) specifies the interrupt priority level for CPU16.
SM --- Saturation mode bit
If either EV or MV is set while SM is set
For data read from accumulator M using TMRT or TMET,
Maximum positive or negative, depending on the state of the sign bit of AM before the overflow occurs
The maximum value is given.
PK [3: 0] --- Program counter address extension field
This field is concatenated with a program counter and is a 20-bit pseudo-linear.
Form an address.

Page 45

2.4 Data type

CPU16 supports the following data types:

- Bit data
- 8-bit (byte) and 16-bit (word) integers
- 32-bit long word integer
- 16-bit and 32-bit signed decimal parts (MAC operations only)
- A 20-bit effective address consisting of a 16-bit page address and a 4-bit extension

One byte is 8-bit wide and can be accessed at any byte location. 1 word is 2 It consists of consecutive bytes and is addressed by the lower byte. Instruction fetch is always word bound It will be accessed by. Word operands are also typically accessed on word boundaries, but with significantly lower performance. You can access even odd-numbered byte addresses if you are prepared for the bottom.

Unaligned word forwarding and unaligned stack access are allowed to maintain compatibility with M68HC11 I am. Transferring misaligned words requires two consecutive byte operations.

2.5 Addressing mode

CPU16 supports 10 different addressing types. One or more for each type There is an addressing mode. Six addressing types for CPU16 are M68HC11 addressing Same as the single type.

ADDR [15: 0] is generated in all addressing modes. In the extended field at this address ADDR [19:16] is combined to form a 20-bit effective address. Extended field to CPU16 It is part of a bank switching method that provides an address space of 1 Mbyte. For most instructions the user vans There is no need to be aware of switching. When accessing across bank boundaries, the effective address ADDR [19:16] changes. However, the value of the relevant extended field depends on the type of instruction, Note that access generally does not change across bank boundaries.

In immediate mode, the byte or word immediately following the instruction contains the argument. fruit The effective address is the address of the byte immediately following the instruction. For AIS, AIX / Y / Z, ADDD, and ADDE instructions It has an extended 8-bit mode and the immediate value is an 8-bit signed number. This number is 16 The execution time is shortened because the sign is expanded to the specified register and then added to the specified register.

The extended mode instruction contains ADDR [15: 0] in the word immediately following the opcode. The effective address is It is formed by concatenating the EK field and the 16-bit extension.

In index mode, registers IX, IY, and IZ are used with their extended fields, Calculate the effective address. Signed 16-bit mode and signed 20-bit mode are available on the M68HC11. An extension of index addressing mode.

In 8-bit index mode, the 8-bit unsigned offset contained in the instruction is in. It is added to the value contained in the dex register and its extended fields.

In 16-bit mode, the 16-bit signed offset contained in the instruction is indexed. It will be added to the values contained in the gista and its extended fields.

In 20-bit mode, the index register contains a 20-bit signed offset. Will be added to the value. This mode is used for JMP and JSR instructions.

Inherent mode instructions use the information available to the processor to determine the effective address. Operands (if any) are system resources and are not fetched from memory.

In accumulator offset mode, the contents of the 16-bit accumulator E are indexed. It is added to one of the registers and its extended field to form an effective address. In this mode Using index registers and accumulators in a loop without destroying the contents of accumulator D to come.

Relative mode is used for branch and long branch instructions. When the branch condition is satisfied, bytes Or the signed 2's complement offset of the word is added to the program counter. This PK fee The value of the new PC concatenated with the rud is the effective address.

Post-modified index mode is used with MOVB and MOVW instructions. XK After the effective address formed by IX and IX is used, an 8-bit signed offset is indexed. It is added to register X.

On the M68HC11 system, page 0 ($ 0000 to $ 00FF) using direct addressing mode. Allows for fast access to RAM or I / O mapped to. However, CPU16 is at the beginning of page 0

512 bytes are used for the exception vector. Instead of losing direct mode, ZK feel A reset initialization vector is assigned to the register and index register Z. The programmer is ZK Reconfigure the selected page in the field and use 8-bit unsigned index mode in IZ to You can access useful data structures in the address map.

2.6 Instruction set

CPU16 has an 8-bit instruction set. Also, a multi-page operation using pre-byte Supports de-map. This configuration allows an 8-bit opcode at the same time as the page 0 opcode. It is possible to fetch Perand. Programmatically 8-bit offset index add Taking full advantage of the lessing mode can significantly reduce the instruction space.

The instruction set for CPU16 is based on the instruction set for M68HC11, but on the 16-bit data bus. The opcode map has been repositioned for best performance. All instructions for M68HC11 It is also supported by CPU16, but the execution contents are not always the same. M68HC11 code Many will work on CPU16 if reassembled. Users have different instruction processing times, different arbitration methods, And it is necessary to take into account the new interrupt stack frame.

CPU16 has a wealth of 16-bit arithmetic instructions and logical instructions, including signed and unsigned multiplication and division. There is an order. Also added new instructions to support extended addressing and digital signal processing. Has been added.

Table summarizes the CPU16 instruction setShown in 9. The LSB of the condition code register has an effect Not listed because there are few instructions to receive. Note to instructions that affect interrupt masks and PK fields It is attached.

Table 9 Instruction set summary

NEMO	Operation	Explanation	Addressin	order			Condition code							
Nick			Mode	Opcode	Operand	Cycle S		MV	H	EV	N	Z	V	C
ABA	Add B to A	(A) + (B) ⇒ A	INH	370B	——	2	——	——	△	——	△	△	△	△
ABX	Add B to X	(XK: IX) + (000: B) ⇒ XK: IX	INH	374F	——	2	——	——	——	——	—	—	—	—
ABY	Add B to Y	(YK: IY) + (000: B) ⇒ YK: IY	INH	375F	——	2	——	——	——	——	—	—	—	—
ABZ	Add B to Z	(ZK: IZ) + (000: B) ⇒ ZK: IZ	INH	376F	——	2	——	——	——	——	—	—	—	—
ACE	Add E to AM [31:15]	(AM [31:15]) + (E) ⇒ AM	INH	3722	——	2	——	△	——	△	—	—	—	—
ACED	Add concatenated E and D to AM	(E: D) + (AM) ⇒ AM	INH	3723	——	Four	——	△	——	△	—	—	—	—
			IND8, X	43	ff	6								
			IND8, Y	53	ff	6								
			IND8, Z	63 63	ff	6								
			IMM8	73	ii	2								
			IND16, X	1743	gggg	6								
ADCA	Add with Carry to A	(A) + (M) + C ⇒ A	IND16, Y	1753	gggg	6	——	——	△	——	△	△	△	△
			IND16, Z	1763	gggg	6								
			EXT	1773	hh ll	6								
			E, X	2743	——	6								

			E, Y	2753	——	6								
			E, Z	2763	——	6								
ADCB	Add with Carry to B	(B) + (M) + C ⇒ B	IND8, X	C3	ff	6	——	——	△	——	△	△	△	△
			IND8, Y	D3	ff	6								
			IND8, Z	E3	ff	6								
			IMM8	F3	ii	2								
			E, X	27C3	——	6								
			E, Y	27D3	——	6								
			E, Z	27E3	——	6								
			IND16, X	17C3	gggg	6								
			IND16, Y	17D3	gggg	6								
			IND16, Z	17E3	gggg	6								
			EXT	17F3	hh ll	6								
ADCD	Add with Carry to D	(D) + (M: M + 1) + C ⇒ D	IND8, X	83	ff	6	——	——	——	——	△	△	△	△
			IND8, Y	93	ff	6								
			IND8, Z	A3	ff	6								
			E, X	2783	——	6								
			E, Y	2793	——	6								
			E, Z	27A3	——	6								
			IMM16	37B3	jj kk	Four								
			IND16, X	37C3	gggg	6								
			IND16, Y	37D3	gggg	6								
			IND16, Z	37E3	gggg	6								
			EXT	37F3	hh ll	6								
ADCE	Add with Carry to E	(E) + (M: M + 1) + C ⇒ E	IMM16	3733	jj kk	Four	——	——	——	——	△	△	△	△
			IND16, X	3743	gggg	6								
			IND16, Y	3753	gggg	6								
			IND16, Z	3763	gggg	6								
			EXT	3773	hh ll	6								

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code							
				Opcode	Operand	Cycle	S	MV	H	EV	N	Z	V	C
ADDA	Add to A	(A) + (M) ⇒ A	IND8, X	41 41	ff	6	——	——	△	——	△	△	△	△
			IND8, Y	51	ff	6								
			IND8, Z	61	ff	6								
			IMM8	71 71	ii	2								
			E, X	2741	——	6								
			E, Y	2751	——	6								
			E, Z	2761	——	6								
			IND16, X	1741	gggg	6								
			IND16, Y	1751	gggg	6								
			IND16, Z	1761	gggg	6								
			EXT	1771	hh ll	6								
ADDB	Add to B	(B) + (M) ⇒ B	IND8, X	C1	ff	6	——	——	△	——	△	△	△	△
			IND8, Y	D1	ff	6								
			IND8, Z	E1	ff	6								
			IMM8	F1	ii	2								
			E, X	27C1	——	6								
			E, Y	27D1	——	6								
			E, Z	27E1	——	6								
			IND16, X	17C1	gggg	6								
			IND16, Y	17D1	gggg	6								
			IND16, Z	17E1	gggg	6								
			EXT	17F1	hh ll	6								
ADDD	Add to D	(D) + (M: M + 1) ⇒ D	IND8, X	81	ff	6	——	——	——	——	△	△	△	△
			IND8, Y	91	ff	6								
			IND8, Z	A1	ff	6								
			IMM8	FC	ii	2								
			E, X	2781	——	6								
			E, Y	2791	——	6								
			E, Z	27A1	——	6								
			IMM16	37B1	jj kk	Four								
			IND16, X	37C1	gggg	6								

			IND16, Y	37D1	gggg	6	
			IND16, Z	37E1	gggg	6	
			EXT	37F1	hh ll	6	
ADDE	Add to E	(E) + (M: M + 1) ⇒ E	IMM8	7C	ii	2	—— —— —— —— △ △ △ △
			IMM16	3731	jj kk	Four	
			IND16, X	3741	gggg	6	
			IND16, Y	3751	gggg	6	
			IND16, Z	3761	gggg	6	
			EXT	3771	hh ll	6	
ADE	Add D to E	(E) + (D) ⇒ E	INH	2778	——	2	—— —— —— —— △ △ △ △
ADX	Add D to X	(XK: IX) + («D) ⇒ XK: IX	INH	37CD	——	2	—— —— —— —— ————
ADY	Add D to Y	(YK: IY) + («D) ⇒ YK: IY	INH	37DD	——	2	—— —— —— —— ————
ADZ	Add D to Z	(ZK: IZ) + («D) ⇒ ZK: IZ	INH	37ED	——	2	—— —— —— —— ————
AEX	Add E to X	(XK: IX) + («E) ⇒ XK: IX	INH	374D	——	2	—— —— —— —— ————
AEY	Add E to Y	(YK: IY) + («E) ⇒ YK: IY	INH	375D	——	2	—— —— —— —— ————
AEZ	Add E to Z	(ZK: IZ) + («E) ⇒ ZK: IZ	INH	376D	——	2	—— —— —— —— ————
AIS	Add Immediate Data to SP	SK: SP + «IMM ⇒ SK: SP	IMM8	3F	ii	2	—— —— —— —— ————
			IMM16	373F	jj kk	Four	
AIX	Add Immediate Value to X	XK: IX + «IMM ⇒ XK: IX	IMM8	3C	ii	2	—— —— —— —— —△——
			IMM16	373C	jj kk	Four	
AIY	Add Immediate Value to Y	YK: IY + «IMM ⇒ YK: IY	IMM8	3D	ii	2	—— —— —— —— —△——
			IMM16	373D	jj kk	Four	

Page 49

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
AIZ	Add Immediate Value to Z	ZK: IZ + «IMM ⇒ ZK: IZ	IMM8	3E	ii	2	—— —— —— —— —△——
			IMM16	373E	jj kk	Four	
			IND8, X	46	ff	6	
			IND8, Y	56	ff	6	
			IND8, Z	66 66	ff	6	
			IMM8	76 76	ii	2	
			IND16, X	1746	gggg	6	
ANDA	AND A	(A) • (M) ⇒ A	IND16, Y	1756	gggg	6	—— —— —— —— △△ 0 ——
			IND16, Z	1766	gggg	6	
			EXT	1776	hh ll	6	
			E, X	2746	——	6	
			E, Y	2756	——	6	
			E, Z	2766	——	6	
			IND8, X	C6	ff	6	
			IND8, Y	D6	ff	6	
			IND8, Z	E6	ff	6	
			IMM8	F6	ii	2	
			IND16, X	17C6	gggg	6	
ANDB	AND B	(B) • (M) ⇒ B	IND16, Y	17D6	gggg	6	—— —— —— —— △△ 0 ——
			IND16, Z	17E6	gggg	6	
			EXT	17F6	hh ll	6	
			E, X	27C6	——	6	
			E, Y	27D6	——	6	
			E, Z	27E6	——	6	
			IND8, X	86	ff	6	
			IND8, Y	96	ff	6	
			IND8, Z	A6	ff	6	
			E, X	2786	——	6	
			E, Y	2796	——	6	
ANDD	AND D	(D) • (M: M + 1) ⇒ D	E, Z	27A6	——	6	—— —— —— —— △△ 0 ——
			IMM16	37B6	jj kk	Four	
			IND16, X	37C6	gggg	6	
			IND16, Y	37D6	gggg	6	
			IND16, Z	37E6	gggg	6	
			EXT	37F6	hh ll	6	
			IMM16	3736	jj kk	Four	
			IND16, X	3746	gggg	6	
ANDE	AND E	(E) • (M: M + 1) ⇒ E	IND16, Y	3756	gggg	6	—— —— —— —— △△ 0 ——
			IND16, Z	3766	gggg	6	
			EXT	3776	hh ll	6	
ANDP1	AND CCR	(CCR) • IMM16 ⇒ CCR	IMM16	373A	jj kk	Four	△ △ △ △ △ △ △ △
			IND8, X	04 04	ff	8	

ASL	Arithmetic Shift Left	C b7 b0 0	IND8, Y	14	ff	8	—— —— —— —— —— △ △ △ △
			IND8, Z	twenty four	ff	8	
			IND16, X	1704	gggg	8	
			IND16, Y	1714	gggg	8	
			IND16, Z	1724	gggg	8	
			EXT	1734	hh ll	8	
ASLA	Arithmetic Shift Left A	C b7 b0 0	INH	3704	——	2	—— —— —— —— —— △ △ △ △
ASLB	Arithmetic Shift Left B	C b7 b0 0	INH	3714	——	2	—— —— —— —— —— △ △ △ △

Page 50

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
ASLD	Arithmetic Shift Left D	C b15 b0 0	INH	27F4	——	2	—— —— —— —— —— △ △ △ △
ASLE	Arithmetic Shift Left E	C b15 b0 0	INH	2774	——	2	—— —— —— —— —— △ △ △ △
ASLM	Arithmetic Shift Left AM	C b35 b0 0	INH	27B6	——	Four	—— △ —— △ △ —— △
ASLW	Arithmetic Shift Left Word	C b15 b0 0	IND16, X	2704	gggg	8	—— —— —— —— —— △ △ △ △
			IND16, Y	2714	gggg	8	
			IND16, Z	2724	gggg	8	
			EXT	2734	hh ll	8	
ASR	Arithmetic Shift Right	b7 b0 C	IND8, X	0D	ff	8	—— —— —— —— —— △ △ △ △
			IND8, Y	1D	ff	8	
			IND8, Z	2D	ff	8	
			IND16, X	170D	gggg	8	
			IND16, Y	171D	gggg	8	
			IND16, Z	172D	gggg	8	
			EXT	173D	hh ll	8	
ASRA	Arithmetic Shift Right A	b7 b0 C	INH	370D	——	2	—— —— —— —— —— △ △ △ △
ASRB	Arithmetic Shift Right B	b7 b0 C	INH	371D	——	2	—— —— —— —— —— △ △ △ △
ASRD	Arithmetic Shift Right D	b15 b0 C	INH	27FD	——	2	—— —— —— —— —— △ △ △ △
ASRE	Arithmetic Shift Right E	b15 b0 C	INH	277D	——	2	—— —— —— —— —— △ △ △ △
ASRM	Arithmetic Shift Right AM	b35 b0 C	INH	27BA	——	Four	—— —— —— △ △ —— △
ASRW	Arithmetic Shift Right Word	b15 b0 C	IND16, X	270D	gggg	8	—— —— —— —— —— △ △ △ △
			IND16, Y	271D	gggg	8	
			IND16, Z	272D	gggg	8	
			EXT	273D	hh ll	8	
BCC4	Branch if Carry Clear	If C = 0, branch	REL8	B4	rr	6, 2	—— —— —— —— —— ————
BCLR	Clear Bit (s)	(M) • (Mask) ⇒ M	IND16, X	08 08	mm gggg	8	—— —— —— —— —— △△ 0 ——
			IND16, Y	18 18	mm gggg	8	
			IND16, Z	28 28	mm gggg	8	
			EXT	38	mm hh ll	8	
			IND8, X	1708	mm ff	8	
			IND8, Y	1718	mm ff	8	
			IND8, Z	1728	mm ff	8	
BCLRW	Clear Bit (s) Word	(M: M + 1) • (Mask) ⇒ M: M + 1	IND16, X	2708	gggg mmmm	Ten	—— —— —— —— —— △△ 0 ——
			IND16, Y	2718	gggg mmmm	Ten	
			IND16, Z	2728	gggg mmmm	Ten	
			EXT	2738	hh ll mmmm	Ten	

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
BCS4	Branch if Carry Set	If C = 1, branch	REL8	B5	rr	6, 2	—— —— —— —— ————
BEQ4	Branch if Equal	If Z = 1, branch	REL8	B7	rr	6, 2	—— —— —— —— ————
BGE4	Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL8	BC	rr	6, 2	—— —— —— —— ————
BGND	Enter Background Debug Mode	If BDM enabled enter BDM; else, illegal instruction	INH	37A6	——	——	—— —— —— —— ————
BGT 4	Branch if Greater Than Zero	If Z + (N ⊕ V) = 0, branch	REL8	BE	rr	6, 2	—— —— —— —— ————
BHI 4	Branch if Higher	If C + Z = 0, branch	REL8	B2	rr	6, 2	—— —— —— —— ————
BITA	Bit Test A	(A) • (M)	IND8, X	49	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	59	ff	6	
			IND8, Z	69	ff	6	
			IMM8	79 79	ii	2	
			IND16, X	1749	gggg	6	
			IND16, Y	1759	gggg	6	
			IND16, Z	1769	gggg	6	
			EXT	1779	hh ll	6	
			E, X	2749	——	6	
			E, Y	2759	——	6	
			E, Z	2769	——	6	
BITB	Bit Test B	(B) • (M)	IND8, X	C9	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	D9	ff	6	
			IND8, Z	E9	ff	6	
			IMM8	F9	ii	2	
			IND16, X	17C9	gggg	6	
			IND16, Y	17D9	gggg	6	
			IND16, Z	17E9	gggg	6	
			EXT	17F9	hh ll	6	
			E, X	27C9	——	6	
			E, Y	27D9	——	6	
			E, Z	27E9	——	6	
BLE 4	Branch if Less Than or Equal to Zero	If Z + (N ⊕ V) = 1, branch	REL8	BF	rr	6, 2	—— —— —— —— ————
BLS4	Branch if Lower or Same	If C + Z = 1, branch	REL8	B3	rr	6, 2	—— —— —— —— ————
BLT4	Branch if Less Than Zero	If N ⊕ V = 1, branch	REL8	BD	rr	6, 2	—— —— —— —— ————
BMI 4	Branch if Minus	If N = 1, branch	REL8	BB	rr	6, 2	—— —— —— —— ————
BNE 4	Branch if Not Equal	If Z = 0, branch	REL8	B6	rr	6, 2	—— —— —— —— ————
BPL4	Branch if Plus	If N = 0, branch	REL8	BA	rr	6, 2	—— —— —— —— ————
BRA	Branch Always	If 1 = 1, branch	REL8	B0	rr	6	—— —— —— —— ————
BRCLR4	Branch if Bit (s) Clear	If (M) • (Mask) = 0, branch	IND8, X	CB	mm ff rr	10, 12	—— —— —— —— ————
			IND8, Y	DB	mm ff rr	10, 12	
			IND8, Z	EB	mm ff rr	10, 12	
			IND16, X	0A	mm gggg rrrr	10, 14	
			IND16, Y	1A	mm gggg rrrr	10, 14	
			IND16, Z	2A	mm gggg rrrr	10, 14	
			EXT	3A	mm hh ll rrrr	10, 14	
BRN	Branch Never Never	If 1 = 0, branch	REL8	B1	rr	2	—— —— —— —— ————

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
BRSET4	Branch if Bit (s) Set	If (M) • (Mask) = 0, branch	IND8, X	8B	mm ff rr	10, 12	—— —— —— —— ————
			IND8, Y	9B	mm ff rr	10, 12	
			IND8, Z	AB	mm ff rr	10, 12	
			IND16, X	0B	mm gggg rrrr	10, 14	
			IND16, Y	1B	mm gggg rrrr	10, 14	
			IND16, Z	2B	mm gggg rrrr	10, 14	
			EXT	3B	mm hh ll rrrr	10, 14	
BSET	Set Bit (s)	(M) • (Mask) ⇒ M	IND16, X	09 09	mm gggg	8	—— —— —— —— △△ 0 ——
			IND16, Y	19 19	mm gggg	8	
			IND16, Z	29	mm gggg	8	
			EXT	39 39	mm hh ll	8	
			IND8, X	1709	mm ff	8	
			IND8, Y	1719	mm ff	8	
			IND8, Z	1729	mm ff	8	
BSETW	Set Bit (s) in Word	(M: M + 1) • (Mask) ⇒ M: M + 1	IND16, X	2709	gggg mmmm	Ten	—— —— —— —— △△ 0 ——
			IND16, Y	2719	gggg mmmm	Ten	
			IND16, Z	2729	gggg mmmm	Ten	
			EXT	2739	hh ll mmmm	Ten	
BSR	Branch to Subroutine	(PK: PC) − 2 ⇒ PK: PC Push (PC) (SK: SP) – 2 ⇒ SK: SP Push (CCR) (SK: SP) – 2 ⇒ SK: SP (PK: PC) + Offset ⇒ PK: PC	REL8	36	rr	Ten	—— —— —— —— ————
BVC4	Branch if Overflow Clear	If V = 0, branch	REL8	B8	rr	6, 2	—— —— —— —— ————
BVS4	Branch if Overflow Set	If V = 1, branch	REL8	B9	rr	6, 2	—— —— —— —— ————
CBA	Compare A to B	(A) – (B)	INH	371B	——	2	—— —— —— —— △ △ △ △
CLR	Clear Memory	$ 00 ⇒ M	IND8, X	05 05	ff	Four	—— —— —— —— 0 1 0 0
			IND8, Y	15	ff	Four	
			IND8, Z	twenty five	ff	Four	
			IND16, X	1705	gggg	6	
			IND16, Y	1715	gggg	6	
			IND16, Z	1725	gggg	6	
			EXT	1735	hh ll	6	
CLRA	Clear A	$ 00 ⇒ A	INH	3705	——	2	—— —— —— —— 0 1 0 0
CLRB	Clear B	$ 00 ⇒ B	INH	3715	——	2	—— —— —— —— 0 1 0 0
CLRD	Clear D	$ 0000 ⇒ D	INH	27F5	——	2	—— —— —— —— 0 1 0 0
CLRE	Clear E	$ 0000 ⇒ E	INH	2775	——	2	—— —— —— —— 0 1 0 0
CLRM	Clear AM	$ 000000000 ⇒ AM [32: 0]	INH	27B7	——	2	—— 0 —— 0 ————
CLRW	Clear Memory Word	$ 0000 ⇒ M: M + 1	IND16, X	2705	gggg	6	—— —— —— —— 0 1 0 0
			IND16, Y	2715	gggg	6	
			IND16, Z	2725	gggg	6	
			EXT	2735	hh ll	6	

Page 53

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
CMPA	Compare A to Memory	(A) – (M)	IND8, X	48	ff	6	—— —— —— —— △ △ △ △
			IND8, Y	58	ff	6	
			IND8, Z	68 68	ff	6	
			IMM8	78 78	ii	2	
			IND16, X	1748	gggg	6	
			IND16, Y	1758	gggg	6	
			IND16, Z	1768	gggg	6	

			EXT	1778	hh ll	6	
			E, X	2748	— —	6	
			E, Y	2758	— —	6	
			E, Z	2768	— —	6	
			IND8, X	C8	ff	6	
			IND8, Y	D8	ff	6	
			IND8, Z	E8	ff	6	
			IMM8	F8	ii	2	
			IND16, X	17C8	gggg	6	
CMPB	Compare B to Memory	(B) – (M)	IND16, Y	17D8	gggg	6	—— —— —— —— △△△△
			IND16, Z	17E8	gggg	6	
			EXT	17F8	hh ll	6	
			E, X	27C8	— —	6	
			E, Y	27D8	— —	6	
			E, Z	27E8	— —	6	
			IND8, X	00 00	ff	8	
			IND8, Y	Ten	ff	8	
			IND8, Z	20	ff	8	
COM	One's Complement	$ FF – (M) ⇒ M	IND16, X	1700	gggg	8	—— —— —— —— △△ 0 1
			IND16, Y	1710	gggg	8	
			IND16, Z	1720	gggg	8	
			EXT	1730	hh ll	8	
COMA	One's Complement A	$ FF – (A) ⇒ A	INH	3700	— —	2	—— —— —— —— △△ 0 1
COMB	One's Complement B	$ FF – (B) ⇒ B	INH	3710	— —	2	—— —— —— —— △△ 0 1
COMD	One's Complement D	$ FFFF – (D) ⇒ D	INH	27F0	— —	2	—— —— —— —— △△ 0 1
COME	One's Complement E	$ FFFF – (E) ⇒ E	INH	2770	— —	2	—— —— —— —— △△ 0 1
			IND16, X	2700	gggg	8	
COMW	One's Complement Word	$ FFFF – M: M + 1 ⇒ M: M + 1	IND16, Y	2710	gggg	8	—— —— —— —— △△ 0 1
			IND16, Z	2720	gggg	8	
			EXT	2730	hh ll	8	
			IND8, X	88	ff	6	
			IND8, Y	98	ff	6	
			IND8, Z	A8	ff	6	
			E, X	2788	— —	6	
			E, Y	2798	— —	6	
CPD	Compare D to Memory	(D) – (M: M + 1)	E, Z	27A8	— —	6	—— —— —— —— △△△△
			IMM16	37B8	jj kk	Four	
			IND16, X	37C8	gggg	6	
			IND16, Y	37D8	gggg	6	
			IND16, Z	37E8	gggg	6	
			EXT	37F8	hh ll	6	
			IMM16	3738	jj kk	Four	
			IND16, X	3748	gggg	6	
CPE	Compare E to Memory	(E) – (M: M + 1)	IND16, Y	3758	gggg	6	—— —— —— —— △△△△
			IND16, Z	3768	gggg	6	
			EXT	3778	hh ll	6	

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
			IND8, X	4F	ff	6	
			IND8, Y	5F	ff	6	
			IND8, Z	6F	ff	6	
			IND16, X	174F	gggg	6	
CPS	Compare SP to Memory	(SP) – (M: M + 1)	IND16, Y	175F	gggg	6	—— —— —— —— △△△△
			IND16, Z	176F	gggg	6	
			EXT	177F	hh ll	6	
			IMM16	377F	jj kk	Four	
			IND8, X	4C	ff	6	
			IND8, Y	5C	ff	6	
			IND8, Z	6C	ff	6	
			IND16, X	174C	gggg	6	
CPX	Compare IX to Memory	(IX) – (M: M + 1)	IND16, Y	175C	gggg	6	—— —— —— —— △△△△
			IND16, Z	176C	gggg	6	
			EXT	177C	hh ll	6	
			IMM16	377C	jj kk	Four	
			IND8, X	4D	ff	6	
			IND8, Y	5D	ff	6	

CPY	Compare IY to Memory	(IY) – (M: M + 1)	IND8, Z	6D	ff	6	—— —— —— —— △ △ △ △
			IND16, X	174D	gggg	6	
			IND16, Y	175D	gggg	6	
			IND16, Z	176D	gggg	6	
			EXT	177D	hh ll	6	
			IMM16	377D	jj kk	Four	
CPZ	Compare IZ to Memory	(IZ) – (M: M + 1)	IND8, X	4E	ff	6	—— —— —— —— △ △ △ △
			IND8, Y	5E	ff	6	
			IND8, Z	6E	ff	6	
			IND16, X	174E	gggg	6	
			IND16, Y	175E	gggg	6	
			IND16, Z	176E	gggg	6	
			EXT	177E	hh ll	6	
			IMM16	377E	jj kk	Four	
DAA	Decimal Adjust A	(A) 10	INH	3721	——	2	—— —— —— —— △ △ U △
DEC	Decrement Memory	(M) – $ 01 ⇒ M	IND8, X	01 01	ff	8	—— —— —— —— △ △ △ —
			IND8, Y	11 11	ff	8	
			IND8, Z	twenty one	ff	8	
			IND16, X	1701	gggg	8	
			IND16, Y	1711	gggg	8	
			IND16, Z	1721	gggg	8	
			EXT	1731	hh ll	8	
DECA	Decrement A	(A) – $ 01 ⇒ A	INH	3701	——	2	—— —— —— —— △ △ △ —
DECB	Decrement B	(B) – $ 01 ⇒ B	INH	3711	——	2	—— —— —— —— △ △ △ —
DECW	Decrement Memory Word	(M: M + 1) – $ 0001 ⇒ M: M + 1	IND16, X	2701	gggg	8	—— —— —— —— △ △ △ —
			IND16, Y	2711	gggg	8	
			IND16, Z	2721	gggg	8	
			EXT	2731	hh ll	8	
EDIV	Extended Unsigned Divide	(E: D) / (IX) Quotient ⇒ IX Remainder ⇒ D	INH	3728	——	twenty four	—— —— —— —— △ △ △ △
EDIVS	Extended Signed Divide	(E: D) / (IX) Quotient ⇒ IX Remainder ⇒ ACCD	INH	3729	——	38	—— —— —— —— △ △ △ △
EMUL	Extended Unsigned Multiply	(E) ∗ (D) ⇒ E: D	INH	3725	——	Ten	—— —— —— —— △ △ — △
EMULS	Extended Signed Multiply	(E) ∗ (D) ⇒ E: D	INH	3726	——	8	—— —— —— —— △ △ — △

Page 55

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
EORA	Exclusive OR A	(A) ⊕ (M) ⇒ A	IND8, X	44	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	54	ff	6	
			IND8, Z	64	ff	6	
			IMM8	74 74	ii	2	
			IND16, X	1744	gggg	6	
			IND16, Y	1754	gggg	6	
			IND16, Z	1764	gggg	6	
			EXT	1774	hh ll	6	
			E, X	2744	——	6	
			E, Y	2754	——	6	
			E, Z	2764	——	6	
EORB	Exclusive OR B	(B) ⊕ (M) ⇒ B	IND8, X	C4	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	D4	ff	6	
			IND8, Z	E4	ff	6	
			IMM8	F4	ii	2	
			IND16, X	17C4	gggg	6	
			IND16, Y	17D4	gggg	6	
			IND16, Z	17E4	gggg	6	
			EXT	17F4	hh ll	6	
			E, X	27C4	——	6	
			E, Y	27D4	——	6	
			E, Z	27E4	——	6	
EORD	Exclusive OR D	(D) ⊕ (M: M + 1) ⇒ D	IND8, X	84 84	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	94	ff	6	
			IND8, Z	A4	ff	6	
			E, X	2784	——	6	
			E, Y	2794	——	6	
			E, Z	27A4	——	6	
			IMM16	37B4	jj kk	Four	
			IND16, X	37C4	gggg	6	

			IND16, Y	37D4	gggg	6	
			IND16, Z	37E4	gggg	6	
			EXT	37F4	hh ll	6	
			IMM16	3734	jj kk	Four	
			IND16, X	PLL CA	gggg	6	
EORE	Exclusive OR E	(E) ⊕ (M: M + 1) ⇒ E	IND16, Y	3754	gggg	6	—— —— —— —— △△ 0 ——
			IND16, Z	3764	gggg	6	
			EXT	3774	hh ll	6	
FDIV	Fractional Unsigned Divide	(D) / (IX) ⇒ IX Remainder ⇒ D	INH	372B	——	twenty two	—— —— —— —— —△△△
FMULS	Fractional Signed Multiply	(E) ∗ (D) ⇒ E: D [31: 1] 0 ⇒ D [0]	INH	3727	——	8	—— —— —— —— △△△△
IDIV	Integer Divide	(D) / (IX) ⇒ IX; Remainder ⇒ D	INH	372A	——	twenty two	—— —— —— —— —△ 0 △
			IND8, X	03 03	ff	8	
			IND8, Y	13	ff	8	
			IND8, Z	twenty three	ff	8	
INC	Increment Memory	(M) + $ 01 ⇒ M	IND16, X	1703	gggg	8	—— —— —— —— △△△—
			IND16, Y	1713	gggg	8	
			IND16, Z	1723	gggg	8	
			EXT	1733	hh ll	8	
INCA	Increment A	(A) + $ 01 ⇒ A	INH	3703	——	2	—— —— —— —— △△△—
INCB	Increment B	(B) + $ 01 ⇒ B	INH	3713	——	2	—— —— —— —— △△△—

Page 56

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
			IND16, X	2703	gggg	8	
INCW	Increment Memory Word	(M: M + 1) + $ 0001 ⇒ M: M + 1	IND16, Y	2713	gggg	8	—— —— —— —— △△△—
			IND16, Z	2723	gggg	8	
			EXT	2733	hh ll	8	
		< Ea > ⇒ PK: PC	IND20, X	4B	zg gggg	8	
			IND20, Y	5B	zg gggg	8	
JMP	Jump	Note: Index cash register K when using the star	IND20, Z	6B	zg gggg	8	—— —— —— —— ————
		The field is ignored vinegar.	EXT20	7A	zb hh ll	6	
		Push (PC)	IND20, X	89	zg gggg	12	
		(SK: SP) – 2 ⇒ SK: SP Push (CCR)	IND20, Y	99	zg gggg	12	
		(SK: SP) – 2 ⇒ SK: SP < Ea > ⇒ PK: PC	IND20, Z	A9	zg gggg	12	
JSR	Jump to Subroutine						—— —— —— —— ————
		Note: Index cash register K when using the star The field is ignored vinegar.	EXT20	FA	zb hh ll	Ten	
LBCC4	Long Branch if Carry Clear	If C = 0, branch	REL16	3784	rrrr	6, 4	—— —— —— —— ————
LBCS4	Long Branch if Carry Set	If C = 1, branch	REL16	3785	rrrr	6, 4	—— —— —— —— ————
LBEQ4	Long Branch if Equal	If Z = 1, branch	REL16	3787	rrrr	6, 4	—— —— —— —— ————
LBEV4	Long Branch if EV Set	If EV = 1, branch	REL16	3791	rrrr	6, 4	—— —— —— —— ————
LBGE4	Long Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL16	378C	rrrr	6, 4	—— —— —— —— ————
LBGT 4	Long Branch if Greater Than Zero	If Z; (N ⊕ V) = 0, branch	REL16	378E	rrrr	6, 4	—— —— —— —— ————
LBHI 4	Long Branch if Higher	If C; Z = 0, branch	REL16	3782	rrrr	6, 4	—— —— —— —— ————
LBLE 4	Long Branch if Less Than or Equal to Zero	If Z; (N ⊕ V) = 1, branch	REL16	378F	rrrr	6, 4	—— —— —— —— ————
LBLS4	Long Branch if Lower or Same	If C; Z = 1, branch	REL16	3783	rrrr	6, 4	—— —— —— —— ————
LBLT 4	Long Branch if Less Than Zero	If N ⊕ V = 1, branch	REL16	378D	rrrr	6, 4	—— —— —— —— ————
LBMI 4	Long Branch if Minus	If N = 1, branch	REL16	378B	rrrr	6, 4	—— —— —— —— ————
LBMV4	Long Branch if MV Set	If MV = 1, branch	REL16	3790	rrrr	6, 4	—— —— —— —— ————
LBNE 4	Long Branch if Not Equal	If Z = 0, branch	REL16	3786	rrrr	6, 4	—— —— —— —— ————
LBPL4	Long Branch if Plus	If N = 0, branch	REL16	378A	rrrr	6, 4	—— —— —— —— ————
LBRA	Long Branch Always	If 1 = 1, branch	REL16	3780	rrrr	6	—— —— —— —— ————
LBRN	Long Branch Never	If 1 = 0, branch	REL16	3781	rrrr	Four	—— —— —— —— ————
LBSR	Long Branch to Subroutine	Push (PC) (SK: SP) – 2 ⇒ SK: SP Push (CCR) (SK: SP) – 2 ⇒ SK: SP (PK: PC) + Offset ⇒ PK: PC	REL16	27F9	rrrr	Ten	—— —— —— —— ————

LBVC4	Long Branch if Overflow Clear	If V = 0, branch	REL16	3788	rrrr	6, 4	—— —— —— —— ————
LBVS4	Long Branch if Overflow Set	If V = 1, branch	REL16	3789	rrrr	6, 4	—— —— —— —— ————

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
LDAA	Load A	(M) ⇒ A	IND8, X	45 45	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	55 55	ff	6	
			IND8, Z	65 65	ff	6	
			IMM8	75	ii	2	
			IND16, X	1745	gggg	6	
			IND16, Y	1755	gggg	6	
			IND16, Z	1765	gggg	6	
			EXT	1775	hh ll	6	
			E, X	2745	——	6	
			E, Y	2755	——	6	
			E, Z	2765	——	6	
LDAB	Load B	(M) ⇒ B	IND8, X	C5	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	D5	ff	6	
			IND8, Z	E5	ff	6	
			IMM8	F5	ii	2	
			IND16, X	17C5	gggg	6	
			IND16, Y	17D5	gggg	6	
			IND16, Z	17E5	gggg	6	
			EXT	17F5	hh ll	6	
			E, X	27C5	——	6	
			E, Y	27D5	——	6	
			E, Z	27E5	——	6	
LDD	Load D	(M: M + 1) ⇒ D	IND8, X	85	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	95	ff	6	
			IND8, Z	A5	ff	6	
			E, X	2785	——	6	
			E, Y	2795	——	6	
			E, Z	27A5	——	6	
			IMM16	37B5	jj kk	Four	
			IND16, X	37C5	gggg	6	
			IND16, Y	37D5	gggg	6	
			IND16, Z	37E5	gggg	6	
			EXT	37F5	hh ll	6	
LDE	Load E	(M: M + 1) ⇒ E	IMM16	3735	jj kk	Four	—— —— —— —— △△ 0 ——
			IND16, X	3745	gggg	6	
			IND16, Y	3755	gggg	6	
			IND16, Z	3765	gggg	6	
			EXT	3775	hh ll	6	
LDED	Load Concatenated E and D	(M: M + 1) ⇒ E (M + 2: M + 3) ⇒ D	EXT	2771	hh ll	8	—— —— —— —— ————
LDHI	Initialize H and I	(M: M + 1) X ⇒ HR (M: M + 1) Y ⇒ IR	EXT	27B0	——	8	—— —— —— —— ————
LDS	Load SP	(M: M + 1) ⇒ SP	IND8, X	CF	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	DF	ff	6	
			IND8, Z	EF	ff	6	
			IND16, X	17CF	gggg	6	
			IND16, Y	17DF	gggg	6	
			IND16, Z	17EF	gggg	6	
			EXT	17FF	hh ll	6	
			IMM16	37BF	jj kk	Four	

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order Opcode	Operand	Cycle S	Condition code MV H EV N Z V C
LDX	Load IX	(M: M + 1) ⇒ IX	IND8, X	CC	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	DC	ff	6	
			IND8, Z	EC	ff	6	
			IND16, X	17CC	gggg	6	
			IND16, Y	17DC	gggg	6	
			IND16, Z	17EC	gggg	6	
			EXT	17FC	hh ll	6	
			IMM16	37BC	jj kk	Four	
LDY	Load IY	(M: M + 1) ⇒ IY	IND8, X	CD	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	DD	ff	6	
			IND8, Z	ED	ff	6	
			IND16, X	17CD	gggg	6	
			IND16, Y	17DD	gggg	6	
			IND16, Z	17ED	gggg	6	
			EXT	17FD	hh ll	6	
			IMM16	37BD	jj kk	Four	
LDZ	Load IZ	(M: M + 1) ⇒ IZ	IND8, X	CE	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	DE	ff	6	
			IND8, Z	EE	ff	6	
			IND16, X	17CE	gggg	6	
			IND16, Y	17DE	gggg	6	
			IND16, Z	17EE	gggg	6	
			EXT	17FE	hh ll	6	
			IMM16	37BE	jj kk	Four	
LPSTOP	Low Power Stop	If S then STOP else NOP	INH	27F1	——	4, 20	—— —— —— —— ————
LSR	Logical Shift Right	0 b7 b0 C	IND8, X	0F	ff	8	—— —— —— —— 0 △ △ △
			IND8, Y	1F	ff	8	
			IND8, Z	2F	ff	8	
			IND16, X	170F	gggg	8	
			IND16, Y	171F	gggg	8	
			IND16, Z	172F	gggg	8	
			EXT	173F	hh ll	8	
LSRA	Logical Shift Right A	0 b7 b0 C	INH	370F	——	2	—— —— —— —— 0 △ △ △
LSRB	Logical Shift Right B	0 b7 b0 C	INH	371F	——	2	—— —— —— —— 0 △ △ △
LSRD	Logical Shift Right D	0 b15 b0 C	INH	27FF	——	2	—— —— —— —— 0 △ △ △
LSRE	Logical Shift Right E	0 b15 b0 C	INH	277F	——	2	—— —— —— —— 0 △ △ △
LSRW	Logical Shift Right Word	0 b15 b0 C	IND16, X	270F	gggg	8	—— —— —— —— 0 △ △ △
			IND16, Y	271F	gggg	8	
			IND16, Z	272F	gggg	8	
			EXT	273F	hh ll	8	

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
MAC	Multiply and Accumulate Signed 16-Bit Fractions	(HR) ∗ (IR) ⇒ E: D (AM) + (E: D) ⇒ AM Qualified (IX) ⇒ IX Qualified (IY) ⇒ IY (HR) ⇒ IZ (M: M + 1) X ⇒ HR (M: M + 1) Y ⇒ IR	IMM8	7B	xoyo	12	— — △ — — △ — — △ —
			IXP to EXT	30	gg hh ll	8	
MOVB	Move Byte	(M1) ⇒ M2	EXT to IXP	32	gg hh ll	8	— — — — — — — — △△ 0 — —
			EXT to EXT	37FE	hh ll hh ll	Ten	
			IXP to EXT	31	gg hh ll	8	
MOVW	Move Word	(M: M + 11) ⇒ M: M + 12	EXT to IXP	33 33	gg hh ll	8	— — — — — — — — △△ 0 — —
			EXT to EXT	37FF	hh ll hh ll	Ten	
MUL	Multiply	(A) ∗ (B) ⇒ D	INH	3724	— —	Ten	— — — — — — — — — — — △
			IND8, X	02 02	ff	8	
			IND8, Y	12	ff	8	
			IND8, Z	twenty two	ff	8	
NEG	Negate Memory	$ 00 – (M) ⇒ M	IND16, X	1702	gggg	8	— — — — — — — — △ △ △ △
			IND16, Y	1712	gggg	8	
			IND16, Z	1722	gggg	8	
			EXT	1732	hh ll	8	
NEGA	Negate A	$ 00 – (A) ⇒ A	INH	3702	— —	2	— — — — — — — — △ △ △ △
NEGB	Negate B	$ 00 – (B) ⇒ B	INH	3712	— —	2	— — — — — — — — △ △ △ △
NEGD	Negate D	$ 0000 – (D) ⇒ D	INH	27F2	— —	2	— — — — — — — — △ △ △ △
NEGE	Negate E	$ 0000 – (E) ⇒ E	INH	2772	— —	2	— — — — — — — — △ △ △ △
			IND16, X	2702	gggg	8	
NEGW	Negate Memory Word	$ 0000 – (M: M + 1) ⇒ M: M + 1	IND16, Y	2712	gggg	8	— — — — — — — — △ △ △ △
			IND16, Z	2722	gggg	8	
			EXT	2732	hh ll	8	
NOP	Null Operation	— —	INH	274C	— —	2	— — — — — — — — — — — —
			IND8, X	47 47	ff	6	
			IND8, Y	57 57	ff	6	
			IND8, Z	67 67	ff	6	
			IMM8	77 77	ii	2	
			IND16, X	1747	gggg	6	
ORAA	OR A	(A); (M) ⇒ A	IND16, Y	1757	gggg	6	— — — — — — — — △△ 0 — —
			IND16, Z	1767	gggg	6	
			EXT	1777	hh ll	6	
			E, X	2747	— —	6	
			E, Y	2757	— —	6	
			E, Z	2767	— —	6	
			IND8, X	C7	ff	6	
			IND8, Y	D7	ff	6	
			IND8, Z	E7	ff	6	
			IMM8	F7	ii	2	
			IND16, X	17C7	gggg	6	
ORAB	OR B	(B); (M) ⇒ B	IND16, Y	17D7	gggg	6	— — — — — — — — △△ 0 — —
			IND16, Z	17E7	gggg	6	
			EXT	17F7	hh ll	6	
			E, X	27C7	— —	6	
			E, Y	27D7	— —	6	
			E, Z	27E7	— —	6	

Page 60

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
			IND8, X	87	ff	6	
			IND8, Y	97	ff	6	
			IND8, Z	A7	ff	6	
			E, X	2787	— —	6	
			E, Y	2797	— —	6	
ORD	OR D	(D); (M: M + 1) ⇒ D	E, Z	27A7	— —	6	— — — — — — — — △△ 0 — —

			IMM16	37B7	jj kk	Four	
			IND16, X	37C7	gggg	6	
			IND16, Y	37D7	gggg	6	
			IND16, Z	37E7	gggg	6	
			EXT	37F7	hh ll	6	
ORE	OR E	(E); (M: M + 1) ⇒ E	IMM16	3737	jj kk	Four	—— —— —— —— △△ 0 ——
			IND16, X	3747	gggg	6	
			IND16, Y	3757	gggg	6	
			IND16, Z	3767	gggg	6	
			EXT	3777	hh ll	6	
ORP 1	OR Condition Code Register	(CCR); IMM16 ⇒ CCR	IMM16	373B	jj kk	Four	△ △ △ △ △ △ △ △
PSHA	Push A	(SK: SP) + 1 ⇒ SK: SP Push (A) (SK: SP) – 2 ⇒ SK: SP	INH	3708	——	Four	—— —— —— —— ————
PSHB	Push B	(SK: SP) + 1 ⇒ SK: SP Push (B) (SK: SP) – 2 ⇒ SK: SP	INH	3718	——	Four	—— —— —— —— ————
PSHM	Push Multiple Registers Mask bits: 0 = D 1 = E 2 = IX 3 = IY 4 = IZ 5 = K 6 = CCR 7 = (reserved)	For mask bits 0 to 7: If mask bit set Push register (SK: SP) – 2 ⇒ SK: SP	IMM8	34	ii	4 + 2N N = number of iterations	—— —— —— —— ————
PSHMAC	Push MAC State	MAC Registers ⇒ Stack	INH	27B8	——	14	—— —— —— —— ————
PULA	Pull A	(SK: SP) + 2 ⇒ SK: SP Pull (A) (SK: SP) – 1 ⇒ SK: SP	INH	3709	——	6	—— —— —— —— ————
PULB	Pull B	(SK: SP) + 2 ⇒ SK: SP Pull (B) (SK: SP) – 1 ⇒ SK: SP	INH	3719	——	6	—— —— —— —— ————
PULM 1	Pull Multiple Registers Mask bits: 0 = CCR [15: 4] 1 = K 2 = IZ 3 = IY 4 = IX 5 = E 6 = D 7 = (reserved)	For mask bits 0 to 7: If mask bit set (SK: SP) + 2 ⇒ SK: SP Pull register	IMM8	35	ii	4 + 2 (N + 1)) N = number of iterations	△ △ △ △ △ △ △ △
PULMAC	Pull MAC State	Stack ⇒ MAC Registers	INH	27B9	——	16 16	—— —— —— —— ————
RMAC	Repeating Multiply and Accumulate Signed 16-Bit Fractions	Repeat until (E) < 0 (AM) + (H) ∗ (I) ⇒ AM Qualified (IX) ⇒ IX; Qualified (IY) ⇒ IY; (M: M + 1) X ⇒ H; (M: M + 1) Y ⇒ I (E) – 1 ⇒ E	IMM8	FB	xoyo	6 + 12 per per iteration	—— △ —— △ ————

Page 61

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
			IND8, X	0C	ff	8	
			IND8, Y	1C	ff	8	
			IND8, Z	2C	ff	8	
ROL	Rotate Left	C b7 b0	IND16, X	170C	gggg	8	—— —— —— —— △ △ △ △
			IND16, Y	171C	gggg	8	
			IND16, Z	172C	gggg	8	
			EXT	173C	hh ll	8	
ROLA	Rotate Left A	C b7 b0	INH	370C	——	2	—— —— —— —— △ △ △ △
ROLB	Rotate Left B	C b7 b0	INH	371C	——	2	—— —— —— —— △ △ △ △
ROLD	Rotate Left D	C b15 b0	INH	27FC	——	2	—— —— —— —— △ △ △ △

ROLE	Rotate Left E	C b15 b0	INH	277C	——	2	—— —— —— —— △ △ △ △
ROLW	Rotate Left Word	C b15 b0	IND16, X	270C	gggg	8	—— —— —— —— △ △ △ △
			IND16, Y	271C	gggg	8	
			IND16, Z	272C	gggg	8	
			EXT	273C	hh ll	8	
ROR	Rotate Right	b7 b0 C	IND8, X	0E	ff	8	—— —— —— —— △ △ △ △
			IND8, Y	1E	ff	8	
			IND8, Z	2E	ff	8	
			IND16, X	170E	gggg	8	
			IND16, Y	171E	gggg	8	
			IND16, Z	172E	gggg	8	
			EXT	173E	hh ll	8	
RORA	Rotate Right A	b7 b0 C	INH	370E	——	2	—— —— —— —— △ △ △ △
RORB	Rotate Right B	b7 b0 C	INH	371E	——	2	—— —— —— —— △ △ △ △
RORD	Rotate Right D	b15 b0 C	INH	27FE	——	2	—— —— —— —— △ △ △ △
RORE	Rotate Right E	b15 b0 C	INH	277E	——	2	—— —— —— —— △ △ △ △
RORW	Rotate Right Word	b15 b0 C	IND16, X	270E	gggg	8	—— —— —— —— △ △ △ △
			IND16, Y	271E	gggg	8	
			IND16, Z	272E	gggg	8	
			EXT	273E	hh ll	8	
RTI2	Return from Interrupt	(SK: SP) + 2 ⇒ SK: SP Pull CCR (SK: SP) + 2 ⇒ SK: SP Pull PC (PK: PC) – 6 ⇒ PK: PC	INH	2777	——	12	△ △ △ △ △ △ △ △
RTS3	Return from Subroutine	(SK: SP) + 2 ⇒ SK: SP Pull PK (SK: SP) + 2 ⇒ SK: SP Pull PC (PK: PC) – 2 ⇒ PK: PC	INH	27F7	——	12	—— —— —— —— ————
SBA	Subtract B from A	(A) – (B) ⇒ A	INH	370A	——	2	—— —— —— —— △ △ △ △

Page 62

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
SBCA	Subtract with Carry from A	(A) – (M) – C ⇒ A	IND8, X	42	ff	6	—— —— —— —— △ △ △ △
			IND8, Y	52 52	ff	6	
			IND8, Z	62	ff	6	
			IMM8	72	ii	2	
			IND16, X	1742	gggg	6	
			IND16, Y	1752	gggg	6	
			IND16, Z	1762	gggg	6	
			EXT	1772	hh ll	6	
			E, X	2742	——	6	
			E, Y	2752	——	6	
			E, Z	2762	——	6	
SBCB	Subtract with Carry from B	(B) – (M) – C ⇒ B	IND8, X	C2	ff	6	—— —— —— —— △ △ △ △
			IND8, Y	D2	ff	6	
			IND8, Z	E2	ff	6	
			IMM8	F2	ii	2	
			IND16, X	17C2	gggg	6	
			IND16, Y	17D2	gggg	6	
			IND16, Z	17E2	gggg	6	
			EXT	17F2	hh ll	6	
			E, X	27C2	——	6	
			E, Y	27D2	——	6	
			E, Z	27E2	——	6	
SBCD	Subtract with Carry from D	(D) – (M: M + 1) – C ⇒ D	IND8, X	82	ff	6	—— —— —— —— △ △ △ △
			IND8, Y	92	ff	6	
			IND8, Z	A2	ff	6	
			E, X	2782	——	6	
			E, Y	2792	——	6	
			E, Z	27A2	——	6	
			IMM16	37B2	jj kk	Four	

			IND16, X	37C2	gggg	6	
			IND16, Y	37D2	gggg	6	
			IND16, Z	37E2	gggg	6	
			EXT	37F2	hh ll	6	
SBCE	Subtract with Carry from E	(E) – (M: M + 1) – C ⇒ E	IMM16	3732	jj kk	Four	—— —— —— —— △△△△
			IND16, X	3742	gggg	6	
			IND16, Y	3752	gggg	6	
			IND16, Z	3762	gggg	6	
			EXT	3772	hh ll	6	
SDE	Subtract D from E	(E) – (D) ⇒ E	INH	2779	——	2	—— —— —— —— △△△△
STAA	Store A	(A) ⇒ M	IND8, X	4A	ff	Four	—— —— —— —— △△ 0 ——
			IND8, Y	5A	ff	Four	
			IND8, Z	6A	ff	Four	
			IND16, X	174A	gggg	6	
			IND16, Y	175A	gggg	6	
			IND16, Z	176A	gggg	6	
			EXT	177A	hh ll	6	
			E, X	274A	——	Four	
			E, Y	275A	——	Four	
			E, Z	276A	——	Four	

Page 63

Table 9 Instruction set summary (continued)

NEMO	Operation	Explanation	Addressin	order			Condition code
Nick			Mode	Opcode	Operand	Cycle S	MV H EV N Z V C
STAB	Store B	(B) ⇒ M	IND8, X	CA	ff	Four	—— —— —— —— △△ 0 ——
			IND8, Y	DA	ff	Four	
			IND8, Z	EA	ff	Four	
			IND16, X	17CA	gggg	6	
			IND16, Y	17DA	gggg	6	
			IND16, Z	17EA	gggg	6	
			EXT	17FA	hh ll	6	
			E, X	27CA	——	Four	
			E, Y	27DA	——	Four	
			E, Z	27EA	——	Four	
STD	Store D	(D) ⇒ M: M + 1	IND8, X	8A	ff	6	—— —— —— —— △△ 0 ——
			IND8, Y	9A	ff	6	
			IND8, Z	AA	ff	6	
			E, X	278A	——	6	
			E, Y	279A	——	6	
			E, Z	27AA	——	6	
			IND16, X	37CA	gggg	Four	
			IND16, Y	37DA	gggg	Four	
			IND16, Z	37EA	gggg	Four	
			EXT	37FA	hh ll	6	
STE	Store E	(E) ⇒ M: M + 1	IND16, X	374A	gggg	6	—— —— —— —— △△ 0 ——
			IND16, Y	375A	gggg	6	
			IND16, Z	376A	gggg	6	
			EXT	377A	hh ll	6	
STED	Store Concatenated D and E	(E) ⇒ M: M + 1 (D) ⇒ M + 2: M + 3	EXT	2773	hh ll	8	—— —— —— —— △△——
STS	Store SP	(SP) ⇒ M: M + 1	IND8, X	8F	ff	Four	—— —— —— —— △△ 0 ——
			IND8, Y	9F	ff	Four	
			IND8, Z	AF	ff	Four	
			IND16, X	178F	gggg	6	
			IND16, Y	179F	gggg	6	
			IND16, Z	17AF	gggg	6	
			EXT	17BF	hh ll	6	
STX	Store IX	(IX) ⇒ M: M + 1	IND8, X	8C	ff	Four	—— —— —— —— △△ 0 ——
			IND8, Y	9C	ff	Four	
			IND8, Z	AC	ff	Four	
			IND16, X	178C	gggg	6	
			IND16, Y	179C	gggg	6	
			IND16, Z	17AC	gggg	6	
			EXT	17BC	hh ll	6	
			IND8, X	8D	ff	Four	

STY	Store IY	(IY) ⇒ M: M + 1	IND8, Y	9D	ff	Four	— — — — — — — — △△ 0 — —
			IND8, Z	AD	ff	Four	
			IND16, X	178D	gggg	6	
			IND16, Y	179D	gggg	6	
			IND16, Z	17AD	gggg	6	
			EXT	17BF	hh ll	6	
STZ	Store Z	(IZ) ⇒ M: M + 1	IND8, X	8E	ff	Four	— — — — — — — — △△ 0 — —
			IND8, Y	9E	ff	Four	
			IND8, Z	AE	ff	Four	
			IND16, X	178E	gggg	6	
			IND16, Y	179E	gggg	6	
			IND16, Z	17AE	gggg	6	
			EXT	17BE	hh ll	6	

Page 64

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
SUBA	Subtract from A	(A) – (M) ⇒ A	IND8, X	40	ff	6	— — — — — — — — △ △ △ △
			IND8, Y	50	ff	6	
			IND8, Z	60	ff	6	
			IMM8	70	ii	2	
			IND16, X	1740	gggg	6	
			IND16, Y	1750	gggg	6	
			IND16, Z	1760	gggg	6	
			EXT	1770	hh ll	6	
			E, X	2740	— —	6	
			E, Y	2750	— —	6	
			E, Z	2760	— —	6	
SUBB	Subtract from B	(B) – (M) ⇒ B	IND8, X	C0	ff	6	— — — — — — — — △ △ △ △
			IND8, Y	D0	ff	6	
			IND8, Z	E0	ff	6	
			IMM8	F0	ii	2	
			IND16, X	17C0	gggg	6	
			IND16, Y	17D0	gggg	6	
			IND16, Z	17E0	gggg	6	
			EXT	17F0	hh ll	6	
			E, X	27C0	— —	6	
			E, Y	27D0	— —	6	
			E, Z	27E0	— —	6	
SUBD	Subtract from D	(D) – (M: M + 1) ⇒ D	IND8, X	80	ff	6	— — — — — — — — △ △ △ △
			IND8, Y	90	ff	6	
			IND8, Z	A0	ff	6	
			E, X	2780	— —	6	
			E, Y	2790	— —	6	
			E, Z	27A0	— —	6	
			IMM16	37B0	jj kk	Four	
			IND16, X	37C0	gggg	6	
			IND16, Y	37D0	gggg	6	
			IND16, Z	37E0	gggg	6	
			EXT	37F0	hh ll	6	
SUBE	Subtract from E	(E) – (M: M + 1) ⇒ E	IMM16	3730	jj kk	Four	— — — — — — — — △ △ △ △
			IND16, X	3740	gggg	6	
			IND16, Y	3750	gggg	6	
			IND16, Z	3760	gggg	6	
			EXT	3770	hh ll	6	
SWI	Software Interrupt	(PK: PC) + 2 ⇒ PK: PC Push (PC) (SK: SP) – 2 ⇒ SK: SP Push (CCR) (SK: SP) – 2 ⇒ SK: SP $ 0 ⇒ PK SWI Vector ⇒ PC	INH	3720	— —	16 16	— — — — — — — — — — — —
SXT	Sign Extend B into A	If B7 = 1 then A = $ FF else A = $ 00	INH	27F8	— —	2	— — — — — — — — △ △ — —
TAB	Transfer A to B	(A) ⇒ B	INH	3717	— —	2	— — — — — — — — △△ 0 — —
TAP	Transfer A to CCR	(A [7: 0]) ⇒ CCR [15: 8]	INH	37FD	— —	Four	△ △ △ △ △ △ △ △
TBA	Transfer B to A	(B) ⇒ A	INH	3707	— —	2	— — — — — — — — △△ 0 — —
TBEK	Transfer B to EK	(B) ⇒ EK	INH	27FA	— —	2	— — — — — — — — — — — —
TBSK	Transfer B to SK	(B) ⇒ SK	INH	379F	— —	2	— — — — — — — — — — — —
TBXK	Transfer B to XK	(B) ⇒ XK	INH	379C	— —	2	— — — — — — — — — — — —

TBYK	Transfer B to YK	(B) ⇒ YK	INH	379D	——	2	—— —— —— —— ————
TBZK	Transfer B to ZK	(B) ⇒ ZK	INH	379E	——	2	—— —— —— —— ————
TDE	Transfer D to E	(D) ⇒ E	INH	277B	——	2	—— —— —— —— △△ 0 ——

Page 65

Table 9 Instruction set summary (continued)

NEMO Nick	Operation	Explanation	Addressin Mode	order			Condition code
				Opcode	Operand	Cycle S	MV H EV N Z V C
TDMSK	Transfer D to XMSK: YMSK	(D [15: 8]) ⇒ X MASK (D [7: 0]) ⇒ Y MASK	INH	372F	——	2	—— —— —— —— ————
TDP1	Transfer D to CCR	(D) ⇒ CCR [15: 4]	INH	372D	——	Four	△ △ △ △ △△△△
TED	Transfer E to D	(E) ⇒ D	INH	27FB	——	2	—— —— —— —— △△ 0 ——
TEDM	Transfer E and D to AM [31: 0] Sign Extend AM	(D) ⇒ AM [15: 0] (E) ⇒ AM [31:16] AM [35:32] = AM31	INH	27B1	——	Four	—— 0 —— 0 ————
TEKB	Transfer EK to B	$ 0 ⇒ B [7: 4] (EK) ⇒ B [3: 0]	INH	27BB	——	2	—— —— —— —— ————
TEM	Transfer E to AM [31:16] Sign Extend AM Clear AM LSB	(E) ⇒ AM [31:16] $ 00 ⇒ AM [15: 0] AM [35:32] = AM31	INH	27B2	——	Four	—— 0 —— 0 ————
TMER	Transfer AM to E Rounded	Rounded (AM) ⇒ Temp If (SM • (EV; MV)) then Saturation ⇒ E else Temp [31:16] ⇒ E	INH	27B4	——	6	—— △ —— △ △△——
TMET	Transfer AM to E Truncated	If (SM • (EV; MV)) then Saturation ⇒ E else AM [31:16] ⇒ E	INH	27B5	——	2	—— —— —— —— △△——
TMXED	Transfer AM to IX: E: D	AM [35:32] ⇒ IX [3: 0] AM35 ⇒ IX [15: 4] AM [31:16] ⇒ E AM [15: 0] ⇒ D	INH	27B3	——	6	—— —— —— —— ————
TPA	Transfer CCR MSB to A	(CCR [15: 8]) ⇒ A	INH	37FC	——	2	—— —— —— —— ————
TPD	Transfer CCR to D	(CCR) ⇒ D	INH	372C	——	2	—— —— —— —— ————
TSKB	Transfer SK to B	(SK) ⇒ B [3: 0] $ 0 ⇒ B [7: 4]	INH	37AF	——	2	—— —— —— —— ————
			IND8, X	06 06	ff	6	
			IND8, Y	16 16	ff	6	
			IND8, Z	26	ff	6	
TST	Test for Zero or Minus	(M) – $ 00	IND16, X	1706	gggg	6	—— —— —— —— △△ 0 0
			IND16, Y	1716	gggg	6	
			IND16, Z	1726	gggg	6	
			EXT	1736	hh ll	6	
TSTA	Test A for Zero or Minus	(A) – $ 00	INH	3706	——	2	—— —— —— —— △△ 0 0
TSTB	Test B for Zero or Minus	(B) – $ 00	INH	3716	——	2	—— —— —— —— △△ 0 0
TSTD	Test D for Zero or Minus	(D) – $ 0000	INH	27F6	——	2	—— —— —— —— △△ 0 0
TSTE	Test E for Zero or Minus	(E) – $ 0000	INH	2776	——	2	—— —— —— —— △△ 0 0
			IND16, X	2706	gggg	6	
TSTW	Test for Test for Zero or Minus Word	(M: M + 1) – $ 0000	IND16, Y	2716	gggg	6	—— —— —— —— △△ 0 0
			IND16, Z	2726	gggg	6	
			EXT	2736	hh ll	6	
TSX	Transfer SP to X	(SK: SP) + 2 ⇒ XK: IX	INH	274F	——	2	—— —— —— —— ————
TSY	Transfer SP to Y	(SK: SP) + 2 ⇒ YK: IY	INH	275F	——	2	—— —— —— —— ————
TSZ	Transfer SP to Z	(SK: SP) + 2 ⇒ ZK: IZ	INH	276F	——	2	—— —— —— —— ————
TXKB	Transfer XK to B	$ 0 ⇒ B [7: 4] (XK) ⇒ B [3: 0]	INH	37AC	——	2	—— —— —— —— ————
TXS	Transfer X to SP	(XK: IX) – 2 ⇒ SK: SP	INH	374E	——	2	—— —— —— —— ————
TXY	Transfer X to Y	(XK: IX) ⇒ YK: IY	INH	275C	——	2	—— —— —— —— ————
TXZ	Transfer X to Z	(XK: IX) ⇒ ZK: IZ	INH	276C	——	2	—— —— —— —— ————
TYKB	Transfer YK to B	$ 0 ⇒ B [7: 4] (YK) ⇒ B [3: 0]	INH	37AD	——	2	—— —— —— —— ————
TYS	Transfer Y to SP	(YK: IY) – 2 ⇒ SK: SP	INH	375E	——	2	—— —— —— —— ————
TYX	Transfer Y to X	(YK: IY) ⇒ XK: IX	INH	274D	——	2	—— —— —— —— ————
TYZ	Transfer Y to Z	(YK: IY) ⇒ ZK: IZ	INH	276D	——	2	—— —— —— —— ————
TZKB	Transfer ZK to B	$ 0 ⇒ B [7: 4] (ZK) ⇒ B [3: 0]	INH	37AE	——	2	—— —— —— —— ————
TZS	Transfer Z to SP	(ZK: IZ) – 2 ⇒ SK: SP	INH	376E	——	2	—— —— —— —— ————

Page 66

Table 9 Instruction set summary (continued)

NEMO	Operation	Explanation	Addressin	order			Condition code						
Nick			Mode	Opcode	Operand	Cycle S	MV	H	EV	N	Z	V	C
TZX	Transfer Z to X	(ZK: IZ) ⇒ XK: IX	INH	274E	——	2	——	——	——	——	——	——	
TZY	Transfer Z to Y	(ZK: IZ) ⇒ ZK: IY	INH	275E	——	2	——	——	——	——	——	——	
WAI	Wait for Interrupt	WAIT	INH	27F3	——	8	——	——	——	——	——	——	
XGAB	Exchange A with B	(A) ⇔ (B)	INH	371A	——	2	——	——	——	——	——	——	
XGDE	Exchange D with E	(D) ⇔ (E)	INH	277A	——	2	——	——	——	——	——	——	
XGDX	Exchange D with X	(D) ⇔ (IX)	INH	37CC	——	2	——	——	——	——	——	——	
XGDY	Exchange D with Y	(D) ⇔ (IY)	INH	37DC	——	2	——	——	——	——	——	——	
XGDZ	Exchange D with Z	(D) ⇔ (IZ)	INH	37EC	——	2	——	——	——	——	——	——	
XGEX	Exchange E with X	(E) ⇔ (IX)	INH	374C	——	2	——	——	——	——	——	——	
XGEY	Exchange E with Y	(E) ⇔ (IY)	INH	375C	——	2	——	——	——	——	——	——	
XGEZ	Exchange E with Z	(E) ⇔ (IZ)	INH	376C	——	2	——	——	——	——	——	——	

Note: 1. CCR [15: 4] changes according to the result of the operation. PK fields are not affected.
2. The CCR [15: 0] changes according to the copy of the CCR pulled from the stack.
3. 3. The PK field changes according to the state pulled from the stack. The rest of the CCR Minutes are not affected.
Four. The cycle times of conditional branching are listed in the order of "taken, not taken".

Table 10 Instruction set abbreviations and symbols

A	-Accumulator A	IMM8 -8 Bit Immediate
AM	-Accumulator M	IMM16 -16 Bit Immediate
B	-Accumulator B	IND8, X-unsigned 8-bit offset IX
CCR	-Condition code register	IND8, Y-Unsigned 8-bit offset IY
D	-Accumulator D	IND8, Z-Unsigned 8-bit offset IZ
E	-Accumulator E	IND16, X-Signed 16-bit offset IX
EK	-Extended addressing extended field	IND16, Y-Signed 16-bit offset IY

IR	-MAC multiplicand register
HR	-MAC multiplier register
IX	-Index register X
IY	-Index register Y
IZ	-Index register Z
K	-Address extension register
PC	-Program counter
PK	-Program counter expansion field
SK	-Stack pointer extension field
SL	-Multiplication / cumulative sign latch
SP	-Stack pointer
XK	-Index register X extended field
YK	-Index register Y extended field
ZK	-Index register Z extended field
XMSK-Modulo Addressing Index	
	Register X mask
YMSK-Modulo Addressing Index	
	Register Y mask
S	--Stop Disable Control bit
MV	-AM overflow indicator
H	-Half carry indicator
EV	-AM Extended Overflow Indicator
N	-Negative indicator
Z	-Zero indicator
V	-2's complement overflow indicator
C	-Carry / Borrow Indicator
IP	-Interrupt priority field
SM	-Saturation mode control bit
PK	-Program counter expansion field
————	Bits do not change
△	-Bits change as specified
U	-Bits are uncertain (UNKNOWN)
0	-Clear bits
1	-Set of bits
M	-Memory location used for operations
R	-Result of operation
S	-Source data
X	-Registers used for operations
M	-1 Memory byte address
M + 1	-Address of bytes at M + $ 0001
M: M + 1 -1 Memory word address	
(...) Contents of the address pointed to by X-IX	
(...) Contents of the address pointed to by Y-IY	
(...) Contents of the address pointed to by Z-IZ	
X	-E with offset IX
Y	-E Offset IY
Z	-E Offset IZ
EXT	-Extended
EXT20 -20 bit extension	

IND16, Z-Signed 16-bit offset IZ	
IND20, X-Signed 20-bit offset IX	
IND20, Y-Signed 20-bit offset IY	
IND20, Z-Signed 20-bit offset IZ	
INH	--Inherent
IXP	--Post-modified index
REL8 -8 bit relative	
REL16 -16 bit relative	
b b	-4 Bit address extension
ff	-8-bit unsigned offset
gg	-8-bit signed offset
gggg -16 bit signed offset	
ggggg -20 bit signed offset	
hh	-16 High-order byte of bit-extended address
ii	-8 Bit Immediate Data
jj	-16 Bit Immediate Data Higher byte
kk	-16 Bit Immediate Data Low byte
ll	-16 low-order bytes of a bit-extended address
mm	-8 bit mask
mmmm -16 bit mask	
rr	-8-bit unsigned relative offset
rrrr -16 bit signed relative offset	
xo	-X off MAC index register set
yo yo	-Y off MAC index register set
z z	-4 bit zero extension
+	--Addition
————	-Subtraction or negation (2's complement)
*	--Multiplication
/ /	--Division
>	--Greater than
<	--Less than
= =	- equal
≧	--Greater than or equal to
≤	--Less than or equal to
≠	--Not equal
•	--Logical product (AND)
;	--Inclusive OR (OR)
#	--Exclusive OR (EOR)
NOT	--Complement
::	- Linking
#	- send on
#	- exchange
±	--Sign bit. Also used to show tolerance Used
"	--Sign extension
%%	-Binary price
$	Hexadecimal

2.7 Extraordinary

Exceptions are events that intercept normal instruction processing. When entering exception handling, an exception is thrown from the execution of a normal instruction. Move on to the execution of the routine to be handled.

Each exception is assigned a vector that points to the corresponding processing routine. For exception handling, processing route
Contains all the operations needed to transfer control to Chin, but includes the execution of the processing routine itself
Not done.

2.7.1 Exception vector

The exception vector is the address of the routine that handles the exception. The exception vector is in the first 512 bytes of bank 0
It is contained in a data structure called an instruction vector table to be placed.

All vectors, except the reset vector, consist of one word and are located in the data space. Reset
The vector consists of 4 words and is located in the program space. 52 predefined or reserved
There is a vector and 200 user-defined vectors.

Each vector is assigned an 8-bit value. The vector number is generated by an external device or
It is supplied by the Rossessa. There is a direct mapping from the vector number to the vector table address.

CPU16 shifts the vector number 1 bit to the left (multiplies by 2) and translates it into an address.

Table 11 Exception Vector Table

Vector number	Vector address	Address space	Exception type
0	0000	P	Reset --Initial ZK [11: 8], SK [7: 4], & PK [3: 0]
	0002	P	Reset --Initial PC
	0004	P	Reset --Initial SP
	0006	P	Reset --Initial IZ (Direct Page)
Four	0008	D	Breakpoint
Five	000A	D	Bus Error
6	000C	D	Software Interrupt
7	000E	D	Illegal Instruction
8	0010	D	Division by Zero
9-E	0012-001C	D	Unassigned, Reserved
F	001E	D	Uninitialized Interrupt
Ten	0020	D	Unassigned, Reserved
11 11	0022	D	Level 1 Interrupt Autovector
12	0024	D	Level 2 Interrupt Autovector
13	0026	D	Level 3 Interrupt Autovector
14	0028	D	Level 4 Interrupt Autovector
15	002A	D	Level 5 Interrupt Autovector
16 16	002C	D	Level 6 Interrupt Autovector
17 17	002E	D	Level 7 Interrupt Autovector
18 18	0030	D	Spurious Interrupt
19-37	0032-006E	D	Unassigned, Reserved
38-FF	0070-01FE	D	User-defined Interrupts

2.7.2 Exception stack frame

During exception handling, the contents of the program counter and the contents of the condition code register are changed.
SK: Stacked at the location pointed to by SP. If not changed during exception handling, it will be stuck
PK: The PC value is the address of the next instruction in the current instruction stream + $ 0006. Exceptions in the figure below
Indicates a tack frame.

Low Address ⇒ SP After Exception Stacking

Condition Code Register

High Address Program Counter ⇒ SP Before Exception Stacking

Figure 4 Exception stack frame

2.7.3 Exception handling sequence

Exception handling is performed in four clearly distinct phases:

- The priority of all pending exceptions is evaluated and the highest priority exception is handled first.
  vinegar.
- The processor state is stacked and the CCR PK extended field is cleared.

-The exception vector number is acquired and converted to the vector address.

- The contents of the vector address are loaded on the PC and the processor jumps to the exception handling routine.

Within each phase there are variations depending on the type of exception. But the reset vector
All but all vectors contain 16-bit addresses and the PK fields are cleared. example
The outer handler must be in bank 0 or the vector points to the jump table.

2.7.4 Exception type

Exceptions are generated internally or externally. External exceptions are defined as asynchronous, interrupt, bus There are raises, breakpoints, and resets. Internal exceptions are defined as synchronous and are software discounted There are include (SWI) instructions, background (BGND) instructions, injustice instruction exceptions, and division by zero exceptions. reset For more information on interrupts and interrupts, see Chapter 3, Streamline Low Power Integration Mode. See Jules .

Asynchronous exceptions occur regardless of CPU16 or IMB clock, but exception handling is done synchronously vinegar. For all asynchronous exceptions except reset, exception handling is the first instruction boundary after the exception is recognized. It starts with.

Synchronous exception handling is defined as part of the instruction. Exception handling of synchronous exceptions is always complete and interrupts The first instruction of the processing routine is always executed before is detected.

Due to pipeline processing, the return PK: PC value stacked for asynchronous exceptions other than reset is the current It will be the address of the next instruction in the current instruction stream plus $ 0006. All RTI instructions Exception handling routine needs to be terminated and was interrupted by subtracting $ 0006 from the stacked values Resume execution of the instruction stream. The PK: PC value at the time of executing the synchronous exception is the ad of the instruction that caused the exception. It will be less + $ 0006. RTI always subtracts $ 0006 on return, so execution resumes with the next instruction. So you have to adjust the PK: PC where the instruction that caused the exception is stacked. PK: PC value is $ 0002 is added before being stacked.

2.8 Development support

CPU16 has powerful built-in tools for tracking program execution and system debugging. increase. These tools provide deterministic opcode tracking, breakpoint exceptions, and back.

Gundo debug mode. To use this feature easily, a simple serial interface, And a terminal is required. Also, if you need advanced debugging, a simple serial interface In-circuit emulation by adding a bus analyzer to the computer and terminal And system debugging is possible.

2.8.1 Freeze

When developing a system using a debugger etc., the CPU may be stopped. At this time, from the CPU The Leeds signal is asserted to each module.

Each module uses this freeze signal to stop the module itself when the CPU stops (freeze). It is possible to. When the module itself is stopped, the clock is supplied to the inside of the module (counter, etc.). It is done by stopping. When the CPU started operating, the freeze signal was negated, so it was stopped. The module will also start working.

Each module can also ignore the freeze signal. At that time, even if the CPU stops, it is still in the module. Functions (such as counters) continue to work.

2.8.2 BDM connection pinout

The pins are shown below.

DS	1	2	BERR
GND	3	Four	BKPT / DSCLK
GND	Five	6	FREEZE
RESET	7	8	IPIPE1 / DSI
VDD	9	Ten	IPIPE0 / DSO

Figure 5 BDM connection pinout

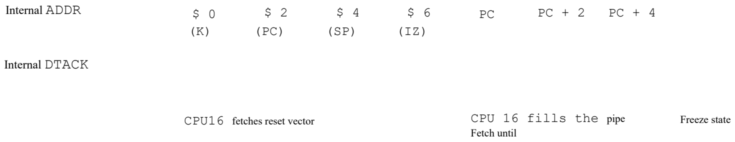
2.8.3 Mechanism for entering BDM after reset

The waveform below shows the waveform from entering BDM (freezing state) immediately after reset negating.

RESET

BKPT

FREEZE



CPU16 has a minimum of 7 words (4 words of the reset vector and a pipe) to enter the BDM immediately after a reset. Requires fetch (bus access) (3 words) to fill.

For more information on BDM, see the CPU16 Reference Manual (CPU16RMJ / AD).

Chapter 3
Streamline low power Integration module

Streamline Low Power Integration Module (SLIM) is a system boot,
It consists of six functional blocks that perform initialization, configuration, and external bus control. Figure 6 shows "SLIM" Block diagram of .

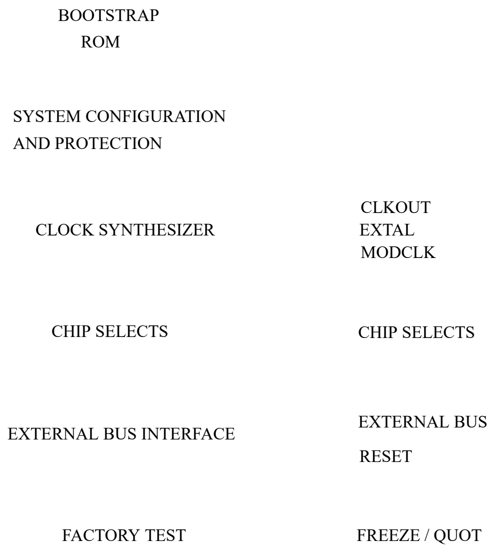


Figure 6 SLIM block diagram

3.1 Overview

Bootstrap ROM loads SRAM with small programs for use in system initialization.

You can start it

System configurations and protection blocks control the configuration and mode of operation of the MCU. This block is with the bus We also provide a software watchdog monitor.

The system clock is a clock signal used by SLIM, other IMB modules, and external devices. To generate. It also has a built-in periodic interrupt generator, which is a time-strict control route.

Page 72

Supports execution.

The chip select block provides two chip select signals. For each chip select Has its own base address register and option register.

The external bus interface handles the transfer of information between the IMB module and the external address space.

The system test block contains the hardware needed to test the MCU. this Is intended for use in factory testing and is not supported for use in normal applications. plug.

Table 12 shows the SLIM address map that occupies 128 bytes. In the 128-byte address space Reading an unused register returns zero.

Table 12 SLIM address map

Address Address	15 8	7 0
$ YFFA00 [1]	SLIM MODULE CONFIGURATION REGISTER (SLIMCR)	
$ YFFA02	FACTORY TEST REGISTER (SLIMTR)	
$ YFFA04	CLOCK SYNTHESIZER CONTROL REGISTER (SYNCR)	
$ YFFA06	NOT USED	RESET STATUS REGISTER (RSR)
$ YFFA08	NOT USED	MODULE TEST E (SLIM TRE)
$ YFFA0A	MODULE CONFIGURATION SHADOW REGISTER (MCRC)	
$ YFFA0C	NOT USED	SUB-MODULE DISABLE (SMD)
$ YFFA0E	PORT / CLOCK CONFIGURATION SHADOW REGISTER (PCON)	
$ YFFA10	PORT A OUTPUT DATA (PORTA)	PORT B OUTPUT DATA (PORTB)
$ YFFA12	PORT A PIN DATA (PORTAP)	PORT B PIN DATA (PORTBP)
$ YFFA14	NOT USED	PORT A / B DATA DIRECTION (DDRAB)
$ YFFA16	NOT USED	NOT USED
$ YFFA18	PORT C OUTPUT DATA (PORTC)	PORT D OUTPUT DATA (PORTD)
$ YFFA1A	PORT C PIN DATA (PORTCP)	PORT D PIN DATA (PORTDP)
$ YFFA1C	PORT C DATA DIRECTION (DDRC)	PORT D DATA DIRECTION (DDRD)
$ YFFA1E	PORT C PIN ASSIGNMENT (PCPAR)	PORT D PIN ASSIGNMENT (PDPAR)
$ YFFA20	NOT USED	PORT E OUTPUT DATA (PORTE)
$ YFFA22	NOT USED	PORT E PIN DATA (PORTEP)
$ YFFA24	NOT USED	PORT E DATA DIRECTION (DDRE)
$ YFFA26	NOT USED	PORT E PIN ASSIGNMENT (PEPAR)
$ YFFA28	PORT G OUTPUT DATA (PORTG)	PORT H OUTPUT DATA (PORTH)
$ YFFA2A	PORT G PIN DATA (PORTGP)	PORT H PIN DATA (PORTHPP)
$ YFFA2C	PORT G DATA DIRECTION (DDRG)	PORT H DATA DIRECTION (DDRH)
$ YFFA2E	NOT USED	NOT USED
$ YFFA30	NOT USED	PORT F OUTPUT DATA (PORTF)
$ YFFA32	NOT USED	PORT F PIN DATA (PORTFP)
$ YFFA34	NOT USED	PORT F DATA DIRECTION (DDRF)
$ YFFA36	PORT F PIN ASSIGNMENT (PFPAR)	
$ YFFA38	NOT USED	PORT F EDGE FLAGS (PORTFE)
$ YFFA3A	NOT USED	PORT F EDGE-DETECT ENABLE (PFEER)

Page 73

Table 12 SLIM Address Map (continued)

Address Address	15 8	7 0
$ YFFA3C	PORT F LEVEL (PFLVR)	PORT F INTERRUPT VECTOR (PFIVR)
$ YFFA3E	NOT USED	NOT USED
$ YFFA40	TEST MODULE MASTER SHIFT A (TSTMSRA)	
$ YFFA42	TEST MODULE MASTER SHIFT B (TSTMSRB)	
$ YFFA44	TEST MODULE SHIFT COUNT (TSTSC)	
$ YFFA46	TEST MODULE REPETITION COUNTER (TSTRC)	
$ YFFA48	TEST MODULE CONTROL (CREG)	
$ YFFA4A	TEST MODULE DISTRIBUTED REGISTER (DREG)	
$ YFFA4C	NOT USED	
$ YFFA4E	NOT USED	
$ YFFA50	SYSTEM PROTECT CONTROL (SYPCR)	
$ YFFA52	TIMER CONTROL (TIC)	TIMER INTERRUPT VECTOR (TIV)
$ YFFA54	NOT USED	SOFTWARE SERVICE (SWSR)
$ YFFA56	PRESCALER (READ-ONLY)	
$ YFFA58	SOFTWARE WATCHDOG PERIOD REGISTER (SWP)	
$ YFFA5A	TIMER PERIOD REGISTER (TIP)	
$ YFFA5C	SOFTWARE WATCHDOG DOWNCOUNTER (SWDC) (READ-ONLY)	
$ YFFA5E	REAL-TIME DOWNCOUNTER (RTDC) (READ-ONLY)	
$ YFFA60	CHIP-SELECT BASE A (CSBARA)	
$ YFFA62	CHIP-SELECT OPTION A (CSORA)	
$ YFFA64	CHIP-SELECT BASE B (CSBARB)	
$ YFFA66	CHIP-SELECT OPTION B (CSORB)	
$ YFFA68	CHIP-SELECT BASE C (CSBARC)	
$ YFFA6A	CHIP-SELECT OPTION C (CSORC)	
$ YFFA6C	CHIP-SELECT CONTROL REGISTER (CSCR)	
$ YFFA6E	NOT USED	
$ YFFA70	RESERVED	
$ YFFA72	RESERVED	LEVEL 1 IACK CYCLE READ (ICRR)
$ YFFA74	RESERVED	LEVEL 2 IACK CYCLE READ (ICRR)
$ YFFA76	RESERVED	LEVEL 3 IACK CYCLE READ (ICRR)
$ YFFA78	RESERVED	LEVEL 4 IACK CYCLE READ (ICRR)
$ YFFA7A	RESERVED	LEVEL 5 IACK CYCLE READ (ICRR)
$ YFFA7C	RESERVED	LEVEL 6 IACK CYCLE READ (ICRR)
$ YFFA7E	RESERVED	LEVEL 7 IACK CYCLE READ (ICRR)

Note: 1. Y = M111. However, M is the logic of the SLIMCR's module mapping (MM) bits.
Indicates the status.

Page 74

3.2 System configuration and protection block

This functional block provides configuration control over the entire MCU. Also, interrupt arbitration, bus monitoring
It also performs running and system testing functions. Bus monitor, HALT moni for MCU system protection
Includes data, spurious monitor, and software watchdog timer. These machines
Noh is part of a microcontroller to reduce the number of external components required for a complete control system
It is built in as.

Figure 7 shows a “system protection block” .

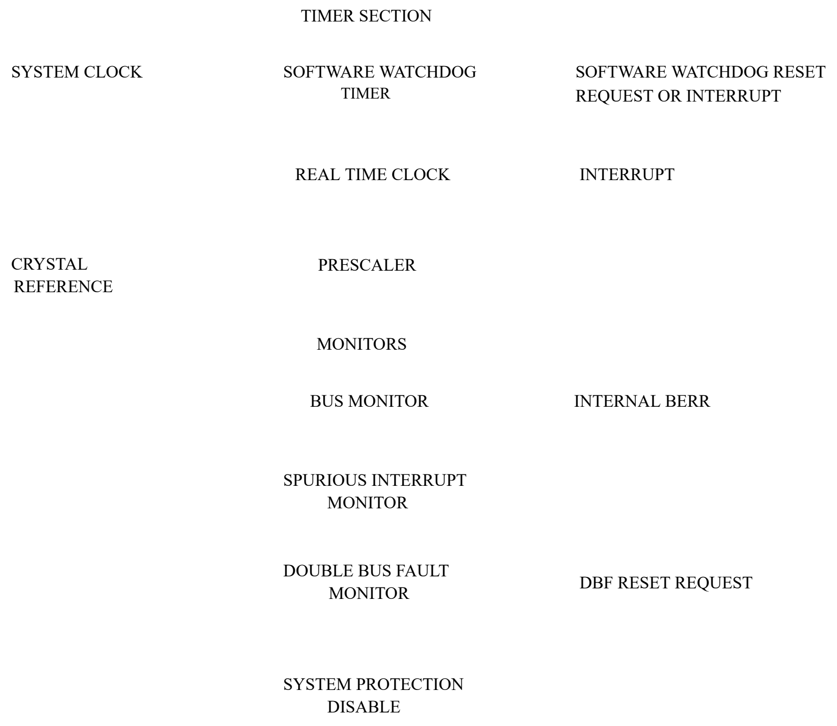


Figure 7 System protection block

Page 75

3.2.1 System configuration

SLIM controls the configuration of the MCU during normal operation and internal testing.

SLIMCR—SLIM module configuration register $ YFFA00

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
LOAD	FRZ		MOTO		MODE		SHEN	SUPV	MM	ROMSU ROMEN			IARB		
RESET: (DSI = 1)															
MCRC15	MCRC [14:13]		MCRC12		MCRC [11: 9]		MCRC8	MCRC7	MCRC6	MCRC5	MCRC4		MCRC [3: 0]		
1	0	0	1	1	1	1	0	0	1	0	0	1	1	1	1
RESET: (DSI = 0)															
MCRC15	MCRC [14:13]		MCRC12	DS	SIZE	R / W	MCRC8	MCRC7	MCRC6	MCRC5	AD11		MCRC [3: 0]		
1	0	0	1	Pin state	Pin state	Pin state	0	0	1	0	Pin state	1	1	1	1

Module configuration registers control the system configuration. Handling of this register
Or it is as follows.

LOAD: Can be written once after reset

FRZ [1: 0]: Read / write at any time
MOTO : Read-only
MODE [2: 0]: Read-only
SHEN : Read / write at any time
SUPV : Can be written once after reset
MM : Can be written once after reset
ROMSU : Can be written once after reset
ROMEN : Read / write at any time
IARB [3: 0]: Read / write at any time
The MM bit can only be written once and must be left set thereafter.

LOAD — Pad driver load

0 = One pair of drivers is enabled. (Half drive)
1 = Both pairs of drivers are enabled. (Full drive)

Some pads by choosing whether to enable both pairs of pad drivers
Controls the drive strength of the output driver. This bit can only be written once after a reset,
You can read it at any time. Table 13 shows the signal names controlled by the LOAD bit.

Table 13 Pins controlled by the LOAD bit

LOAD		MC68HC916Y5	MC68HC916Y6, MC68HC16Y5
1		All output drivers are full drive	All output drivers are full drive
0	Full drive	CLKOUT / PE4	CLKOUT / PE4, RMISO / PSP4, RMOSI / PSP5, Signals other than RSCK / PSP6 and RSS / PSP7
	Half drive	Signals other than CLKOUT / PE4	CLKOUT / PE4, RMISO / PSP4, RMOSI / PSP5, RSCK / PSP6, RSS / PSP7

FRZ [1: 0] — Software enable freeze

Table 14 shows that FREEZE assertions are software watchdogs, real-time clocks,
And the effect on the prescaler.

Table 14 FREEZE Impact of assertions

FRZ [1: 0]		Features to be disabled
0	0	None None
0	1	Bus monitor
1	0	Software watchdog, real-time clock, prescaler
1	1	Both

MOTO— Motorola configuration

0 = Architecture other than Motorola
1 = Motorola architecture

This bit pertains to Motorola and non-Motorola architectures and CPU modules
Used to configure options. Currently this bit is absent for MOTO / non-MOTO operations
It is effective. SLIM always does so regardless of the state of this bit or the corresponding MCRC shadow register.
It works in MOTO mode.

MODE—Mode configuration

As shown in Table 15, these status bits allow you to see the current mode.
Controls the configuration of external address bus and data bus operating modes.
Writing to this field will be ignored.

Table 15 Mode configuration bit fields

MODE [2: 0]			composition
0	0	0	Peripheral mode
0	0	1	spare
0	1	0	Slave factory test mode
0	1	1	Master factory test mode (MISR enabled)
1	0	0	Non-multiplexed master mode
1	0	1	Multiplexing master mode
1	1	X	Single chip mode

Note: MC68HC (9) 16Y5 / 916Y6 is in non-multiplexed master mode, multiplexed master mode,
It is possible to operate in nguru chip mode. Other modes are Motorola

This mode cannot be used for shipping test or function.
This master mode is synonymous with extended mode.

SHEN—Show Cycle Enable

0 = Disable show cycle.
1 = Enable show cycle.

This bit determines what EBI does to the external bus during the internal transfer operation. show·
You can use the cycle to monitor internal transfers from the outside.
IMB when show cycle is enabled in master mode with 8-bit data bus width
Only the operation of the upper half of the data bus is driven to the upper data bus pin.

SUPV—Supervisor / Unconstrained data space

The SUPV bit holds the SLIM global register in the supervisor data space or user data.
Place it in one of the spaces. CPU16 of MC68HC (9) 16Y5 / 916Y6 works only in supervisor mode
SUPV is invalid because it is not.

MM—Module mapping

0 = Internal modules are addressed with $ 7FF000- $ 7FFFFF.
1 = Internal module is addressed by $ FFF000- $ FFFFFF.

The IMB address line ADDR [23:20] follows the logical state of ADDR19 unless driven externally.
is. MM corresponds to ADDR23 of IMB. If the MM is cleared, SLIM will install the IMB module.
Maps to dress space $ 7FF000 to $ 7FFFFF, but CPU16 should access this address space
can not. To make it accessible, it is possible by resetting the MM to be set.
I will. MM is a bit that can be written only once. Initialization software logically leveles MM
Must be set to Le 1.

ROMSU—ROM Supervisor / Unconstrained Mode

0 = Boot ROM responds to supervisor access and user access regardless of FC2 state
increase.
1 = Boot ROM responds to supervisor access when FC2 = 1.

Page 77

This bit controls whether the SLIM boot ROM responds to supervisor access.
will be used. ROMSU is disabled because CPU16 only operates in supervisor mode.

ROMEN—ROM enabled

0 = Boot ROM does not respond. CSB is enabled at this time.
When 1 = MOTO = 1, the boot ROM responds to the highest lowest address ($ 000000- $ 0001FF). child
CSB is disabled when. When MOTO = 0, the boot ROM is the highest level address.
Responds to ($ FFFE00-$ FFFFFF). Both CSBs are disabled.

IARB [3: 0] — Interrupt arbitration field

This field is SLIM's external interrupt request, interrupt by port F edge detection, real-time.
Down counter (periodic interrupt) or software watchdog down counter (periodic interrupt)
(When used only) is used when an interrupt occurs.

Each module that can generate an interrupt request has an interrupt arbitration (IARB) field.
Mediation between interrupt requests of the same priority is tuned by the bit value of the IARB field. for example
Even if there is only one pending request, if arbitration does not occur every time an interrupt request is confirmed
It will not be. The IARB field must have a non-zero value for the conflict to occur. IARB feel
If an interrupt request from a module with a value of% 0000 is recognized, CPU16 raises a spurious interrupt exception.
To process. SLIM forwards the external interrupt request to CPU16, so the SLIM IARB field values are the same.
Used for arbitration of priority internal and external interrupts. The reset value of IARB for SLIM is
% 1111 (highest priority) and IARB reset value for all other modules is% 0000
vinegar. This prevents SLIM interrupts from being dropped during initialization.

SLIMTR—SLIM test register $ YFFA02

SLIMTR is for factory testing only.

SMD—Submodule disable register $ YFFA0D

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	UNUSED								RESERVED				BRKEN	CHIP SELECT	TEST	SYS PROT
RESET:													0	0	0	0

The submodule disable register (SMD) is SLIM when the submodule is not in use.
Used to disable within. When the submodule sets the corresponding bit to 1,
It will be disabled. Disabling a submodule provides a clock to the submodule
The supply is stopped and the submodule has the lowest power consumption.

BRKEN— Breakpoint Enable

0 = Enable breakpoint logic.
1 = Disable breakpoint logic.

CHIP SELECT — Chip Select Submodule Disable

0 = Enable the chip select submodule.

1 = Disable the chip select submodule.

TEST—Test submodule disabled

0 = Enables the test submodule.

1 = Disable the test submodule.

SYSPROT—System protection submodule disabled

0 = Enables the system protection submodule.

1 = Disables the system protection submodule.

3.2.2 System protection

The system protection submodule is a software watchdog timer, real-time clock
It consists of a timer section that includes a key and a prescaler. Also, bus, HALT, and
It also includes a monitor section consisting of spurious interrupt monitors on the external bus. These features are complete
Assembled as part of a microcontroller to reduce the number of external components required for the entire control system
It is expected.

SYPCR — System protection control register $ YFFA50

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
REN	SREN	PCLK	SLPC	RZ	SZ	IRSEL	TIEN	0		TIQL [2: 0]		DBE	BME	BMT [1: 0]	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The system protection control register is a system monitor function, software watch dock.
Controls prescaling for clocks and bus monitor timing. Test mo
This register can only be written once after a power-on reset or an external reset, except during a power-on reset.
No, but you can read it at any time. In test mode you can write at any time
increase.

REN—Real-time reset enable

0 = Disables prescaler reset.

1 = Enables prescaler reset.

SREN—Software watchdog reset enable

This bit controls whether writing to the SWP register resets the prescaler.

0 = Disables prescaler reset.

1 = Enables prescaler reset.

PCLK—Prescaler Clock Select

This bit selects the clock source to the prescaler.

0 = Select the system clock.

1 = Select the crystal standard.

SLPC—Software watchdog count operation select during LPSTOP

0 = Software watchdog does not count with LPSTOP. Resume counting after reset
do.

1 = Software watchdog is counted by LPSTOP.

RZ—Timer zero flag

This bit is a flag set after the RTC down counter reaches zero. To clear
Writes zero after reading this register.

SZ—Software Watchdog Zero Flag

This bit is set after the software watchdog down counter reaches zero.
Flag. To clear, write zero after reading this register.

IRSEL— Interrupt / Reset Select

This bit counts down to zero, the software watchdog down cow

Select whether the input issues a reset or an interrupt.

0 = The counter causes a reset.

1 = Counter raises the interrupt specified in the TIQL field of SYPCR.

Page 79

TIEN— Timer interrupt enabled

This bit indicates whether the real-time down counter issues an interrupt after a timeout.
Control

No interrupt occurs when 0 = zero is reached.

Occurs when 1 = zero is reached.

TIQL [2: 0] — Timer interrupt request level

This field is a real-time down counter or software watchdog dow
Defines the priority of interrupts generated when the counter times out. See Table 16
please.

Table 16 Timer interrupt request level fields

TIQL [2: 0]			Explanation
0	0	0	Interrupt Disabled
0	0	1	Interrupt Request Level 1
0	1	0	Interrupt Request Level 2
0	1	1	Interrupt Request Level 3
1	0	0	Interrupt Request Level 4
1	0	1	Interrupt Request Level 5
1	1	0	Interrupt Request Level 6
1	1	1	Interrupt Request Level 7

Real-time down counter / software watchdog counter with the same priority
Real-time down counter / software if both busy and external IRQ interrupts occur at the same time.
Watchdog counter interrupts are processed first.

DBE—Double bass fault enable

DBE has a double bass four, similar to checking for double bass faults on the IMB.
Controls the monitor. (See “3.2.5 Holt Monitor” on page 79)

0 = Disable double bass fault monitor.

1 = Enable double bass fault monitor.

BME—Bus monitor enable

0 = Disables the external bus monitor function.

1 = Enables the external bus monitor function.

This bus monitor enable is for external access and is for internal modules.
The monitor is always working. Note that this bit is valid only in master mode. This feature
See 3.2.3 Bus Monitor for an overview of .

In single chip mode, it is always functioning depending on this bit.

BMT [1: 0] — Bus monitor timing

These bits select the bus monitor timeout period. See Table 17 .

Table 17 Bus monitor timeout period

BMT [1: 0]	time out
00 00	64 System Clocks
01 01	32 System Clocks
Ten	16 System Clocks
11 11	8 System Clocks

Page 80

After a system reset, the bus monitor timeout value defaults to 64 system clocks.
It will be.
Note: To perform external access, this timeout value is a chip select access
Must be longer than the time.

3.2.3 Bus monitor

The bus monitor function usually checks that the response time of the bus cycle does not become too long. agreement
If the answer time is too long, the bus monitor function asserts the BERR signal and kills the bus cycle.
increase. CPU16 handles bus error exceptions when the normal bus cycle ends with a BERR signal.
The longer response time of the bus cycle means that the time to assert DTACK is slower.
Means The response delay time is measured in clock cycles and is measured in the BMT field (see Table 17) .
You can select the maximum allowable response time by setting (illuminate).
The bus monitor constantly monitors the internal bus cycle, but in the external bus cycle, the SYPCR
The BME bit (bit 2 of $ YFFA50) allows you to disable / enable the monitor.
In a system with multiple bus masters, the bus master should have an external bus monitor function.
The external bus monitor function must be disabled on the non-master side.

3.2.4 Spurious interrupt monitor

The spurious interrupt monitor is interrupt arbitrated during the IACK cycle (see “IARB [3: 0] — Interrup[redacted]n page 76 ”. Field) was $ 0, or the IACK cycle was in the BMT field (see Table 17).
If it does not finish within the configured maximum allowed response time, SLIM asserts BERR to force it t
CPU16 handles spurious interrupt exceptions when the IACK cycle ends with BERR.
This IARB (interrupt arbitration field) is owned by all interrupt resource modules, so interrupts
Must be set ($ 1 to $ F) when using only.

3.2.5 Holt monitor

Holt monitors have a double bass folder on the internal bus when the SYPCR DBE bit is enabled.
If this happens, assert a reset and restart the system. Reset at that time
The Tatus Register (RSR) flag indicates that the Holt monitor caused the last reset.
increase.
If for some reason the normal bus cycle is killed with BERR, CPU16 handles the bus error exception.
CPU16 asserts an internal HALT signal if it is killed by BERR while handling the exception.
Stop execution. This is called a double bass fault. SLIM monitors the internal HALT signal
Therefore, when CPU16 asserts the internal HALT signal with a double bus fault, it immediately resets.
And restart. This is called a Holt monitor.

3.2.6 Software Watchdog

The software watchdog monitors the system software and software on the system
Protects against looping and out-of-control errors. Software wow
Chidog regularly runs a special service sequence to tie the software watchdog.
It prevents mouts and does not issue an error response. This real-time service operation
If not done, the software watchdog times out and issues a reset or interrupt.
increase.

The features of the software watchdog are as follows.

-It is possible to select whether to issue an interrupt or a reset at the end of the timeout period.
-The input clock to the software down counter (SWDC) is an externally generated clock.
Either crystal frequency is possible. The time-out period range is 512 µs to 524 ms (4.000 MHz).
Crystal frequency), 125ns ~ 131ms (16.000MHz system clock) or 83.3ns ~
87ms (24.000MHz system clock).
• SWDC checks with a real-time down counter to extend the timeout period.
You can create a 32-bit down counter.
・ SWDC can be read at any time.

Page 81

SWSR—Software Watchdog Service Register $ YFFA55

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
				UNUSED												
RESET:									0	0	0	0	0	0	0	0

The software watchdog service register is written by the SWDOG service sequence.
It is a location where you can get around. This register is affected by system reset and power on reset
I can't. Reading this register always returns zero.

The software watchdog timer service sequence runs in the following order:

- Write $ 55 in SWSR.
- Write $ AA to SWSR.

These writes must be done in the order they are listed before the timeout,
You can execute any number of instructions during the period until the end of the period.

When the proper service sequence occurs, the software down counter will turn to software.
It reloads the value of the watchdog cycle register and starts this process again.

If the timeout period expires before the service process is executed, the software watchdog will
Generates a system reset or interrupt based on the state of the IRSEL bit. Reset
The status register (RSR) flag is also set.

3.2.7 Timer section

The SLIM system protection timer section contains a pair of down cows that share a system prescaler.
There is an input. See Figure 8 .

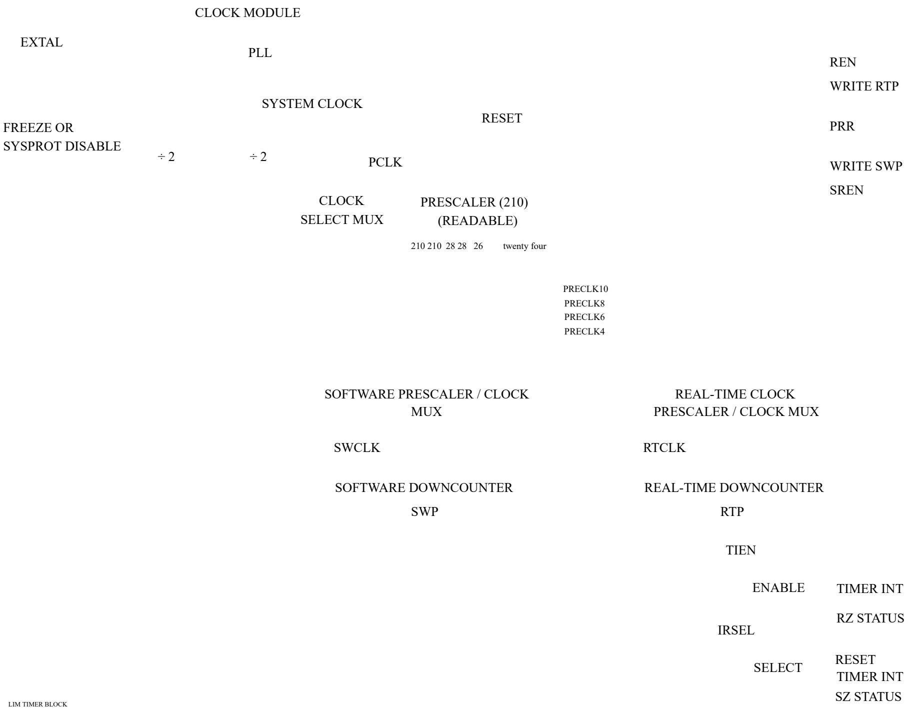


Figure 8 Software watchdog, real-time clock and prescaler

Page 82

PRE—System protection prescaler $ YFFA56

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SWDOG and RTC share a prescaler, which is a 10-bit synchronous counter. SWDOG again
Can apply different prescaler taps to the RTC down counter at the same time. Prescaler supply
The clock selected for is divided by 2 before being sent to the prescaler.

Both SWCLK and RTCLK are prescaled with counts of 1024, 256, 64, or 16 respectively.
vinegar. Crystal frequency, system clock, and prescaler output (PRECLK10, PRECLK8,
All of PRECLK6, or PRECLK4) can be supplied to the SWDOG or RTC prescaler clock mux.
increase. Software watchdog control field in the timer controller register (TIC)
The (SWC [2: 0]) bits are clocked or pressed as the SWCLK of the software watchdog.
Select which of the kale inputs to use. Rear of timer control register (TIC)
The rutime control field (RTC [2: 0]) bits perform the same control over the real-time clock.

To do. The value of the prescaler can be read from the prescaler register (PRE) at any time.
increase. Performing a 16-bit read on the prescaler returns a 10-bit value. Upper 6 bits
Is zero.

Phase-locked loops (PLLs) can be selected whether they are running or not. PCLK If is set but the PLL is not working, the system protection prescaler will not work. Hmm. Therefore, the system protection prescaler tap in the SWDOG and RTC clock control fields. Or even if the crystal frequency is selected, it is invalid.

SWP—Software watchdog cycle register $ YFFA58

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The software watchdog cycle register contains the software watchdog down cow. Contains the value to be loaded into the data. Start software watchdog down counter To do this, you must write to the lower byte of SWP.

SWP is always readable and writable. This register if the SREN bit of SYPCR is set Writing to will reset the prescaler and stop it.

Loading zero into this register clears the software watchdog down counter Will be done.

SWP is not affected by a system reset. Therefore, before a system reset occurs The value programmed into SWP is preserved. But with a power-on reset Only then SWP will be reset to zero.

Table 18 Software watchdog cycle range

SWC [2: 0] [1] clock Control settings			Clock source (SWCLK) [2]		Standard timeout range [3]		
			PCLK = 1	PCLK = 0	Crystal At 4.000 MHz	System CLK is At 16.000 MHz	System CLK is At 24.000 MHz
0	0	0	System CLK / 2		————	125 ns ~ 8.19 ms	83.3 ns ~ 5.46 ms
0	0	1	Crystal Ref / 2		512 ns ~ 32.7 ms	————	————
0	1	0	Crystal Ref / 2 1024	System CLK / 2 1024	512 ∝ s to 33.5 s	128 ∝ s to 8.38 s	85.3 ∝ s to 5.59 s
0	1	1	Crystal Ref / 2 256	System CLK / 2 256	128 ∝ s to 8.38 s	32.0 ∝ s ~ 2.09 s	21.3 ∝ s ~ 1.39 s
1	0	0	Crystal Ref / 2 64	System CLK / 2 64	32.0 ∝ s ~ 2.09 s	8.00 ∝ s ~ 524 ms	5.33 ∝ s ~ 349 ms

Table 18 Software watchdog cycle range (continued)

SWC [2: 0] [1] clock Control settings			Clock source (SWCLK) [2]		Standard timeout range [3]		
			PCLK = 1	PCLK = 0	Crystal At 4.000 MHz	System CLK is At 16.000 MHz	System CLK is At 24.000 MHz
1	0	1	Crystal Ref / 2 16 16	System CLK / 2 16 16	8.00 ∝ s ~ 524 ms	2.00 ∝ s ~ 131 ms	1.33 ∝ s ~ 87 ms
1	1	0	Used as an extension of RTC		————	————	————
1	1	1	Software Watchdog stopped		————	————	————

Note: 1. See TIC (Timer Control Register).

2. The clock source is shown in Figure 8 "S [redacted]
Ku and Prescaler ” and Table 21“ Softw
See Im Clock Control Settings .

3. 3. The minimum timeout value is SWP = 1, the clock source is one clock, and the maximum value is SWP = $ FFFF and the clock source is 2^{16} = 65538 clock minutes.

SWDC—Software watchdog down counter $ YFFA5C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Software watchdog down counter outputs SWC prescaler multiplexer It consists of a 16-bit down counter that is clocked from (SWCLK signal). SWCLK signal Provides the clock to the SWDC. SWDC counts down to zero when the SWP value is loaded vinegar. SWDC does not start counting until the lower byte of SWP is written.

When the software watchdog times out, depending on the state of the IRSEL bit in SYPCR,
Assert a reset or interrupt. Next, the SWDOG down counter is the current tie from SWP.
Load the muout value and start the countdown again. You can change the timeout value at any time
increase.

After the timeout occurs, the SYPCR software watchdog zero flag (SZ) is set.
It will be set.

The SWDC can be read at any time.

The SWDC is unaffected when a system reset is asserted.

The watchdog timeout period (COP period) is calculated by the following formula.

Without prescaler, when using crystal clock:

$$\text{COP cycle =}\ \frac{\text{(SWP count value)} \times 2}{\text{Crystal clock}}$$

Without prescaler, when using system clock:

$$\text{COP cycle =}\ \frac{\text{(SWP count value)} \times 2}{\text{System clock}}$$

If you have a prescaler and want to use a crystal clock:

$$\text{COP cycle =}\ \frac{\text{(SWP count value)} \times 2}{\text{(Crystal Clock) / (Prescaler)}}$$

If you have a prescaler and want to use the system clock:

$$\text{COP cycle =}\ \frac{\text{(SWP count value)} \times 2}{\text{(System clock) / (Prescaler)}}$$

Page 84

3.2.8 Real-time clock

The real-time clock (RTC) is clock driven by the RTC prescaler multiplexer output.
It consists of a 16-bit down counter. RTCLK signals are sourced from selectable clock sources
It will be.

The RTCLK signal is supplied to the real-time clock down counter (RTDC), which is the RTP.
When the value is loaded, it counts down to zero. RTDC continues until the low byte of RTP is written
Does not start counting.

RTP—Real-time periodic register $ YFFA5A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0

RESET:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The real-time periodic register contains a value to load into the RTC down counter. chain
To start a real-time down counter, whether it is done or not, RTP
Must be written to the lower byte of.

When writing to this register, the prescaler will be reset if the REN bit of SYPCR is set.
Will be Loading zero into this register clears the RTC down counter and stops it.

RTDC—Real-time clock down counter $ YFFA5E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0

RESET:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When the down counter reaches zero, the TIEN bit is set and TIQL is non-zero.
An interrupt is generated (if the timing down counter detects zero, the system protection register
The star (SYPCR) real-time zero flag RZ is always set). RTDC sets the current RTP value
Load and start counting again.

The RTC period (without prescale, using the crystal clock) is calculated as follows:

$$\text{RTC cycle =}\ \frac{\text{(RTP count value)} \times 2}{\text{Crystal clock}}$$

The RTC period (without prescaling, using the system clock) is calculated as follows:

$$\text{RTC cycle} = \frac{(\text{RTP count value}) \times 2}{\text{System clock}}$$

The RTC period (with prescale and using the crystal clock) is calculated as follows:

$$\text{RTC cycle} = \frac{(\text{RTP count value}) \times 2}{(\text{Crystal Clock}) / (\text{Prescaler})}$$

The RTC period (with prescale and using the system clock) is calculated as follows:

$$\text{RTC cycle} = \frac{(\text{RTP count value}) \times 2}{(\text{System clock}) / (\text{Prescaler})}$$

These calculations give the real-time period range shown in Table 19 .

Table 19 Real-time cycle range (timers are not chained) 1

RTC [2: 0] 1 clock Control settings			Clock source (RTCLK) 2		Standard timeout range 3		
			PCLK = 1	PCLK = 0	Crystal At 4.000 MHz	System CLK is At 16.000 MHz	System CLK is At 24.000 MHz
0	0	0	System CLK / 2		————	125 ns ~ 8.19 ms	83.3 ns ~ 5.46 ms
0	0	1	Crystal Ref / 2		512 ns ~ 32.7 ms	————	————
0	1	0	Crystal Ref / 2 1024	System CLK / 2 1024	512 ∝ s to 33.5 s	128 ∝ s to 8.38 s	85.3 ∝ s to 5.59 s
0	1	1	Crystal Ref / 2 256	System CLK / 2 256	128 ∝ s to 8.38 s	32.0 ∝ s ~ 2.09 s	21.3 ∝ s ~ 1.39 s
1	0	0	Crystal Ref / 2 64	System CLK / 2 64	32.0 ∝ s to 2.09 s	8.00 ∝ s ~ 524 ms	5.33 ∝ s ~ 349 ms
1	0	1	Crystal Ref / 2 16 16	System CLK / 2 16 16	8.00 ∝ s to 524 ms	2.00 ∝ s ~ 131 ms	1.33 ∝ s ~ 87 ms
1	1	0	Stop Real-Time Clock		————	————	————
1	1	1	Stop Real-Time Clock		————	————	————

Note: 1. See TIC (Timer Control Register).

2. The clock source is shown in Figure [redacted]
And Prescaler ” and Table 21“ Sof
See Setting Lock Control .

3. 3. The minimum timeout value is RTP = 1, the clock source is one clock, and the maximum value is.
When RTP = $ FFFF and the clock source is 2 16 = 65538 clock minutes.

3.2.9 RTC and SWDOG in chain mode

32-bit down counter by concatenating the real-time clock and software watchdog
Can be formed. At this time, SWDC becomes the highest word (MSW) and goes down in real time.
The counter is the lowest word (LSW). 32-bit down counters that reach zero reload
Don't count down SWDC by 1 in the lowest word, roll it over to $ FFFF, then cow
Continue to

To trigger a downcounter chain, one 32-bit value or two 16-bit values, SWP
You can write to registers and RTP registers.

Note: The down counter chain starts counting until it writes to the lower byte of RTP.
not.
This counter can be read at any time. However, after reading 32-bit
The outside may not be accurate.

The chained timer period (without prescale, when using a crystal clock) is as follows:
Is calculated:

$$\text{Timer cycle} = \frac{(\text{32-bit SWP\_RTP count value}) \times 2}{\text{Crystal clock}}$$

The chained timer period (without prescaling, using the system clock) is as follows:
Calculated:

$$\text{Timer cycle} = \frac{\text{(32-bit SWP\_RTP count value)} \times 2}{\text{System clock}}$$

The chained timer cycle (with prescale and using crystal clock) is as follows:

Page 86

Calculated:

$$\text{Timer cycle} = \frac{\text{(32-bit SWP\_RTP count value)} \times 2}{\text{(Crystal Clock) / (Prescaler)}}$$

The chained timer cycle (with prescale and system clock) is as follows:
Is calculated:

$$\text{Timer cycle} = \frac{\text{(32-bit SWP\_RTP count value)} \times 2}{\text{(System clock) / (Prescaler)}}$$

These calculations give the real-time periodic range shown in Table 20 .

Table 20 Real-time cycle range (timer chained)

SWC [2: 0] [1] clock Control Configuration	RTC [2: 0] [1] clock Control settings			Clock source (RTCLK) [2]		Standard timeout range [3]		
				PCLK = 1	PCLK = 0	Crystal At 4.000 MHz	System CLK is At 16.000 MHz	System CLK is At 24.000 MHz
1 1 0	0	0	0	System CLK / 2		————	125 ns ~ 536 s	83.3 ns ~ 357.9 s
	0	0	1	Crystal Ref / 2		512 ns ~ 2147 s	————	————
	0	1	0	Crystal Ref / 2 1024	System CLK / 2 1024	512 ∝ s ~ 25.4 days	128 ∝ s ~ 6.36 days	85.3 ∝ s ~ 4.24 days
	0	1	1	Crystal Ref / 2 256	System CLK / 2 256	128 ∝ s ~ 6.36 days	32.0 ∝ s ~ 38.1 hrs	21.3 ∝ s ~ 25.4 hrs
	1	0	0	Crystal Ref / 2 64	System CLK / 2 64	32.0 ∝ s ~ 38.1 hrs	8.00 ∝ s ~ 9.54 hrs	5.33 ∝ s ~ 6.36 hrs
	1	0	1	Crystal Ref / 2 16 16	System CLK / 2 16 16	8.00 ∝ s ~ 9.54 hrs	2.00 ∝ s ~ 2.38 hrs	1.33 ∝ s ~ 1.59 hrs
	1	1	0	Stop Real-Time Clock		————	————	————
	1	1	1	Stop Real-Time Clock		————	————	————

Note: 1. See TIC (Timer Control Register).

2. The clock source is shown in Figure [redacted]
And Prescaler " and Table 21" Sof
See Setting Lock Control .

3. 3. The minimum timeout value is SWP_RTP = 1, the clock source is one clock, and the maximum value is
SWP_RTP = $ FFFFFFFF when the clock source is 2 32 clocks.

3.2.10 Timer control

The timer control function consists of a timer control register and a timer interrupt vector register.
I am.

TIC—Timer control register $ YFFA52

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PRR		SWC [2: 0]		DIV2R		RTC [2: 0]					TIV				
RESET:															
0	U	U	U	0	U	U	U	U	U	U	U	U	U	U	U

This register is invalid for RESET (a reset source other than POW), but in POW it sets a value of "$ 00".
To See "3.5 Resets " on page 100 for the types of resets .
Timer control registers are software watchdog down counters and realtors
Select the clock input of the im clock down counter.

Page 87

PRR— Prescaler reset

0 = No reset operation is performed.

1 = Reset the prescaler at the time of writing.

The PRR resets the circuit of the prescaler. Reading this bit always returns zero.
A circuit that divides this bit by 2 by writing 1 to DIV2R at the same time, and this pre connected to it.
It is possible to reset the entire frequency divider circuit of the scaler.

SWC [2: 0] — Software watchdog clock control

The SWC [2: 0] field is used when selecting the input for the software watchdog down counter.
Provides seven possible settings for. See Table 19.

DIV2R— Prescaler (1/2) clock reset

0 = No reset operation is performed.

1 = Resets the circuit that divides by 2 at the time of writing.

DIV2R resets the circuit that divides by 2 immediately before being supplied to the prescaler. Lee this bit
Will always return zero.

RTC [2: 0] — Real-time clock control

The RTC [2: 0] field is available when selecting the input for the real-time clock down counter.
Provides 6 settings. See Table 21 .

Table 21 Software watchdog and real-time clock control settings

Clock control settings			SWC [2: 0]	RTC [2: 0]
0	0	0	System Clock / 2	System Clock / 2
0	0	1	Crystal Reference / 2	Crystal Reference / 2
0	1	0	2^{10} Prescaler Tap	2^{10} Prescaler Tap
0	1	1	2^{8} Prescaler Tap	2^{8} Prescaler Tap
1	0	0	2^{6} Prescaler Tap	2^{6} Prescaler Tap
1	0	1	2^{4} Prescaler Tap	2^{4} Prescaler Tap
1	1	0	For RTC expansion	RTC stop
1	1	1	SWDOG stop	RTC stop

For the SWC [2: 0] field, the real-time down counter is set to zero by setting the RTC extension.
Clock drive the software watchdog down counter each time you decrement
Can be done. Enabling this setting disables the software watchdog feature.
Will be done.

SWDOG or RTC down counters, depending on the system clock and crystal reference settings.
It is possible to drive the clock at the crystal frequency or the frequency obtained by dividing the external clock by two, respectively.
to come. On the other hand, depending on the prescaler setting, the down counter on one of the four prescaler taps.
Can be clock driven.

Note: You can select whether the PLL is running or not. PCLK bit (see “3.2.2” on page 77
If System Protection) is set and the PLL is not running, then the system
The protection prescaler cannot operate. Therefore, SWDOG and RTC clocks
Control field selects system protection prescaler tap or crystal frequency
Even if you select it, it cannot work.

Exiting LPSTOP by reset does not affect the SWC or RTC fields. in this way,
SWDOG and RTC continue to operate with the same clock configuration during LPSTOP and normal operation.

Page 88

TIV—Timer interrupt vector register $ YFFA53

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			TIC									TIV [7: 0]			
RESET:															

0	U	U	U	0	U	U	U	U	U	U	U	U	U	U	U

The timer interrupt vector register should be generated in response to the timer interrupt during the IACK cycle. There is a Kuta number. In POW, set the value of "$ 0F". 100 for reset types See "3.5 Reset " on the page.

3.3 System clock

SLIM's system clock sends timing signals to the IMB module and the external peripheral bus. Supply The MCU has a completely static design, so registers and notes even if the clock rate fluctuates. It does not affect the contents of the list. The hardware and software of the system is the clock in operation. It can respond to rate fluctuations.

The system clock signal can be generated in three ways. Internal Phase Locked Lou The PLL is a clock signal from the internal reference (low speed reference mode) or internal reference (high speed reference mode). You can do it. Alternatively, you can enter an external frequency source (external clock mode). this Keep these clock sources in mind as you read the rest of the chapter. Figure 9 shows Sith It is a block diagram of the system clock .

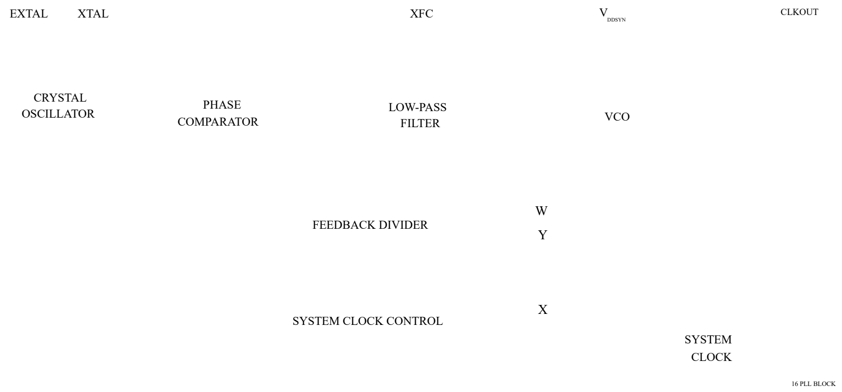
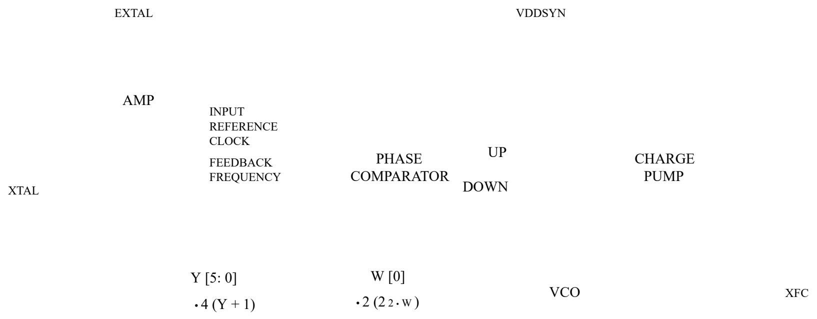


Figure 9 System clock block diagram



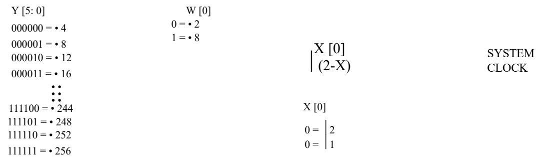


Figure 9 (a) System clock block diagram (low speed reference mode)

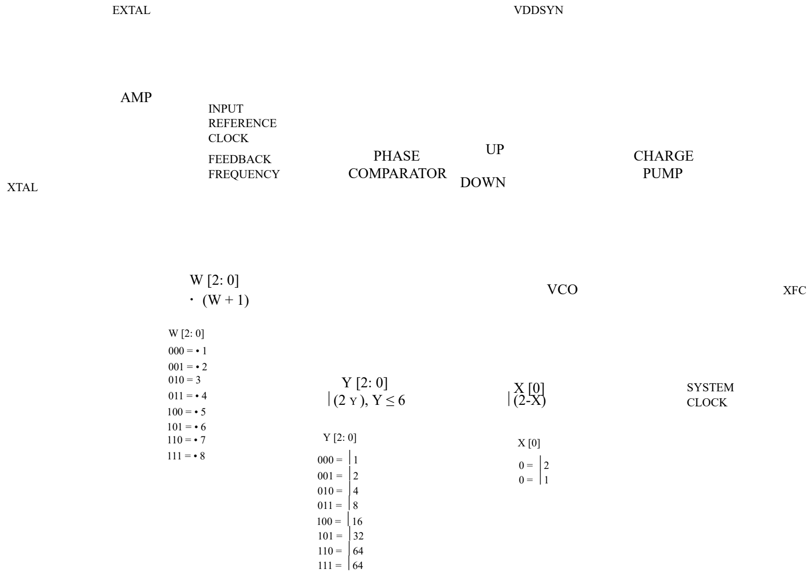


Figure 9 (b) System clock block diagram (high-speed reference mode)

3.3.1 Clock source

The source of the system clock is the port / clock configuration register on reset.
Determined by the state of the two shadow bits of the star (PCON [2] and PCON [0]) and the V DDSYN / MODCLK pin.
Will be fixed. See Table 22 .

Table 22 PLL mode

MODE: MODE: V DDSYN / MODCLKPCON0	SYNCR X / W / Y bit allocation	X / W / Y Reset value	X / W / Y function	Clock formula

External Clock: 0 / X	X
	Y [2: 0]
		X = 1
		Y = 0
			X: divide
			W: no effect
			Y: divide
				FSYSTEM = EXTAL / (2-X) (2^{Y}): (Y ≤ 6)

Fast Reference: 1/0	X
	W [2: 0]
	Y [2: 0]
		PCON [2] = 0[1]
		X = 1
		W = 0
		Y = 0
		PCON [2] = 1[1]
		X = 0
		W = 3
		Y = 0
			X: divide
			W: multiply
			Y: divide
				FSYSTEM = EXTAL (W + 1) / (2-X) (2^{Y}): (Y ≤ 6)

Slow Reference: 1/1	X
	W
	Y [5: 0]
		X = 0
		W = 0
		Y = 63
			X: divide
			W: multiply
			Y: multiply
				FSYSTEM = EXTAL (4) (Y + 1) $(2^{2W + X})$

Note: 1. If “L” is set while the IPIPE1 / DSI pin is reset, the PCON2 shadow bit
The IRQX pin at reset is more than the default value mask-programmed to

Give priority to the state.

PCON [0] Reset from the default reset state mask programmed into the shadow bit
You can prioritize the state of the FREEZE pin at the time. If MODCLK is held at “H” during a reset,
The clock synthesizer produces a clock signal from a crystal oscillator or an external reference input. nine
Lock synthesizer control register (SYNCR) determines operating frequency and various operating modes
To do. If V_{DDSYN} / MODCLK is held at “L” during a reset , the clock synthesizer will be desaved.
Therefore, the external system clock signal must be supplied. Synthesizer is dissed
When bulled, the SYNCR control bit is invalid.

To generate a reference frequency using an inter[redacted]reference crystal between the EXTAL and XTAL pins.
You have to connect. Figure 10 (a) and Figure 10 (nded circuit.

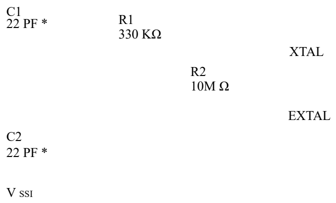
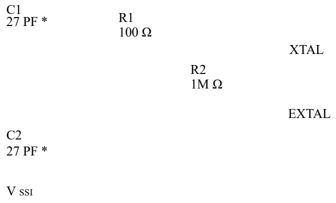


Figure 10

* Resistance and capacitance values are manufactured using DAISHINKU DMX-38 32.768KHz crystal.
Based on the test circuit. The actual part value is determined according to the crystal type
give me. Contact the crystal manufacturer for the exact circuit.

Figure 10 (a) Internal oscillator circuit (low speed reference mode)



* Resistance and capacitance values are tests made using KDS041-18 4.194 MHz crystal.
Based on the circuit. Determine the actual part value according to the crystal type.
Contact the crystal manufacturer for the exact circuit.

Figure 10 (b) Internal oscillator circuit (high-speed reference mode)

If an external reference signal or an external system clock signal is supplied through the EXTAL pin, the XTAL pin will
Must be in a floating state. The external reference signal frequency is greater than or equal to the specified maximum reference frequency.
Must be below. External system clock signal frequency is the specified maximum system clock
Must be below or below the frequency.

Input duty cycle when supplying an external system clock signal (without using a PLL)
It is important to be strict, especially near the maximum operating frequency. Clock signal duty
The relationship between the cycle and the clock signal period is expressed by the following equation.

$$\text{Minimum period of external clock} = \frac{\text{Minimum “H” / “L” time of external clock}}{\text{50\% -external clock input duty cycle Percentage fluctuation}}$$

Each clock mode is explained below.

3.3.2 Clock mode

In slow reference mode, the system clock is generated by the PLL from the internal 32.768kHz reference frequency.
It will be. The frequency of the system clock is controlled by programming the X, Y, W bits of the SYNCR.
Will be done.

Note: In low speed reference mode, crystals other than 32.768 kHz can be used, but on-chip.
The crystal oscillator must be in the 25kHz to 50kHz range for operation.

In high speed reference mode, the system clock is generated by the PLL from a standard 4.194MHz external reference. However, the range is from the specified minimum crystal frequency to the specified maximum system clock frequency. Reset The system clock frequency is determined by the PCON2 shadow bit. High speed reference mode Provides a system clock with a 50% duty cycle, regardless of the reference duty cycle. I will give you.

In external clock mode, the EXTAL pin must be supplied with a clock source. This The lock is used to generate the system clock directly (VCO is turned off). At reset Does not use a divider because the system clock frequency and the input clock frequency are equal. this Strict minimum duty cycle requirements if the frequency is the specified maximum system clock frequency It should not violate.

3.3.3 Clock synthesizer operation

When the internal or external reference frequency is supplied , use V DDSYN / MODCLK to the clock circuit. Power is supplied. If a separate power supply is prepared, the noise margin of the MCU will be improved, and the MCU will be powered down.

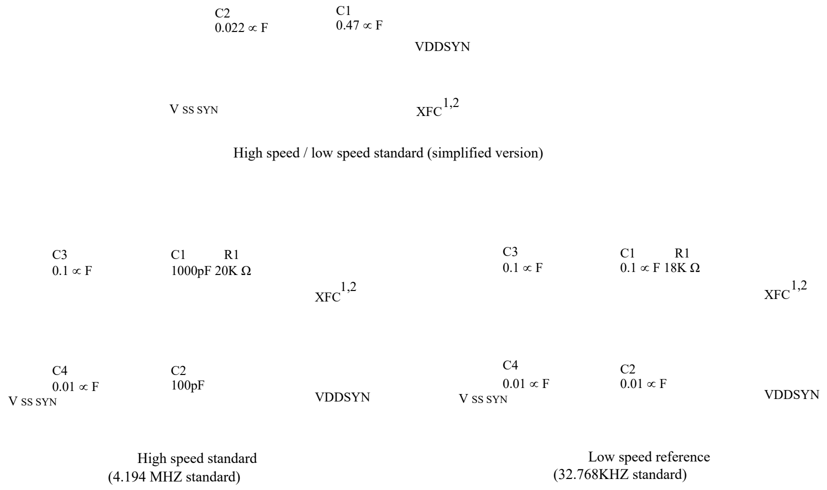
Page 92

You can operate the clock circuit at any time. As the power supply for V DDSYN / MODCLK, use a low noise type power supply. Must be used. As close to the V DDSYN / MODCLK pin as possible for a stable operating frequency Install enough external bypass capacitors. An external system clock signal is applied And when the PLL is disabled, V DDSYN / MODCLK must be connected to GND.

A voltage controlled oscillator (VCO) produces a system clock signal. 50% clock duty To maintain the wheel, the VCO frequency should be twice the clock frequency, depending on the state of the X-bit of SYNCR. Is quadrupled. A portion of this clock signal is fed back to the divider / counter. Divider Controls the frequency of one input of the phase comparator. The other input of the phase comparator is Crysta Reference signal from an oscillator or an external source. The comparator is proportional to the phase difference between its two inputs Generates the control signal. This signal is filtered by a low pass filter to correct the output frequency of the VCO. Used to do.

The configuration of the filter circuit can be changed according to the external environment and the required clock stability. Figure 11 shows the recommended system clock filter network in low speed and high speed reference modes. To do. XFC pin leaks must be kept within specified limits to ensure optimum stability and PLL performance. Must be.

On the XFC pin when an external system clock signal is applied and the PLL is disabled. No need to connect an external filter network. In this case, the XFC pin should be in the float state. Put it down please.



High speed / low speed standard (simplified version)

High speed standard
(4.194 MHZ standard)

Low speed reference
(32.768KHZ standard)

Figure 11 System Clock Filter Network

Note: 1. Maintain low leaks on XFC nodes. For details, see "Appendix A Electrical Characteristics."
Please refer to.
2. Recommended loop filter to reduce sensitivity to low frequency noise.
3. Make sure that V DD SYN and V SS SYN do not get noise from other power supplies.

The synthesizer locks when the divided VCO frequency is equal to the EXTAL frequency. Lock time Is affected by the time constant of the filter and the difference between the two comparator inputs. Comparator input

The synthesizer must be relocked each time it changes. Lock status is SYNCR
Indicated by the SLOCK bit. During power-up, the MCU will reset until the synthesizer locks.
You will not be released from the state. Depending on the crystal type, characteristic frequency, and layout of the external oscillator circuit
The lock time is affected.

When using a clock synthesizer, the control register SYNCR has an operating frequency and various types.
Determines the operating mode.

In slow reference mode, the PLL feedback path has one W bit and six Y bits, up to 2048.

Page 93

You can multiply the frequency up to twice. The X-bit is in the VCO clock output path and interferes with the PLL.
The system clock frequency can be divided by two. The clock frequency is calculated from the following equation.
It will be.

$$F_{SYSTEM} = EXTAL\ [4 (Y + 1) \cdot (2^{2W + X})]$$

In fast reference mode, the PLL feedback path has three W bits, multiplying the frequency by 1-8 times.
can. The three Y and X bits are in the VCO clock output path and interfere with the PLL.
You can slow down the system clock without. The clock frequency can be calculated from the following equation.

$$F_{SYSTEM} = EXTAL\ (W + 1) / ((2 - X) \cdot (2^{Y})): (Y \leq 6)$$

In external clock mode, there are three Y and X bits between the EXTAL input and the system clock.
Therefore, the clock can be slowed down to reduce power consumption. The clock frequency is calculated from the following equation.
It will be.

$$F_{SYSTEM} = EXTAL / ((2 - X) \cdot (2^{Y})): (Y \leq 6)$$

The clock frequency selected by the W, X, and Y bits for the device to operate properly.
Must be within the limits specified for the MCU.

The VCO frequency can be calculated from the following equation.

$$FVCO = F_{SYSTEM}\ (2 - X)$$

3.3.4 Clock control

The clock control circuit is a system in special situations such as the disappearance of synthesizer reference or low power operation.
Determines the system clock frequency and clock operation. The clock source is the port at reset
/ Two shadow bits in the clock configuration registers (PCON2 and PCON0), and
And V DDSYN / MODCLK Determined by the state of the pin.

SYNCR—Clock synthesizer control register $ YFFA04

Slow reference mode

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
X	Y			W [5: 0]				EDIV	STOSC	LOSCD	SLIMP	SLOCK	RSTEN	STSLIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	0	1	0	0	0

High speed reference mode

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
X		W [2: 0]		NOT USED		Y [2: 0]		EDIV	STOSC	LOSCD	SLIMP	SLOCK	RSTEN	STSLIM	STEXT
RESET: (DSI = 1)															
PCON2 (0)	0	PCON2 (1)	PCON2 (1)	0	0	0	0	0	0	0	0	1	0	0	0
RESET: (DSI = 0)															
IRQX Pin state	0	IRQX Pin state	IRQX Pin state	0	0	0	0	0	0	0	0	1	0	0	0

External clock mode

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
X		NOT USED				Y [2: 0]		EDIV	STOSC	LOSCD	SLIMP	SLOCK	RSTEN	STSLIM	STEXT
RESET:															
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Page 94

The clock synthesizer control register (SYNCR) is read in supervisor mode. It is possible to write. The encoding and default values for the high-order bytes of SYNCR are selected at reset. It depends on the clock mode.

In slow reference mode, the default value for high-order bytes is $ 3F, which uses a 32.768kHz crystal. If this is the case, the operating frequency will be 8.32MHz.

When PCON2 is set to zero in fast reference mode, the default value for high-order bytes is $ 80, which is the reference. There is a one-to-one correspondence to frequencies. If PCON2 is set to 1, the default value for high-order bytes is $ 30. The operating frequency is twice the reference frequency, and if a 6MHz crystal is used, the operating frequency is 12MHz. It will be the wave number.

In external clock mode, the default value for high-order bytes is $ 80, which is the input frequency on EXTAL. Corresponds to the operating frequency. The default value is the default of the other bits in the register at reset. It is forced to be set to SYNCR along with the value.

W— Frequency control bit

The field length and function of this bit depends on the clock mode. In slow reference mode Multiplies the reference frequency using only the W bit. In fast reference mode, 3 W bits Use to multiply the reference frequency.

X— Frequency control bit

This bit controls the 1-bit divider that drives the system clock in all modes. vinegar. When X is set, the divider is bypassed, and when it is cleared, the system The clock is divided by two. This bit does not affect the VCO's lock bit.

Y— Frequency control bit

The field length and function of this bit depends on the clock mode.

In slow reference mode, the six Y bits are used to multiply the reference frequency.

In fast reference mode, the three Y bits are used to divide the PLL output frequency. External clock The mode uses three Y bits to divide the input clock frequency. This mode is for the VCO Does not affect the lock bit.

EDIV—E Clock division rate

0 = ECLK frequency is system clock ÷ 8
1 = ECLK frequency is system clock ÷ 16

STOSC— Oscillator stopped

0 = The crystal oscillator circuit continues to operate even during LPSTOP.
1 = Crystal oscillator circuit and PLL are turned off during LPSTOP to save power. On the tip All clocks stop.

The STOSC bit does not affect the system clock in external clock mode. STOSC bit If set, the MCU will wake up with an external interrupt or reset. STOSC bit If is cleared, the interrupt priority field in the condition code register All interrupts higher than (IP) wake up the MCU.

LOSCD — Clock Loss Oscillator Disable

0 = Enables the clock loss oscillator.
1 = Clock loss The oscillator is disabled and is not operating.

SLIMP—Limp mode flag

0 = The system clock is normally supplied by an external clock from the PLL or EXTAL input. Have been paid.
1 = System clock disappearance detected, system clock is based on crystal disappearance oscillator It is supplied from.

When using the built-in synthesizer, SLIMP is set when the reference frequency is lost. The VCO continues to operate using the reference control voltage.

SLOCK—Synthesizer lock flag

0 = Not locked when VCO is enabled.
1 = VCO is locked when enabled. Or it is always set when the VCO is disabled ing.

Page 95

RSTEN—Reset Enable

When 0 = SLIMP = 1, the MCU continues to operate in limp mode.
A system reset occurs when 1 = SLIMP = 1.

STSLIM — Stop mode SLIM clock

0 = When the LPSTOP instruction is executed, the SLIM clock is driven from the crystal oscillator and the VCO is stopped. Stop to save power.

When the 1 = LPSTOP instruction is executed, the SLIM clock is driven from the VCO.

STEXT—Stop mode external clock

0 = When the LPSTOP instruction is executed, the external clock is not driven.

1 = When the LPSTOP instruction is executed, the external clock is driven according to the state of the STSLIM bit.

3.4 External bus interface

The external bus interface (EBI) is used by the CPU inside the MCU to access external devices. vinegar. The external bus has 16 multiplexed address lines and 16 data lines.

EBI can function as a bus master (master mode).

When the CPU inside the MCU is enabled, the EBI acts as the bus master and in the internal bus cycle. Allows access to external memory resources. EBI is a chip select base address Programming the gista and option registers, and programming the port D pin assignment registers Operates an external bus cycle based on the chip select pin options given.

EBI supports byte, word, and long word transfers. The port is a data transfer Asynchronous cycle controlled by send (SIZE) and data transfer acknowledge (DTACK) pins. You will be access. 16-bit wide (full) allows access to 8-bit and 16-bit data ports vinegar. With 8-bit width (half), only 8-bit ports can be accessed. Switching to an 8-bit port Sending may require multiple bus cycles.

To increase flexibility and require less external logic, the MCU uses a chip select circuit as an external bus. You can sync to access. Chip select logic also provides these accesses. On the other hand, the bus control signal generated internally can be supplied. For more information, see 3.8 Chips. See Lect .

3.4.1 Master mode

The external bus has five master operating modes:

- 16-bit bus width, non-multiplexed master mode
- 8-bit bus width, non-multiplexed master mode
- 16-bit bus width, multiplexing master mode
- 8-bit bus width, multiplexing master mode
- Single chip

External bus operation is synchronized with the clock (CLKOUT) output signal. External bus speed is DTACK 2 to 14 clock cycles (0 to 12 waits) due to optional external bus termination using the number Tate) can be programmed. Bus speed is the Chip Select Option Register (CSOR) Programmed in the DTACK field of.

External chip select pins and DTACK pins may not be available in some configurations, but few. Allocate at least one chip select channel and provide bus cycle end timing You have to do it. The chip select channel internally terminates the bus cycle. Can be done. Of course, using the optional chip select pin externally, the appropriate memory mode You can also select joules and peripherals.

EBI has its own external bus controller. When SLIM is not in single-chip mode The internal and external buses are not separated. Limited support for another external master.

The external bus sets the MODE bit in the module configuration register (SLIMCR). Is configured. If you configure SLIM as a bus master, the chip select channel is also a bus. Cycle length, port size (8-bit or 16-bit data bus), strobe characteristics, etc. Must be programmed to indicate the bus attributes of.

Table 23 shows the “master mode signals ”.

Table 23 Master mode signals

				action mode				
Credit	pin Number of	port	In need Pin	MUX 16-bit	Non-MUX 16-bit	MUX 8-bit	Non-MUX 8-bit	single Tip
A23 / ECLK	1	C	No	Optional	Optional	Optional	Optional	Optional
ADDR [22:16] [2]	8	C	No	Optional	Optional	Optional	Optional	No
AD [15: 0]	16 16	A, B	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No
ALE	1	F	No	Yes Yes	No	Yes Yes	No	No
AS	1	E	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No
BKPT / DSCLK / CSE0	1	D	Yes Yes	CPU Pin	CPU Pin	CPU Pin	CPU Pin	CPU Pin

CLKOUT	1	E	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
CSA	1	D	No	Optional	Optional	Optional	Optional	No
CSB	1	D	No	Optional	Optional	Optional	Optional	No
CSC	1	D	No	Optional	Optional	Optional	Optional	No
DATA [15: 8]	8	G	No	No	Yes Yes	No	Yes Yes	No
DATA [7: 0]	8	H	No	No	Yes Yes	No	No	No
DTACK	1	D	No	Optional	Optional	Optional	Optional	No
DS [1]	1	E	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No
EBR / TSC	1	————	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
EXTAL, XTAL	2	————	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
FC [2: 0] [2]	3	E	No	Optional	Optional	Optional	Optional	No
FREEZE / QUOT / CSE1	1	D	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
IPIPE0 / DSO	1	D	Yes Yes	CPU Pin	CPU Pin	CPU Pin	CPU Pin	CPU Pin
IPIPE1 / DSI	1	D	Yes Yes	CPU Pin	CPU Pin	CPU Pin	CPU Pin	CPU Pin
IRQ [6: 2]	Five	F	No	Optional	Optional	Optional	Optional	Optional
IRQX / BERR, IRQ7	2	F	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
RESET	1	————	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
R / W [1]	1	E	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	No
SIZE	1	E	Yes Yes	Yes Yes	Yes Yes	No	No	No
V_{DDSYN} , V_{SSSYN}	2	————	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes
XFC	1	————	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes	Yes Yes

Note: 1. R / W and DS can be replaced with RD and WR.
2. This signal or some signals in the bus are not connected on the MC68HC (9) 16Y5 / 916Y6.

3.4.2 Address Bus

The address bus signal AD [15: 0] uses the high-order byte address during the bus cycle. The MCU drives the address on the bus when it begins the bus cycle. This address is an assassin for AS It is valid while it is being played.

3.4.3 Data Bus

In multiplex mode, 16 bits whenever address latch enable (ALE) is asserted. A valid address is loaded on the address bus (AD [15: 0]). After the ALE is negated DS is asserted and valid data is loaded on the AD [15: 0] bus.

In non-multiplexed master mode, the data bus signal D [15: 0] becomes a bidirectional parallel bus with the MCU. Transfer data between. 8-bit or 1 bus cycle by read or write operation Transfers 16-bit data. In the read cycle, the last clock in that bus cycle The MCU latches the data at the falling edge. In the write cycle, the port width and the operand size All 16 bits of the data bus are driven regardless of the size (16-bit bus width master). mode). The MCU data data after a half-clock cycle in which the AS is asserted in the write cycle. Drive to Ta Bus.

3.4.4 Bus control signal

In master mode, the internal CPU has the address, size, function code, and read / file. Start the bus cycle by driving the output.

At the beginning of the cycle, the SIZE signal is driven along with the function code signal. SIZE signal Is used only for 16-bit port sizes. The size signal is then in the operand cycle Indicates word access or byte access for (see Table 23). The size signal is an address straw Valid as long as the (AS) is asserted. Read / write (R / W) signals are forwarded during the bus cycle Determine the direction of. This signal changes state at the beginning of the bus cycle when requested and the AS is asserted. It is valid while it is. The R / W signal is from “H” if there is a read cycle before the write cycle. Change the state to “L”. This signal will continue to be in the “L” state during two consecutive write cycles. I will continue.

3.4.4.1 Address Strobe

Address Strobe (AS) indicates that the address on the address bus and various control signals are valid.
It is a timing signal. This signal is asserted half a clock after the bus cycle begins. Many
In overlay mode, the valid state of address, function code, size, and R / W is ALE.
Affected by the signal.

3.4.4.2 Data strobe

The data strobe (DS) is a timing signal. In the read cycle, the MCU asserts the DS
Instruct the external device to drive the data on the bus. In the read cycle, DS is AS
It is asserted at the same time. In the write cycle, the DS tells you that the data on the data bus is valid.
Notify external devices. In write cycles, the MCU asserts AS for one clock cycle.
Later assert DS.

3.4.4.3 Size signal

Size (SIZE) indicates the size of the transferred data. The size signal is valid while AS is asserted
vinegar. The size encodings are shown in Table 24 .

Table 24 SIZE signal encoding

Port size	SIZE	Transfer size
16 bit	1	Part-Time Job
16 bit	0	word

Page 98

3.4.4.4 Function code

The function code (FC [2: 0]) is automatically generated by CPU16. CPU16 is always FC2
Due to the high output, FC1 and FC0 effectively select one of the four address spaces. (FC2 is CPU16
This is the signal used by the CPU32, which is higher than the above.) FC [1: 0] =% 00 is reserved. The remaining three skies
The space is called program space, data space, and CPU space. CPU space is normal read or lie
Used for specific control information that is not related to the bus cycle. The function code signal is AS
Valid while is asserted. The function code encodings are shown in Table 25 .

Table 25 Function Code Encoding

FC2	FC1	FC0	Address space
1	0	0	spare
1	0	1	Data space
1	1	0	Program space
1	1	1	CPU space

3.4.4.5 Read / write signal

The R / W determines the forwarding direction during the bus cycle. This signal is a bus cycle as needed
It changes state at the beginning of and is valid while AS is asserted.

3.4.4.6 Bus cycle end signal

In master mode, as programmed in the Chip Select Options Register (CSOR)
The internal chip select channel ends the bus cycle. The DTACK pin (if available) is
It can be used to end the cycle early. When the DTACK pin is assigned as a function of DTACK
If the active low is asserted, the EBI will be chip select timer wait size.
End the bus cycle without waiting for the number of cars. The DTACK pin is assigned as a generic port,
Or if it is assigned as a DTACK function but not asserted, the chip select is
End the cycle internally.

3.4.5 Data transfer mechanism

The MCU architecture supports byte or word operand transfers. EBI is 8-bit
Data bus (8-bit bus width master mode) or 16-bit data bus (16-bit bus)

Width master mode), configured to support 8-bit or 16-bit port peripherals
The data bus cannot be dynamically sized to accommodate rals.

Each port (a port is the data bus width used by an external device during data transfer) is the data.
Assigned to a specific bit on the bus. 16-bit ports are assigned to data bus bits [15: 0]
It is allocated and the 8-bit port is assigned to the data bus bit [15: 8].

In multiplex mode, the address and data share the same bus. At the beginning of the bus cycle,
The dress will be driven onto the address / data bus. At that time, the slave addresses the address.
I have to do it. The address / data bus is then used for data transfer.

Table 26 Port size configuration in master mode

SZA	SZB	SZC	Port size
0	0	0	8 bit
1	X	X	16 bit
X	1	X	16 bit
X	X	1	16 bit

Note: The SZA, SZB, and SZC bits are in the chip select control register.

Page 99

The MCU always attempts to transfer the maximum amount of data in every bus cycle. in short,
Word operations assume that the port size is 16 bits wide when the bus cycle begins.
increase. The operand bytes are named as shown in Figure 12. OP0 is a long word opera
The most significant byte of the input, OP3 is the least significant byte. The 2 bytes of the word length operand are OP0 (most significant).
It is OP1. One byte of the byte length operand is OP0.

operand	Byte order			
	31 24	23 16	15 8	7 0
Long word	OP0	OP1	OP2	OP3
3 bytes		OP0	OP1	OP2
word			OP0	OP1
Part-Time Job				OP0

Figure 12 Operand byte order

3.4.6 Operand alignment

The data multiplexer makes the required connections for different combinations of addresses and data sizes.
I will do it. The data multiplexer takes 2 bytes of the 16-bit bus and places them where they are needed.
Forward to. For alignment, the data size is transferred word by word. See Table 23
Please.

AD0 also affects the operation of the data multiplexer. During operand transfer, AD [23: 1] is accessed
Indicates the word-based address portion of the operand to be struck, where AD0 is byte-off from that base.
Indicates a set. Note that AD [23:20] follows the state of AD19 on the MCU.

Note: AD [22:19] is enabled on the IMB on the MC68HC (9) 16Y5 / 916Y6, but externally.
Not connected to a pin.

3.4.7 Operand misalignment

The CPU16 processor architecture uses a 16-bit basic operand size. Opera
Ends are misaligned when crossing word boundaries. This depends on the value of AD0.
When AD0 = 0 (even address), the address is on word and byte boundaries. AD0 = 1 (odd number a)
When dressed), the address is only on the byte boundary. Byte operand at any address
Align correctly. Word operands or long word operands are mi at odd addresses
It will be a alignment.

In MC68HC (9) 16Y5 / 916Y6, the maximum amount of data that can be transferred in one bus cycle is one aligned word.
vinegar. If the MCU forwards longword operands through a 16-bit port, the first bus server
The most significant operand word is transferred in the cycle, and the least significant operand word is transferred in the next bus cycle.
Will be transferred.

CPU16 can perform misaligned word transfers. With this feature, M68HC11 CPU and software
Can be made compatible with. CPU16 miss-aligns long word transfers with two miss-alignments
Treat as a word transfer.

3.4.8 Operand transfer case

EBI by address bus and data bus (multiplexed or non-multiplexed bus) and various control signals
And perform data transfer between external devices. A signal issued by the bus master, which is a signal in the synchronous bus structure.

Controls the movement of data in the space

Table 27 summarizes the operand alignment status for various transfers. Items indicated by OPn are 1
The part of the request operand that is read or written in a cycle, for that bus cycle
It is defined by SIZE and AD0.

Page 100

Table 27 Operand Alignment

Transfer case	SIZE	AD0 [2]	Number of cycles	Cycle number	DATA [15: 8] [2]	DATA [7: 0] [2]
Byte to 8-Bit Port (Even / Odd)	X	X	1	1'st	OP0	XXX
Byte to 16-Bit Port (Even)	1	0	1	1'st	OP0	XXX
Byte to 16-Bit Port (Odd)	1	1	1	1'st	XXX	OP0
Word to 8-Bit Port (Aligned)	X	0	2	1'st	OP0	XXX
	X	1		2'nd	OP1	XXX
Word to 8-Bit Port (Misaligned) [1]	X	1	2	1'st	OP0	XXX
	X	0		2'nd	OP1	XXX
Word to 16-Bit Port (Aligned)	0	0	1	1'st	OP0	OP1
Word to 16-Bit Port (Misaligned) [1]	1	1	2	1'st	XXX	OP0
	1	0		2'nd	OP1	XXX
Long Word to 8-Bit Port (Aligned)	X	0	Four	1'st	OP0	XXX
	X	1		2'nd	OP1	XXX
	X	0		3'rd	OP2	XXX
	X	1		4'th	OP3	XXX
Long Word to 8-Bit Port (Misaligned) [1]	X	1	Four	1'st	OP0	XXX
	X	0		2'nd	OP1	XXX
	X	1		3'rd	OP2	XXX
	X	0		4'th	OP3	XXX
Long Word to 16-Bit Port (Aligned)	0	0	2	1'st	OP0	OP1
	0	0		2'nd	OP2	OP3
Long Word to 16-Bit Port (Misaligned) [1]	1	1	Four	1'st	XXX	OP0
	1	0		2'nd	OP1	XXX
	1	1		3'rd	XXX	OP2
	1	0		4'th	OP3	XXX

Note: 1. CPU16 transfers misaligned long word to two misaligned words
Treat as sending.
2. In multiplex mode, DATA [15: 0] corresponds to AD [15: 0].

3.4.9 External bus in low power stop mode interface

In low power stop mode (LPSTOP), the EBI is based on the state of the EBR pin to the external bus pin. Controls the state of If EBR is asserted, SLIM will address, address / data, AS, DS, R / Does not drive the W, SIZE, BERR, ALE, and DTACK pins. If the EBR is negated, SLIM will Dress, address / data, AS, DS, R / W, SIZE, BERR, and DTACK pins at the end of each known state Drive to the state. The EBR does not need to be held in the “L” state during LPSTOP. Narrow pulse Only one (25ns) is required.

Page 101

3.5 reset

The reset procedure initializes the system and recovers from a catastrophic failure. This MCU is hard
A Reset is executed by the combination of software and software. SLIM first determines if the reset is enabled
Judge and assert the control signal. And the basic system configuration and hardware mode
After making a boot ROM selection based on the lect input, it passes control to CPU16.

A reset occurs when the active “L” logic level of the RESET pin is clocked into SLIM.
vinegar. The reset is gated by the CLKOUT signal. Asynchronous reset is for catastrophic situations
is. Asynchronous resets can occur at any clock edge. Synchronous reset is a bus cycle
It is timed to occur at the end of. If there is no clock when RESET is asserted
If so, the reset will not occur until the clock is started. When the reset is clock driven, RESET is
The write cycle that was in progress when asserted can be completed.

Reset is the highest priority exception handling for CPU16. Therefore, a reset is issued while the program is in progress.
If it occurs, it will be aborted by reset exception handling and will be restarted from the reset. Reset
Only basic tasks are performed during exception handling. Other initialization tasks are Handler Lou
Chin has to do.

RSR—Reset status register $ YFFA07

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								EXT	POW	SW	DBF	0	LOC	SYS	TST
RESET:																

The reset status register has a bit corresponding to each reset source on the MCU.
Setting the bit to 1 shows the type of reset that occurred. Same multiple reset sources
If this happens at times, you can set multiple bits in the RSR. The reset status register is
It is updated by the reset control circuit when the MCU is released from the reset state. This register
Can be read at any time. Writing has no effect.

EXT—External reset

A reset was caused by an external signal.

POW—Power-up reset

A reset occurred due to the power-up reset circuit. POW powers on reset
Synonymous with (POR).

SW—Software Watchdog Reset

The software watchdog circuit caused a reset.

DBF—Double bass fault reset

A reset has occurred due to the system protection submodule Holt Monitor.

LOC—Clock loss reset

A reset occurred due to the loss of the clock submodule reference frequency. Clock submo
This reset occurs only when the Joule RSTEN bit is set and the VCO is enabled.
I can do it.

SYS—system reset

The CPU reset instruction causes a reset, but this is because CPU16 does not have a reset instruction.
Bits are not used and are always read at zero.

TST—Test submodule reset

The test submodule caused a reset.

Page 102

3.5.1 Reset mode selection

Each time SLIM is reset, its behavior is set internally or from external logic. SLIM

The configuration consists of the SLIM module configuration shadow register (MCRC) and the port controller.
Figuration shadow register (PCON), and BKPT and V DDSYN / MODCLK pixels at reset.
It is set based on the status of the computer.

Table 28 shows " External pin configuration control during reset ".

Table 28 External pin configuration control at reset

Configuration pin	Affected pins	The state of the pin at reset For “H”	The state of the pin at reset For “L”
BKPT / DSCLK / CSE0 / PD2	FREEZE / QUOT / CSE1 / PD3 BKPT / DSCLK / CSE0 / PD2 IPIPE1 / DSI / PD1 IPIPE0 / DSO / PDO	Background mode disabled, pin function is digital I / O (PD [3: 0])	Background mode enabled
V DDSYN / MODCLK	————	Crystal reference mode	External clock mode
IPIPE1 / DSI / PD1	————	Default reset configuration controlled by MCRC and PCON	Default configuration controlled by external pins. See Table 30
AD14 / PA6 1	————	Flash enabled (if STOP bit = 0) / ROM enabled	Flash disabled / ROM disable

Note: 1. Valid only when DSI = 0. If DSI = 1,'Flash Enable' (when shadow STOP bit = 0) and
Is on the'ROM Enable'side.

Table 29 Default configuration values

option	register	value	Explanation
High or Low Current Drive on Output Drivers	MCRC15	1	High Drivers
Freeze Software Enable & Freeze Bus Monitor Enable	MCRC [14:13]	00 00	Both Enabled
Motorola / Non-Motorola Configuration	MCRC12	1	Motorola Config.
Mode Configuration Multiplexed Master Mode 1 Nonmultiplexed Master Mode Single Chip Mode	MCRC [11: 9]	111	Single Chip Mode
Show Cycle Enable	MCRC8	0	Disabled Disabled.
Supervisor / Unrestricted SLIM Register Access	MCRC7	0	Unrestricted
MODMAP	MCRC6	1	$ FFF000- $ FFFFFF
SLIM Boot ROM Array Supervisor / Unrestricted Access	MCRC5	0	Unrestricted
SLIM Boot ROM Enabled / Disabled	MCRC4	0	Disabled Disabled.
Initial Interrupt Arbitration Level	MCRC [3: 0]	1111	Highest Priority
Configuration of A23 / ECLK / PC7 Pin Function	PCON [15:14]	00 00	PC7
Configuration of A [22:16] / PC [6: 0] Pins 2	PCON [13:11]	011	A [18:16], PC [6: 3]
CSA / PD7 Pin Function	PCON10	0	PD7
CSA Port Size	PCON9	1	16-bit port
CSB / PD6 Pin Function	PCON8	0	PD6
CSB Port Size	PCON7	1	16-bit port
CSC / PD5 Pin Function	PCON6	0	PD5
CSC Port Size	PCON5	1	16-bit port
DTACK / PD5 Function	PCON4	0	PD4
FC [2: 0] / PE [7: 5] Function 2	PCON3	0	PE [7: 5]

Table 29 Default configuration values (continued)

option	register	value	Explanation
2X / 1X Reference System Clock	PCON2	1	2X
CLKOUT / PE4 Pin Function	PCON1	0	PE4
Fast / Slow Crystal Reference	PCON0	0	Fast Reference
Contents of the Boot ROM Array		————	Generic Code
Security Mode Keys, Addresses, Sequence & Enable / Disable		0	Disabled Disabled.
R / W / WR / PE0 & DS / RD / PE1 Pin Functions	CSCR0	0	R / W & DS
Whether CSB is synchronized to AS or DS at reset	CSORB10	0	AS

Note: 1. The size of the CSA, CSB, and CSC determines the size of the external data bus.

If any CS is set to 16 bits, the external data bus will be 16
Bit width.

2. This signal or some signals in the bus are not connected on the MC68HC (9) 16Y5 / 916Y6.

MCRC—SLIM MCR configuration shadow register $ YFFA0A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MCRC 15	MCRC [14:13]		MCRC 12	MCRC [11: 9]			MCRC 8	MCRC 7	MCRC 6	MCRC Five	MCRC Four	MCRC [3: 0]			
1	0	0	1	1	1	1	0	0	1	0	0	1	1	1	1

For MC68HC (9) 16Y5 / 916Y6 , this register is the "default configuration value " in Table 29.
It becomes the value shown in. This register is a mask option.

The MCR configuration shadow register is the default register for the corresponding bit of the MCR.
A register containing set values. If the DSI pin was not asserted at reset, the mass
A data reset loads data from this register into the MCR. DSI pin is assassin at reset
If so, on the address / data bus at reset rather than some shadow bits on the MCRC
The information driven to is prioritized.

PCON—Port / Clock Configuration Shadow Register $ YFFA0E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PCON [15:14]		PCON [13:11]			PCON10	PCON9	PCON8	PCON7	PCON6	PCON5	PCON4	PCON3	PCON2	PCON1	PCON0
0	0	0	1	1	0	1	0	1	0	1	0	0	1	0	0

For MC68HC (9) 16Y5 / 916Y6 , this register is the "default configuration value " in Table 29.
It becomes the value shown in. This register is a mask option.

3.5.2 Set the default setting at reset from the external pin

When IPIPE1 / DSI / PD1 is driven externally to logic 0, it will reveal certain pins at reset.
By driving to the first defined logical level, in the PCON and MCRC shadow registers
You can disable the default reset value for a bit. SLIM is held in the reset state
In the meantime, AD [15: 8] / PA [7: 0], AD [7: 0] / PB [7: 0], DS / RD / PE1, R / W / WR / PE0, SIZE / PE3, and FREEZE / QUOT /
The CSE1 / PD3 pin is configured as an input with a weak pull-up . Table 30 is set by external pins
It is a summary of the contents.

Table 30 Set the default setting at reset from the external pin (DSI = 0)

to be influenced pin	Reset Sha Doe Bit	Override Pin (s)	If DSI = 0, the mode being reset Influence of select "H"			If DSI = 0, the mode being reset Influence of select "L"
			State of Select Pins			Mode Selected
			DS	SIZE	R / W	
All Ports	MCRC [11: 9]	DS / RD / PE1, SIZE / PE3, R / W / WR / PE0	1	0	0	Non-multiplexed Master
			1	0	1	Multiplexed Master
			1	1	X	Single Chip
			Other combinations			Do not use

Table 30 Set the default setting at reset from the external pin (DSI = 0) (continued)

to be influenced pin	Reset Sha Doe Bit	Override Pin (s)	If DSI = 0, the mode being reset Influence of select "H"			If DSI = 0, the mode being reset Influence of select "L"
			State of Select Pins			Clock Source
			V DDSYN	FREEZE		
XTAL, EXTAL	PCON0	V DDSYN , FREEZE	0	x x		External Clock Input
			1	0		Fast Reference
			1	1		Slow Reference
	TPU FLASH Shadow STOP Bit (MC68HC916Y5 / 916Y6)	AD [12]	1 (If shadow STOP = 0)			TPU Flash ROM Normal Mode
			0			TPU Flash ROM Disable
	FLASH32K / 64K Ito's Shadow STOP Bit (MC68HC916Y5 / 916Y6)	AD [14]	1 (If shadow STOP = 0)			Flash ROM Normal Mode
			0			Flash ROM Disable
			Mask ROM Configuration			Mask ROM status
			AD14	AD13	AD10	
	(MC68HC16Y5)	AD [14:13] / PA [6: 5], AD10 / PA2	1	1	X	Normal Mode
			1	X	1	Normal Mode
			1	0	0	Enable Emulator mode
			0	X	X	Disable Disable
	MCRC4	AD11 / PA3	Boot ROM Enabled (CSBoot Disabled)			Boot ROM Disabled (CSBoot Enabled)

		AD10	Emulator Mode Disable			Emulator Mode Enable
FC [2: 0] / PE [7: 5}[1]	PCON3	AD9 / PA1	FC [2: 0]			PE [7: 5]
DTACK / PD5	PCON4	AD8 / PA0	DTACK			PD4
CSB / PD6	PCON8	AD7 / PB7	CSB			PD6
	PCON7	AD6 / PB6	CSB = 16-bit Chip Select			CSB = 8-bit Chip Select
A23 / PC7 / ECLK	PCON [15:14]	AD [5: 4] / PB [5: 4]	AD5		AD4	Pin function selected
			0		0	PC7
			0		1	ECLK
			1		X	A23
A [22:16] / PC [6: 0][1]	PCON [13:11]	AD [3: 1] / PB [3: 1]	AD3	AD2	AD1	Pin function selected
			0	0	0	PC [6: 0]
			0	0	1	PC [6: 1], A16
			0	1	0	PC [6: 2], A [17:16]
			0	1	1	PC [6: 3], A [18:16]
			1	0	0	PC [6: 4], A [19:16]
			1	0	1	PC [6: 5], A [20:16]
			1	1	0	PC [6], A [21:16]
			1	1	1	A [22:16]
CLKOUT / PE4	PCON1	AD0 / PB0	CLKOUT			PE4
	PCON2	IRQX / BERR / PF1	Fast Reference Mode SYNCR Bits			Fast Reference Mode SYNCR Bits
			X	W [2: 0]	Y [2: 0]	X W [2: 0] Y [2: 0]
			0	0 1 1	0 0 0	1 0 0 0 0 0 0

Note: 1. This signal or some signals in the bus are not connected in MC68HC (9) 16Y5 / 916Y6 is.

In normal operation, the user should pull up the TSC pin (# 113) to "H".

3.5.2.1 Port / clock configuration

Port / clock configuration shadow registers (PCON) are ports C, D, E, F
Default reset values and defaults for the corresponding bits and fields in the pinout register
A mask programmable register that determines the lock configuration. See Table 31 .

On a master reset, the configuration specified based on the configuration information programmed into PCON is loaded.
Will be done. Asserting DSI at reset for some port pin assignment bits and clock selection
On the other hand, the PCON setting is ignored, so the setting is determined by the state of the external pin at reset.
Some bits in the port pin assignment register do not have a shadow bit corresponding to PCON.

Table 31 Port / Clock Configuration Shadow Register (PCON)

Bit (s)	State	register	Affected pins	Selected function
PCON [15:14]	0 0	PCPAR [7: 6]	ADDR23 / ECLK / PC7	PC7
	0 1			ECLK
	1 X			ADDR23
PCON [13:11][1]	0 0 0	PCPAR [2: 0]	ADDR [22:16] / PC [6: 0]	PC [6: 0]
	0 0 1			PC [6: 1], ADDR16
	0 1 0			PC [6: 2], ADDR [17:16]
	0 1 1			PC [6: 3], ADDR [18:16]
	1 0 0			PC [6: 4], ADDR [19:16]
	1 0 1			PC [6: 5], ADDR [20:16]
	1 1 0			PC [6], ADDR [21:16]
	1 1 1			ADDR [22:16]
PCON10	1	PDPAR7	CSA / PD7	CSA
	0			PD7
PCON9	1	CSCR3	PH [7: 0]	CSA = 16-Bit Port
	0			CSA = 8-Bit Port
PCON8	1	PDPAR6	CSB / PD6	CSB
	0			PD6
PCON7	1	CSCR2	PH [7: 0]	CSB = 16-Bit Port
	0			CSB = 8-Bit Port
PCON6	1	PDPAR5	CSC / PD5	CSC
	0			PD5
PCON5	1	CSCR1	PH [7: 0]	CSC = 16-Bit Port
	0			CSC = 8-Bit Port
PCON4	1	PDPAR4	DTACK / PD4	DTACK
	0			PD4
PCON3[1]	1	PEPAR [7: 5]	FC [2: 0] / PE [7: 5]	FC [2: 0]
	0			PE [7: 5]

PCON2	1	SYNCR [15,13:12]	N ONE	System Clock = 2 X Reference
	0			System Clock = 1 X Reference
PCON1	1	PEPAR4	CLKOUT / PE4	CLKOUT
	0			PE4
PCON0	1	————	N ONE	Slow Crystal Reference Mode
	0			Fast Crystal Reference Mode

: Default configuration

Note: 1. This signal or some signals in the bus are not connected in MC68HC (9) 16Y5 / 916Y6 is.

3.5.3 Reset timing

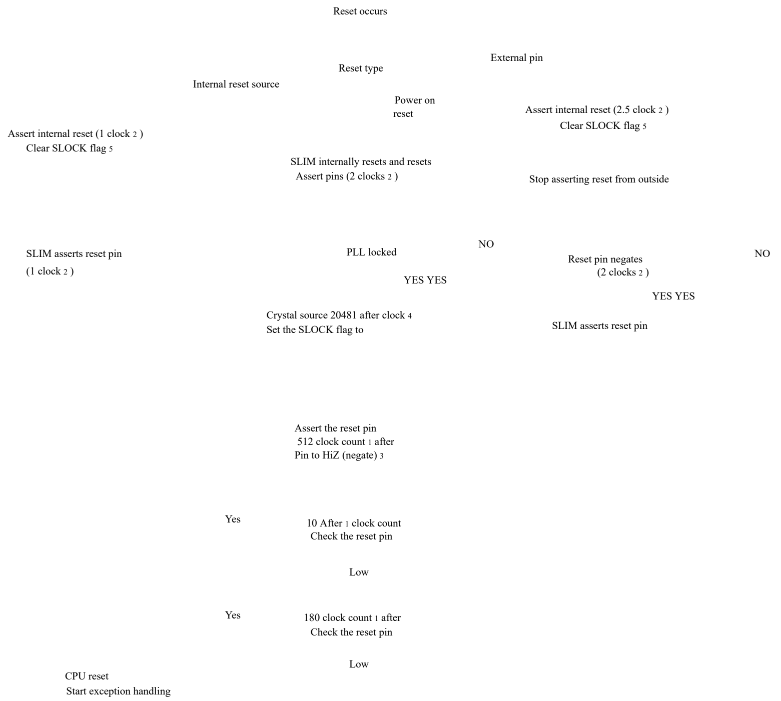
The RESET input must be asserted for the specified minimum period for the reset to occur. Ri
To prevent the light cycle from being aborted by the set, the longest bus cycle time (or
Internally delays the external RESET assertion for a period equal to the bus monitor timeout period)
You can postpone it. While RESET is asserted, the SLIM pin is inactive high in
It is in a pedance state or is driven into an inactive state.

When the external device asserts RESET for a reasonable period of time, the reset control circuit puts the signal on the internal latch.
Clock drive. The control circuit has detected that the RESET signal is no longer driven externally.
Then, for another 512 CLKOUT cycles, drive the RESET pin to “L” to do this for the entire system.
Guarantees a length reset.

If the internal source asserts the reset signal, the reset control circuit will reset for a minimum of 512 cycles.
Is asserted. Internal reset if the reset signal is still asserted at the end of 512 cycles
The control circuit continues to assert RESET until the signal is negated.

After 512 cycles, the reset input pin has a high impedance that is inactive for 10 cycles.
You will be in a dance state. At the end of this 10-cycle period, the reset input is tested. Input is a theory
When at level 1, reset exception handling begins. However, the reset input is at logic level 0
Occasionally, the reset control circuit goes into a high impedance state for an additional 180 cycles and ends.
If the reset input is logical level 1, reset exception handling will start. But reset input
If is logic level 0, the reset circuit again drives the pin to “L” for 512 cycles. child
The process of is repeated until RESET is negated.

The reset sequence flow is shown below.



Note: 1 Clock count is the number of system clocks.
2 The number of clocks shown in parentheses is the delay time.
3 If the external pull-up resistor / external capacitance of the reset pin (open-drain type I / O) is too large, SLIM will have 512 systems.
After asserting the clock, the reset pin negates (the pin is HiZ, so it is set to logic level 1 by an external pull-up resistor.

Please note that it may not start up within 190 system clocks after (trying to). At that time, the CPU is running Unable to start. Also note that when SLIM negates the reset pin, the MCU port will change to the post-set behavior. Is required. 190 The port returns to the reset state when it is determined to be low after the system clock has elapsed.

[4] This number is for first reference mode only. In slow reference mode, this is 328 clocks of crystal source vibration.

[5] The SLOCK flag is cleared on reset. For reset sources other than power-on reset CPU operation without being affected by PLL lock In order to get into the work, the SLOCK flag will be set while the CPU is running. The SLOCK flag is set after the PLL is locked. Fixed number of clocks (First reference mode: 20481 clocks of crystal source / Slow reference mode: Cli It takes 328 clocks) of Star Source Vibration. If a reset occurs during system operation, the VCO gain must be changed. This clears the SLOCK flag but does not unlock the PLL. The SLOCK flag is on page 93, “SLOCK—Synthesizer Lock. See Flags .

Figure 13 Reset control flow diagram

Page 107

3.5.4 Power on reset

Power-on reset when using the system clock with the SLIM clock synthesizer Requires a special environment for powering on the system and clock synthesizer. MCU works To do this, regardless of the clock source, the clock synthesizer power input pin V DDSYN / MODCLK pin Voltage must be applied to. If no voltage is applied to the V DDSYN / MODCLK pin, the crystal Oscillation does not occur. If V DDSYN / MODCLK is applied at power up, the startup time is Crysta It is affected by the specific parameters of the oscillator and the design of the oscillator circuit. In addition, the ramp-up time of VDD is also renewed. Affects the pin condition during setting.

During a power-on reset, SLIM's internal circuitry cuts the IMB's internal and external reset lines. Live When VDD ramps up to the specified minimum, this circuit releases the internal reset line. And the SLIM pin is initialized. When the VDD reaches the minimum value, the clock synthesizer VCO operates. It starts and the clock frequency ramps up to the limp mode frequency. Clock synthesizer The RESET signal is negated after the PLL locks and 512 system cycles have elapsed.

The SLIM clock synthesizer feeds the clock signal to other MCU modules. Clock moves These modules are reset after the internal reset signal has been asserted for 4 clock cycles. Will be The duration of these four cycles is determined by the DMA ramp time and the VCO frequency ramp time. I will. In the worst case it is about 15ms. During this time, the module port pins may be in an indefinite state. I will. The input-only pin can be put into a known state by an external pull-up resistor, but during this time, it is turned on. External logic on the output pin or the output-only pin must condition the line. Prevent conflict Therefore, the active driver requires a high impedance buffer or isolation resistor.

3.5.5 Usage of three-state control pins

When the three-state control (TSC) input is asserted at the same time as the RESET pin, the MCU outputs all outputs. Places the river in an inactive high impedance state. In order for the driver to change the state, this Signal must be asserted for 10 clock cycles. Power-up reset There are certain restrictions on how to use the TSC inside, as described below.

When using the internal clock synthesizer (V DDSYN / MODCLK is kept at “H” during reset) The synthesizer ramp-up time affects the time required for 10 clock cycles. terrible The case is about 20ms from the TSC assertion.

When an external clock signal is applied (V DDSYN / MODCLK is held at “L” during reset), the TSC answer As soon as 10 clock pulses are applied to the EXTAL pin after the operation, the pin is high impedance. It will be in a state.

When the TSC assertion is enabled, the internal signal is forced to a certain value, which happens to be a mode. May be selected. To resume normal operation, reset and restart the MCU. Must be.

3.6 Interrupt

A central processing unit, SLIM, and interrupt service are required for interrupt recognition and service. It involves complex interactions between vices or modules.

CPU16 has 7 levels of interrupts, with 7 being the highest level and 1 being the lowest level (level 0 is a percentage). Means no crowd). Interrupts are 7 automatic interrupt vectors and 200 assignable interrupt vectors. I will provide you. All interrupts with a priority lower than 7 are assigned to the condition code register. It can be masked by the included priority (IP) field. CPU16 makes interrupts a kind of asynchronous exception Treat as.

The external interrupt input has 7 pins of IRQ [7: 2] and IRQX, and if the level of IRQX is set to 1, it looks like IRQ1. It is possible to use it for. Interrupts are affected by the mask level of CPU16. If the mask of CPU16 If the bell (IP mask value) is 0, IRQ1 is the lowest priority interrupt and IRQ7 is the highest priority interrupt.

The IP mask field consists of 3 bits (CCR [7: 5]). Binary values% 000 to% 111 are eight Provides a priority mask. Depending on the mask, interrupt request with priority equal to or lower than the mask value (excluding IRQ7) Can be prevented from being recognized and processed. Interrupts are masked when the IP has% 000 I can't. During exception handling, the priority of serviced interrupts is set in the IP field.

The interrupt request signal can be asserted by an external device or microcontroller module. Need The line is internally connected by a wired NOR. Same multiple requests with different priorities

You can do it from time to time. The internal assertion of the interrupt request signal is in the logical state of the corresponding MCU pin.
Does not affect.

External interrupt requests are sent to CPU16 through the external bus interface and SLIM interrupt control circuitry.
vinegar. The CPU treats external interrupt requests as if they were sent by SLIM.

IRQ [7: 2] and IRQX are configured as active “L” level sensitive or edge detection inputs.
Can be In addition, these inputs can be configured as general purpose I / O pins or edge detection pins.
In that case, the IRQ feature (level sensitive) is not supported. Port F pin assignment register
Control bits are used to assign functions. For more information, see 3.7.5 Port F Behavior .
please refer to.

IRQ7 is a non-maskable interrupt and is available in all SLIM configurations. Also, IRQ [6: 2]
IRQX is also available in all SLIM configurations. In addition, the IRQX pin is a port F priority register (PFLVR).
It is possible to set any level with. For more information, see "3. 7.5 operation of the port F See"
Please.

The IRQ [6: 2] can be masked when configured with level sensitive. IRQX has a level setting of 6 or less
If you have one, you can mask it. IRQ7 cannot be masked. IRQ7 input is a redundant service and stack overflow
It is transition sensitive to prevent lows. Non-maskable interrupt is IRQ7
IP mask level changes from% 111 to a smaller value each time is asserted and during IRQ7 assertion
It is generated every time it is converted.

Interrupt requests are sampled at the continuous falling edges of the system clock. Interrupt request input
The force circuit has a hysteresis characteristic. The request signal must be asserted for two consecutive clock periods or more.
Will not be valid. Even if the request becomes valid, exception handling is not performed immediately, and once Pendin
It will be Requests that are pending are processed at instruction boundaries or are higher priority exceptions.
It will be processed when the exception handling for is completed.

CPU16 does not latch the priority of pending interrupt requests. Low priority request is pendy
If a high-priority interrupt source makes a service request while it is busy, the high-priority interrupt source makes a service request.
Request is serviced. If an interrupt request with a priority equal to or lower than the current IP mask value is made, the CPU
Does not recognize the request.

3.6.1 Interrupts in master mode Acknowledgment cycle NS

The interrupt acknowledge bus cycle is generated during exception handling. CPU16 interrupt priority
CPU16 addresses when it detects one or more interrupt requests that have a higher priority than the tee mask value.
Performs CPU spatial read from A [23: 0] = $ FFFFF: [IP]: 1.

CPU16 has two functions by performing a CPU spatial read cycle. As the first function
Then, the interrupt value corresponding to the highest priority interrupt request is placed on the address bus, and the next function is interrupt.
Gets the exception vector number from the source. Where CPU16 performs an interrupt acknowledge bus cycle
If so, the interrupt level at that time is set to the same value in the interrupt mask field of CCR [7: 5]. That
Therefore, even if an interrupt lower than the mask level occurs, it will not be accepted.

The module that requested the interrupt service is on the address bus at the beginning of the interrupt acknowledge cycle.
Decodes the interrupt value placed above, and if it is the interrupt level of the module itself, in that cycle
Respond. This response means that when there are multiple modules with the same interrupt level, the interrupts are tied together.
It is for mediation when it occurs in ming. As this arbitration method, the interrupt adjustment of the module
Performed by a serial conflict between the bit values of the stop (IARB) field.

Each module that can make an interrupt service request, including SLIM, is in the configuration register.
Has an IARB field. The IARB field has% 0001 (lowest priority) to% 1111 (highest)
You can assign values up to (priority). If the value of the IARB field is% 0000, that mod
CPU16 handles spurious interrupt exceptions when an interrupt from an interrupt is recognized.

SLIM's IARB is used for arbitration between internal interrupt requests and external interrupt requests because EBI manages external interrupt requests.
The value is used. The reset value of IARB for SLIM is% 1111 and for all other modules
The IARB reset value for it is% 0000. Different initialization software to set the arbitration method
You must assign an IARB value.

Each module must have a unique IARB value. Same non-zero for multiple IARB fields
When the same value and the same interrupt level, CPU16 reads the vector number from multiple modules at the same time.
The result is unpredictable.

Mediation must always be done, even if only one source is requesting service. this
Is important for two reasons. First, unless SLIM survives the competition, the CPU interrupt acknowledge rhino

Kuru is not driven to an external bus. If there is no conflict, an interrupt will occur due to a bus error.

The Knoridge bus cycle ends and a spurious interrupt exception is thrown.

When the arbitration is completed, the priority module sends the interrupt vector number from the data bus to the CPU, which is the bus support. I have to finish the ikuru. For external interrupt requests, the interrupt acknowledge cycle is out Since it is transferred to the local bus, the external device decodes the interrupt value, returns the vector number, and bus-said. The end signal must be generated. Spurious discount if the device does not respond in time A crowded exception is thrown.

3.6.2 Summary of interrupt processing

The following is a summary of the interrupt processing sequence. When this sequence is started, a valid interrupt server The screw request is detected and processing continues.

A. The CPU either finishes high priority exception handling or reaches the instruction boundary.

B. The processor state is stacked and the PK extended fields in the CCR are cleared.

C. The interrupt acknowledge cycle begins as follows:

1. FC [2: 0] is driven to% 111 (CPU space) encoding.
2. The address bus is driven as follows: ADDR [23:20] =% 1111; ADDR [19:16] = % 1111 indicates that the cycle is an interrupt acknowledge CPU spatial cycle. ADDR [15: 4] =% 11111111111; ADDR [3: 1] = Recognized interrupt request priority. ADDR0 =% 1.

   Note: In MC68HC (9) 16Y5 / 916Y6, ADDR [22:19] is not connected to the external pin, but The input module bus (IMB) is driven in this way.
3. The request priority is latched from the address bus to the IP field of the CCR.

D. Modules or external peripherals requesting interrupt service have ADDR [3: 1] priority values. Decode. If the priority of the request is the same as the priority value of this address, an IARB conflict will occur. vinegar. If there is no conflict, the spurious interrupt monitor asserts BERR and handles the spurious interrupt exception. It will be done.

E. After arbitration, the interrupt acknowledge cycle can be terminated by any of the following three methods: increase.

1. The main interrupt source supplies the vector number and DTACK signal according to the access. CPU16 is that Get the Kuta number.
2. The AVEC signal is asserted by the function of CSA, and CPU16 is an auto vector corresponding to the interrupt priority. Generate a data number.
3. The bus monitor asserts BERR and CPU16 generates the spurious interrupt vector number.

F. Vector numbers are converted to vector addresses.

G. The contents of the vector address are loaded on the PC and the exception handling routine is executed.

3.7 General-purpose input / output

Most of the pins associated with SLIM can be used for several functions. These main functions are off-chip Providing an external bus interface for applications that require access to resources is. These pins can be used as digital I / O pins when not used as a primary function. vinegar. SLIM pins are grouped into 8-bit ports for easy I / O functionality. each The port has associated registers that are used to configure the pins to the desired function.

SLIM contains eight general purpose I / O ports, A, B, C, D, E, F, G, and H. Enter both Can be used as output. All ports are used to monitor or control the state of the pins There are two related data registers.

The port output data register is readable and writable. Write to port output data register When plugged in, the data direction register is driven to the pad of the pin if it is set to output. Output It is possible to read the output data register regardless of the actual state of the data.

The pin data register is all about whether the pin consists of an input or an output. Returns the current state of all pins. The reset clears the data direction register and all pins Is the input, so after a reset all pin data registers reflect the actual state of the pin. increase. Writing to this register has no effect.

Ports C, D, E, and F have pin assignments that indicate whether individual pins are digital I / O pins or other functions. There is a gista. Ports A, B, and G do not have pin assignment registers and are SLIM configured at reset. It is configured as a digital I / O, depending on the mode of. Port H does not have a pinout register

Page 110

Program to SLIM mode and chip select (CSA, CSB, CSC) data port size bits It is configured as a digital I / O based on the rammed value.

Every port has an associated data direction register that outputs each port or an individual pin on the port. Or used to configure as input. In addition to the output data and pin data registers, the port F has an edge detection flag register that indicates whether a transition has occurred on any of the pins.

Table 32 lists the key features of a generic I / O port and the modes in which it can be used.

Table 32 Port features and modes

Port[1]	Modes that can be used as ports	Main function
A	Single chip	AD [15: 8]
B	Single chip	AD [7: 0]

C	Single chip, non-multiplexed master, multiplexed master	A23 / ECLK, A [18:16]
D	Single chip, non-multiplexed master, multiplexed master	Chip Select, Debug
E	Single chip, non-multiplexed master, multiplexed master	Bus Control (excluding FC [2])
F	Single chip, non-multiplexed master, 16-bit server Non-multiplexing master of width, multiplexing master	IRQ [7: 2], IRQX / BERR, ALE
G	Single chip, multiplexing master	D [15: 8]
H	Single chip, 8-bit bus wide non-multiplexed machine Star, multiplexing master	D [7: 0]

3.7.1 Behavior of ports A and B

Ports A and B can be configured with the entire port as an input or an output. At reset
Based on the mode of the SLIM that is configured, port A and port B will be digitized during reset negate.
Can be set to input or address bus pin.

PORTA—Port A output data register $ YFFA10

PORTB—Port B output data register $ YFFA11

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

Port A and Port B output data registers are configured with ports A and B as outputs.
If so, it is used to latch the data to drive to the corresponding port output pin. port
Reading the A and B output data registers is always in the data latch state. This Les
Gista is invalid for RESET (a reset source other than POW), but in POW it sets a value of "$ FFFF".
vinegar. See “3.5 Resets ” on page 100 for the types of resets .

PORTAP—Port A pin data register $ YFFA12

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PAP7	PAP6	PAP5	PAP4	PAP3	PAP2	PAP1	PAP0	PORTBP							
RESET:															
Current state of the corresponding pin															

PORTBP—Port B pin data register $ YFFA13

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PORTAP								PAB7	PAB6	PAB5	PAB4	PAB3	PAB2	PAB1	PAB0
RESET:															
								Current state of the corresponding pin							

Reading pin data registers A and B puts them in the corresponding pin state.

Writing to PORTAP and PORTBP has no effect.

DDRAB—Port A / B Data Direction Register $ YFFA15

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED														DDA	DDB
RESET (single chip mode):															
U	U	U	U	U	U	U	U	0	0	0	0	0	0	0	0

Port A and B data direction registers are respectively when the pin is configured as an I / O.
Controls the direction of the pin driver for ports A and B. When DDA or DDB is set to 1,
All pins on the corresponding ports are configured as outputs. Also, DDA or DDB is cleared to zero
If so, all pins on the corresponding port will be configured as inputs.

3.7.2 Operation of port C

The port C pin can be used as a digital I / O or address signal. PC7 has a clock output pin
You can also configure it. A23, A [18:16] are used to form the address space. Reset
According to the SLIM mode configured at time, the pin on port C will function normally when the reset is released.
Or you can set it to discrete input.

PORTC—Port C output data register $ YFFA18

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PC7	NOT USED	NOT USED	NOT USED	NOT USED	PC2	PC1	PC0	PORTD							

RESET:

U	U	U	U	U	U	U	U

The port C output data register latches the data to drive to the port C output pin. Reading PORTC is always in the data latch state. This register is RESET (a register other than POW) It is invalid for the set source), but in POW, it sets the value of "$ FF". About the type of reset See “3.5 Reset ” on page 100 .

PORTCP—Port C pin data register $ YFFA1A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PCP7	NOT USED	NOT USED	NOT USED	NOT USED	PCP2	PCP1	PCP0	PORTDP							

RESET:

Current state of the corresponding pin	

Reading the port C pin data register puts it in the port C pin state. Write line There is no effect even if it becomes.

DDRC — Port C data direction register $ YFFA1C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
DDC7	NOT USED	NOT USED	NOT USED	NOT USED	DDC2	DDC1	DDC0	DDRD							

RESET:

0	0	0	0	0	0	0	0

Bits in the port C data direction register are pinned when the pin is configured as an I / O pin. Controls the direction of the driver. When any bit is set to 1, the corresponding pin can be output Will be made. When any bit is cleared to zero, the corresponding pin is configured as an input. It will be. A master reset clears all bits to zero. Affects system reset It will not be.

PCPAR—Port C pin assignment register $ YFFA1E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PCPAR [15:14]		0	0	0	PCPAR [10: 8]			PDPAR							

RESET (DSI = 1, Single-chip configuration):

0	PCON14 0				0	0	0

RESET (DSI = 0, Single-chip configuration):

0	AD4 Pin state				0	0	0

RESET (DSI = 1, Master configuration) [1] :

PCON15 0	PCON14 0				PCON13 0	PCON12 1	PCON11 1

RESET (DSI = 0, Master configuration):

AD5 Pin state	AD4 Pin state				AD3 Pin state	AD2 Pin state	AD1 Pin state

[1] MC68HC (9) 16Y5 / 916Y6 is set to single chip mode when DSI = 1. Therefore, this condition does not exist.

This port C pin assignment register controls the function of all pins on port C. PCPAR [15:14] PC7 is set as an address pin (A23), clock output (ECLK), or digital I / O pin. You can choose. See Table 33 .

Table 33 PC7 Pin Allocation Encoding

PCPAR [15:14]	PC7 pin function
0 0	Digital I / O (PC7)
0 1	ECLK
1 X	A23

Reading the PCPAR puts it in the current configuration of the port pin. PCPAR [13:11] cannot be written and always returns zero when read. In extended mode, the pinout register is the port / clock configuration shadow. C. Resets to the default reset state programmed in the register (PCON [15:11]). PC [6: 0], you can configure the address pins (A [22:16]) or as a digital I / O . table

See 34 . A series of values stored in PCPAR [10: 8] that can be configured as an address line
Select the number of pins. The remaining pins are configured as digital I / O pins. Selected address
Lines are selected from the lowest pin (A16) and I / O pins are selected from the highest pin (PC6) in succession.

However, in this MC68HC (9) 16Y5 / 916Y6, A [22:19] / PC [6: 3] is assigned as an external pin.
plug.

Table 34 Port C pin assignments

PCPAR [10: 8]	Pin function	
	Port CI / O pin	Address pin
0 0 0	PC [6: 0]	none
0 0 1	PC [6: 1]	A16
0 1 0	PC [6: 2]	A [17:16]
0 1 1	PC [6: 3]	A [18:16]
1 0 0	PC [6: 4]	A [19:16]
1 0 1	PC [6: 5]	A [20:16]
1 1 0	PC6	A [21:16]
1 1 1	none	A [22:16]

Page 113

3.7.3 Operation of port D

Port D pins can be digital I / O or external bus control / handshake and debug signals
Can be selected individually. Pins on port D are reset according to the SLIM mode configured at reset.
When released, it can be set to normal function or discrete input.

PORTD—Port D output data register $ YFFA19

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	PORTC								PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
RESET:																
									U	U	U	U	U	U	U	1

The port D output data register latches the data to drive to the port D output pin.
Reading PORTD is always in the data latch state. This register is RESET (a register other than POW)
Only PD0 is set in the set source), and the others are invalid. In POW, set the value of "$ FF".
See “3.5 Resets ” on page 100 for the types of resets .

PORTDP—Port D pin data register $ YFFA1B

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	PORTC								PDP7	PDP6	PDP5	PDP4	PDP3	PDP2	PDP1	PDP0
RESET:																
									Current state of the corresponding pin							

Reading the port D pin data register puts it in the port D pin state. Write line
There is no effect even if it becomes.

DDRD—Port D data direction register $ YFFA1D

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	DDRC								DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET:																
									0	0	0	0	0	0	0	1

The bits in the port D data direction register are pinned when the pin is configured as an I / O pin.
Controls the direction of the driver. When any bit is set to 1, the corresponding pin can be output
Will be made. When any bit is cleared to zero, the corresponding pin is configured as an input.
It will be. Reset clears the non-DDO bits to zero.

PDPAR—Port D pin assignment register $ YFFA1F

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	PCPAR								PDPA7	PDPA6	PDPA5	PDPA4	PDPA3	PDPA2	PDPA1	PDPA0
RESET (Single-chip configuration):																
									0	0	0	0	BKPT	BKPT	BKPT	BKPT
RESET (Master configuration):																
									PCON10 0	PCON8 0	PCON6 0	PCON4 0	BKPT	BKPT	BKPT	BKPT

The port D pin assignment register bit controls the function of each pin on port D. See Table 34
please. If the control bit is 1, the corresponding pin is used for normal function. This bit is

In the case of (b), the corresponding pin is configured as a digital I / O pin. Configured as an I / O pin
Pins are controlled by the corresponding bits in the port D data register and the data direction register.
increase.

SLIM mode and port / clock configuration shadow configured on reset
The bit state of the register is used to configure PDPAR [7: 4] when resetting in master mode.
It will be. The shadow register configuration can be changed by the external pin at reset.
PDPAR [3: 0] is always reset to the inverted state of the BKPT pin.

Table 35 Port D pin assignment

PDPAR bit	Pin function	
	Port D signal (= 0)	Bus control signal (= 1)
PDPA7	PD7	CSA
PDPA6	PD6	CSB
PDPA5	PD5	CSC
PDPA4	PD4	DTACK
PDPA3	PD3	FREEZE
PDPA2	PD2	BKPT
PDPA1	PD1	IPIPE1
PDPA0	PD0	IPIPE0

3.7.4 Port E behavior

Port E has pins that are used as bus control signals or digital I / O. At reset
According to the configured SLIM mode, the pin on port E goes up to the bus control signal when the reset is released.
Or it can be configured for discrete input.

PORTE—Port E output data register $ YFFA21

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								NOT USED	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:															
								U	U	U	U	U	U	U	U

The port E output data register latches the data that drives the port E pin. This Les
When you read the gista, it is always in the data latch state. It is not affected by the reset. this
The register is invalid for RESET (a reset source other than POW), but in POW it sets the value of "$ FF".
vinegar. See “3.5 Resets ” on page 100 for the types of resets .

PORTEP—Port E pin data register $ YFFA23

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								NOT USED	PEP6	PEP5	PEP4	PEP3	PEP2	PEP1	PEP0
RESET:															
									Current state of the corresponding pin						

Reading the port E-pin data register puts it in the port E-pin state. Write line
There is no effect even if it becomes.

DDRE—Port E data direction register $ YFFA25

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								NOT USED	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
RESET:															
								0	0	0	0	0	0	0	0

The bit of the port E data direction register is the port E pin when the pin is configured as an I / O.
Controls the direction of the driver. If you set any bit in this register to 1, the corresponding pin
Is configured as an output. Clearing any bit of this register to zero will result in the corresponding pin.
Is configured as an input. A master reset clears all bits to zero. System
It is not affected by reset.

PEPAR—Port E pin assignment register $ YFFA27

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								NOT USED	PEPA6	PEPA5	PEPA4	PEPA3	PEPA2	PEPA1	PEPA0
RESET (Single-chip configuration):									0	0	0	PCON10	0	0	0	0
RESET (Master configuration):									PCON30	PCON30	PCON30	PCON10	1	1	1	1

The bits in the port E pin assignment register control the function of each pin on port E. See Table 36 please. If any bit of this register is set to 1, the corresponding pin will be configured in the main function. It will be. If any bit of this register is cleared to zero, the corresponding pin will be on the port E output day. It is configured as an I / O pin controlled by data and data direction registers.

Table 36 Port E pin assignments

PEPAR bit	Pin function	
	Port E signal (= 0)	Bus control signal (= 1), which is the main function
PEPA6	PE6	FC1
PEPA5	PE5	FC0
PEPA4	PE4	CLKOUT
PEPA3	PE3	SIZE
PEPA2	PE2	AS
PEPA1	PE1	DS
PEPA0	PE0	R / W

3.7.5 Port F behavior

Port F has an interrupt pin (IRQ [7: 2], IRQX) and an address latch enable pin (ALE). I will. The function of the pin of port F is that the IRQ [7: 2] is level-sensitive interrupt request input, edge sensitivity. Either the civic interrupt request input, the rising edge detection I / O pin or the falling edge detection I / O pin It can be programmed to work. PF1 is a bus error signal (BERR), level sensitive, or Or edge-sensitive interrupt request input, priority 1-7, rising edge detection I / O pin, Or it can be programmed to act as one of the falling edge detection I / O pins. PF0 is an address With a slatch enable signal (ALE), rising edge detection I / O pin or falling edge detection I / O pin Can be programmed to work. Table 37 summarizes the behavior of port F.

Table 37 Port F behavior

pin	Pin function
IRQ [7: 2]	Level detection interrupt input
	Edge detection interrupt input
	General-purpose I / O port (with edge detection interrupt function)
IRQX	Level detection interrupt input
	Edge detection interrupt input
	General-purpose I / O port (with edge detection interrupt function)
ALE	General-purpose I / O port (with edge detection interrupt function)
	Multiplex bus control signal

PORTF—Port F output data register $ YFFA31

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:															
								1	1	1	1	1	1	1	1

The port F output data register latches the data driven to the port F output pin. Reading this register always prints the current state of the data latch. By reset All latches are set to “1”.

PORTFP—Port F pin data register $ YFFA33

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								PFP7	PFP6	PFP5	PFP4	PFP3	PFP2	PFP1	PFP0
RESET:															
								Current state of the corresponding pin							

Reading the port F pin data register prints the current state of the port F pin. Writing does not affect it.

DDRF port F data direction register $ YFFA35

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:															
								0	0	0	0	0	0	0	0

The bits in the port F data direction register are port F when the pin is configured as an I / O. Controls the direction of the pin driver. Setting any bit in this register to 1 corresponds Pins are configured as outputs. Clearing any bit of this register to zero will result in the corresponding pin. Is configured as an input. This register can be read or written at any time. reset Clears the bit to zero.

PFPAR—Port F pin allocation register $ YFFA36

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PFPA7		PFPA6		PFPA5		PFPA4		PFPA3		PFPA2		PFPA1[1]		PFPA0	
RESET															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	*	*

*: When the multiplexing master mode SLIM configuration is selected, "0", that In other cases, "1".

Note: 1. RQX / BERR / PF1 pin changed from level / edge detection interrupt mode to general purpose port If low is input at the time of operation, the interrupt level will be kept internally. child To avoid holding the port, change this pin to a BERR pin and then change the port function. please do it.

Each of the ei[redacted] register controls the function of the pins on port F. About the function of each pin See Table 38 and ore information.

Table 38 Port F pin assignments

Pin name	BITS 1 in the PFPAR field	Pin assignment		Impact of DDRF
		Signal name	function	
	PFPA [7: 2]			
IRQ [7: 2] / PF [7: 2]	0 0	IRQ [7: 2]	interrupt	DDRF [7: 2] disabled (fixed input)
	0 1	PF [7: 2]	General purpose port	DDRF [7: 2] valid
	Ten	PF [7: 2]	General purpose port	DDRF [7: 2] valid
	1 1	IRQ [7: 2]	interrupt	DDRF [7: 2] disabled (fixed input)

Table 38 Port F pin assignments (continued)

Pin name	BITS 1 in the PFPAR field		Pin assignment		Impact of DDRF
			Signal name	function	
	PFPA1	IRQXL			
IRQX / BERR / PF1[2]	0 0	1-7	IRQX	interrupt	DDRF1 disabled (fixed input)
	0 1		PF1	General purpose port	DDRF1 enabled
	Ten		PF1	General purpose port	DDRF1 enabled
	1 1		IRQX	interrupt	DDRF1 disabled (fixed input)
	XX	0	BERR	Control signal	DDRF1 disabled (fixed input)
	PFPA0				

ALE / PF0	0 0	ALE 3	Control signal	DDRF0 disabled (fixed input)
	0 1	PF0	General purpose port	DDRF0 enabled
	Ten	PF0	General purpose port	DDRF0 enabled
	1 1	PF0	General purpose port	DDRF0 enabled

Note: 1. The contents of this field are listed in Table 39, "Port F Pin Allocation Register Fields."
See Encoding .
2. The input state is low when changing this pin from interrupt (IRQX) to port (PF1).
Then, the interrupt will be held internally. To avoid retention, once
Change to BERR and then change to port.
3. Valid only in multiplex master mode

Table 39 Port F Pin Allocation Register Field Encoding

PFPAR feel BITS		function	Explanation	Vector source
MSB	LSB			
0	0	Edge detection interrupt input	SLIM interrupts when it detects a falling edge from the outside Continues to assert internal interrupts to the CPU while holding increase. When an IACK cycle occurs from the CPU, SLIM Clears the held interrupt and negates the internal interrupt signal. To At least to detect the falling edge of an interrupt Requires low for 2 system clocks.	From the data bus Give or With auto vector handle.
0	1	General purpose I / O pin (With rising edge detection interrupt function)	General purpose I / O port. This port is an edge detection interrupter Has the ability to interrupt the detection to the CPU It is possible to.	Supports from PFIVR NS.
1	0	General purpose I / O pin (With falling edge detection interrupt function)	At least 2 system clocks are required to detect edges It is important.	
1	1	Level detection interrupt input	The assertion of the interrupt from the outside (low Akti Lee blanking) is If so, SLIM does not hold the interrupt signal and is an internal interrupt. To the CPU. Therefore, interruption from the outside Only assert is external until the IACK cycle occurs You need to keep pulling. If the IACK cycle is If you cancel the interrupt from the outside before it occurs, the sp Please note that it will be a rear interrupt.	From the data bus Give or sometimes With auto vector handle.

PORTFE—Port F edge detection flag register $ YFFA39

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					PFE7	PFE6	PFE5	PFE4	PFE3	PFE2	PFE1	PFE0
RESET:															
								0	0	0	0	0	0	0	0

The port F edge detection flag register causes the appropriate transition on the port F input or output pin.
Indicates that. The corresponding edge detection flag is the input or output edge of the corresponding port F pin.
Set when edge detected when set as a detection pin. Pin is level sen
If set as civic (or BERR, ALE), the flag for that pin, even if there is a transition

Is not set.

PFEER—Port F Edge Detection I / O Interrupt Enable Register $ YFFA3B

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					PFEE7	PFEE6	PFEE5	PFEE4	PFEE3	PFEE2	PFEE1	PFEE0
	RESET:														
								0	0	0	0	0	0	0	0

Port F edge-detection I / O interrupt enable register bits are separate edge-detection I / O pins.
Used to enable interrupts from. If a bit is 1, specify it as the corresponding pin
When an edge is generated, it is specified in the PFEL [2: 0] field of the port F interrupt level register.
An interrupt request is generated according to the interrupt priority. If the bit is zero, the pair is paired even if an edge is detected.
The corresponding edge detection I / O pin does not generate an interrupt request. This register is set by master reset.
Resets to b and disables edge detection I / O interrupts.

PFLVR—Port F interrupt level register $ YFFA3C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	IRQXL [2: 0]			0	PFEL [2: 0]						PFIVR				
RESET:															
0	0	0	1	0	0	0	0								

The port F interrupt level register contains a 3-bit field that controls IRQX and a port F array.
There is a 3-bit field that determines the priority of the detected interrupt. Read PFLVR7 and PFLVR3

Will always return zero, and writing will have no effect.

IRQXL [2: 0] —IRQX configuration level [2: 0]

This field controls whether the pin is IRQX or BERR.

000 = Functions as BERR.

XXX = Level sensitivity corresponding to an interrupt of priority XXX (XXX is non-zero) when asserted
Serves as a tib or edge-sensitive interrupt input. IRQX interrupt by reset
Only level is set to XXX = 001 as a level 1 interrupt.

PFEL [2: 0] — Port F Edge Detection I / O Interrupt Level

This field specifies the priority of port F edge detection I / O interrupts 0-7. To reset
The value 000 is set to indicate that PFEL [2: 0] is disabling interrupts. SLIM?
Port F edge detection interrupts when all of these interrupt sources are competing for the same priority
Has the lowest arbitration priority in SLIM.

PFIVR—Port F Edge Detection I / O Interrupt Vector Register $ YFFA3D

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PFLVR								PFIVR							
RESET:															
								0	0	0	0	1	1	1	1

The port F edge detection I / O interrupt vector register is generated by the port F edge detection circuit.
Determines which vector in the exception vector table to use for interrupts. Edge interrupt
The logic uses bit [7: 0] of this register as an interrupt vector.

3.7.6 Operation of ports G and H

Ports G and H are pins used as data buses or general purpose I / O. Configured at reset
Configure to data bus or generic input when reset is released according to the SLIM mode
Can be

PORTG—Port G output data register $ YFFA28

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTH							
RESET:															
U	U	U	U	U	U	U	U								

The port G output data register latches the data that drives the port G pin. This Les
When you read the gista, it is always in the data latch state. It is not affected by the reset. this
The register is invalid for RESET (a reset source other than POW), but in POW it sets the value of "$ FF".
vinegar. See “3.5 Resets ” on page 100 for the types of resets .

PORTGP—Port G pin data register $ YFFA2A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PGP7	PGP6	PGP5	PGP4	PGP3	PGP2	PGP1	PGP0	PORTHP							
RESET:															
Current state of the corresponding pin															

Reading the port G-pin data register puts it in the port G-pin state. Write line
There is no effect even if it becomes.

DDRG—Port G data direction register $ YFFA2C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRH							
RESET:															
0	0	0	0	0	0	0	0								

The bit of the port G data direction register is the port G pin when the pin is configured as an I / O.
Controls the direction of the driver. If you set any bit in this register to 1, the corresponding pin
Is configured as an output. Clearing any bit of this register to zero will result in the corresponding pin.
Is configured as an input. Reset clears all bits to zero.

PORTH—Port H output data register $ YFFA29

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
PORTG								PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
RESET:															

U U U U U U U U

The port H output data register latches the data that drives the port H pin. This Les When you read the gista, it is always in the data latch state. It is not affected by the reset. this The register is invalid for RESET (a reset source other than POW), but in POW it sets the value of "$ FF". vinegar. See “3.5 Resets ” on page 100 for the types of resets .

PORTHP—Port H pin data register $ YFFA2B

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
				PORTGP					PHP7	PHP6	PHP5	PHP4	PHP3	PHP2	PHP1	PHP0
RESET:																
												Current state of the corresponding pin				

Reading the port H pin data register puts it in the port H pin state. Write line There is no effect even if it becomes.

DDRH—Port H data direction register $ YFFA2D

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
				DDRG					DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	PE0
RESET:																
									0	0	0	0	0	0	0	0

The bit of the port H data direction register is the port H pin when the pin is configured as an I / O. Controls the direction of the driver. If you set any bit in this register to 1, the corresponding pin Is configured as an output. Clearing any bit of this register to zero will result in the corresponding pin. Is configured as an input Reset clears all bits to zero.

3.8 chip select

Standard microcontrollers use external chip select and address decode signals. Additional hardware is required to supply. The MCU has three individually programmable tips Select (CSA, CSB, CSC pins) is available. These chip select are with external memory Provides the peripheral device with bus timing and bus cycle termination.

Note: The MCU implements the optional DTACK pin in master mode (external bus).
If not included in the mentation, use chip select and external cycle
Must be programmed to exit (corresponding chip select pi)
Whether it is used, used, or unused).

Base address registers (CSBRA, CSBRB, CSBRC) and options for each chip select individually There are registers (CSORA, CSORB, CSORC). These registers have specific chip select Contains the programmable characteristics of. Figure 13 shows the programmable chip select circuit. Shows the lock diagram.

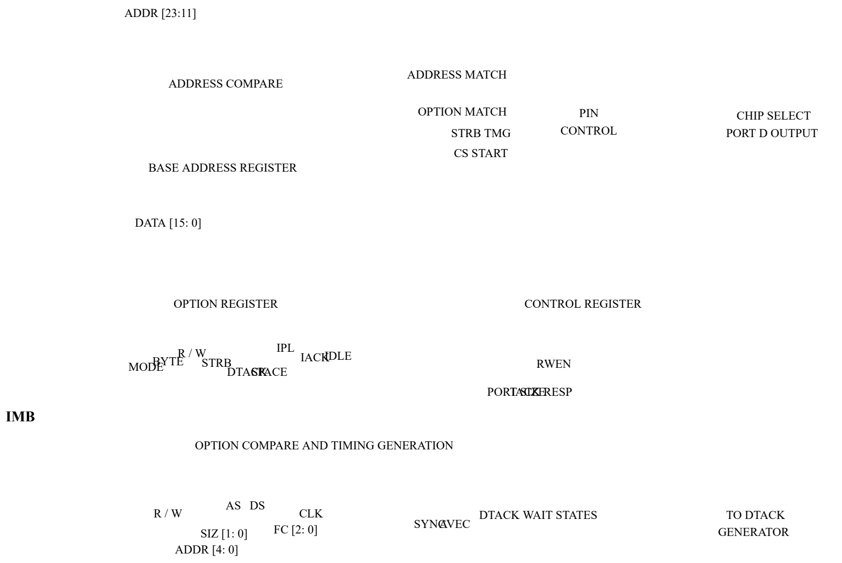


Figure 14 Chip select block diagram

3.8.1 Programmable chip select circuit

Chip select asserts can be output enable, read / write strobe, or interrupt a.
It can be synchronized with the bus control signal to provide the knoverage signal. The logic circuit is inside
It is also possible for the unit to generate a DTACK signal. One DTACK generator is shared by all circuits. Same a
Wait state if multiple chip select is assigned to the dress and control circuit

Must be the same number.
You can select an address block size of 2K to 1M bytes.
When memory access occurs, the chip select circuit will use the address space type, address, and access.
Seth type, transfer size, and interrupt priority (for interrupt acknowledge), chip selection
Compares with the parameters set in the register. If all the parameters match, click
Assert the select signal. The select signal is active "L". CSB is reset
It can also be used to select the initial boot ROM that contains the data and the initial program.
Same add to the chip select function as a microcontroller module or internal memory array
If less is given, access to that address is an internal module or internal memory array.
The chip select signal is not externally asserted.
Each chip select pin has multiple functions. Chip select structure at reset negate
The result is determined by the reset mode setting. If CSB is enabled in the mode setting
The boot ROM select signal (CSB) is automatically asserted during reset negate. Single click
In mode, all chip select pins are configured as port D [7: 5]. Data bar
The size is controlled by the port size programmed in CSA, CSB and CSC. CSA,
If CSB and CSC are all programmed to 8-bit size, D [7: 0] is the I / O port (port H).
And D [15: 8] is used as the 8-bit data bus. Chip select is 16-bit po
When configured with size, D [15: 0] is configured as a 16-bit data bus.

Note: In master mode, the bus size can be set to 8-bit or 16-bit.
At this time, the pin set on the bus cannot be used as a general-purpose port.

Table 40 shows "Chip Select and Discrete Output Allocation" .

Table 40 Chip Select and Discrete Output Allocation

pin	Pin function	
	Chip select	Discrete output
CSA	CSA	PD7
CSB	CSB	PD6
CSC	CSC	PD5

3.8.2 Chip select configuration

The mode, pin function, and port size are set to the register bits described in "3.7 General Purpose I / O".
Based on. In single chip mode, the chip select is configured as PD [7: 6]. trout
In data mode, the data bus size is the port size programmed into CSA, CSB and CSC.
Is controlled by.

Note: In master mode, with or without chip select pins and whether they are in use.
Program the chip select port size to determine the size of the data bus
Must be determined.

The three independent bits of the chip select control register (CSCR) are each chip select.
Determines the port size of the port as 8-bit or 16-bit. Port D pin assignment register
Use each chip select as a digital I / O or chip select with 3 bits of (PDPAR)
It is decided to do (see Table 34). The state of these bits at reset is the port configuration.
Determined by the mask-programmed shadow bit corresponding to the ration register (PCON)
Will be done. Deformation of some bits in the shadow register depending on the state of a particular pin at reset
You can disable the reset value (see Table 30).

3.8.3 Chip select pin assignment

The bits in the port D pin assignment register (PDPAR) control the function of the chip select pins.
Bits in the chip select control register (CSCR) make them chip select.
Determines the port size when used. The reset state of these bits is the port confit.
Determined by the mask-programmed shadow bits of the regulation register (PCON)

increase. These pins are assigned as chip select or port D I / O according to Table 41 .

Table 41 Chip Select Reset Allocation

bit	State	register	pin	Default function
PCON10	1 0	PDPAR7	CSA / PD7	CSA PD7

PCON9	1
	0
		CSCR3	————	CSA = 16-bit port
				CSA = 8-bit port

PCON8	1 0	PDPAR6	CSB / PD6	CSB PD6

PCON7	1
	0
		CSCR2	————	CSB = 16-bit port
				CSB = 8-bit port

PCON6	1 0	PDPAR5	CSC / PD5	CSC PD5

PCON5	1
	0
		CSCR1	————	CSB = 16-bit port
				CSB = 8-bit port

3.8.4 Chip select control register

The chip select control register contains the access support performed by each chip select. Bits that control is, IACK when chip select is not programmed to match the cycle Reply to cycle, and R / W and DS pins to "lead strobe" and "write strobe" You have the option to reprogram to be.

CSCR—Chip Select Control Register $ YFFA6C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			IPL [7: 1]				IACK	MUXA	MUXB MUXC ADRDIS			SZA	SZB	SZC	RWEN
RESET: (DSI = 1)															
0	0	0	0	0	0	0	0	0	0	0	0	PCON9	PCON7	PCON5	0 *
												(1)	(1)	(1)	
RESET: (DSI = 0)															
0	0	0	0	0	0	0	0	0	0	0	0	PCON9	AD6	PCON5	0 *
												(1) Pin state		(1)	

*: The reset state is mask programmable.

IPL [7: 1] — Interrupt priority level

This field responds to the IACK cycle at the interrupt level supported by the chip select logic. Controls whether to answer. Any number of these bits can be set from 1 to 7. child If none of these are set, the chip select logic responds to the IACK cycle. I will not. If one or more of these are set, the chip select is the IACK bit, It responds according to the options programmed in CSCR8, CSA.

IACK— Interrupt acknowledge response

This bit, along with CSORA0, determines the response of the chip select logic to the IACK cycle. Set. Table 42 summarizes the responses by the values programmed into these two bits.

Table 42 IACK cycle response to external IRQ input assertions

IACK cycle state				Answer			
IRQ pin Assert Be done	To the IRQ level handle IPL bit state	CSCR8 IACK response field	CSORA0 IACK / CSA field	cycle Termination	AS	CSA	Vector source
N	————	————	————	BIU IBERR	1	1	None None
Y	0	0	0	None / Bus Monitor / IBERR	1	1	None None

Table 42 IACK Cycle Responses to External IRQ Input Assertions (continued)

IACK cycle state					Answer		
IRQ pin Assert Be done	To the IRQ level handle IPL bit state	CSCR8 IACK response field	CSORA0 IACK / CSA field	cycle Termination	AS	CSA	Vector source
Y	1	0	0	Internal AVEC	1	1	Corresponding Autovector
Y	————	0	1	External DTACK	0	1	External
Y	————	1	0	Internal DTACK [1]	0	1	External
Y	————	1	1	Internal DTACK [1]	0	0	External

Note: 1. You can also exit with the DTACK pin (if available).

MUX [A: C] — Multiplexed bus priority for non-multiplexed buses

These three independent bits allow a multiplexed bus cycle (cycle) on a non-multiplexed external bus. It can be generated (in units). Chip select options and base address address If a match occurs in the Gista pair and the MUX bit of the corresponding chip select is asserted Access to that memory block is the MCU's external bus implementation (multiplexing or It runs using the multiplexed address and data, regardless of (non-multiplexed).

Note: For MCUs with a multiplexed external bus, these bits are invalid and can be set. You can clear it.

ADRDIS—Address Bus Disable

This bit disables the external address bus during internal access. This bit is set When the show cycle function is used, the external address is used in the internal cycle. The bus will not be driven.

Note: When this feature is enabled, 1 way for all internal and external cycles. It state is added.

0 = Disables the external address bus during internal access.
1 = Enables the external address bus during internal access.

SZ [A: C] — Chip Select Port Size

These bits determine the port size of each chip select as 8 bits or 16 bits. Set. The state of these bits at reset is the port configuration register. Determined by the corresponding mask-programmed shadow bit in (PCON).

0 = 8-bit port
1 = 16-bit port

RWEN — Read and Write Strobe Enable

0 = R / W and DS pins are the original functions.
1 = R / W becomes WR and DS becomes RD.

3.8.5 Chip Select Base Register

The base address is the starting address of the block enabled in chip select. The block size determines the range of address space from the base address. Each chip selection The channel has an associated base address register and is efficient for each application. You can build a good address map.

CSBARA—Chip Select Base Register A $ YFFA60

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR twenty three *	ADDR twenty two *	ADDR twenty one *	ADDR 20 *	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11 11		BLKSZ	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSBARB—Chip Select Base Register B $ YFFA64

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR		BLKSZ	

twenty three*	twenty two*	twenty one*	20 *	19 19	18 18	17 17	16 16	15	14	13	12	11 11			
RESET: (MOTO = 1)															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
RESET: (MOTO = 0)															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: The state of MOTO is "1" for MC68HC (9) 16Y5 / 916Y6.

CSBARC—Chip Select Base Register C $ YFFA68

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR twenty three*	ADDR twenty two*	ADDR twenty one*	ADDR 20 *	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11 11		BLKSZ	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* In MC68HC (9) 16Y5 / 916Y6, ADDR [23:20] follows the state of ADDR19. ADDR [23:20]
If does not match ADDR19, the chip select will not be activated.

CSBAR [15: 3] — Base Address Field

This field sets the starting address of a particular address space. Address comparison logic Compares the addresses in the block using only the most significant bit. The base address value is Must be a multiple of the lock size. In the base address register diagram, the base register Shows the correspondence between a bit and an address line.

Note: For CPU16, ADDR [23:20] = ADDR19, so the maximum block size is 512K. It's a byte. Therefore, you cannot access addresses from $ 080000 to $ F7FFFF. The base of the block can be placed above this dead zone, but the impact of ADDR19 Must be considered.

BLKSZ— Block size field

The block size is determined by bits [2: 0] in each base register. This field Use to determine the size of the address space enabled by chip select. table See 43 .

Table 43 Block size field encoding

bit	Block size (bytes)	Address line to be compared
000	2K	ADDR [23:11]
001	8K	ADDR [23:13]
010	16K	ADDR [23:14]
011	64K	ADDR [23:16]
100	128K	ADDR [23:17]
101	256K	ADDR [23:18]
110	512K	ADDR [23:19]
111	512K	ADDR [23:20]

Note: ADDR [23:20] has the same logical level as ADDR19 during normal operation.

3.8.6 Option register

There are eight option registers that determine the timing and conditions of asserting the chip select signal. There is a field of, and it is now possible to generate peripheral control signals using chip select. is. To assert the chip select signal, the base address register and options All bits of the register must be satisfied. To support DTACK or autovector The conditions for these bits must be met.

CSORA—Chip Select Option Register A $ YFFA62

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MODE	BYTE		R / W		STRB	DTACK				SPACE		PROG / DATA		IDLE	IACK / CS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSORB—Chip Select Option Register B $ YFFA66

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MODE	BYTE		R / W		STRB	DTACK				SPACE		PROG / DATA		IDLE	UNUSED
RESET:															
ROMEN = 1															
0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0

ROMEN = 0 *

0	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0

* The reset status of ROMEN is "0" for MC68HC (9) 16Y5 / 916Y6.

CSORC—Chip Select Option Register C $ YFFA6A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MODE	BYTE		R / W		STRB	DTACK				SPACE		PROG / DATA		IDLE	UNUSED
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CSB (CSORB) option register is set to the peripheral memory device after reset.
There is a default value to support the bootstrap behavior of.

MODE—Asynchronous / Synchronous mode

0 = Select asynchronous mode (chip select assert to internal or external bus control signal
Therefore, it is decided).

1 = Select synchronization mode (chip select is asserted synchronously with the ECLK signal).

The MCU must include the optional ECLK pin to enable synchronous mode. Not the same
In period mode, chip select is asserted synchronously with AS or DS.
The DTACK field is used in synchronous mode because the bus cycle runs only as an asynchronous operation.
Not used. When a match condition occurs in a chip select programmed into a synchronous operation
Chip select informs EBI that the E clock cycle is pending.

BYTE—Higher / Lower Byte Options

If you select the chip select 16-bit port option in the pinout register, this
The field is used. Table 44 lists the high / low byte options.

Table 44 Top / Bottom Byte Options

Part-Time Job	Explanation
00 00	Disable Disable
01 01	Lower Byte
Ten	Upper Byte
11 11	Both Bytes

Page 126

If the device requesting the interrupt does not provide the vector number, use autovector acknowledge.
Must be generated. The bus cycle ends when AVEC is asserted. This is AVEC internally
Can be generated and done using the chip select option register.

R / W—Lead / Write

This field makes chip select read-only, write-only, or both read and write.
Assert. See Table 45 for available options .

Table 45 R / W encoding

R / W	Explanation
00 00	Reserved
01 01	Read Only
Ten	Write Only
11 11	Read / Write

Combining R / W fields with address strobe / data strobe fields
Allows you to generate various control signals for external devices. Day in the BYTE field
Unless a sable option is detected, each time an address match occurs, the specified block
Chip select is asserted at all addresses in the lock.

STRB—Address Strobe / Data Strobe

0 = address strobe

1 = Data strobe

This bit controls the timing of chip select assertions in asynchronous mode. Ah
If you select a dress strobe, the chip select will be asserted in sync with the address strobe.
It will be. If you select a data strobe, the chip select will synchronize with the data strobe.
Will be sart.

DTACK — Data transfer acknowledge

This field specifies the source of DTACK in asynchronous mode. The optional DTACK pin
When not included in a particular version of SLIM, use chip select to internally DTACK
Generate. You can also use this field to maximize bus speed for a particular application.
User generates internal DTACK by controlling the number of wait states inserted for optimization
You can adjust the bus timing according to. Table 46 shows the encoding of the DTACK field.
To do.

Table 46 DTACK field encoding

DTACK	Explanation	
	1 bus cycle Wait number	1 bus cycle Number of system clocks
0000	No Wait States	2 system clock
0001	1 Wait State	3 system clock
0010	2 Wait States	4 system clock
0011	3 Wait States	5 system clock
0100	4 Wait States	6 system clock
0101	5 Wait States	7 system clock
0110	6 Wait States	8 system clock
0111	7 Wait States	9 system clock
1000	8 Wait States	10 system clock
1001	9 Wait States	11 system clock

Page 127

Table 46 DTACK Field Encoding (continued)

DTACK	Explanation	
	1 bus cycle Wait number	1 bus cycle Number of system clocks
1010	10 Wait States	12 system clock
1011	11 Wait States	13 system clock
1100	12 Wait States	14 system clock
1101	13 Wait States	15 system clock
1110	14 Wait States	16 system clock
1111	Until the assertion of the DTACK pin	Until the assertion of the DTACK pin

SPACE—Address space

This option field is indicated by the CPU-generated function code.
Used to check the dress space. CPU16 always operates in supervisor space, but it is relatively
At the time of built-in acknowledge, it operates in CPU space. Table 47 shows the encoding of the SPACE field
Indicates.

Table 47 SPACE field encoding

SPACE field	Address space
00 00	CPU space
01 01	User space
Ten	Supervisor space
11 11	Supervisor / User Space

PROG / DATA—Program / Data Space

The PROG / DATA bit is set to decode the program and data space as shown in Table 48.
It is justified. Chip selection if the space field is set to CPU space (% 00)
Logic can be used to check for interrupts. Table 48 shows the program / data field d.
Indicates programming.

Table 48 Program / Data Field Encoding

PROG / DATA field	Program / data space
00 00	Data or program space
01 01	Data space
Ten	Program space
11 11	spare

This field only affects the chip select response and affects the CPU's perception of interrupts. I will not give it. Any level is a chip select assassin, regardless of the level of the IACK cycle. It means to be to.

IDLE—Addition of idle clock

This field is for external devices on the bus before SLIM drives the next address on the bus. Add an idle clock at the end of the external cycle to allow time to stop the drive Decide if.

0 = Do not add idle clock.
1 = Add one idle clock.

IACK / CS—CSA assertion on IACK

This bit, along with CSCR8, determines whether the external IACK cycle is terminated externally or internally. It also decides whether to assert CSA during the external IACK cycle. Table 49 shows IACK / CS fees Indicates the rud.

Table 49 IACK / CS fields

IACK / CS field	Explanation
0	CSA not asserted with external IACK cycle
1	CSA asserted with external IACK cycle if CSCR8 = 1

3.8.7 Boot ROM Chip Select and Boot ROM Submo Jules

Initialization software accesses an external memory device from the boot ROM chip select. CSB is used for this. The boot ROM submodule occupies 512 bytes of system memory. It can be accessed by it or word. Module Configuration Register (MCR) The MOTO, ROMSU, and ROMEN bits control boot ROM behavior. For [redacted] The Temu configuration " , for the description of the behavior and how to please.

3.9 Factory test block

The test submodule supports scan-based testing of various MCU modules. child The test submodule is built into SLIM, which supports protection testing.

3.9.1 Test register

The test submodule registers are used in Motorola's factory tests. Register name and The display of addresses is to indicate that these addresses are occupied.

SLIMTR—System Integration Test Register	$ YFFA02
SLIMTRE—System Integration Test Register (E-Clock)	$ YFFA09
TSTMSRA—Master shift register A	$ YFFA40
TSTMSRB—Master shift register B	$ YFFA42
TSTSC—Test Module Shift Count	$ YFFA44
TSTRC—Test module iteration count	$ YFFA46
CREG—Test submodule control register	$ YFFA48
DREG—Distributed Register	$ YFFA4A

Chapter 4
Auxiliary chip select

Auxiliary chip select modules (ACS) can be used to add chip select is. The MC68HC (9) 16Y5 / 916Y6 has three chip select modules with an auxiliary chip select module. The code has been added.

IMB

BUS INTERFACE

Auxiliary
Cihp Select 0

PCS0 / ACS0

Auxiliary
Cihp Select 1

Port
CS

PCS1 / ACS1

Auxiliary
Cihp Select 2

PCS2 / ACS2

Figure 15 ACS block diagram

4.1 Overview

Standard microcontrollers have additional hardware to supply external chip select signals I need a. This MCU contains three programmable chip select circuits, 2-16. External memory and peripheral devices can be accessed in a clock cycle. Chit given by ACS A subset of the functionality provided by Select is supported.

The features supported by the auxiliary chip select module are:

- Three programmable chip select
  Chip select ACS0, ACS1 and ACS2 can be individually programmed with a variety of selectable features. Can be done. All chip select has the same selectable features.
- Various block sizes
  The size of the block starting from the base address is 4K, 8K, 16K, 32K, 64 <128K, 256K, 512K bytes. You can blog.
- Read-only, write-only, or read / write select
  Memory blocks can be read-only, write-only, or both read and write I can do it.
- Address strobe and data strobe timing
  Chip select signals can be synchronized to AS or DS, thus controlling CS, OE, WE, etc. You can easily generate a signal.

- Insert wait state to generate internal DTACK
  This option allows DTACK to be generated internally. Users interface with various devices
  You can program the appropriate number of wait states to do this.
- Checking the address space
  You can optionally check supervisor, user, program and data space access
  increase.
- Three general-purpose output pins for discrete output
  The three chip select pins can be used as general purpose output pins.

All chip select supports 16-bit data bus wide non-multiplexed master mode However, it does not support 8-bit data buses.

Below is the flow chart for asserting a chip select. External bus cycle data Be careful to properly program the base and options registers for the ip. Please.

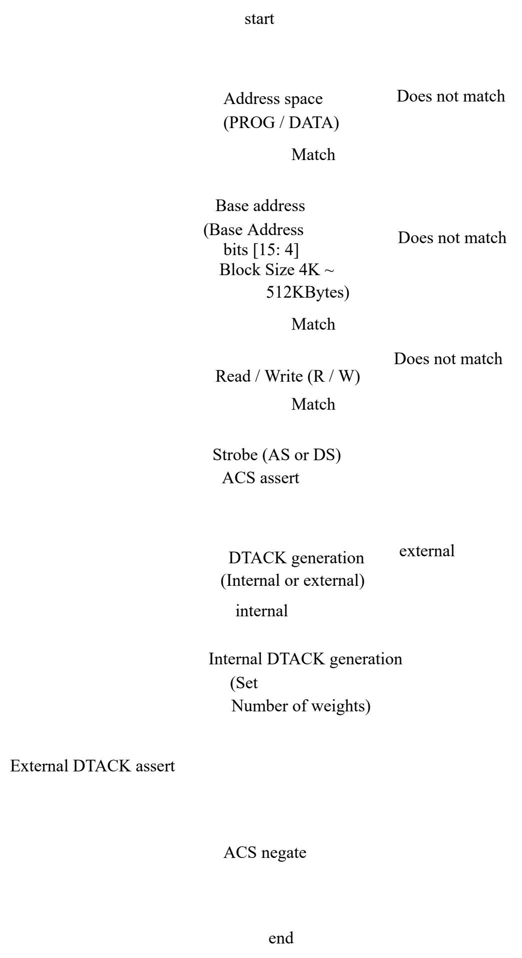


Figure 16 Chip Select Match Flow

Even if the chip select pin is not available, use the chip select logic, You can end the external cycle. Chip select needs to generate DTACK internally Use the DTACK field in any address space to determine if.

Table 50 ACS address map

Address Address	15	8	7	0
$ YFFA80	Auxiliary Chip Select Module Configuration Register (ACSMCR)			

$ YFFA82	Auxiliary Chip Select Module Test Register (ACSTR)	
$ YFFA84	Reserved	
$ YFFA86	Reserved	
$ YFFA88	Reserved	ACS Output Data Register (ACSODR)
$ YFFA8A	Reserved	
$ YFFA8C	Reserved	
$ YFFA8E	Reserved	ACS Pin Assignment Register (ACSPAR)
$ YFFA90	Auxiliary Chip Select Base Address Register 0 (ACSBSR0)	
$ YFFA92	Auxiliary Chip Select Option Register 0 (ACSOR0)	
$ YFFA94	Auxiliary Chip Select Base Address Register 1 (ACSBSR1)	
$ YFFA96	Auxiliary Chip Select Option Register 1 (ACSOR1)	
$ YFFA98	Auxiliary Chip Select Base Address Register 2 (ACSBSR2)	
$ YFFA9A	Auxiliary Chip Select Option Register 2 (ACSOR2)	

Note: Y = M111. However, M is the logical state of the SLIMCR modemap (MM) bits. child
In the MCU, Y must be $ F. Reset if M is cleared
You will not be able to access the IMB module until M is once after reset
Can only be written.
SLIM may be in non-multiplexed master mode for the ACS module to work properly.
The SLIM chip select control register SZA / SZB / SZC (chip)
Select port size) At least one bit to 1 (= 16-bit port)
Please set it.

4.2 ACS Global Register

ACS global registers define the parameters that ACS interfaces with the CPU. reset After that, ACS is set as a generic output.

ACSMCR — ACS configuration register $ YFFA80

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP				NOT USED				SUPV				NOT USED			
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

STOP Stop enable
0 = Normal ACS clock operation.
1 = Stop ACS clock operation.
STOP disables the system clock on most of the modules, putting ACS in a low power state.
To do.

Page 132

SUPV — Supervisor / Unrestricted
0 = unconstrained access
1 = Supervisor Access
On systems with control access levels, SUPV supervises assignable ACS registers.
Place it in either the dedicated data space or the unconstrained data space. All ACS registers are
It will be placed in the supervisor space. CPU16 only works in supervisor mode, so SUPV
Has no meaning.

ACSTR — ACS module test register $ YFFA82

ACSODR — ACS output data register $ YFFA89

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					0	0	0	0	0	ODR2	ODR1	ODR0
RESET:															
								0	0	0	0	0	1	1	1

ACSODR is driven to the PORT CS output pins if they are configured as general purpose outputs. Used to latch the data to be used. Reading ACSODR always reads the current value of the data latch It will be squeezed out.

ACSPAR — ACS pin assignment register $ YFFA8F

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					0	0	0	0	0	PAR2	PAR1	PAR0
RESET:															
								0	0	0	0	0	1	1	1

ACSPAR assigns which ACS pin to chip select function or general purpose output in 1-pin increments. Will be decided by. Chip select pins configured as general purpose outputs in ACSPAR are configured by ACSODR. Only controlled, the chip select circuit does not affect these pins.

0 = general-purpose output
1 = Chip select function

Table 51 ACS pin controls

ACSPAR bit	Port CS signal	Bus control signal
PAR2	PCS2	ACS2
PAR1	PCS1	ACS1
PAR0	PCS0	ACS0

4.3 ACS base address register

The base address is the starting address of the block enabled by chip select. BLKSZ Determines the block range above the base address. Each chip select has a related base It has an address register and efficiently builds an address map for each application. I can do it. If the chip select address range overlaps the address of the internal module, Internal modules take precedence. Chip select does not respond to access.

ACSBAR0—ACS base address register 0 $ YFFA90
ACSBAR1—ACS base address register 1 $ YFFA94
ACSBAR2—ACS base address register 2 $ YFFA98

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12		BLKSZ		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPU16 drives A [23:20] into the same logical state as A19. Chip select for CPU16 For it to work properly, the value programmed by the user at A [23:20] must match A19.

ACSBAR [15: 4] — Base Address Field

This field sets the starting address for a particular address space. The address comparison logic is Compares the addresses in the block using only the most significant bit. The base address value is block Must be a multiple of the size. In the base address register diagram, the base register Shows the correspondence between bits and address lines.

BLKSZ— Block size field

This field is above the base address that must be enabled in chip select Determines the size of the block. The following table shows the block size field bit en Shows the coding.

Table 52 BLKSZ Bit Encoding

Block size field Block size		Address line to compare
0000	4 K	A [23:12]
0001	8 K	A [23:13]
0010	16 K	A [23:14]
0011	32 K	A [23:15]
0100	64 K	A [23:16]
0101	128 K	A [23:17]

0110	256 K	A [23:18]
0111	512 K	A [23:19]
1000	512 K	A [23:20]
1001	512 K	A [23:21]
1010	512 K	A [23:22]
1011	Reserved	Reserved
1100	Reserved	Reserved
1101	Reserved	Reserved
1110	Reserved	Reserved
1111	Reserved	Reserved

Note: ACS is not asserted when you select the Reserved setting.

A [23:20] is driven to the same logical state as A19, so the maximum block size is 512K bytes.
vinegar. Even if you use all 24 address lines, addresses from $ 080000 to $ F7FFFF are accessible.
I can't.

4.4 Option register

There are six option registers that determine the timing and conditions of asserting the chip select signal.
There is a field of. Required for interface to external devices using option registers
It can generate as many different signals as it needs, so it can be properly programmed depending on the type of access being performed.
Please be careful to ram.

Page 134

ACSOR0—ACS option register boot 0	$ YFFA92
ACSOR1—ACS option register boot 1	$ YFFA96
ACSOR2—ACS option register boot 2	$ YFFA9A

CSOR [0:10] — Chip select option register $ YFFA4E-$ YFFA76

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED	BYTE		R / W		STRB		DTACK (WAITS)			SPACE		PROG / DATA		NOT USED	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MBYTE — Top / Bottom Byte Options

The high / low byte option is used to properly handle bus forwarding on 16-bit ports.
will be used.

Table 53 Top / Bottom Byte Options

Part-Time Job	Explanation
00 00	Disable Disable
01 01	Lower Byte
Ten	Upper Byte
11 11	Both Bytes

R / W—Lead / Write

This field can make chip select read-only, write-only, or both read and write.
Sart

Table 54 R / W encoding

R / W	Explanation
00 00	Reserved
01 01	Read Only
Ten	Write Only
11 11	Read / Write

STRB—Address Strobe / Data Strobe

0 = address strobe
1 = Data strobe

This bit controls the timing of chip select assertions in asynchronous mode. Adre
If you select a strobe, the chip select is asserted in sync with the address strobe.
increase. If you select a data strobe, the chip select will synchronize with the data strobe.
Will be sart.

Note: A chip whose DTACK is synchronized to the DS at the end of 2 clocks (no wait state).

In a select configuration, the chip select is not asserted in the write cycle.

DTACK—Data Strobe Acnorridge

This field specifies the source of DTACK in asynchronous mode. In addition, the user inserts By controlling the number of state states, the user adjusts the bus timing to the generation of the internal DTACK. You can adjust it. This feature optimizes bus speed for specific applications It is possible. The following table shows the encoding of the DTACK field. No weight The state encoding (% 0000) corresponds to the 2-clock cycle bus.

If you want to program multiple ACS chip select with the same base address, the DTACK way The state must be set to the same value.

Whenever a DTACK pin is available, an externally generated DTACK rather than an internally generated DTACK Note that is prioritized. This is when the external DTACK occurs earlier than the internal DTACK. It means that the bus cycle is terminated by the external DTACK instead of the internal DTACK.

Table 55 DTACK field encoding

	Explanation	
DTACK	1 bus cycle Wait number	1 bus cycle Number of system clocks
0000	No Wait States	2 system clock
0001	1 Wait State	3 system clock
0010	2 Wait States	4 system clock
0011	3 Wait States	5 system clock
0100	4 Wait States	6 system clock
0101	5 Wait States	7 system clock
0110	6 Wait States	8 system clock
0111	7 Wait States	9 system clock
1000	8 Wait States	10 system clock
1001	9 Wait States	11 system clock
1010	10 Wait States	12 system clock
1011	11 Wait States	13 system clock
1100	12 Wait States	14 system clock
1101	13 Wait States	15 system clock
1110	14 Wait States	16 system clock
1111	Until the assertion of the DTACK pin	Until the assertion of the DTACK pin

SPACE—Address space

This option field is used to select the address space of the chip select logic. I will use it. CPU16 normally operates in supervisor space. See the table below.

Table 56 SPACE field encoding

SPACE field	Address space
00 00	Reserved
01 01	User space
Ten	Supervisor space
11 11	Supervisor / User Space

PROG / DATA—Program data space

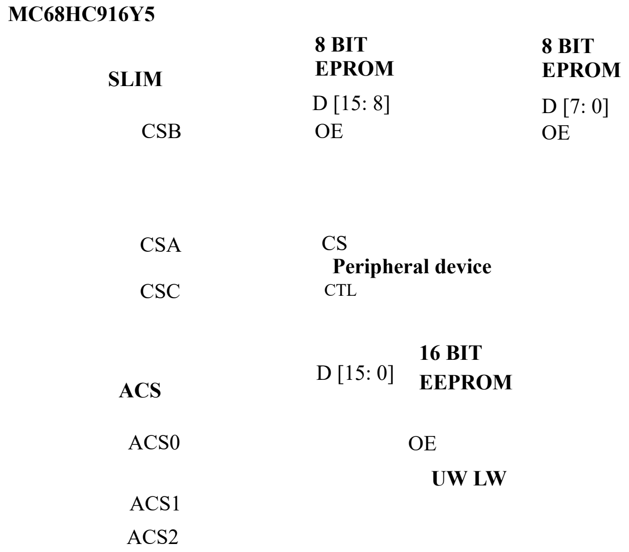
The PROG / DATA field is used to select the program and data space. The table below please refer to.

Table 57 PROG / DATA field encoding

PROG / DATA field	Address space
00 00	Program or data
01 01	Data space
Ten	Program space
11 11	Reserved

4.5 System connection example

The following figure shows an example of connecting a system with auxiliary chip select and SLIM chip select.



	Chip select base address		Optional state
	CSB	ADDRESS1	16BIT PORT, BOTH BYTE, Read
SLIM	CSA	ADDRESS2	8BIT PORT, MUX mode UPPER BYTE, Read
	CSC	ADDRESS2	8BIT PORT, MUX mode UPPER BYTE, Write

	Chip select base address		Optional state
	ACS0	ADDRESS3	16BIT PORT, BOTH BYTE, Read
	ACS1	ADDRESS3	16BIT POTE, UPPER BYTE, Write
ACS Module	ACS2	ADDRESS3	16BIT PORT, LOWER BYTE, Write

Figure 17 ACS system connection example

Chapter 5
Time processor unit 2

Motorola's High Performance Time Processor Unit 2 (TPU2) is designed for timing control. An intelligent semi-autonomous microcontroller. TPU2 runs in parallel with the CPU and is CPU-mediated ROM microinstruction processing, real-time hardware event scheduling and processing Performs science, I / O operations, and access to shared data. As a result, the CPU for each timer event Setup and service times have been reduced or eliminated. Next, a simplified block diagram of TPU2 indicate.

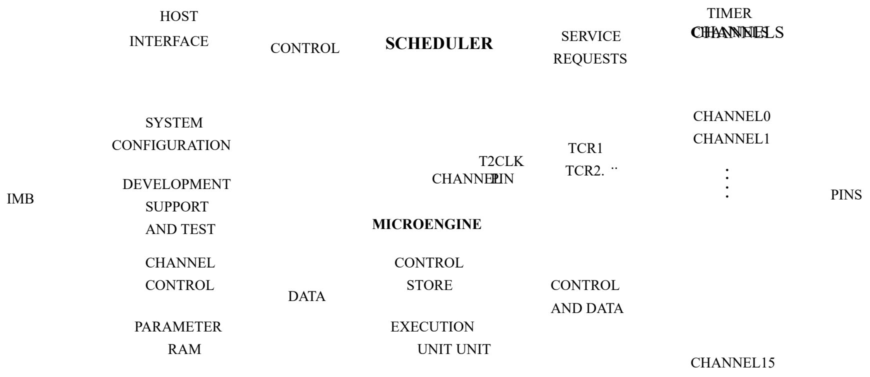


Figure 18 Block diagram of TPU2

5.1 Overview

The TPU2 feature replaces software features that require the host CPU's interrupt service. This book Does not touch on the details of the timer function. For a detailed description of a particular feature, see Using the TPU. Function Library and TPU Emulation Mode (TPUPN00 / D) "and other TPU programming notes please refer to. The standard TPU2 microcode ROM has the following factory-programmed ties. The function is implemented.

- Discrete input / output (DIO)
- New Input Capture / Transition Counter (NITC)
- Output compare (OC)
- Pulse width modulation (PWM)
- Synchronous pulse width modulation (SPWM)
- Multi-channel pulse width modulation (MCPWM)

- Table stepper motor (TSM)
- Period / Pulse width accumulator (PPWA)
- High-speed quadracha decoding (FQD)
- Queued output match (QOM)
- Programmable Time Accumulator (PTA)
- Universal asynchronous receiver / transmitter (UART)
- Brushless motor commutation (COMM)
- Frequency measurement (FQM)

・Hall effect decoding (HALLD)
• Serial input / output port (SIOP)

5.2 Programmer's model

The address map of the TPU2 control register occupies 512 bytes. 512 byte ad
Reads an unused register in less space and returns zero.

Table 58 TPU2 address map

Address Address	15 8 7 0
$ YFFE00	TPU2 MODULE CONFIGURATION REGISTER (TPUMCR)
$ YFFE02	TEST CONFIGURATION REGISTER (TCR)
$ YFFE04	DEVELOPMENT SUPPORT CONTROL REGISTER (DSCR)
$ YFFE06	DEVELOPMENT SUPPORT STATUS REGISTER (DSSR)
$ YFFE08	TPU INTERRUPT CONFIGURATION REGISTER (TICR)
$ YFFE0A	CHANNEL INTERRUPT ENABLE REGISTER (CIER)
$ YFFE0C	CHANNEL FUNCTION SELECTION REGISTER 0 (CFSR0)
$ YFFE0E	CHANNEL FUNCTION SELECTION REGISTER 1 (CFSR1)
$ YFFE10	CHANNEL FUNCTION SELECTION REGISTER 2 (CFSR2)
$ YFFE12	CHANNEL FUNCTION SELECTION REGISTER 3 (CFSR3)
$ YFFE14	HOST SEQUENCE REGISTER 0 (HSQR0)
$ YFFE16	HOST SEQUENCE REGISTER 1 (HSQR1)
$ YFFE18	HOST SERVICE REQUEST REGISTER 0 (HSRR0)
$ YFFE1A	HOST SERVICE REQUEST REGISTER 1 (HSRR1)
$ YFFE1C	CHANNEL PRIORITY REGISTER 0 (CPR0)
$ YFFE1E	CHANNEL PRIORITY REGISTER 1 (CPR1)
$ YFFE20	CHANNEL INTERRUPT STATUS REGISTER (CISR)
$ YFFE22	LINK REGISTER (LR)
$ YFFE24	SERVICE GRANT LATCH REGISTER (SGLR)
$ YFFE26	DECODED CHANNEL NUMBER REGISTER (DCNR)
$ YFFE28	TPU2 MODULE CONFIGURATION REGISTER 2 (TPUMCR2)

Y = M111. Where M represents the logical state of the SLIMCR module map bits.
vinegar. For M68HC16 devices, M must be 1.

5.3 TPU2 components

The TPU2 module has two 16-bit timebases, 16 independent timer channels, and a task.
It consists of a scheduler, a microengine, and a host interface. Moreover,
A dual that TPU2 uses to store data and pass parameters between the module and the host CPU.
There is also an al-port parameter RAM.

5.3.1 Time base

Two 16-bit counters are the basis for all output compare and input capture events
Provides a quasi-time base. The prescaler for the two timebases is the TPU2 module controller.
Host CPU controls using bit fields in the regulation registers (TPUMCR, TPUMCR2)
I will control it. The TCR1 clock can be obtained from the system clock. TCR2 clock is system
It can be obtained from the clock or from an external clock input through the T2CLK pin.

5.3.2 Timer channel

TPU2 has 16 independent channels, each connected to an MCU pin. these

The channel hardware is the same except for channel 15. Output disabled for channel 15

Gick has been added. The direction of each pin (output or input) is controlled by TPU2.

5.3.3 Scheduler

The scheduler determines the TPU channel that the microengine serves. Host system
Assigns each active channel one of three priorities: high, medium, and low. Multiple sir
If the service request is received at the same time, the priority scheduling mechanism is assigned the priority.
Allow services based on rank and channel number.

5.3.4 Micro engine

The microengine consists of a control store and an execution unit. Controls
The tore ROM contains the microcode for each factory-masked timer function.

5.3.5 Host interface

The CPU can control TPU2 using host interface registers. Excluding CISR
All TPU2 registers must be accessed by word operation. CISR is a byte or word
You can access it by operation. Host interface registers are accessible from the IMB.

5.3.6 Parameter RAM

Parameter RAM is dual-port communication RAM for TPU2 and host CPU, TPU2
Occupies the first 256 bytes of the module address map. Parameter RAM is used by TPU2
It also stores constants and variables. Channel parameters are configured as 128 x 16-bit words
vinegar. The following parameter RAM address map shows the organization of parameter words in memory.

Page 140

Table 59 TPU2 Parameter RAM Address Map

Channel	Base	Parameter Address							
Number	Address Address	0	1	2	3	Four	Five	6	7
0	$ YFFFF ##	00 00	02 02	04 04	06 06	08 08	0A	0C	0E
1	$ YFFFF ##	Ten	12	14	16 16	18 18	1A	1C	1E
2	$ YFFFF ##	20	twenty two	twenty four	26	28 28	2A	2C	2E
3	$ YFFFF ##	30	32	34	36	38	3A	3C	3E
Four	$ YFFFF ##	40	42	44	46	48	4A	4C	4E
Five	$ YFFFF ##	50	52 52	54	56	58	5A	5C	5E
6	$ YFFFF ##	60	62	64	66 66	68 68	6A	6C	6E
7	$ YFFFF ##	70	72	74 74	76 76	78 78	7A	7C	7E
8	$ YFFFF ##	80	82	84 84	86	88	8A	8C	8E
9	$ YFFFF ##	90	92	94	96	98	9A	9C	9E
Ten	$ YFFFF ##	A0	A2	A4	A6	A8	AA	AC	AE
11 11	$ YFFFF ##	B0	B2	B4	B6	B8	BA	BC	BE
12	$ YFFFF ##	C0	C2	C4	C6	C8	CA	CC	CE
13	$ YFFFF ##	D0	D2	D4	D6	D8	DA	DC	DE
14	$ YFFFF ##	E0	E2	E4	E6	E8	EA	EC	EE
15	$ YFFFF ##	F0	F2	F4	F6	F8	FA	FC	FE

5.4 Features of TPU2

5.4.1 Event timing

Match and capture events are handled by independent channel hardware
vinegar. Therefore, the event is accurate with one time-based clock period, regardless of the number of active channels.
Will be processed.

In many timer systems, the number of functions assigned to each pin is constant and functionally constrained.
I did. All TPU2 channels have the same built-in hardware except channel 15, and they are also operational.
Since there is no functional difference, you are free to configure any channel to perform the required timer function.
Channel 15 is logged so that it can be configured as a regular TPU2 channel or an output disable pin.
A circuit is added.

5.4.2 Communication between channels

In TPU2, one channel affects the operation of one or more channels without CPU intervention.
It is possible to further enhance the uniqueness. Multiple channels on one channel due to interchannel communication
You can control channels and multiple functions can interact with each other.

5.4.3 Service Priority for Programmable Channels

TPU2 provides services so that demanding features are frequently serviced and regardless of priority level.
Two preferred schemes are provided to ensure that the minimum time is allocated to all requesting channels.
doing. One is the priority setting for multiple request channels with different priority levels, and the other

The first is setting the priority for request channels with the same priority level. Multiple with the same priority
When channels request services at the same time, arbitration is performed according to the channel number.

5.4.4 Cache coherency

In order for the data to be coherent, all available parts of the data are of the same or logical generation.
Must be related to. For example, 32 bits read and written as two 16-bit words
Suppose you have a counter value. This 32-bit value is when both 16-bit parts are updated at the same time.
Only read coherent and write only if both 16-bit parts are enabled at the same time.
It's coherent. The hardware of the parameter RAM is a co-op of two adjacent 16-bit parameters.
Supports healthy access. Host CPU uses long word operation to coffee
You must guarantee the license.

5.5 Timer function

The following sections describe the TPU features implemented in the MC68HC (9) 16Y5 / 916Y6. This in this book
I will omit a detailed explanation of these functions. For more information, see Using the TPU Function Library and TPU.
See Emulation Mode (TPUPN00 / D) and other TPU programming notes.

5.5.1 Discrete Input / Output (DIO)

When using a pin as a discrete input, the parameters are the current input level of the pin and its
Indicates the previous 15 levels. Bit 15, the most significant bit of the parameter, is up to date. Bi
14 indicates the new state next. The same applies below. The programmer chooses one of the following three conditions
You can update the parameters.

1. When a transition occurs
2. When the host CPU makes a request
3. When a match occurs at the rate specified for another parameter

"H" or "L" only at the request of the CPU when using pins as discrete outputs
Cannot be set to.

5.5.2 Output compare (OC)

The output compare feature uses one of three methods: rising edge, falling edge, or
You can generate a flipped edge (toggle behavior) of the previous edge.

- Immediately after the CPU starts, it produces a pulse with a length equal to the programmable delay time.
- Generated after a programmable delay from the time specified by the user.
- Continuous generation. Upon receiving a link from the channel, the output compare feature is specified without CPU intervention.
  The offset is calculated by the following formula with reference to the period.

Offset = period * ratio

The ratio in this case is a parameter set by the user.

This algorithm has a 50% duty, where each “L” “H” / “L” time is equal to the calculated offset value.
Generates a continuous square wave of cycles. Initially before continuous pulse generation is started due to offset calculation
There is a link time.

5.5.3 Pulse Width Modulation (PWM)

TPU2 has a pulse width variation of 0 to 100% duty cycle (within the resolution and latency of TPU2).
You can generate a PWM waveform. The CPU has a parameter that indicates the period to define PWM.
We have another parameter that indicates the “H” time. Update one or both of these parameters to
You can instruct the waveform to change immediately or in synchronization with the next “L” to “H” transition of the pin.

5.5.4 Synchronous Pulse Width Modulation (SPWM)

TPU2 produces a pulse width modulation (PWM) waveform, and the CPU changes the period and “H” time of this waveform at any time.
I can do it. Synchronized with the timer function of the 2nd channel, the transition of the synchronous PWM from “L” to “H” and the 2nd cha
The transition of flannel can have a temporal relationship.

5.5.5 Period / Pulse Width Accumulator (PPWA)

The period / pulse width accumulator algorithm has a programmable number of periods or pulses (from 1).
Accumulates 16-bit or 24-bit total values for the period or pulse width of the input signal over a period of up to 255).
vinegar. After the cumulative period expires, the algorithm reverts to sequential blocks up to 8 channels.
You can generate a link. The user points to the starting channel of the block and the number of channels in the block.
Set. Link generation depends on the mode of operation. Input signal using any channel
It is possible to measure the cumulative number of cycles of. Up to 24 bits can be used for cumulative parameters. 1 to 255 cycles
If you take a measurement and add it to the previous measurement before the TPU2 interrupts the CPU, you can measure the frequency instantly.
Makes average measurements and up-to-date complete cumulative measurements (over a period of programmed cycles).
I can.

The pulse width (“H” time portion) of the input signal is measurable (up to 24 bits) and has a programmable period.
It can be added to previous measurements over a period of numbers (1 to 255). This makes it momentary
Or an average pulse width measurement capability can be achieved, and the latest complete cumulative number is always parametric (during a specified number of cycles).
You can get it at The specified input signal by using the output compare function in combination with PPWA.
You can generate an output signal that is proportional to. The ratio of input to output frequency is programmable. Different frequencies
One or more output signals that have a number but are proportionally synchronized to one input signal
It can be generated with flannel.

5.5.6 Table Stepper Motor (TSM)

TSM linearly accelerates and decrements stepper motors with up to 58 programmable step rates.
Performs speed control. TSM uses a table of PRAM, not an algorithm, to add a stepper motor
A fast profile is defined, allowing the user to fully define the profile. In addition,
The loop rate parameter reduces the terminal operating speed of the motor independently of the acceleration table.
You can control it in this way. The CPU only has to write the desired position, and the TPU2 motors to the required position.
Accelerates, speeds up, and decelerates. Full-step and half-step operation is supported for 2-phase motors
It has been. In addition, the slew rate parameter allows the motor to start regardless of the acceleration table.
You can finely control the minal operation speed.

5.5.7 New Input Capture / Transition Coun Ta (NITC)

For any channel in TPU2, each time a transition occurs, or when a specified number of transitions occur.
Captures the value of the specified location in the specified TCR or parameter RAM and issues an interrupt request on the bus.
You can notify the master. The time of the last two transitions is kept in the parameter RAM
increase. Channels can perform input captures without interruption. Also, one transition or a specified number of transitions
Once detected, channel activity can be stopped until it is reinitialized. After each transition
Or after a specified number of transitions, a channel can generate links to other channels.

5.5.8 Queued Output Match (QOM)

QOM generates one or more output match events from the offset table in parameter RAM
Can be In loop mode, compound pulse trains can be generated once, a specified number of times, or consecutively.
This feature can be triggered by a link from another TPU2 channel. In addition, of the match sequence
You can get the reference time for from another channel. QOM is a wave with 0% or 100% high time

You can generate pulse width modulated waveforms that include shapes. In addition, QOM allows the TPU2 channel to be output discretely.
It can also be used as a computer.

5.5.9 Programmable Time Accumulator (PTA)

The PTA is a programmable period that has a 32-bit sum of the total “H” time, “L” time, or the period of the input signal.
Accumulate by number or number of pulses. Periodic accumulation starts at rising or falling edge
I can do it. After the specified number of cycles or pulses, the PTA issues an interrupt request and optionally another cha.
Generates a link to the flannel.

It is possible to make periodic measurements from 1 to 255 and add them to the previous measurements before the TPU2 interrupts the CPU.
Therefore, the instantaneous or average frequency measurement function and the latest complete accumulation (programmed).
Period of the number of cycles) is realized.

5.5.10 Multi-channel pulse width modulation (MCPWM)

MCPWM has a full 0% to 100% duty size, regardless of the activity of other TPU2s.
Generates a pulse width modulated output in the circle range. This includes two TPU2 channels and one PWM channel.
An external gate is required (simple 1-channel PWM capability is supported by the QOM feature).

There are two types of high ties, edge-aligned and center-aligned, for multiple PWMs generated by MCPWM.
There is alignment. Edge-aligned mode has n + 1 TPU2 chas for n PWMs.
It uses flannel and center-aligned mode uses 2n + 1 channels. Center align
In mode, specify a user-defined dead time to avoid catastrophic current spikes. 2
You can drive an H-bridge with multiple PWMs. This feature is important in motor control applications
It is important.

5.5.11 High Speed Quadrature Decode (FQD)

FQD is a position feedback function for motor control. Two signals from a slotted encoder
Decodes to provide the CPU with a 16-bit free running position counter. High speed for FQD
Built-in CPU selection “speed switch” that disables one of the channels in the high speed signal
Can be decoded. Timestamps are provided with each counter update, so position interpolation
It is possible, and the speed can be determined more appropriately at low speeds or when using a low resolution encoder.
increase. The third index channel on some encoders is supported by the NITC feature.

5.5.12 Universal Asynchronous Receiver / Transmitter (UART)

The UART feature uses one or two TPU2 channels for asynchronous communication. Data word
The length is programmable from 1 to 14 bits. This feature with even, odd, and no parity detection
Supports generation. The baud rate can be freely programmed and can be 100K baud or higher. In TPU2
It can provide eight bidirectional UART channels operating at 9600 baud and above.

5.5.13 Brushless motor commutation (COMM)

This feature produces phase commutation signals for various brushless DC motors. Position decoded to FQD,
Or the commutation state is obtained directly from the HALLD function.

The state sequence is realized as a user-configurable state machine, thereby allowing other pans.
Allows a flexible approach in your application. CPU offset parameters are available
This allows the CPU to advance or delay all switching angles during operation.

5.5.14 Frequency measurement (FQM)

FQM counts the number of input pulses to the TPU2 channel during the user-defined window cycle. this
Features include single shot mode and continuous mode. In continuous mode, sample we
No pulse is lost between the windows. The user can use either the rising edge or the falling edge.
Select whether to detect the pulse with. This function is intended for high-speed measurement and has a noise reduction function.
Slow pulse measurements can be performed with PTA.

Page 144

5.5.15 Hall effect decoding (HALLD)

This function combines the sensor signal from the brushless motor with the directional input from the CPU and converts it to the status number.
Code This feature supports 2-sensor or 3-sensor decoding. Decoded
The status number is written to the COMM channel and outputs the required commutation drive signal. This feature
Not only for Lasiles motor applications, but also for optional switch decoding, etc.
It can be applied to general applications.

5.5.16 Serial Input / Output Port (SIOP)

The SIOP feature uses two or three TPU2 channels for one-way or two-way synchronous serial ports.
Can be used to form and communicate with a wide variety of peripheral devices. Baud rate and data transfer support
Features such as is can be programmed by the user. In this case, only one TPU2 channel is used
increase. This function can also generate only the clock.

5.6 TPU2 register

The TPU2 memory map includes system configuration registers and channel controls.
Le register and status register, and development support register and test verification register
There is a data.

5.6.1 System configuration registers

TPUMCR — TPU2 module configuration register $ YFFE00

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	TCR1P		TCR2P		EMU	T2CG	STF	SUPV	PSCK	TPU2	T2CSL		IARB		
RESET:															
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

STOP — Stop bit
0 = TPU2 works normally.
1 = Internal clock stops.

TCR1P — Timer count register 1 Prescaler control
The TCR1 counter has a system clock rate of 2, 4, 8, 16, 32, 64, 128, or 256.
You can do it. Set the prescaler division ratio to 1, 2, 4, and 8 with the TCR1P bit.
To do. Also, the TCR1P, PSCK, and DIV2 of the TPUMCR2 register determine the rate of the TCR1 counter.
increase. When DIV2 is zero, the increment rate for TCR1 is shown in the table below (system clock is:
For 16MHz).

Table 60 TCR1 Increment Rate

TCR1P	Divide By	PSC = 0		PSC = 1	
		Number of Clocks	Rate at 16 MHz	Number of Clocks	Rate at 16 MHz
00 00	1	32	2 ∝ s	Four	250 ns
01 01	2	64	4 ∝ s	8	500 ns
Ten	Four	128 128	8 ∝ s	16 16	1 ∝ s
11 11	8	256	16 ∝ s	32	2 ∝ s

Page 145

DIV2 CLOCK

÷ 2

CSEL
MUX

0 15

TCR1

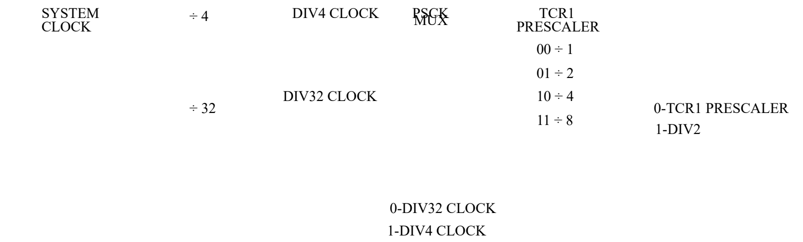


Figure 19 Prescaler control 1

TCR2P — Timer control register 2 prescaler control

The TCR2 is clocked from the output of the prescaler. If T2CG = 0, go to TCR2 prescaler The input to is an external TCR2 clock source. If T2CG = 1, the input is the TPU2 system clock. It becomes ÷ 8. The TCR2P field can be specified as a prescaler value of 1, 2, 4, or 8. to come. Channels using TCR2 have the ability to decompose the TPU system clock up to 8 divisions. I am. The table below summarizes the prescaler output.

Table 61 Prescaler output

TCR2P	Division ratio	Internal clock division ratio	External clock division ratio
00 00	1	8	1
01 01	2	16 16	2
Ten	Four	32	Four
11 11	8	64	8

EMU — emulation control

The EMU bit is used for microcode emulation.

T2CG — TCR2 clock / gate control

This control bit and the T2CSL control bit determine the clock source for TCR2. increase. See Table 62 .

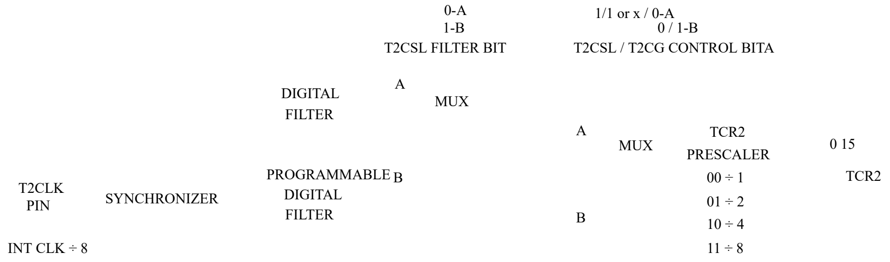


Figure 20 Prescaler control 2

STF — Stop flag

0 = TPU is running.
1 = TPU stopped (STOP bit asserted).

SUPV — Supervisor data space

CPU16-based devices always operate in supervisor mode.

0 = Allocatable registers are unconstrained (FC2 is ignored).
1 = Allocatable registers are constrained (FC2 is decoded).

PSCK — Prescaler clock

0 = System clock / 32 is the input to the TCR1 prescaler.
1 = System clock / 4 is the input to the TCR1 prescaler.

TPU2 — TPU2 enabled

Used when executing TPU code on TPU2. This bit can only be written once The number of bits that µPC and the breakpoint register scan in bits (9 bits or 11 bits)) And controls the parameter preload.

0 = TPU mode. Parameters determined by bit 10 of the micro ROM entry point

RAM preload, and 9-bit scan of μPC and breakpoint registers.
1 = TPU2 mode. 11-bit scan of μPC and breakpoint registers.

T2CSL — TCR2 counter clock source

This bit and the TPUMCR T2CG control bit allow the TCR2 clock source and edge. Determine the detection.

Table 62 TCR2 counter clock source

T2CSL	T2CG	TCR2 Clock
0	0	Rise transition T2CLK
0	1	Gated
1	0	Fall transition T2CLK
1	1	Rise and fall transition T2CLK

IARB — Interrupt arbitration number

This field has multiple modules or peripheral devices with the same priority level interrupt. When present, there is a TPU2 arbitration number used to arbitrate the intermodule bus.

TPUMCR2 — TPU2 module configuration register 2 $ YFFE28

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
							DIV2	SOFTRST	ETBANK [1: 0]		FPSCK [2: 0]			T2CF	DTPU
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DIV2 — ÷ 2 control

0 = TCR1 depends on the control bits of the TCR1P and PSCK fields of the TPUMCR register.
It is incremented at the determined rate.
1 = TCR1 counter is incremented at the rate of 2 system clocks.

Controls in Table 63 (÷ 2)

DIV2	PSCK	TCR1P	Clocks	Rate at 16.7 MHz	Rate at 20 MHz
1	X	XX	2	125 ns	100 ns

Page 147

SOFTRST — Soft reset

When this bit and the TPUMCR STOP bit are set, TPU2 performs an internal reset. To do. To free TPU2 from reset, the CPU must write zero to this bit. Hmm. SOFTRST must be asserted for 9 clocks.

0 = normal operation
1 = Put TPU2 in the reset state until the bit is cleared.

ETBANK [1: 0] — Entry Table Bank Select

Determines the bank location of the entry table. These control bits are Written only once after setting. ETBANK contains multiple entry tables in microcode Used when rare. ETBANK must be 00 to perform the ROM function.

Table 64 Bank location of entry table

ETBANK	BANK
00 00	0
01 01	1
Ten	2
11 11	3

FPSCK [2: 0] — Filter prescaler clock

Determines the ratio of the system clock frequency to the digital filter clock. these The bit reset value is zero, defining the filter clock as the four system clocks.

Table 65 System Clock Frequency / Minimum Guaranteed Detection Pulse

Filter control	Division ratio	16.7 MHz	20 MHz

000	2	240 ns	200 ns
001	Four	360 ns	300 ns
010	8	600 ns	500 ns
011	16 16	1.08 ∝ s	900 ns
100	32	2.04 ∝ s	1.7 ∝ s
101	64	3.96 ∝ s	3.3 ∝ s
110	128 128	7.8 ∝ s	6.5 ∝ s
111	256	15.48 ∝ s	12.9 ∝ s

T2CF — T2CLK pin filter control

This control bit can only be written once after a reset.
0 = Use a fixed 4 clock filter.
1 = T2CLK Filters the input pin with the same filter clock that is supplied to the channel.

DTPU — TPU2 pin disabled

This control bit can only be written once after a reset.
0 = TP15 acts as a normal TPU2 channel.
1 = TP15 pin is configured as an output disable pin.
When the TPU15 pin is zero, all TPU2 output pins are high impedance, regardless of pin function.
It becomes a state.

TICR — TPU2 interrupt configuration register $ YFFE08

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
		NOT USED				CIRL			CIBV				NOT USED		
RESET:															
					0	0	0	0	0	0	0				

CIRL - channel interrupt request level

The interrupt request level for all channels is specified in this 3-bit encode field.
vinegar. Level 7 in this field indicates a non-maskable interrupt. Level zero is all cha
Indicates that flannel interrupts are disabled.

CIBV — Channel interrupt-based vector

TPU2 is assigned 16 unique interrupt vector numbers, one for each channel. CIBV
The field specifies the most significant nibble for all 16 TPU2 channel interrupt vector numbers. TPU 20%
The lower nibble of the embedded vector number is determined by the number of the channel on which the interrupt occurred.
The information to be sent must be written to the sent data in right-justified format. QSPI sends data RAM
You cannot change the information in. QSPI copies the information to the data serializer in preparation for transmission
To do. The information remains in the outbound RAM until it is overwritten.

5.6.2 Channel control register

CIER - channel interrupt enable register $ YFFE0A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH [15: 0] — Channel interrupt enable / disable

0 = Disable channel interrupts.
1 = Enable channel interrupts.

CISR — Channel interrupt status register $ YFFE20A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH [15: 0] — Channel interrupt status bit

0 = Channel interrupt is not asserted.
1 = Channel interrupt is asserted.

CFSR0 — Channel function select register 0 $ YFFE0C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CHANNEL15				CHANNEL14				CHANNEL13				CHANNEL12			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CFSR1 — Channel function select register 1 $ YFFE0E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CHANNEL11				CHANNEL10				CHANNEL9				CHANNEL8			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CFSR2 — Channel Function Select Register 2 $ YFFE10

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CHANNEL7				CHANNEL6				CHANNEL5				CHANNEL4			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CFSR3 — Channel Function Select Register 3 $ YFFE12

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CHANNEL3				CHANNEL2				CHANNEL1				CHANNEL0			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Channels [15: 0] — Timer function encoded for each channel

The encoded 4-bit field in the channel feature selection register is the actual in the corresponding channel.
Specifies one of the 16 timer functions to be performed.

HSQR0 — Host sequence register 0 $ YFFE14

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH15		CH14		CH13		CH12		CH11		CH10		CH9		CH8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HSQR1 — Host sequence register 1 $ YFFE16

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH [15: 0] — Encoded host sequence

The host sequence field selects the operating mode of the timer function selected on a channel.
To do. The meaning of the host sequence bits depends on the timer function specified.

HSRR0 — Host service request register 0 $ YFFE18

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH15		CH14		CH13		CH12		CH11		CH10		CH9		CH8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HSRR1 — Host service request register 1 $ YFFE1A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH [15: 0] — Type of encoded host service

The host service request field is the host for the timer function selected on a channel.
Select the type of service request. The meaning of the host service request bit is the specified timer function.

Depends on. Host service request fields cleared to% 00 are microengines
Notifies the host that it has completed service on that channel. The host is the corresponding host
In the channel by writing one of the three non-zero states to the service request field.
You can request the service of. The CPU monitors the host service request register and TPU2 needs service.
Wait for the request to be cleared, then change the parameters or require a new service for that channel

You must issue a request.

CPR0 — Channel priority register 0 $ YFFE1D

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH15		CH14		CH13		CH12		CH11		CH10		CH9		CH8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPR1 — Channel priority register 1 $ YFFE1E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH [15: 0] — One of the three encoded channel priority levels

Table 66 channel priority levels

CHX [1: 0]	service
00 00	Disabled Disabled.
01 01	Low
Ten	Middle
11 11	High

5.6.3 Development Support Registers and Test Registers

These registers are used for custom microcode development or factory testing. This book Does not explain how to use these registers. Register name and address for reference I will post it.

DSCR — Development Support Control Register	$ YFFE04
DSSR — Development Support Status Register	$ YFFE06
LR — Link register	$ YFFE22
SGIR — Service grant latch register	$ YFFE24
DCNR — Decorated channel number register	$ YFFE26
TTCR — TPU2 test configuration register	$ YFFE02

Page 151

Chapter 6
Configurable timer
Module 3

Configurable Timer Module 3 (CTM3) is a Motorola Modular Microcontrol
It belongs to the timer module family of the roller family. The timer architecture is inside
Numerous timebases (counter submodules) and channels (action submodules) stored
It is modular with respect to joules or timer I / O pins).

CTM3 is a plurality of submodules located on either side of the CTM3 internal submodule bus (SMB).
It is composed of All data and control signals in CTM3 are transmitted through this bus. SMB
Is the bus interface unit sub connected to the intermodule bus (IMB) and CPU.
It connects to an external device through a module (BIUSM). With this configuration, the CPU is on each SMB.
You can access the data and control registers in the CTM3 submodule
increase. Four 16-bit wide timebase buses (TBB1, TBB2, TBB3, TBB4), each with a counter
Used to transfer timing information from to the action submodule.

Figure 21 shows a block diagram of the CTM3.

6.1 Overview

The time-based bus exits the counter submodule and is the action submodule.
Uses. Each submodule has access to two timebase buses.

All CTM3 submodules with the Bus Interface Unit Submodule (BIUSM)
Can communicate with IMB through the submodule bus (SMB).

The counter prescaler submodule (CPSM) can be used with the counter submodule 6
Generates a kind of clock frequency. This submodule is included in BIUSM.

The Modular Counter Submodule (MCSM) is modular in a wide range of applications.
Multi-function timer sub-mods capable of performing complex counting and timing functions such as counting
This is Le. The modulus register recycles the counter at counts other than 64K clock cycles.
Flexibility has been added. CTM3 has four built-in MCSMs.

Double Action Submodules (DASMs) can occur automatically without software intervention
It provides two 16-bit input captures or two 16-bit output compare functions. 13 for CTM3
It has 3 built-in DASMs.

The Pulse Width Modulation Submodule (PWMSM) allows for a wide range of frequencies, regardless of other CTM output signals.
Therefore, it is possible to generate a pulse width modulated signal. PWMSM affects the operation of the time-based bus
I will not receive it. CTM3 has five built-in PWMSMs.

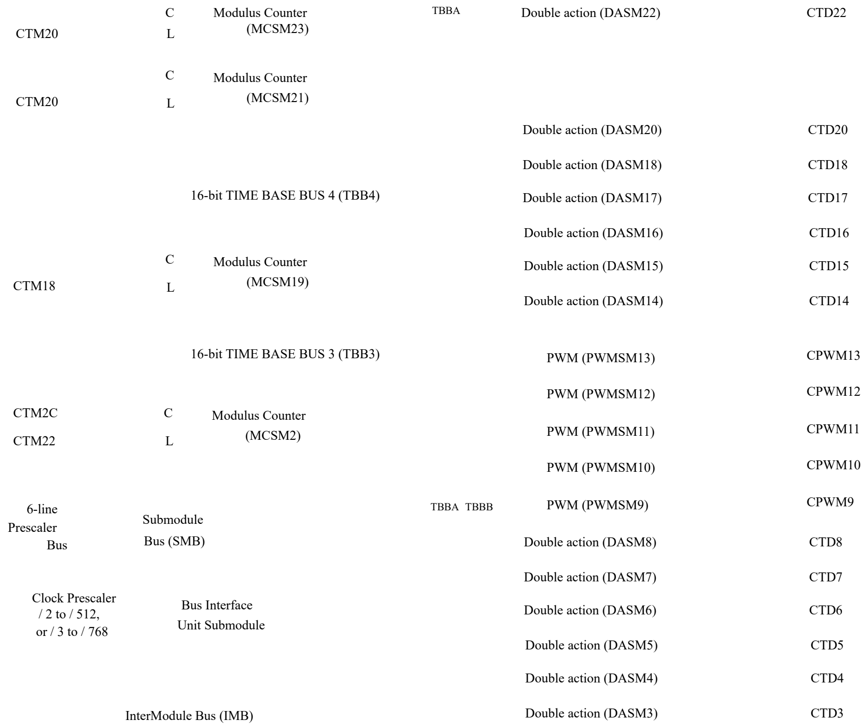
Page 152

16-bit TIME BASE BUS 1 (TBB1)

16-bit TIME BASE BUS 2 (TBB2)

Channel
I / O Pins

TBBB



Note: CTM18,20,22 are CTD18,20,22 respectively
It is shared with the pins of.

Figure 21 Block diagram of CTM3

6.2 Address map

The CTM3 address map occupies 256 bytes.

Table 67 Address Map for CTM3

Address Address	Submodule name	15	8 7	0
$ YFF900	BIUSM		BIUSM MODULE CONFIGURATION REGISTER (BIUMCR)	
$ YFF902			BIUSM TEST REGISTER (BIUTEST)	
$ YFF904			BIUSM TIME BASE REGISTER (BIUTBR)	
$ YFF906			RESERVED	
$ YFF908	CPSM		CPSM CONTROL REGISTER (CPCR)	
$ YFF90A			CPSM TEST REGISTER (CPTR)	

Table 67 Address Map for CTM3 (continued)

Address Address	Submodule name	15	8 7	0
$ YFF90C			RESERVED	
$ YFF90E			RESERVED	
$ YFF910	MCSM2		MCSM2 STATUS / INTERRUPT / CONTROL REGISTER (MCSMSIC)	
$ YFF912			COUNTER (MCSMCNT)	
$ YFF914			MODULUS LATCH (MCSMML)	
$ YFF916			RESERVED	
$ YFF918	DASM3		DASM3 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)	
$ YFF91A			REGISTER A (DASMA)	
$ YFF91C			REGISTER B (DASMB)	

$ YFF91E		RESERVED
$ YFF920	DASM4	DASM4 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF922		REGISTER A (DASMA)
$ YFF924		REGISTER B (DASMB)
$ YFF926		RESERVED
$ YFF928	DASM5	DASM5 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF92A		REGISTER A (DASMA)
$ YFF92C		REGISTER B (DASMB)
$ YFF92E		RESERVED
$ YFF930	DASM6	DASM6 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF932		REGISTER A (DASMA)
$ YFF934		REGISTER B (DASMB)
$ YFF936		RESERVED
$ YFF938	DASM7	DASM7 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF93A		REGISTER A (DASMA)
$ YFF93C		REGISTER B (DASMB)
$ YFF93E		RESERVED
$ YFF940	DASM8	DASM8 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF942		REGISTER A (DASMA)
$ YFF944		REGISTER B (DASMB)
$ YFF946		RESERVED
$ YFF948	PWM SM9	PWMSM9 STATUS / INTERRUPT / COUNTROL (PWMSIC)
$ YFF94A		PERIOD (PWMA)
$ YFF94C		PULSE WIDTH (PWMB)
$ YFF94E		COUNTER (PWMC)

Table 67 Address Map for CTM3 (continued)

Address Address	Submodule name	15	8 7	0
$ YFF950	PWM SM10		PWMSM10 STATUS / INTERRUPT / COUNTROL (PWMSIC)	
$ YFF952			PERIOD (PWMA)	
$ YFF954			PULSE WIDTH (PWMB)	
$ YFF956			COUNTER (PWMC)	
$ YFF958	PWM SM11		PWMSM11 STATUS / INTERRUPT / COUNTROL (PWMSIC)	
$ YFF95A			PERIOD (PWMA)	
$ YFF95C			PULSE WIDTH (PWMB)	
$ YFF95E			COUNTER (PWMC)	
$ YFF960	PWM SM12		PWMSM12 STATUS / INTERRUPT / COUNTROL (PWMSIC)	
$ YFF962			PERIOD (PWMA)	
$ YFF964			PULSE WIDTH (PWMB)	
$ YFF966			COUNTER (PWMC)	
$ YFF968	PWM SM13		PWMSM13 STATUS / INTERRUPT / COUNTROL (PWMSIC)	
$ YFF96A			PERIOD (PWMA)	
$ YFF96C			PULSE WIDTH (PWMB)	
$ YFF96E			COUNTER (PWMC)	
$ YFF970			DASM14 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)	

$ YFF972	DASM14	REGISTER A (DASMA)
$ YFF974		REGISTER B (DASMB)
$ YFF976		RESERVED
$ YFF978		DASM15 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF97A	DASM15	REGISTER A (DASMA)
$ YFF97C		REGISTER B (DASMB)
$ YFF97E		RESERVED
$ YFF980		DASM16 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF982	DASM16	REGISTER A (DASMA)
$ YFF984		REGISTER B (DASMB)
$ YFF986		RESERVED
$ YFF988		DASM17 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF98A	DASM17	REGISTER A (DASMA)
$ YFF98C		REGISTER B (DASMB)
$ YFF98E		RESERVED
$ YFF990		DASM18 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)
$ YFF992	DASM18	REGISTER A (DASMA)
$ YFF994		REGISTER B (DASMB)
$ YFF996		RESERVED

Table 67 Address Map for CTM3 (continued)

Address Address	Submodule name	15	8 7	0
$ YFF998		MCSM19 STATUS / INTERRUPT / CONTROL REGISTER (MCSMSIC)		
$ YFF99A	MCSM19	COUNTER (MCSMCNT)		
$ YFF99C		MODULUS LATCH (MCSMML)		
$ YFF99E		RESERVED		
$ YFF9A0		DASM20 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)		
$ YFF9A2	DASM20	REGISTER A (DASMA)		
$ YFF9A4		REGISTER B (DASMB)		
$ YFF9A6		RESERVED		
$ YFF9A8		MCSM21 STATUS / INTERRUPT / CONTROL REGISTER (MCSMSIC)		
$ YFF9AA	MCSM21	COUNTER (MCSMCNT)		
$ YFF9AC		MODULUS LATCH (MCSMML)		
$ YFF9AE		RESERVED		
$ YFF9B0		DASM22 STATUS / INTERRUPT / CONTROL REGISTER (DASMSIC)		
$ YFF9B2	DASM22	REGISTER A (DASMA)		
$ YFF9B4		REGISTER B (DASMB)		
$ YFF9B6		RESERVED		
$ YFF9B8		MCSM23 STATUS / INTERRUPT / CONTROL REGISTER (MCSMSIC)		
$ YFF9BA	MCSM23	COUNTER (MCSMCNT)		
$ YFF9BC		MODULUS LATCH (MCSMML)		
$ YFF9BE- $ YFF9FE		RESERVED		

Note: Y = M111. Where M represents the logical state of SLIMCR's MODMAP. MC68HC16
On the device, M must be 1. If M is cleared, reset
You will not be able to access the IMB module until it occurs. M is once after reset
Can only be written.

6.3 Time-based bus system

The time-based bus system consists of four 16-bit buses, TBB1, TBB2, TBB3, and TBB4.
I am. The time-based bus is pulled from the counter submodule and is an action
Used by submodules. Each submodule has two timebase buses (TBBA, TBBB).
Access is possible (see Figure 22). These counters dodge one of the two timebase buses
You can eve and the channel references one of the two timebase buses. Within each CTM3 submodule
The control bits allow the software to connect the submodule to the desired timebase bus.
I can.

Page 156

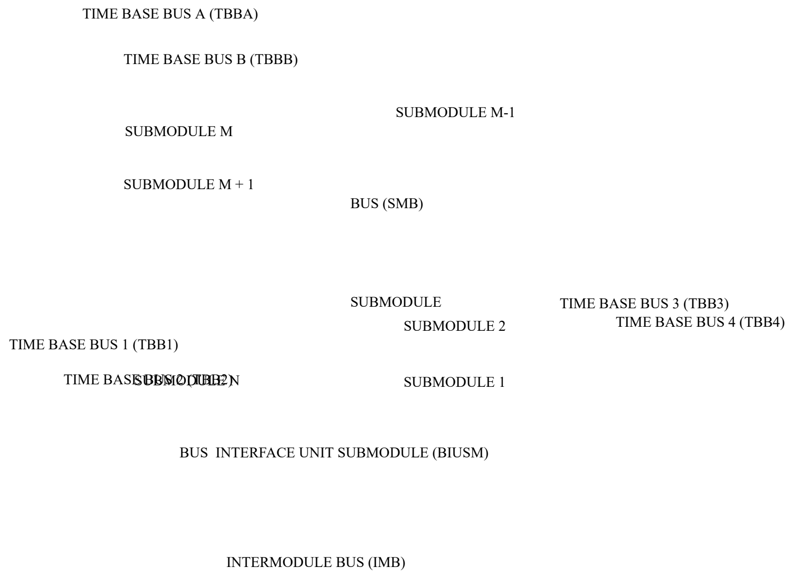


Figure 22 Time-based bus configuration

MC68HC (9) 16Y5 / 916Y6 CTM3 implementation has two time-based buses glowing
It is a bal and has access to all submodules. Table 68 in which each submodule CTM3
Indicates whether the time-based bus can be used.

Table 68 CTM3 Time Base Bus Allocation

Submodule name	Global / local time base Bus allocation	
	Time-based bus A	Time-based bass B
MCSM 2	TBB4	TBB3
DASM 3	TBB4	TBB3
DASM 4	TBB4	TBB3
DASM 5	TBB4	TBB3
DASM 6	TBB4	TBB3
DASM 7	TBB4	TBB3

DASM 8	TBB4	TBB3
DASM 14	TBB4	TBB2
DASM 15	TBB4	TBB2
DASM 16	TBB4	TBB2

Page 157

Table 68 CTM3 Time Base Bus Allocation (continued)

Submodule name	Global / local time base Bus allocation	
	Time-based bus A	Time-based bass B
DASM 17	TBB4	TBB2
DASM 18	TBB4	TBB2
MCSM 19	TBB4	TBB2
DASM 20	TBB4	TBB2
MCSM 21	TBB4	TBB2
DASM 22	TBB1	TBB2
MCSM 23	TBB1	TBB2

The time-based bus transfers timing information from the counter to the action submodule. Used for Each CTM3 submodule becomes a clock source module (one or Drive two time-based buses) or become an action submodule (tie) You can read timing information from the base bus and react to it).

The time-based bus is a pre-charge / discharge type bus that can be wired OR, and has multiple counts. No damage to the hardware will occur if the data drives the same bus at the same time.

6.4 Clock source for counter submodule

The clock source for each counter submodule is from one of the seven clock sources. Selected by the software. Six of them are prescaler taps from on-chip oscillators. vinegar. The maximum frequency that can be supplied to the counter is the frequency obtained by dividing the MCU system clock by two. The other 5 Four of the taps are divided by 2, 4, 16, and 32 from the system clock cycle. vinegar. Another input clock to the counter is the software-defined MCU clock 64, 128, 256, and also Is divided by 512. Another pre that divides the MCU clock by 3, 6, 12, 24, 48, 96, 192, 384, 768 There is also a scaler option. The seventh clock source that can be selected is the rising edge of the input signal. An external pin that can be triggered at the edge or falling edge. If you use an external input, the counter will be micro You can use frequencies that are based on something other than a controller oscillator. The external clock source is up to the event It can also be used for pulse counting.

Note: Each counter shares the same clock pin (CTM2C). Modular counter
Load pins are shared with CTD18, CTD20 and CTD22 pins.

6.5 Bus Interface Unit Submodule (BIUSM)

BIUSM connects the SMB to the IMB so that the CTM3 submodule can communicate with the bus master (CPU). increase. BIUSM also communicates with the interrupt request from the CTM submodule to the IMB, and interrupts the interrupt request. Transfers the interrupt level, priority bits, and vector number to the CPU during the cycle.

6.5.1 BIUSM Registers

BIUSM includes module configuration registers, time-based bus registers, Contains test registers. The BIUSM register block is always the first 4 in the CTM register area. It is located at a register location and you cannot move the register position within the CTM3 configuration. Reading unused bits and reserved address locations from software always returns "0" increase. Writes to unused bits and reserved address locations are ignored.

BIUMCR—BIU module configuration register $ YFF900

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	FRZ	NOT USED	VECT [7: 6]			IARB [2: 0]		NOT USED		TBS1		NOT USED			TBRS0
RESET:															
0	0		1	1	0	0	0			0					0

STOP—Stop Enable

When STOP = 1, the SMB FREEZE signal is output regardless of the FREEZE signal status on the IMB, and CTM3 Operation stops completely. BIUSM will continue to operate and the CPU will access the registers of the submodule. Allows you to set. The SMB FREEZE signal is either reset or the CPU is (through the IMB). It remains active until the STOP bit is negated.

0 = CTM works.
1 = CTM operation stops.

FRZ—Freeze Enable

When FREEZE = 1, the FREEZE signal on the SMB is output when the FREEZE signal on the IMB is active. Then, CTM3 will stop working completely.

Note: Some submodules have this FREEZE signal due to the internal enable bit. There is something that works.

BIUSM continues to operate, allowing the CPU to access the submodule's registers. The SMB FREEZE signal has the FRZ bit cleared or the IMB FREEZE signal negated. Stays active until.

0 = Ignore the IMB FREEZE signal.
1 = Stop the CTM3 submodule when the FREEZE signal is output to the IMB.

See “2.3.1 Condition Code Registers” on page 43 .

VECT [7: 6] — Interrupt-based vector number field

Interrupt-based vector number For the bit field, select the interrupt-based vector number of CTM3. increase. Of the 8 bits required to define the vector number, the lower 6 bits are hard for each submodule. It is programmed in the ware and the remaining 2 bits are specified in VECT [7: 6]. By this designation, CTM3 Select one of the four positions in the interrupt vector table as the vector (see Table 69).

Table 69 Interrupt-based vector number bit field

VECT7	VECT6	Base vector number
0	0	$ 00
0	1	$ 40
1	0	$ 80
1	1	$ C0

IARB [2: 0] — Interrupt priority specification field

This bit field and the IARB3 bits in each submodule have 15 priority identifiers. Can be specified to control the priority of multiple interrupt requests that occur at the same priority level on the IMB. I can do it.

The IARB field is reset to the default of zero, interrupt arbitration acknowledge cycle
Prevents the module from controlling priority during IACK. IMB during the IACK cycle
If there is no priority control for, "Streamline Low Power Integration.
Spurious interrupt vector is generated from "module" , interrupt priority number is not initialized
Notify the system. The seven levels of interrupts are the basic order of interrupt priority.
I am. The 4-bit interrupt priority number is the second order, with a maximum of 15 requests for each basic order.
make it possible. During the IACK cycle, the interrupt request with the highest priority number acquires service.
(% 1111 is the highest and% 0001 is the lowest). CTM3 has its own IARB3 for each submodule
Two priorities by providing bits (which can be set or cleared from software)
You can use it. After IARB [2: 0] is assigned in BIUSM, these are all CTMs.
It can be used in interrupt requests. Therefore, the interrupt of the CTM3 submodule is the same interrupt.
It can be separated from interrupt requests from other modules at the only level. IARB [2: 0] is reset
It will be rear.

TBRS1, TBRS0—Time-based register bus selection bits

Use these bits to access when reading the timebase register (BIUTBR)
Specifies the time base bus. See Table 70 .

Table 70 Time-based register bus selection bits

TBRS1	TBRS0	Time-based bass
0	0	TBB1 (TBBA)
0	1	TBB2 (TBBB)
1	0	TBB3 (TBBB)
1	1	TBB4 (TBBA)

BIUTEST—BIUSM configuration register $ YFF902

BIUTEST is used for shipping testing of CTM3. Access to BIUTEST is in test mode on the MCU
Must be executed when is.

BIUTBR—BIUSM time base register $ YFF904

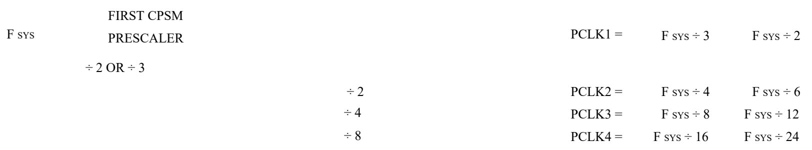
15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MOST SIGNIFICANT BYTE (MSB)								LEAST SIGNIFICANT BYTE (LSB)							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIUTBR is a read-only register used to read a value on the timebase bus.
vinegar. The time-based bus to access is specified by TBRS1 and TBRS0 in the BIUMCR register.
Writes to BIUTBR are ignored except in certain test modes.

6.6 Counter Prescaler Submodule (CPSM)

The Counter Prescaler Module (CPSM) is a programmable divider system, an MCU system.
Divides the main clock (f SYS) of the clock and provides 6 clock options (PCLKx) to the CTM counter.
To do. Five of these frequencies are obtained from the fixed frequency divider. The division ratio of the last clock frequency is
You can select one of the four division ratios in the software.

CPSM is built into BIUSM . Figure 23 shows a block diagram of the CPSM.


F SYS
FIRST CPSM
PRESCALER
÷ 2 OR ÷ 3
÷ 2
÷ 4
÷ 8
PCLK1 = F SYS ÷ 3 F SYS ÷ 2
PCLK2 = F SYS ÷ 4 F SYS ÷ 6
PCLK3 = F SYS ÷ 8 F SYS ÷ 12
PCLK4 = F SYS ÷ 16 F SYS ÷ 24

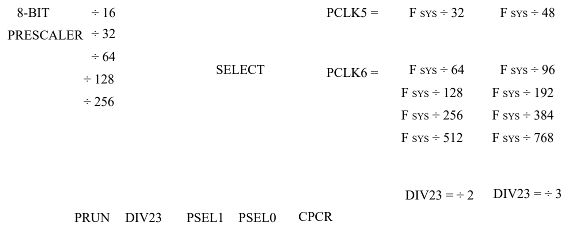


Figure 23 Block diagram of CPSM

6.6.1 CPSM registers

CPSM has built-in control and test registers. Unused bits and forecast
When the contracted address is read from the software, "0" is returned. Unused bits and reserved
Writes to finished addresses are ignored.

CPCR—CPSM control register $ YFF908

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
					NOT USED							PRUN	DIV23	PSEL [1: 0]	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PRUN— Prescaler activation bit

The PRUN bit is a read / write control bit. With this bit, software
You can turn the prescaler counter on and off from a. With this bit,
You can synchronize the counters in various CTM3 submodules. This bit is reset
It will be cleared by

0 = Reset the prescaler divider to stop the counter.
1 = Specify to start the prescaler.

DIV23— 2 divisions / 3 divisions

The DIV23 bit is a read / write control bit. Using this bit, the first stage Pris
Select the division ratio of the Kera counter. This bit can be changed at any time by software
vinegar. This bit is cleared by reset.

0 = Specify the first stage prescaler as 2 divisions.
1 = Specify the first stage prescaler as 3 divisions.

Page 161

PSEL [1: 0] — Prescaler division ratio selection field

Use this bit field to select the output signal (PCLK6) of the programmable prescaler.
I will choose. See Table 71 .

Table 71 Prescaler division ratio selection field

Prescaler control register bit				Prescaler division ratio					
PRUN	DIV23	PSEL1	PSEL0	PCLK1	PCLK2	PCLK3	PCLK4	PCLK5	PCLK6
0	X	X	X	0	0	0	0	0	0
1	0	0	0	2	Four	8	16 16	32	64
1	0	0	1	2	Four	8	16 16	32	128 128
1	0	1	0	2	Four	8	16 16	32	256
1	0	1	1	2	Four	8	16 16	32	512
1	1	0	0	3	6	12	twenty four	48	96
1	1	0	1	3	6	12	twenty four	48	192
1	1	1	0	3	6	12	twenty four	48	384
1	1	1	1	3	6	12	twenty four	48	768

CPTR—CPSM test register $ YFF90A

This test register is used for CTM3 shipping testing. Access to the CPTR is done by the MCU
Must be done while in mode.

6.7 Modular Counter Submodule (MCSM)

The modulus counter submodule (MCSM) is a 16-bit modulus latch, 16-bit.
Loadable up counter, counter loading logic, clock selector, time
It has a built-in base bus driver and an interrupt interface. The modulus register is
Gives you the flexibility to recycle the counter at counts other than 64K clock cycles. Modular
The state of the register is transferred to the counter in the following three states:

- When overflow occurs
- When a predetermined edge is detected on the external input pin (CTD18 / CTD20 / CTD22)
- When the program writes to the modulus register. This value loads into the counter immediately
  It is also possible.

The software writes a value to the modulus register and later cows it with one of the first two criteria.
You can also load it into the data.

Four MCSMs are built into CTM3 . Figure 24 shows a block diagram of the MCSM.

Note: A CPSM clock signal must be applied for MCSM to count.
When released from the reset, the CPSM prescaler starts working (software)
MCSM cows internal or external events until (when the PRUN bit is set)
I will not do it. This synchronizes all counters in the CTM3 submodule
Can be

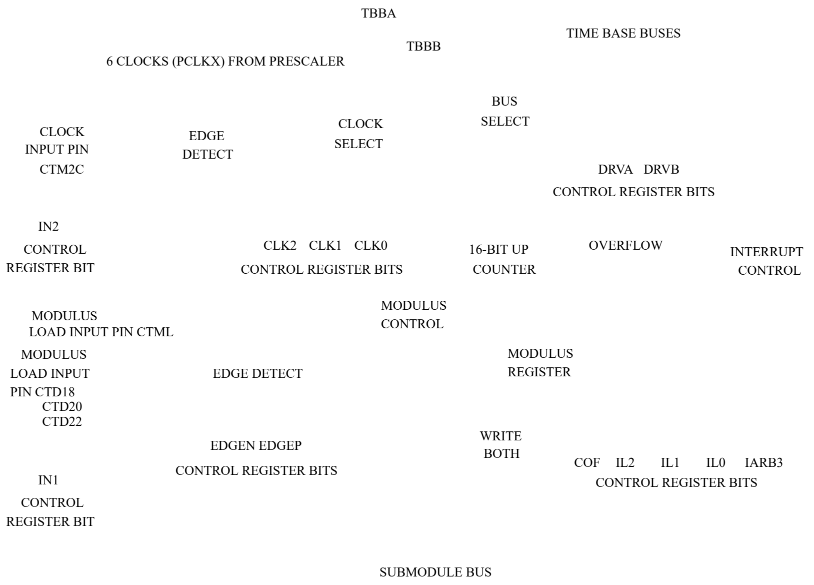


Figure 24 MCSM block diagram

6.7.1 MCSM register

MCSM has status / interrupt / control registers, counters, and modulus latches.
increase. When reading unused bits and reserved addresses from the software, all "0" s are returned.
vinegar. Writes to unused bits and reserved addresses are ignored. CTM3 has 4 MCSMs
The MCSM has its own set of registers.

MCSMSIC—MCSM status / interrupt / control register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
COF		IL [2: 0]		IARB3	NOT USED	DRV [A: B]		IN2	IN1	EDGEN	EDGEP	NOT USED		CLK [2: 0]	
RESET:															
0	0	0	0	0	0	0	0	U	U	0	0	0	0	0	0

Table 72 shows the address locations of the four MCSMSICs.

Table 72 MCSMSIC address location

MCSMSIC register	Address Address
MCSM2	$ YFF910
MCSM19	$ YFF998
MCSM21	$ YFF9A8
MCSM23	$ YFF9B8

Page 163

COF—Counter overflow flag

This status flag bit indicates whether a counter overflow has occurred. The MCSM counter overflow occurs when it changes from $ FFFF to $ xxxx. Where $ xxxx is The value of the modulus latch. If the IL field is non-zero, then the COF bit is set At that time, an interrupt request is generated.

0 = No counter overflow.
1 = With counter overflow.

This flag bit is set by the hardware and is software or system reset. Will only be cleared by. To clear this flag, first read this bit ("1" is (Read), then write "0" to this bit.

Note: This is only possible if there is no event to set the flag between read and write. The flag clear method works. Event to set a flag between read and write This COF flag will not be cleared if an error occurs.

IL [2: 0] — Interrupt level field

This bit field is a read / write control bit for interrupt requests issued by the MCSM. Select a priority level. These bits are always readable / writable and can be reset by reset It will be cleared. See Table 73 .

Table 73 Interrupt level bit fields

IL2	IL1	IL0	Priority level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

IARB3 — Interrupt arbitration bit 3

This bit feels like IARB [2: 0] in the BIUSM module configuration register. It works in combination with the device. Each module that issues an interrupt request on the IMB is placed in the arbitration field. Must have a neat value. Multiple modules request interrupts at the same time with the same priority If so, this interrupt arbitration identification number is used to control the priority of the IMB. The IARB3 bit is cleared on reset. For more information on IARB [2: 0], see “6.5.1” on page 157. See BIUSM Registers .

DRV [A: B] — Time-based bus-driven bit field

This readable / writable bit field includes FCSM and timebase buses A and B. Controls the connection of. These bits are cleared by reset. See Table 74 .

Table 74 Time-based bus-driven bit fields

DRVA	DRVB	Bus selected
0	0	Neither time base bus A nor B is driven

0	1	Time base bus B is driven
1	0	Time base bus A is driven
1	1	Both time base bus A and bus B are driven

WARNING: Do not drive both timebases at the same time.
For example, prohibiting MCSM21 and MCSM19 from driving on the same timebase (TBB2)

Page 164

Is to stop. If you drive to the same timebase, that value will be destroyed,
There is no damage to the hardware.

IN2—Clock input pin status bit

This read-only status bit indicates the logical state of clock input pin CTM2C. this Writing a “1” or “0” to a bit is ignored. This bit is affected by the reset plug.

IN1—Modular load input pin status bit

This read-only status bit is the logical state of the modulus load input pin CTS16B / CTD20. Indicates. Writing “1” or “0” to this bit is ignored. This bit is reset Not affected by

EDGEN, EDGEP—Modular load edge detection selection bit

This readable / writable control bit has the effect detected by the modulus load pin CTS16B. Select the See Table 75 .

Table 75 Modular load edge detection selection bits

EDGEN	EDGEP	Edge detected by modulus load pin / IN1
0	0	None None
0	1	Positive edge only
1	0	Negative edge only
1	1	Positive and negative edge

CLK [2: 0] — Counter clock selection field

This readable / writable control bit is one of six internal clock signals (PCLKx). Lock selection, or two external conditions on the input pin (ie rising / falling edges) Make a selection. The maximum frequency of the external clock signal is f SYS / 4 . See Table 76.

Table 76 Counter clock selection fields

CLK2	CLK1	CLK0	Free running counter clock source
0	0	0	Prescaler output 1 (/2 or / 3)
0	0	1	Prescaler output 2 (/4 or 6)
0	1	0	Prescaler output 3 (/8 or / 12)
0	1	1	Prescaler output 4 (/16 or / 24)
1	0	0	Prescaler output 5 (/ 32 or / 48)
1	0	1	Prescaler output 6 (/ 64 or / 768)
1	1	0	CTM2C pin input, negative edge
1	1	1	CTM2C pin input, positive edge

MCSMCNT—MCSM counter register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			MSB								LSB				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCSM counter registers are readable / writable registers. Table 77 shows four MCSMCNT ads Indicates a less location.

Table 77 MCSMCNT Address Location

MCSMCNT register	Address Address
MCSM2	$ YFF912
MCSM19	$ YFF99A
MCSM21	$ YFF9AA
MCSM23	$ YFF9BA

MCSMML—MCSM Modular Latch

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			MSB								LSB				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The MCSM modulus latch register is a read / write register. Table 78 shows four MCSMML Indicates the address and location of.

Table 78 MCSMML Address Locations

MCSMML register	Address Address
MCSM2	$ YFF914
MCSM19	$ YFF99C
MCSM21	$ YFF9AC
MCSM23	$ YFF9BC

6.8 Double Action Submodule (DASM)

Double Action Submodule (DASM) can occur automatically without software intervention It has two 16-bit input captures or two 16-bit output compare functions. Input edge The detector can be programmed to generate the capture function at the desired edge. Output flip flo Is set at one of the output compare signals and reset at the other one.

Depending on the input capture and output compare modes, the software takes advantage of optional interrupts. I can do it. The software either captures or outputs either of the two input time-based buses. Decide whether to use it for the pair.

With six modes of operation, the software uses DASM's input capture and output compare capabilities. Pulse width measurement, time width measurement, single pulse generation, continuous pulse generation, and standard input cap You will be able to perform cha and output compare. DASM can also act as a single I / O pin I can do it. The DASM operating mode is the mode select bit field of the DASMSIC register (MODE [3: 0]). Is determined by. Table 79 shows the different DASM operating modes.

Table 79 DASM operating modes

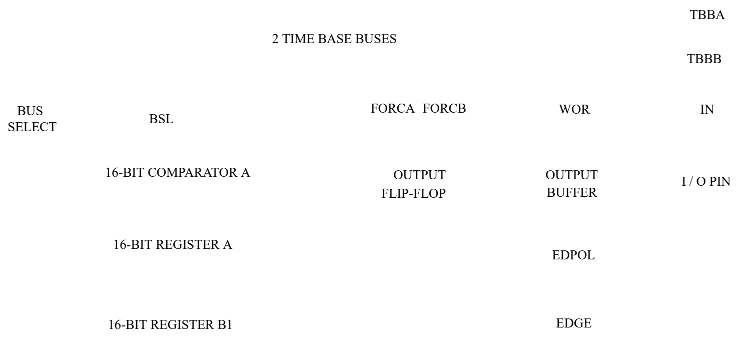
MODE [3: 0]	mode	Mode description
0000	DIS	Disabled-Input pin is high impedance; IN gives state of input pin
0001	IPWM	Input pulse width measurement-Capture on leading edge and the trailing edge of an input pulse
0010	IPM	Input period measurement-Capture two consecutive rising / falling edges
0011	I c	Input capture-Capture when the designated edge is detected
0100	OCB	Output compare, flag set on B compare-Generate leading and trailing edges of an output pulse and set the flag
0101	OCAB	Output compare, flag set on A and B compare-Generate leading and trailing edges of an output pulse and set the flag
1xxx	OPWM	Output pulse width modulation-Generate continuous PWM output with 7, 9, 11, 12, 13, 14, 15, or 16 bits of resolution

DASM has two timing channels (A and B), an output flip-flop, an input edge detector, and so on.
It consists of several control circuits and interrupt circuits. All control bits and status bits are
It is located in the DASMSIC register.
Channel A consists of one 16-bit data register and one 16-bit comparator.
is. Channel B also has one 16-bit data register and one 16-bit controller from the user.
It looks like it is made up of paretas, but internally channel B has two data registers (B1).
And B2). The mode of operation determines which registers are accessed by the software.
See Table 80 .

Table 80 Channel B data register access

mode	Data register
Input Capture (IPWM, IPM, IC)	Registers A and B2 are used to hold the captured values. In these modes, the B1 register is used as a temporary latch for channel B
Output Compare Compare (OCB, OCAB)	Registers A and B2 are used to define the output pulse. Registers B1 is not used in these modes
Output Pulse Width Modulation Mode (OPWM)	Registers A and B1 are used as primary registers and hidden register B2 is used as a double buffer for channel B

The minimum pulse (measured or generated) is 1 data because the register contents are always automatically transferred at the correct time.
Im base bass count. A and B data registers are always readable / writable registers
It is accessible via the CTM3 submodule bus.
CTM3 has 13 built-in DASMs . Figure 25 shows a block diagram of DASM.



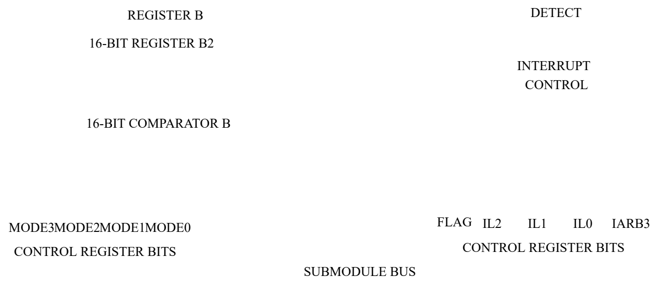


Figure 25 DASM block diagram

6.8.1 DASM registers

DASM has one status / interrupt / control register and two data registers (A and two). And B). All zeros when reading unused bits and reserved addresses from the software Is returned. Writes to unused bits and reserved addresses are ignored. 13 for CTM3 There are DASMs, and each DASM has its own set of registers.

DASMSIC—DASM status / interrupt / control register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
FLAG		IL [2: 0]		IARB3	NOT USED	WOR	BSL	IN	FORCA	FORCB	EDPOL		MODE [3: 0]		
RESET:															
0	0	0	0	0	0	0	0	U	0	0	0	0	0	0	0

Table 81 shows the address locations of the 13 DASMSICs.

Table 81 DASMSIC address location

DASMSIC register	Address Address
DASM3	$ YFF918
DASM4	$ YFF920
DASM5	$ YFF928
DASM6	$ YFF930
DASM7	$ YFF938
DASM8	$ YFF940
DASM14	$ YFF970
DASM15	$ YFF978
DASM16	$ YFF980
DASM17	$ YFF988
DASM18	$ YFF990
DASM20	$ YFF9A0
DASM22	$ YFF9B0

FLAG—Flag status bit

This status bit is where an input capture event or an output compare event occurs.
Indicates whether or not it was. If the IL field is other than “0” and the FLAG bit is set, an interrupt request will be issued.
Occurs.

0 = No input capture event or output compare event has occurred.
1 = An input capture event or output compare event has occurred.

Table 82 shows the status of the flag status bits in different modes.

Table 82 DASM mode flag status bit status

mode	Flag status bit state
DIS	FLAG bit is reset
IPWM	FLAG bit is set each time there is a capture on channel A
IPM	FLAG bit is set each time there is a capture on channel A, except for the first time
I c	FLAG bit is set each time there is a capture on channel A
OCB (when MODE 0 = 0)	FLAG bit is set each time there is a successful comparison on channel B
OCAB (when MODE 0 = 1)	FLAG bit is set each time there is a successful comparison on either channel A or B
OPWM	FLAG bit is set each time there is a successful comparison on channel A

This set of FLAG bits is done by hardware and cleared by software or system.
It is done by reset. The first thing to do when clearing the FLAG bit from software
Read the bit of "1" and write "0" to this bit, or select DIS mode.

Note: This flag clear method sets a flag between read and write operations.
It works only if there is no venting. Between read and write operations
When an event that sets the FLAG bit occurs, the FLAG bit is not cleared.

IL [2: 0] — Interrupt level field

This bit field is a read / write control bit for interrupt requests generated by DASM.
Used when selecting a priority level. These bits are always readable / writable and can be read / written at any time.
It will be cleared by reset. See Table 83 .

Table 83 Interrupt Level Bit Fields

IL2	IL1	IL0	Priority level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

IARB3 — Interrupt arbitration bit 3

This bit is the IARB [2: 0] fee in the BIUSM module configuration register.
Works in combination with Ludo. Each module that generates an interrupt request on the IMB is in the arbitration field.
Must have a unique value. Multiple modules need interrupts at the same time with the same priority
If requested, the interrupt arbitration identification number is used to control the priority for the IMB.
The IARB3 bit is cleared by reset. For more information on IARB [2: 0], see page 157.
See 6.5.1 BIUSM Registers .

WOR— Wired OR Bit

The WOR bit is not used in the DIS, IPWM, IPM, and IC modes. Before reading this bit
Returns the value written once.

In OCB, OCAB, and OPWM modes, the WOR bit is used to open and drain the output buffer.
Specify whether to make it a totem pole or a totem pole.

0 = Set the output buffer to a totem pole.
1 = Make the output buffer open / drain.

The WOR bit is cleared by reset.

BSL—Bus selection bit
Use this control bit to select the timebase bus to connect to DASM.
0 = DASM is connected to timebase bus A.
1 = DASM is connected to time base bus B.

IN—Input pin status bit

In DIS, IPWM, IPM, and IC modes, this read-only status bit is the logic of the input pin. The level is displayed.

In each mode of OCB, OCAB, and OPWM, when this bit is read, the output flip after the polarity is determined by EDPOL. A value latched on the desktop is returned.

Writes to this bit are ignored.

FORCA — Mandatory control bit A

In OCB, OCAB, and OPWM modes, this FORCA bit is used by the software and is a channel. Make the output flip-flop behave as if a match occurred at A (but the FLAG bit is set). Will not be). Writing "1" to the FORCA bit sets the output flip-flop. "0" Writes are ignored.

In DIS, IPWM, IPM, and IC modes, the FORCA bit is not used and writes to this bit It will be ignored. The FORCA bit is cleared by reset and always returns "0" when read.

Note: Writing “1” to both the FORCA and FORCB bits at the same time will flip the output. The flop will be reset.

Page 170

FORCB — Mandatory control bit B

In OCB, OCAB, and OPWM modes, this FORCB bit is used by the software and is a channel. Makes the output flip-flop behave as if a match occurred at B (but the FLAG bit is set). Will not be). Writing "1" to the FORCB bit sets the output flip-flop. “0” Writes are ignored.

In DIS, IPWM, IPM, and IC modes, the FORCB bit is not used and writes to this bit It will be ignored.

The FORCB bit is cleared by reset and always returns "0" when read.

Note: Writing “1” to both the FORCA and FORCB bits at the same time will flip the output. The flop will be reset.

EDPOL—Edge detection polarity bit

This bit is not used in DIS mode. When read, the last written value is returned.

In IPWM mode, this bit is used to enable effective capture for channel A and channel B. Select the

0 = Channel A captures the rising edge.
Channel B captures the falling edge.
1 = Channel A captures the falling edge.
Channel B captures the rising edge.

In IPM and IC modes, the EDPOL bit is used to capture valid input for channel A. Select an edge.

0 = Channel A captures the rising edge.
1 = Channel A captures the falling edge.

In OCB, OCAB, and OPWM modes, the EDPOL bit is used to select the voltage level on the output pin.

0 = The logic level of the output flip-flop is output to the output pin. Compare on channel A When it occurs, set the output pin. Reset the output pin when compare occurs on channel B To

1 = The inversion logic level of the output flip-flop is output to the output pin. Competition on channel A When a occurs, the output pin is reset. Output pin when compare occurs on channel B Set.

The EDPOL bit is cleared on reset.

MODE [3: 0] — Mode selection field

This bit field selects the operating mode of the DASM channel. See Table 84 .

Note: To prevent the occurrence of invalid interrupts, use DASM interrupts when changing the operation mode. We recommend disabling it.

Table 84 DASM mode selection fields

DASM control register bit				Resolution bit number	Time to ignore Base bit	DASM operating mode
MOD3	MOD2	MOD1	MOD0			
0	0	0	0	————	————	DIS-Disabled
0	0	0	1	16 16	————	IPWM-Input pulse width measurement
0	0	1	0	16 16	————	IPM-Input measurement period
0	0	1	1	16 16	————	IC-Input capture

0	1	0	0	16 16	————	OCB-Output compare, flag on B compare
0	1	0	1	16 16	————	OCAB-Output compare, flag on A and B compare
0	1	1	0	————	————	Not used
0	1	1	1	————	————	Not used
1	0	0	0	16 16	————	OPWM-Output pulse with modulation
1	0	0	1	15	15	OPWM-Output pulse with modulation

Page 171

Table 84 DASM Mode Selection Fields (continued)

DASM control register bit				Resolution bit number	Time to ignore Base bit	DASM operating mode
MOD3	MOD2	MOD1	MOD0			
1	0	1	0	14	15-14	OPWM-Output pulse with modulation
1	0	1	1	13	15-13	OPWM-Output pulse with modulation
1	1	0	0	12	15-12-12	OPWM-Output pulse with modulation
1	1	0	1	11 11	15-11	OPWM-Output pulse with modulation
1	1	1	0	9	15-9	OPWM-Output pulse with modulation
1	1	1	1	7	15-7	OPWM-Output pulse with modulation

The mode selection field is cleared by reset.

DASMA—DASM data register A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
				MSB							LSB				
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

DASMA is the data register for channel A. Table 85 shows the addresses of 13 DASMAs. Indicates an application. Table 86 also shows how to use DASMA in different modes of operation.

Table 85 DASMA Address Locations

DASMA register	Address Address
DASM3	$ YFF91A
DASM4	$ YFF922
DASM5	$ YFF92A
DASM6	$ YFF932
DASM7	$ YFF93A
DASM8	$ YFF942
DASM14	$ YFF972
DASM15	$ YFF97A
DASM16	$ YFF982
DASM17	$ YFF98A
DASM18	$ YFF992
DASM20	$ YFF9A2
DASM22	$ YFF9B2

Table 86 DASMA behavior

mode	DASMA operation
DIS	You can access DASMA to select the following modes:
IPWM	DASMA holds the capture value corresponding to the trailing edge of the pulse to be measured.
IPM	The character corresponding to the fixed edge (rising edge or falling edge) detected immediately before. The petit value is kept in DASMA.
I c	The character corresponding to the fixed edge (rising edge or falling edge) detected immediately before. The petit value is kept in DASMA.
OCB	The value corresponding to the leading edge of the generated pulse is loaded into DASMA. Also, OCB In mode and OCAB mode, writing to DASMA will result in the following compare: Enables the comparator corresponding to channel A.
OCAB	The value corresponding to the leading edge of the generated pulse is loaded into DASMA. Also, OCB In mode and OCAB mode, writing to DASMA will result in the following compare: Enables the comparator corresponding to channel A.
OPWM	The value corresponding to the front edge of the generated PWM pulse is loaded into DASMA.

DASMB—DASM data register B

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			MSB								LSB				
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

DASMB is the data register for channel B. Table 87 shows the addresses of 13 DASMBs. Indicates an application. Table 88 also shows how to use DASMB in various operating modes. Selected Depending on the mode of operation, software access is to register B1 or register B2. It will be done.

Table 87 Address location of DASMB

DASMB register	Address Address
DASM3	$ YFF91C
DASM4	$ YFF924
DASM5	$ YFF92C
DASM6	$ YFF934
DASM7	$ YFF93C
DASM8	$ YFF944
DASM14	$ YFF974
DASM15	$ YFF97C
DASM16	$ YFF984
DASM17	$ YFF98C
DASM18	$ YFF994
DASM20	$ YFF9A4
DASM22	$ YFF9B4

Table 88 Behavior of DASMB

mode	Behavior of DASMB
DIS	You can access DASMB to select the following modes: In this mode, in contrast to OPWM mode Register B1 is accessed to provide the value to be used. Unused register B2 is hidden and read It cannot be put out. When writing to register B1, the same value is written to register B1 and register B2. increase.
IPWM	The capture value corresponding to the front edge of the pulse to be measured is retained in DASMB. In this mode, cash register Star B2 is accessed. Buffer register B1 is hidden and inaccessible.
IPM	The capture value corresponding to the edge of the time width (rising edge or falling edge) detected immediately before is It is retained in DASMB. Register B2 is accessed in this mode. Buffer register B1 is hidden It has been and cannot be accessed.
I c	The capture value corresponding to the edge of the time width (rising edge or falling edge) detected immediately before is It is retained in DASMB. Register B2 is accessed in this mode. Buffer register B1 is hidden It has been and cannot be accessed.
OCB	The value corresponding to the trailing edge of the generated pulse is loaded into DASMB. Also, OCB mode and OCAB mode When writing to DASMB, the comparator corresponding to channel B will wait until the next match occurs. It will be navel. In this mode, register B2 is accessed. Buffer register B1 is hidden It is inaccessible.
OCAB	The value corresponding to the trailing edge of the generated pulse is loaded into DASMB. Also, OCB mode and OCAB mode When writing to DASMB, the comparator corresponding to channel B waits until the next match occurs. It is enabled. In this mode, register B2 is accessed. Buffer register B1 is hidden It has been and cannot be accessed.
OPWM	The value corresponding to the trailing edge of the generated PWM pulse is loaded into DASMB. Registers in this mode B1 is accessed. Buffer register B2 is hidden and inaccessible.

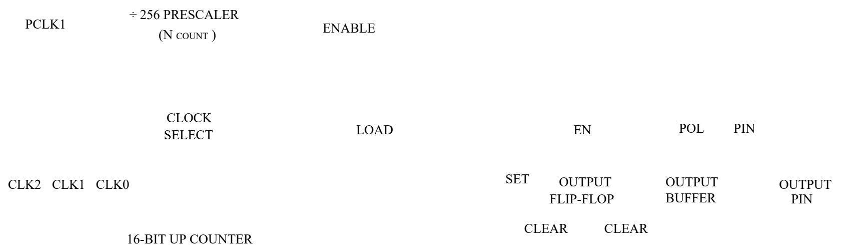
6.9 Pulse Width Modulation Submodule (PWMSM)

PWMSM emits pulse-width modulated signals over a wide frequency range, independent of other CTM output signals. Can be born. The output pulse width can be varied from 0% to 100% with 16-bit accuracy. minimum The pulse width can be up to twice the MCU system clock period (for example, a minimum of 16.78MHz clock). Pulse width 119ns, minimum pulse width 83.3ns with a clock of 24.00MHz).

PWMSM consists of the following functions.

- ・Output flip-flop with specifiable polarity
- ・Clock prescaler and selection circuit
- ・16-bit up counter
- -Two registers that hold the current pulse width and the next pulse width
- -Two registers that hold the current pulse period and the next pulse period
- ・Pulse width comparator
- ・System status sequencer
- ・Circuit that generates 0 to 100% pulse
- ・Interrupt circuit
- ・Status / Interrupt / Control register
- ・Submodule bus interface

PWMSM has its own time-based counter and does not use the CTM time-based bus. However, PWMSM uses the divided PCLK1 clock generated within CPSM. For more information, See “6.6 Counter Prescaler Submodule (CPSM) ” on page 159 . Figure 26 The block diagram of PWMSM is shown in.



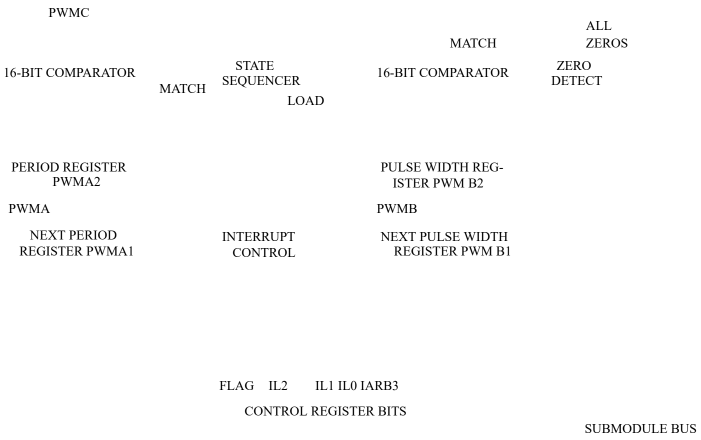


Figure 26 Block diagram of the pulse width modulation submodule

6.9.1 PWM frequency

The relationship between the PWM output frequency (f PWMO) and the MCU system clock frequency (f SYS) is calculated by the following equation.

$$f_{PWMO} = \frac{f_{SYS}}{N_{CLOCK} \cdot N_{COUNTER}}$$

Where N CLOCK is the division ratio of the CPSM clock (2 or 3) and N COUNTER is the division ratio of the PWMSM counter.

Clock selective division ratio (N CLOCK) and bit resolution for a system clock frequency (f SYS)

Given, the minimum PWM output frequency that can be obtained is calculated by the following equation. This expression is less than 16 bits

Applies to resolution.

$$\text{Minimum } f_{PWMO} = \frac{f_{SYS}}{N_{CLOCK} \cdot (2-_{\text{bit resolution} +1} —— 1)}$$

Note: PWMSM is a 16-bit modulus that counts from 1 to the desired preset value.

It has a built-in up counter (see Figure 26). The term "bit resolution" is the same.

Indicates the size of a free running binary counter such as. Worst K

The number of bits is rounded down one number to accommodate the space. For example, coun

Page 175

If the data is preset to count values between 128 and 255, then 7 bits

Has the resolution of Preset to count at any value from 256 to 511

If so, the resolution is 8 bits. The term "resolution" refers to the minimum PWM SM

Used to define the output increment of.

With 16-bit resolution, the maximum possible PWM output frequency is calculated by the following equation.

$$\text{Minimum } f_{PWMO} = \frac{f_{SYS}}{N_{CLOCK} \cdot (2-_{\text{bit resolution} +1})}$$

6.9.2 PWM pulse width

The minimum output pulse width (t PWMIN) and MCU system clock frequency (f SYS) are calculated by the following equations.

$$t_{PWMIN} = \frac{N_{CLOCK}}{f_{SYS}}$$

The maximum output pulse width (t PWMAX) obtained is given by the following equation. This expression is less than 16 bits

Applies to resolution.

$$t_{PWMAX} = N_{CLOCK} \cdot (N_{COUNTRT} -1)$$

f_{SYS}

At 16-bit resolution, the maximum output pulse width (t_{PWMAX}) obtained is given by:

$$t_{PWMAX} = \frac{N_{CLOCK} \cdot (2\text{-}_{\text{bit resolution}} -1)}{f_{SYS}}$$

6.9.3 PWM period register and PWM pulse width register value

The value to be loaded into the PWM period register to obtain a certain period is calculated by the following equation.

$$PWMA = \frac{f_{SYS}}{N_{CLOCK} \cdot f_{PWMO}}$$

The value to be loaded into the PWM pulse width register to obtain a certain pulse width is calculated by the following equation.

$$PWMB = \frac{t_{PWMO}}{t_{PWMIN}} = = \frac{\text{Duty cycle\%}}{100} \cdot PWMA$$

6.9.4 PW[redacted]od and pulse width register value

Achievable usin ÷ 3 options and the clock frequency of 16.78MHz in Table 89 and Table 90.

Page 176

Indicates the pulse width and frequency.

Table 89 ÷ 2 PWM pulse and frequency range (Hz) with option (16.78MHz)

Minimum pulse width	Resolution (number of bits)															
	16 16	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1
0.119 ∝ s (÷ 2)	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K	131K	262K	524K	1049K	2097K	4195K
0.238 ∝ s (÷ 4)	64	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K	131K	262K	524K	1049K	2097K
0.477 ∝ s (÷ 8)	32	64	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K	131K	262K	524K	1049K
0.954 ∝ s (÷ 16)	16 16	32	64	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K	131K	262K	524K
1.91 ∝ s (÷ 32)	8.0 8.0	16 16	32	64	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K	131K	262K
3.81 ∝ s (÷ 64)	4.0 4.0	8.0 8.0	16 16	32	64	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K	131K
7.63 ∝ s (÷ 128)	2.0 2.0	4.0 4.0	8.0 8.0	16 16	32	64	128 128	256	512	1024	2048	4096	8192	16384	32768	65.5K
30.5 ∝ s (÷ 512)	0.5	1.0 1.0	2.0 2.0	4.0 4.0	8.0 8.0	16 16	32	64	128 128	256	512	1024	2048	4096	8192	16384

Table 90 ÷ 3 PWM pulse and frequency range (Hz) with option (16.78MHz)

Minimum pulse width	Resolution (number of bits)															
	16 16	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1
0.179 ∝ s (÷ 3)	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K	87.38K	174.8K	349.5K	699.1K	1398K	2796K
0.358 ∝ s (÷ 6)	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K	87.38K	174.8K	349.5K	699.1K	1398K
0.715 ∝ s (÷ 12)	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K	87.38K	174.8K	349.5K	699.1K
1.431 ∝ s (÷ 24)	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K	87.38K	174.8K	349.5K
2.861 ∝ s (÷ 48)	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K	87.38K	174.8K
5.722 ∝ s (÷ 96)	2.667	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K	87.38K
11.44 ∝ s (÷ 192)	1.333	2.667	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923	21845	43.69K
45.78 ∝ s (÷ 768)	0.333	0.667	1.333	2.667	5.333	10.67	21.33	42.67	85.33	170.7	341.3	682.7	1365	2731	5461	10923

Achievable usin[redacted] ÷ 3 options and the clock frequency of 24.00MHz in Tables 91 and 92.
Indicates the pu frequency.

Table 91 ÷ 2 PWM pulse and frequency range (Hz) with option (24.00MHz)

Minimum pulse width	Resolution (number of bits)															
	16 16	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1
0.083 ∝ s (÷ 2)	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K	187.5K	375K	750K	1.5M	3M	6M
0.166 ∝ s (÷ 4)	91.5	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K	187.5K	375K	750K	1.5M	3M
0.333 ∝ s (÷ 8)	45.77	91.5	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K	187.5K	375K	750K	1.5M
0.666 ∝ s (÷ 16)	22.88	45.77	91.5	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K	187.5K	375K	750K
1.33 ∝ s (÷ 32)	11.44	22.88	45.77	91.5	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K	187.5K	375K
2.66 ∝ s (÷ 64)	5.722	11.44	22.88	45.77	91.5	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K	187.5K
5.33 ∝ s (÷ 128)	2.861	5.722	11.44	22.88	45.77	91.5	183	366	732.4	1464	2929	5859	11718	23437	46.87K	93.75K
21.3 ∝ s (÷ 512)	0.715	1.430	2.861	5.722	11.44	22.88	45.77	91.5	183	366	732.4	1464	2929	5859	11718	23437

Table 92 ÷ 3 PWM pulse and frequency range (Hz) with option (24.00MHz)

Minimum pulse width	Resolution (number of bits)															
	16 16	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1
0.125 ∝ s (÷ 3)	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K	125K	250K	500K	1000K	2000K	4000K
0.250 ∝ s (÷ 6)	61	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K	125K	250K	500K	1000K	2000K
0.500 ∝ s (÷ 12)	30.5	61	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K	125K	250K	500K	1000K
1.000 ∝ s (÷ 24)	15.25	30.5	61	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K	125K	250K	500K
2.000 ∝ s (÷ 48)	7.629	15.25	30.5	61	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K	125K	250K
4.000 ∝ s (÷ 96)	3.814	7.629	15.25	30.5	61	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K	125K
8.000 ∝ s (÷ 192)	1.907	3.814	7.629	15.25	30.5	61	122	244.1	488.2	976.5	1953	3906	7812	15625	31250	62.5K
32.00 ∝ s (÷ 768)	0.476	0.953	1.907	3.814	7.629	15.25	30.5	61	122	244.1	488.2	976.5	1953	3906	7812	15625

6.9.5 PWMSM register

PWMSM has one status / interrupt / control register, periodic register, pulse width register. There is a counter register. Software reads unused bits and reserved addresses Then all "0" will be returned. Writes to unused bits and reserved addresses are ignored vinegar. The CTM3 has five PWMSMs, each with its own set of registers.

PWMSIC—PWM status / interrupt / control register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
COF		IL [2: 0]		IARB3		NOT USED		PIN	NOT USED	LOAD	POL	EN		CLK [2: 0]	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 93 shows the address locations of the five PWMSICs.

Table 93 PWMSIC address location

PWMSIC register	Address Address
PWM9	$ YFF948
PWM10	$ YFF950
PWM11	$ YFF958
PWM12	$ YFF960
PWM13	$ YFF968

FLAG — Cycle exit status

This FLAG bit is set when the PWM output cycle ends.
0 = PWM output cycle has not ended.
1 = PWM output cycle ends.

This FLAG bit is set by the hardware at the end of the PWM output cycle. PWM
Each time it is enabled, the FLAG bit is immediately set in the buffer registers PWMA2 and PWMB2.
Shows that the value has been updated and that a cycle with the new value has started. Also this
The bits are in the user-accessible period register (PWMA1) and pulse width register (PWMB1).

It also shows that you can load values for the next PWM period. Once set, the FLAG bit is
Until it is cleared by software, it will not be affected in the subsequent cycles and will maintain the set state.
To do.

The FLAG bit is cleared by software. To clear the flag, the software
First read this bit with "1" and then write "0" to this bit. For FLAG bits
Writing "1" is ignored. The FLAG bit is in the clear state when PWM is disabled.
To maintain.

Note: This flag clear method sets a flag between read and write operations.
It works only if there is no venting. Between read and write operations
The FLAG bit is not cleared when an event occurs that sets the FLAG bit to
Hmm.

When the interrupt level is set to other than “0” by the interrupt level bit IL [2: 0], FLAG
When the bit is set, an interrupt request is issued. Before returning from the interrupt processing routine,
Clear the FLAG bit on the software so that PWMSM does not immediately generate the next interrupt on the IMB.
need to do it.

IL [2: 0] — Interrupt level bit

This bit field is read / written to select the priority level of interrupt requests generated by PWMSM.
There are control bits that can be included. These bits can be read or written at any time
It will be cleared by reset. See Table 94 .

Table 94 Interrupt level bit fields

IL2	IL1	IL0	Priority level
0	0	0	Interrupt disabled
0	0	1	Interrupt level 1 (lowest)
0	1	0	Interrupt level 2
0	1	1	Interrupt level 3
1	0	0	Interrupt level 4
1	0	1	Interrupt level 5
1	1	0	Interrupt level 6
1	1	1	Interrupt level 7 (highest)

IARB3 — Interrupt arbitration bit 3

This bit is with IARB [2: 0] in the BIUSM module configuration register.
Works in combination. Each module that issues an interrupt request on the IMB is unique within the arbitration field.
Must have a value. When multiple modules request interrupts at the same time with the same priority
Uses the interrupt arbitration number to control the priority over the IMB. This IARB3 bit is
It will be cleared by reset. For more information on IARB [2: 0], see “6.5.1 BIUSM Regis” on page 157.
Please refer to "

PIN — Output pin status

This status bit indicates the logical state of the output pin.
0 = Logic level "0" is being output to the output pin.
1 = Logic level "1" is being output to the output pin.

This bit allows the software to monitor the waveform of the output pin. Read PIN bit
It is a dedicated bit. Writes to this bit are ignored.

LOAD — Period and pulse width register load control

This control bit allows the software to power the PWMSM without glitching the PWM output signal.
It can be reinitialized to start the PWM cycle.
0 = no operation
1 = Load the period register and pulse width register.

Reading this bit always returns "0". As soon as "1" is written to this bit, the following

The operation will start.

- Transfers the value (cycle) of the PWMA1 register to the PWMA2 register.
- Transfers the value (pulse width) of the PWMB1 register to the PWMB2 register.
- Initializes the counter register (PWMC) to $ 0001.
- Resets the control circuit and status sequencer.
- Set the FLAG bit.
- If the new value for PWMB2 is different from $ 0000, set the output flip-flop.

Note: If the EN bit is “0” (that is, PWMSM is disabled), the LOAD bit
Even if you write "1" to, it will be ignored.

POL—Output pin polarity control

This control bit allows the software to specify the polarity of the PWM output signal.
vinegar. This bit works in conjunction with the EN bit and is the non-inverting output or inverting output of the output flip-flop.
Specifies that the force is output to the PWMSM output pin. See Table 95 .

Table 95 PWMSM Output Pin Polarity Selection

Control bit		Output pin status	Synchronous edge	Variable edge	Optional Interrupt ON
POL	EN				
0	0	Always low	————	————	————
1	0	Always high	————	————	————
0	1	High pulse	Rising edge	Falling edge	Rising edge
1	1	Low pulse	Falling edge	Rising edge	Falling edge

Note: If POL and EN are set at the same time (POL = 1, EN = 1), the output polarity will change immediately.
increase. After that 4 clocks, it starts counting and produces the first pulse. pulse·
With respect to the timing, "Table 177 PWMSM timing of page 335 See"
Please.

EN—PWMSM enable control

Use this control bit to enable and disable PWMSM in software.
I can.

0 = Disables PWMSM and stops generating PWM output pulses.
1 = Enables PWMSM and starts generating PWM output pulses.

When PWMSM is disabled (EN = “0”), the following conditions occur:

- The output flip-flop is reset and the output pin level is 1 according to the POL bit state. Or it is set to 0.
- · PWMSM ÷ 256 prescaler is reset.
- The counter stops incrementing and is set to $ 0001.
- The comparator is disabled.
- The PWMA1 and PWMB1 registers transfer values to buffer registers (PWMA2 and PWMB2, respectively). I will continue to send.
- When the EN bit is changed from “0” to “1”, the following actions are performed.
- The output flip-flop is set and the first pulse is started.
- PWMSM ÷ 256 Prescaler reset is released.
- The counter is unreset and starts incrementing from $ 0001.
- The FLAG bit is set (PWMA1 and PWMB1 can be updated with new cycles and pulse widths, respectively. Indicates that there is).

When the EN bit is set, PWMSM is the data in the PWMA2 and PWMB2 registers (this data).
Is updated at the end of each cycle via the PWMA1 and PWMB1 registers) in succession
Generates a pulse width modulated output signal.

Note: Prevent glitches in the output waveform when disabling PWMSM
Therefore, the EN bit is soft until a pulse with a period of 0% (PWMB2 = $ 0000) is output.
Do not clear with wear.

CLK [2: 0] — Clock rate selection

The CLK bit is the control bit and is used to output from the PWMSM prescaler 8
You can choose one of the different clock sources from the software. These bits are so
You can change it at any time from the software. Table 96 shows the counter clock source and clock rate.
Shows the details of.

Table 96 PWMSM clock rate selection

PWMSM CLK bit			CPSM bit DIV23	PWMSM clock	Clock source
CLK2	CLK1	CLK0			
0	0	0	0	f sys / 2	PCLK1
0	0	1	0	f sys / 4	Prescaler (/2)
0	1	0	0	f sys / 8	Prescaler (/4)
0	1	1	0	f sys / 16	Prescaler (/8)
1	0	0	0	f sys / 32	Prescaler (/16)
1	0	1	0	f sys / 64	Prescaler (/ 32)
1	1	0	0	f sys / 128	Prescaler (/ 64)
1	1	1	0	f sys / 512	Prescaler (/ 256)
0	0	0	1	f sys / 3	PCLK1
0	0	1	1	f sys / 6	Prescaler (/2)
0	1	0	1	f sys / 12	Prescaler (/4)
0	1	1	1	f sys / 24	Prescaler (/8)
1	0	0	1	f sys / 48	Prescaler (/16)
1	0	1	1	f sys / 96	Prescaler (/ 32)
1	1	0	1	f sys / 192	Prescaler (/ 64)
1	1	1	1	f sys / 768	Prescaler (/ 256)

PWMA—PWM periodic register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
				MSB								LSB			
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

The PWMA register holds the periodic value for the next cycle of the PWM output waveform. For normal use
Is the current period when the software writes the period value to the PWMA1 register with PWMSM enabled.
At the end of, this value is transferred to the PWMA2 register. When PWMSM is disabled, PWMA1 register
The value written to the star is low in the PWMA2 register in the next half cycle (of the MCU system clock).
Will be done. PWMA2 is a temporary register used to smoothly update the PWM period.
vinegar. This register cannot be read / written directly from the software.

The software can write a new value to PWMA1 at any time, and this new value will be the next PWM cycle.
It becomes valid at the beginning of the period (or when "1" is written to the LOAD bit in the PWMSIC register).
increase. The PWMSM hardware does not change the value of the PWMA1 register.

Table 97 shows the five PWMA address locations.

Table 97 PWMA Address Location

PWMA register	Address Address
PWM9	$ YFF94A
PWM10	$ YFF952
PWM11	$ YFF95A
PWM12	$ YFF962
PWM13	$ YFF96A

PWMB—PWM pulse width register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			MSB								LSB				
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

The PWMB register holds the pulse width value for the next cycle of the PWM output waveform. Normal In applications, with PWMSM enabled, the software writes the pulse width value to the PWMB1 register. This value is transferred to the PWMB2 register at the end of the current cycle. When PWMSM is disabled, The pulse width value written to the PWMB1 register is the next half cycle (of the MCU system clock). It is loaded into the PWMB2 register. PWMB2 is a temporary register that smoothly changes the PWM pulse width. Used to renew. This register cannot be read / written directly from the software.

The software can write a new value to the PWMB at any time, and this new value will be the next PWM cycle. It becomes valid at the beginning of the period (or when "1" is written to the LOAD bit in the PWMSIC register). increase. The PWMSM hardware does not change the value of the PWMB1 register.

Table 98 shows the five PWMB address locations.

Table 98 PWMB Address Location

PWMB register	Address Address
PWM9	$ YFF94C
PWM10	$ YFF954
PWM11	$ YFF95C
PWM12	$ YFF964
PWM13	$ YFF96C

PWMC—PWM counter register

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			MSB								LSB				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

This counter (PWMC register) is a read-only register. This anytime from the software You can read the counter. Writes to this counter are ignored. For PWMC The set loads $ 0001. Also, when PWMSM is disabled (EN = “0”), PWMC is set and held at $ 0001.

Page 182

Table 99 shows the five PWMC address locations.

Table 99 PWMC address location

PWMC register	Address Address
PWM9	$ YFF94E
PWM10	$ YFF956
PWM11	$ YFF95E
PWM12	$ YFF966
PWM13	$ YFF96E

6.10 CTM3 interrupt

CTM3 has the ability to generate various interrupts on the IMB. Each submodule has 7 levels You can request one of these interrupts. Submodules have a 3-bit level number and 1 There are bit arbitration numbers, which can be initialized by software.

The 3-bit level number drives any of the seven interrupt signals on the IMB with a submodule. Select whether to generate an interrupt request. Four priorities prepared on the IMB during arbitration between modules One of the place bits is supplied by the selected submodule and BIUSM supplies the other three. bottom Therefore, CTM3 responds to two of the 15 possible arbitration numbers.

During the IMB arbitration process, BIUSM manages arbitration independently between CTM3 submodules, which submodule

Determines if Le should respond. Have a pending interrupt request at a level being arbitrated on the IMB
Among the submodules, the submodule with the lowest address is given the highest priority to respond.
Will be.

If the IARB is not unique for a module, the interrupts that occur at the same time are as shown in Table 100 .
Priority in hardware according to vector number (or submodule interrupt arbitration sequence number)
Is determined. Following the interrupt arbitration process, CTM3 supplies an 8-bit vector number. 8 bit
6 bits of the data are supplied by the submodule. The submodule has two separate interrupt factors.
It can be identified by a neat interrupt vector. The upper 2 bits of the 8-bit vector are provided by BIUSM.
Will be paid. The lower 6 vector bits are the most pending in CTM3 at the beginning of the arbitration cycle.
Identifies a high priority interrupt request.

Table 100 CTM3 Interrupt Prioritization and Vector / Pin Arrangement

Submodule name	Submodule Base address [1]	Submodule Vector number [2]	Submodule interrupt Mediation sequence [3]	Pin name
BIUSM	$ YFF900	None None	None None	None None
CPSM	$ YFF908	None None	None None	None None
MCSM2	$ YFF910	xx000010	2	CTM2C (Clock) CTD22 (Load)
DASM3	$ YFF918	xx000011	3	CTD4
DASM4	$ YFF920	xx000100	Four	CTD4
DASM5	$ YFF928	xx000101	Five	CTD5
DASM6	$ YFF930	xx000110	6	CTD6
DASM7	$ YFF938	xx000111	7	CTD7
DASM8	$ YFF940	xx001000	8	CTD8
PWSM9	$ YFF948	xx001001	9	CPWM9
PWSM10	$ YFF950	xx001010	Ten	CPWM10

Table 100 CTM3 Interrupt Prioritization and Vector / Pin Arrangement (continued)

Submodule name	Submodule Base address [1]	Submodule Vector number [2]	Submodule interrupt Mediation sequence [3]	Pin name
PWSM11	$ YFF958	xx001011	11 11	CPWM11
PWSM12	$ YFF960	xx001100	12	CPWM12
PWSM13	$ YFF968	xx001101	13	CPWM13
DASM14	$ YFF970	xx001110	14	CTD14
DASM15	$ YFF978	xx001111	15	CTD15
DASM16	$ YFF980	xx010000	16 16	CTD16
DASM17	$ YFF988	xx010001	17 17	CTD17
DASM18	$ YFF990	xx010010	18 18	CTD18
MCSM19	$ YFF998	xx010011	19 19	CTM2C (Clock) CTD18 (Load)
DASM20	$ YFF9A0	xx010100	20	CTD20
MCSM21	$ YFF9A8	xx010101	twenty one	CTM2C (Clock) CTD20 (Load)
DASM22	$ YFF9B0	xx010110	twenty two	CTD22
MCSM23	$ YFF9B8	xx010111	twenty three	CTM2C (Clock) CTD20 (Load)

Note: 1.Y = M111. Where M is the state of the SLIM's module mapping bits (HC16).
For devices, Y = $ F).
2. “xx” stands for VECT [7: 6]. This is the BIUSM module configuration
It is located in the register.
3. Interrupt arbitration # 2 has the highest priority and interrupt arbitration # 26 has the lowest priority.

Chapter 7
Analog / digital converter with cue Module

Queued analog-to-digital converter (QADC) with analog front end and digital control It consists of a subsystem and is an intermodule bus (IMB). Contains interface blocks I am.

See Figure 27 .

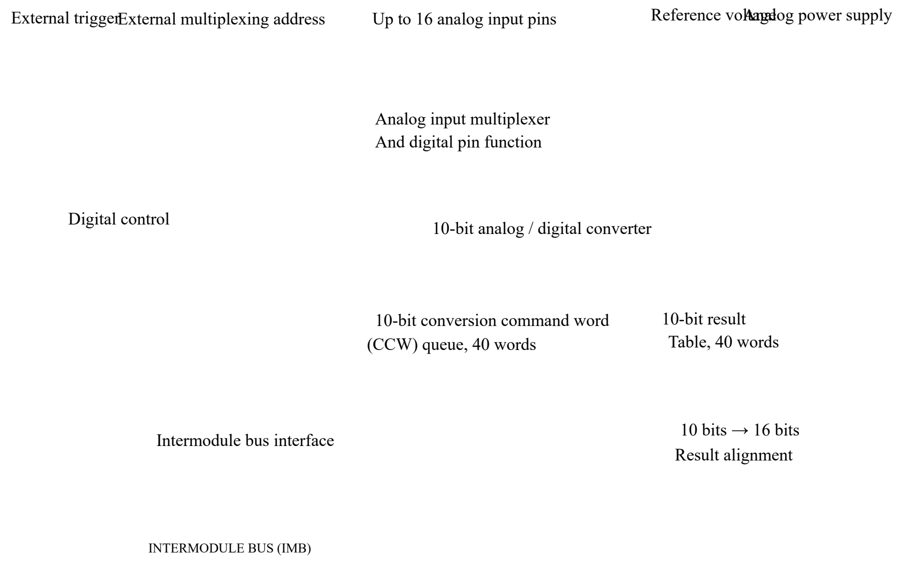


Figure 27 QADC block diagram

7.1 Overview

The queued analog-to-digital converter (QADC) is a 10-bit unipolar, sequential comparison controller.
This is Bata. Internal multiplexer can support up to 16 analog input channels
increase. Also, in enhanced external multiplex mode, up to 41 input channels (including internal channels).
) Can be supported. The actual number of channels depends on the number of pins available on the QADC module.

The analog section has input pins, an analog multiplexer, and a sample and hold analog circuit.
I will. Analog conversions include digital-to-analog converters (DACs), resistors-capacitor arrays, and high gain capacitors.

It runs on a parator.

The digital control unit is a control logic that controls the sequence of conversion processing, channel select logic.
It consists of a serial comparison register (SAR). Also, cycle / interval timer, controller
Troll register, status register, conversion command word (CCW) table RAM, result
Word table RAM is also included.

Application software through the IMB environment with the Bus Interface Unit (BIU)
You can run the QADC with.

The QADC memory map occupies 512 bytes (256 words).

Table 101 QADC Module Address Map

access	address	offset	15	8	7	0
S[1]	$ YFF200[3]	$ 000	Module configuration register (QADCMCR)			
S	$ YFF202	$ 002	Test register (QADCTEST)			
S	$ YFF204	$ 004	Interrupt register (QADCINT)			
S / U[2]	$ YFF206	$ 006	Port A data (PORTQA)		Port B data (PORTQB)	
S / U	$ YFF208	$ 008	Port data direction register (DDRQA)			
S / U	$ YFF20A	$ 00A	Control register 0 (QACR0)			
S / U	$ YFF20C	$ 00C	Control register 1 (QACR1)			
S / U	$ YFF20E	$ 00E	Control register 2 (QACR2)			
S / U	$ YFF210	$ 010	Status register (QASR)			
————	$ YFF212-$ YFF22E	$ 012- $ 02E	Reserve			
S / U	$ YFF230-$ YFF27E	$ 030- $ 07E	Conversion command word (CCW) table			
	$ YFF280- $ YFF2AE	$ 080- $ 0AE	Reserve			
S / U	$ YFF2B0-$ YFF2FE	$ 0B0- $ 0FE	Result word table Right-justified unsigned result register (RJURR)			
————	$ YFF300-$ YFF32E	$ 100- $ 12E	Reserve			
S / U	$ YFF330-$ YFF37E	$ 130- $ 17E	Result word table Left-justified result register (LJSRR)			
————	$ YFF380- $ YFF3AE	$ 180- $ 1AE	Reserve			
S / U	$ YFF3B0-$ YFF3FE	$ 1B0- $ 1FE	Result word table Left-justified unsigned result register (LJURR)			

Note: 1. S = Supervisor only
2. Supervisor or user according to the SUPV bit of S / U = QADCMCR
3. Y = M111 where M is the logic of the SLIMCR module map (MM) bits.
It is a state. For MC68HC (9) 16Y5 / 916Y6, Y must be $ F.
If M is cleared, the IMB module will remain until a reset is performed.
Inaccessible. After reset, M can be written only once.

7.2 Signal description

Figure 28 shows the module pins for the QADC.

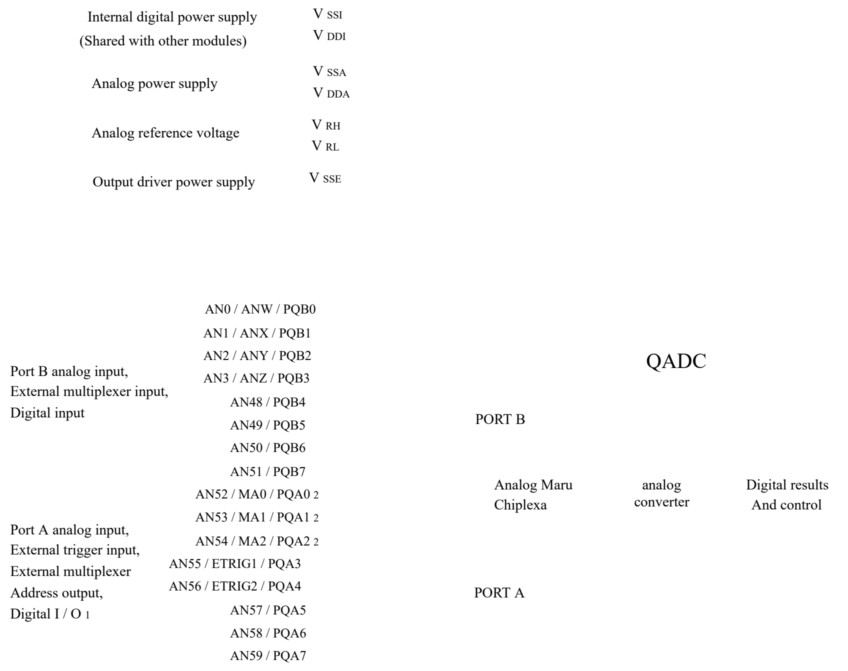


Figure 28 QADC input / output signals

Note: 1. The output pin side of port A is an open-drain type driver.

2. Special care should be taken not to apply a voltage higher than VDDA to this pin.

Table 102 summarizes the functionality of the QADC pins.

Table 102 QADC Pin Features

Pin name	Neemonit nine	Pin function
Port QA analog input pin	AN [59:52]	Eight ports A pin should be AN [59:52] when used as an analog input I will. Port QA is supported by a digital output driver, so the port The analog characteristics of QA and port QB are different. All analog signal input Can be used for other purposes.
Port QA Digital I / O Pin	PQA [7: 0]	Port A pin is used as a bidirectional 8-bit digital I / O port When it is, it becomes PQA [7: 0]. These eight pins are general-purpose digital input signals, Or it can be used for digital open drain output signals. Each PQA pin programs the upper half of the port data direction register (DDRQA). It is set as an input or an output.
Port QB analog input pin	AN [51:48] / AN [3: 0]	Used as an analog input. Port QB pins are analog and digital Since it functions as an input-only port, it has different analog characteristics from port QA. increase. All analog signal input pins can be used for other purposes.

Table 102 QADC Pin Functions (continued)

Pin name	Neemonit nine	Pin function
Port QB digital input pin	PQB [7: 0]	It is used as a dedicated port for 8-bit digital input. PQB pin is analog In addition to functioning as an input pin, it also connects to the synchronizer input during reading. And when the applied voltage meets the VIH and VIL requirements, a general purpose digital It can be used as a tall input.
External trigger input pin	ETRIG [2: 1]	The QADC uses two external trigger pins (ETRIG [2: 1]). With external trigger The force is contained in two multifunction port QA pins (PQA [4: 3]), but these Pins are typically used as analog channel input pins. Outside of 2 inputs Each of the part trigger pins is 1 in the scan queue (queue 1 and queue 2). It corresponds to one. When the queue is in external trigger mode The corresponding external trigger pin is configured as a digital input and the data direction register Software programmed into an external trigger pin in the studio (DDRQA) The output direction is ignored.
Multiplex address output pin	MA [2: 0]	QADC supplies a 3-bit multiplex address output to an external multiplexing chip And allow you to choose one of the eight inputs. Multiplexer add The less output signal (MA [2: 0]) is the multiplexer address output bit, or It can be used as a general-purpose I / O.
Multiplex analog input pin	ANw, ANx, ANy, ANz	In external multiplex mode, the four port B pins are redefined. Each represents eight input channels. ANw: even numbered channels from 0 to 14 ANz: 1 to 15 odd numbered channels ANy: even numbered channels from 16 to 30 ANz: odd numbered channels from 17 to 31
Reference voltage pin	V_{RH} / V_{RL}	V_{RH} and V_{RL} are input pins dedicated to high and low reference voltages.
Dedicated analog power pin	V_{DDA} / V_{SSA}	A dedicated pin that powers the analog subsystem of the QADC module.
External Digital Power Pin (V_{SSE})	V_{SSE}	The V_{SSE} pin provides ground level to the driver for the port QA pin.
Internal digital power pin (with V_{DDI}) V_{SSI})	V_{DDI} / V_{SSI}	The digital part of the QADC and all other digital on the microcontroller chip Powers the Tal module.

7.3 QADC Registers

7.3.1 Global Register

The QADC has three global registers for configuring module operation. module· Configuration register (QADCMCR), test register (QADCTEST), and interrupt register It is a data (QADCINT).

QADCMCR module configuration register $ YFF200

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	FRZ			NOT USED				SUPV		NOT USED			IARB		
RESET:															
1	0							1				0	0	0	0

STOP — Stop mode

0 = Disables stop mode.

1 = Enable stop mode.

FRZ—Freeze Enable

0 = Ignore the IMB FREEZE signal.

When the FREEZE signal of 1 = IMB is output, it freezes after finishing the current conversion process.

See “2.8.1 Freeze” on page 69 .

SUPV—Supervisor / Unconstrained data space

0 = unconstrained access

1 = Supervisor Access

SUPV defines access to assignable ADC registers. CPU16 is in supervisor mode This bit is invalid because it only works with.

IARB [3: 0] — Software interrupt arbitration number

IARB prioritizes interrupt arbitration for QADC.

QADCTEST—QADC test register $ YFF202

ADCTEST is used in combination with the SLIM test register for factory testing of the ADC.

QADCINT—QADC interrupt register $ YFF204

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	—— 1		IRL1		—— 1		IRL2				IVB [7: 2]				IVB [1: 0] 2	
RESET:		0	0	0		0	0	0	0	0	0	0	1	1		

1. Reservation
2. Bits 1 and 0 are provided by the QADC.

IRL1 [14:12] — Queue 1 interrupt level

The IRL1 field sets the interrupt request level for queue 1. State of 000 blocks interrupt request Disables the queue 1 interrupt by stopping it. Queue 1 If you use interrupts, The IRL1 field must be initialized to a non-zero value. Level% 001 is the lowest priority In software interrupt requests, level% 111 is the highest priority request.

IRL2 [10: 8] — Queue 2 interrupt level

The IRL2 field sets the queue 2 interrupt request level. State of 000 blocks interrupt request Disables queue 2 interrupts. Queue 2 If you use interrupts, The IRL2 field must be initialized to a non-zero value. Level 001 is the lowest priority For software interrupt requests, level 111 is the highest priority request.

IVB [7: 0] — Interrupt vector base address

The initialization software inputs the upper 6 bits of the IVB bits of the interrupt register. Interrupt arbitration The vector supplied by QADC to the CPU is the upper 6 bits of the IVB bit and one of the four QADC interrupt requests. It consists of the lower 2 bits supplied to the QADC to identify. The interrupt vector number is divided It has nothing to do with the inclusion level and interrupt arbitration number. The $ 0F vector number is used as the uninitialized interrupt vector. It corresponds. After reset, the lower byte of the interrupt register is read at $ 0F. Once IVB When the field is written, the least significant 2 bits are always read at zero.

Table 103 QADC Vector Assignment by Interrupt Source

IVB [7: 2]	IVB [1: 0]	Interrupt source
×××××× (Set by the user)	1 1	Queue 1 Completion interrupt
	Ten	Queue 1 Pose interrupt
	0 1	Queue 2 Completion interrupt
	0 0	Queue 2 Pose interrupt

Interrupt priority (if IRL1 = IRL2), queue 1 complete → queue 1 pause → queue 2 complete → Queue 2 Pauses are prioritized within the QADC.

7.3.2 General purpose I / O port register

QADC ports QA and QB pass through two 8-bit port data registers (PORTQA and PORTQB). Will be accessed. Data Direction Register (DDRQA) when the pins associated with PORTQA are configured in I / O Bits control the direction of the pin driver. Since PORTQB is input only, it is a data direction register. Is not required. Therefore, the low byte of the port data direction register is not implemented. Preliminary Reading about a bit returns 0, but writing to it has no effect.

PORTQA and PORTQB are not affected by the reset.

PORTQA—Port QA data register $ YFF206

PORTQB—Port QB data register $ YFF207

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	PQA7	PQA6	PQA5	PQA4	PQA3	PQA2	PQA1	PQA0	PQB7	PQB6	PQB5	PQB4	PQB3	PQB2	PQB1	PQB0
RESET:	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
Analog channel ::	AN59	AN58	AN57	AN56	AN55	AN54	AN53	AN52	AN51	AN50	AN49	AN48	AN3	AN2	AN1	N0
External trigger input:				ETRIG2	ETRIG1											
Multiplex address output:						AM2	AM1	AM0								
With multiplex analog:																

ANz ANy ANx ANw

Port QA pins can be used for general purpose digital input signals or digital open drain output signals
PQA [7: 0] when used as a bidirectional 8-bit I / O port. Port QA is analog
Input (AN [59:52], External Trigger Input (ETRIG [2: 1]), and External Multiplex Address Output (MA [2: 0]))
Can also be used as.

Port QB pin is an input-only 8-bit digital port that can be used for general-purpose digital input signals.
When using it, it becomes PQB [7: 0]. Port QB is a non-multiplex analog input (AN [51: 48] /
It can also be used as AN [3: 0]) and multiplex analog inputs (ANz, ANy, ANx, ANw).

DRQA port data direction register $ YFF208

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
DDQA7	DDQA6	DDQA5	DDQA4	DDQA3	DDQA2	DDQA1	DDQA0				RESERVED				
RESET:															
0	0	0	0	0	0	0	0								

The Port Data Direction Register (DDRQA) supports port QA digital I / O pins. Bi-directional
In the case of, the leakage current and capacity specifications will be slightly higher. Set any bit of this register to (1
) When set, the corresponding pin is configured as an output. Set any bit in this register to (0)
When cleared, the corresponding pin will be configured as an input. DDR bits on the pins used for analog inputs
It is done by software to make sure that is not set to 1. DDR bit to 1
When set and the pin is selected for analog conversion, the output digital driver is affected by the load.
The voltage of is the voltage to be sampled.

Note: Be careful when mixing digital and analog inputs. As much as possible
Separate these inputs. Also, make the rise time and fall time as long as possible.
please.

7.3.3 Control and status registers

The QADC has three control registers (QACR0, QACR1, QACR2) and one status register.
There is (QASR).

Page 190

QACR0—Control register 0 $ YFF20A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MUX			RESERVED						PSH			PSA		PSL	
RESET:															
0							0	0	0	1	1	0	0	1	1

MUX—External multiplex mode

The MUX bit allows the software to select an external multiplexing mode. External multi
Plex mode changes the interpretation of channel numbers, forcing MA0, MA1, and MA2 on the output pins.
0 = Up to 16 internally multiplexed channels
1 = Up to 44 externally multiplexed channels

PSH [8: 4] — Prescaler clock "H" time

The PSH field selects the QADC clock (QCLK) "H" time on the prescaler.
Table 104 shows the bit values for the PSH field that specify the QCLK "H" time range.

Table 104 Prescaler clock "H" time

PSH [8: 4]	QCLK "H" time
00000	1 system clock cycle
00001	2 system clock cycle
00010	3 system clock cycle
00011	4 system clock cycle
00100	5 system clock cycle
00101	6 system clock cycle
00110	7 system clock cycle
00111	8 system clock cycle
01000	9 system clock cycle
01001	10 system clock cycle
01010	11 system clock cycle
01011	12 system clock cycle

01100	13 system clock cycle
01101	14 system clock cycle
01110	15 system clock cycle
01111	16 system clock cycle
10000	17 system clock cycle
10001	18 system clock cycle
10010	19 system clock cycle
10011	20 system clock cycle
10100	21 system clock cycle
10101	22 system clock cycle
10110	23 system clock cycle
10111	24 system clock cycle
11000	25 system clock cycle
11001	26 system clock cycle
11010	27 system clock cycle
11011	28 system clock cycle
11100	29 system clock cycle
11101	30 system clock cycle
11110	31 system clock cycle
11111	32 system clock cycle

Page 191

PSA — Prescaler adds half-cycle clock

0 = QCLK “H” and <# 007F> L “time does not change.

1 = QCLK Add half a cycle of one system clock to the “H” time, and one system clock from the “L” time.
Pull half a cycle of lock.

PSL [2: 0] — Prescaler clock “L” time

The PSL field selects the QADC clock (QCLK) “L” time on the prescaler. Table 105 shows PSH
Indicates the “L” time of QCLK enabled by a bit in the field.

Table 105 Prescaler clock “L” time

PSL [2: 0]	QCLK “L” time
000	1 system clock cycle
001	2 system clock cycle
010	3 system clock cycle
011	4 system clock cycle
100	5 system clock cycle
101	6 system clock cycle
110	7 system clock cycle
111	8 system clock cycle

7.3.3.1 QADC clock generation

The conversion clock (QCLK) of the QADC is sourced from the system clock of the MCU. QCLK frequency
Is determined by setting the control bits for PSH [8: 4], PSL [2: 0], PSA.

The maximum frequency of QCLK is 2.1MHz. More clocks depending on the control bit settings
It can be set, but the accuracy of the AD conversion value is not guaranteed. Therefore, the control bit
The setting must be below this maximum frequency. Also, as a duty cycle,
Must be close to 75% (75% on the high side, 25% on the low side).

Changes in QCLK during AD conversion operation can affect the conversion value, so the line is when AD conversion is stopped.
Please become to.

Table 106 and Figure 29 show examples of QC[redacted]

Table 106 QADC clock setting example

	Control Register 0 information			F sys = 24MHz, Input Sample Time (IST) =% 00	
example	PSH [8: 4]	PSA	PSL [2: 0]	QCLK [MHz]	Conversion time [∝ s]
1	8 (=% 01000)	0	2 (=% 010)	2.0 2.0	9.0 9.0
2	8 (=% 01000)	1	2 (=% 010)	2.0 2.0	9.0 9.0

System clock
(F sys = 24Mhz)

QCLK Example 1
(QCLK = 2Mhz)

QCLK Example 2
(QCLK = 2Mhz)

12 cycle

Figure 29 QADC clock example

Page 192

QACR1—Control register 1 $ YFF20C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CIE1	PIE1	SSE1	NOT USED			MQ1					NOT USED				
RESET:															
0	0	0			0	0	0								

CIE1—Queue 1 Completion interrupt enabled

0 = Disables the queue completion interrupt associated with queue 1.
1 = Enable interrupts after conversion of the sample requested by the last CCW in queue 1.

PIE1—Queue 1 pause interrupt enabled

0 = Disables pause interrupts associated with queue 1.
1 = Interrupt after conversion of sample requested by CCW in queue 1 with pause bits set
To enable.

SSE1—Queue 1 single scan enable bit

SSE1 enables single scan for queue 1 and scans after a trigger event occurs.
Will start. Set the MQ1 bit to select one of the single scan queue operating modes
You can set the SSE1 bit to 1 during the same write cycle.
You can write 1 or 0 to the single scan enable bit, but test
It will always be 0 when read unless a mode is selected. SSE1 bit is the trigger event
Enables to start running the queue to handle a single scan on queue 1.
vinegar. When the single scan is complete, the QADC clears the SSE1 bit.

0 = Do not accept trigger events in single scan mode.
1 = Accepts a trigger event in single scan mode and starts executing queue 1.

MQ1 [10: 8] — Queue 1 operating mode

The MQ1 field selects the queue operation mode for queue 1. In Table 103, each of Queue 1
Seed Indicates the bit value of the MQ1 field that enables the mode of operation.

Table 107 Queue 1 Operation mode

MQ1 [10: 8]	action mode
000	Disable mode. No conversion is done.
001	Software-triggered single scan mode (starting with SSE1)
010	External trigger rising edge single scan mode (on ETRIG1 pin)
011	External trigger falling edge single scan mode (on ETRIG1 pin)
100	Reserve. No conversion is done.
101	Continuous scan software trigger mode
110	External trigger rising edge continuous scan mode (on ETRIG1 pin)
111	External trigger falling edge continuous scan mode (on ETRIG1 pin)

QACR2—Control register 2 $ YFF20E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CIE2	PIE2	SSE2			MQ2			RES	NOT USED			BQ2			
RESET:															
0	0	0	0	0	0	0	0	0		1	0	0	1	1	1

CIE2—Cue 2 Complete Software Interrupt Enabled

0 = Disables the queue completion interrupt associated with queue 2.
1 = Enable interrupts after conversion of the sample requested by the last CCW in queue 2.

PIE2—Cue 2 pause software interrupt enabled

0 = Disables pause interrupts associated with queue 2.

1 = Queue 2 pause bit set After sample conversion requested by CCW

Enable inclusion.

SSE2—Cue 2 single scan enable bit

SSE2 enables single scan for queue 2 and scans after a trigger event occurs. Will start. Set the MQ2 bit to select one of the single scan queue operating modes You can set the SSE2 bit to 1 during the same write cycle. Singles You can write 1 or 0 to the can enable bit, but select the test mode. Unless set aside, it will always be 0 when read. SSE2 bit enables trigger event And start running the queue to handle the single scan on queue 2. Singles When the can is complete, the QADC clears the SSE2 bits.

0 = Do not accept trigger events in single scan mode.

1 = Accepts a trigger event in single scan mode and starts executing queue 2.

MQ2 [12: 8] — Queue 2 operating mode

The MQ2 field selects the queue operating mode for queue 2. Table 108 shows the various operations of queue 2. Indicates the bit value of the MQ2 field that enables the mode.

Table 108 Operation mode of queue 2

MQ2 [12: 8]	action mode
00000	Disable mode, no conversion is done.
00001	Software Trigger Single Scan Mode (Starts with SSE2)
00010	External trigger rising edge single scan mode (on ETRIG2 pin)
00011	External trigger falling edge single scan mode (on ETRIG2 pin)
00100	Interval timer single scan mode: Time = QCLK period x 2^{7}
00101	Interval timer single scan mode: Time = QCLK period x 2^{8}
00110	Interval timer single scan mode: Time = QCLK period x 2^{9}
00111	Interval timer single scan mode: Time = QCLK period x 2^{10}
01000	Interval timer single scan mode: Time = QCLK period x 2^{11}
01001	Interval timer single scan mode: Time = QCLK period x 2^{12}
01010	Interval timer single scan mode: Time = QCLK period x 2^{13}
01011	Interval timer single scan mode: Time = QCLK period x 2^{14}
01100	Interval timer single scan mode: Time = QCLK period x 2^{15}
01101	Interval timer single scan mode: Time = QCLK period x 2^{16}
01110	Interval timer single scan mode: Time = QCLK period x 2^{17}
01111	Reserve
10000	Reserve
10001	Continuous scan software trigger mode
10010	External trigger rising edge continuous scan mode (on ETRIG2 pin)
10011	External trigger falling edge continuous scan mode (on ETRIG2 pin)
10100	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{7}
10101	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{8}
10110	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{9}
10111	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{10}
11000	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{11}
11001	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{12}
11010	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{13}
11011	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{14}

Table 108 Operation Modes for Queue 2 (continued)

MQ2 [12: 8]	action mode
11100	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{15}
11101	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{16}
11110	Cycle timer Continuous scan mode: Time = QCLK cycle x 2^{17}
11111	Reserve

RES— Queue 2 resume

0 = After suspend, start execution from queue 2 or the first CCW in the current subqueue.
1 = After suspend, start execution from the aborted CCW in queue 2.

BQ2 [5: 0] — Start of queue 2

The BQ2 field indicates the CCW location where queue 2 begins. Length of queue 1 and queue 2 The location of the CCW table where queue 2 starts with a programmable pointer to make it variable. Identifies The BQ2 field also serves as a queue termination condition for queue 1.

QASR—Status register $ YFF210

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
CF1	PF1	CF2	PF2	TOR1	TOR2		QS					CWP			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

To clear the status bit, read with the specified bit set to “1” and write “0”.

CF1—Queue 1 completion flag

0 = Queue 1 scan is not complete.
1 = Queue 1 scan is complete.

PF1—Queue 1 pause flag

0 = Queue 1 has not reached the pose.
1 = Queue 1 has reached the pose.

CF2—Queue 2 completion flag

0 = Queue 2 Scan is not complete.
1 = Queue 2 scan is complete.

PF2—Cue 2 pause flag

0 = Queue 2 has not reached the pose.
1 = Queue 2 has reached the pose.

TOR1—Queue 1 Trigger Overrun

0 = Unexpected queue 1 No trigger event has occurred.
1 = At least one unexpected queue 1 Trigger event has occurred.

TOR2—Cue 2 Trigger Overrun

0 = No unexpected queue 2 trigger event has occurred.
1 = At least one unexpected queue 2 trigger event has occurred.

QS [9: 6] — Queue status

Table 109 shows the bits in the QS field and their effect on the status of queue 1 and queue 2. indicate.

Table 109 Queue Status

QS [9: 6]	Queue 1 / Queue 2 status
0000	Queue 1 idol, Queue 2 idol
0001	Queue 1 Idol, Queue 2 Pose
0010	Queue 1 idle, queue 2 active
0011	Queue 1 Idle, Queue 2 Trigger Pending

0100	Queue 1 Pose, Queue 2 Idol
0101	Queue 1 pose, Queue 2 pose
0110	Queue 1 Pose, Queue 2 Active
0111	Queue 1 Pose, Queue 2 Trigger Pending
1000	Queue 1 active, queue 2 idle
1001	Queue 1 Active, Queue 2 Pose
1010	Queue 1 Active, Queue 2 Suspend
1011	Queue 1 Active, Queue 2 Trigger Pending
1100	Reserve
1101	Reserve
1110	Reserve
1111	Reserve

CWP — Command word pointer

Depending on the CWP, the software is currently running which CCW or the last completed CCW. You can know which one. The command word pointer has a software read-only feel. There is no effect even if you write.

7.3.4 Conversion command word table

The CCW table is 40 words x 10 bits wide RAM, programmed in software, and one or more. You can request conversion of multiple analog input channels. The entry in the CCW table is 10 bits Conversion command word. CCW tables are written in software and modified in QADC plug. Each CCW requires that analog channels be converted to digital results. CCW is Analo Specifies the channel number, input sample time, and whether the queue will pause after the current CCW. increase. The 10 bits implemented in the CCW word are literate data, but the software does the QADC. Write only once when initializing, and do not change after that. Read unimplemented bits as 0 There is no effect even if it is issued and written. Each location in the CCW table is a result word table It corresponds to the location in the bull. When the conversion of the CCW entry is complete, the corresponding result word A 10-bit result is written to the entry. QADC provides 40 CCW table entries increase. If more results are needed, the software will re-release one or more for multiple channels. Use Zalto table location.

Page 196

Conversion command word
(CCW) table

Result word table

00 00 Head of queue 1

00 00

Channel selection
sample
hold
A / D conversion

End of queue 1
BQ2 Head of queue 2

39 39 End of queue 2

39 39

10-bit conversion command word format

10-bit result (3 types)

9 8 7 6 Five Four 3 2 1 0

P BYP IST CHAN

P = Pause until the next trigger
BYP = Bypass
IST = input sample time
CHAN = channel number and queue exit code

Readable in 16-bit format)

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 RESULT
Right-justified unsigned result

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESULT 0 0 0 0 0 0
Left-justified unsigned result

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
S RESULT 0 0 0 0 0 0
Left-justified signed result

S = sign bit

Figure 30 QADC conversion queue operation

CCW—Conversion command word table $ YFF230— $ YFF27E

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED				P	BYP	IST					CHAN		
RESET:																
							U	U	U	U	U	U	U	U	U	U

P—Pose

0 = Do not enter pause state after running the current CCW.
1 = After executing the current CCW, enter the pause state.

BYP—Sample Amp Bypass

0 = Amp bypass mode is disabled.
1 = Amp bypass mode is enabled.

IST [7: 6] — Input sample time

The IST field allows the software to specify the length of the sample window.
vinegar. Table 110 shows the bits in the IST field that set different input sample conditions.

Table 110 Input sample time

IST [7: 6]	Input sample time
00 00	Input sample time = QCLK period x 2
01 01	Input sample time = QCLK period x 4
Ten	Input sample time = QCLK period x 8
11 11	Input sample time = QCLK period x 16

CHAN [5: 0] — Channel number

The CHAN field selects the input channel number. Table 111 shows Cha in non-multiplex mode.
Indicates the assignment of flannel numbers. Table 112 shows the channel number assignments in multiplex mode.
increase.

Table 111 Non-multiplex channel assignments and pin names

Non-multiplex input pins				CCW CHAN field Channel number	
Port pin name	Analog pin name	Other features	Pin type	Binary number	Decimal number
PQB0	AN0	————	input	000000	0
PQB1	AN1	————	input	000001	1
PQB2	AN2	————	input	000010	2
PQB3	AN3	————	input	000011	3
————	————	invalid	————	0XXXXX	4 to 31
————	————	Reserve	————	10XXXX	32 to 47
PQB4	AN48	————	input	110000	48
PQB5	AN49	————	input	110001	49
PQB6	AN50	————	input	110010	50
PQB7	AN51	————	input	110011	51

PQA0	AN52	————	Input / output	110100	52 52
PQA1	AN53	————	Input / output	110101	53
PQA2	AN54	————	Input / output	110110	54
PQA3	AN55	ETRIG1	Input / output	110111	55 55
PQA4	AN56	ETRIG2	Input / output	111000	56
PQA5	AN57	————	Input / output	111001	57 57
PQA6	AN58	————	Input / output	111010	58
PQA7	AN59	————	Input / output	111011	59
V RL	Low Ref	————	input	111100	60
V RH	High Ref	————	input	111101	61
————	————	V DDA / 2	————	111110	62
————	————	Queue exit code	————	111111	63 63

Table 112 Multiplex channel assignments and pin names

Multiplex input pin				CCW CHAN field Channel number	
Port pin name	Analog pin name	Other features	Pin type	Binary number	Decimal number
PQB0	ANw	————	input	00XXX0	0 to 14 (even)
PQB1	ANx	————	input	00XXX1	1 to 15 (odd number)
PQB2	ANy	————	input	01XXX0	16 to 30 (even)
PQB3	ANz	————	input	01XXX1	17 to 31 (odd number)

Table 112 Multiplex Channel Assignments and Pin Names (continued)

Multiplex input pin				CCW CHAN field Channel number	
Port pin name	Analog pin name	Other features	Pin type	Binary number	Decimal number
————	————	Reserve	————	10XXXX	32 to 47
PQB4	AN48	————	input	110000	48
PQB5	AN49	————	input	110001	49
PQB6	AN50	————	input	110010	50
PQB7	AN51	————	input	110011	51
PQA0	————	MA0	Input / output	110100	52 52
PQA1	————	MA1	Input / output	110101	53
PQA2	————	MA2	Input / output	110110	54
PQA3	AN55	ETRIG1	Input / output	110111	55 55
PQA4	AN56	ETRIG2	Input / output	111000	56
PQA5	AN57	————	Input / output	111001	57 57
PQA6	AN58	————	Input / output	111010	58
PQA7	AN59	————	Input / output	111011	59
V RL	Low Ref	————	input	111100	60
V RH	High Ref	————	input	111101	61
————	————	V DDA / 2	————	111110	62
————	————	Queue exit code	————	111111	63 63

7.3.5 Result Word Table

The result word table is 40 words x 10 bits wide RAM. Corresponding CCW table en
The QADC writes the entry after the analog conversion specified by the bird is complete. soft
The ware can read and write the result word table, but during normal operation the software resurrections.
Read the rut word table to get the analog conversion value from the QADC. Unimplemented bit is 0
There is no effect even if it is read and written.

RJURR— Right-justified unsigned result word table $ YFF2B0— $ YFF2FE

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
		NOT USED								RESULT					

The conversion result is unsigned right-justified data. Bits [9: 0] are used for 10-bit resolution and bits
Reading [15:10] returns zero.

LJSRR—Left-justified result word table $ YFF330— $ YFF37E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
S					RESULT							NOT USED			

S = sign bit

The conversion result is signed left-justified data. Bit [15: 6] is used for 10-bit resolution, with the highest bit
Is inverted to form the sign bit. Reading bit [5: 0] returns zero.

LJURR— Left-justified unsigned result word table $ YFF3B0— $ YFF3FE

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RESULT										NOT USED					

The conversion result is unsigned left-justified data. Bit [15: 6] is used for 10-bit resolution and bits
Reading [5: 0] returns zero.

7.4 Example of cue priority

The two queue commands can indicate the order of AD conversion in parallel, but there is only one AD conversion function. Because of this, the queue priority is specified and conversion is performed sequentially. Each cue is strange It has various triggers to initiate the conversion, which can be done asynchronously.

The following is an example of the QADC queue priority. The figure shows Queue 1 (Q1) and Queue 2 (Q2) conversion is instructed by the queue command (CCW). The terms used in the figure are as follows. is.

— About the trigger

T1: Queue 1 trigger (external trigger, software trigger by Single-Scan, continuous Trigger)

T2: Queue 2 trigger (external trigger, single-scan software trigger, periodic Trigger by target interval, continuous trigger)

— About flags

CF flag: Set when the last queue command ends.

PF flag: Pause Set when the configured queue command is finished.

TOR error flag: Tigger Overrun Error (TOR) triggers the next while performing a queue conversion Is set when

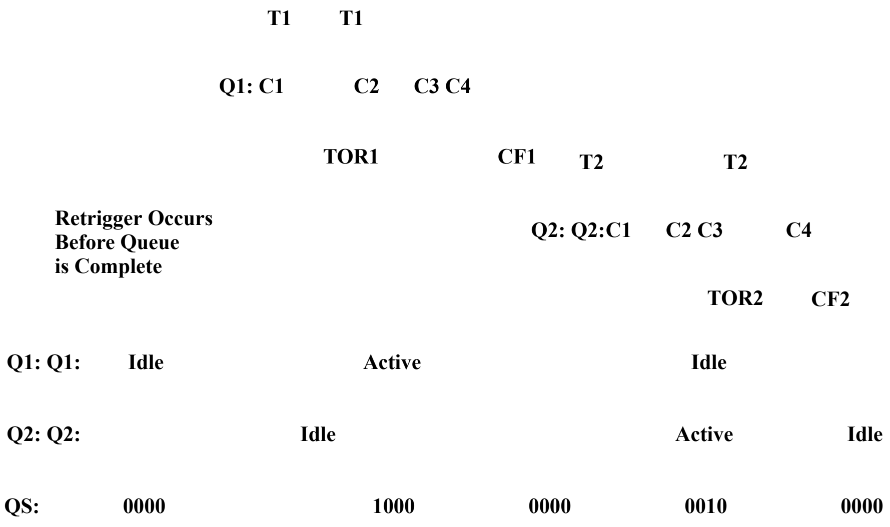


Figure 31 CCW Priority Situation

Figure 31 (a) CCW Priority Situation

In the case of Figure 31 (a), the queue starts performing the AD conversion when it detects the trigger. New during execution If a trigger occurs, the TOR status bit is set and the trigger is ignored.
vinegar. Also, this trigger does not interfere with the running AD conversion.

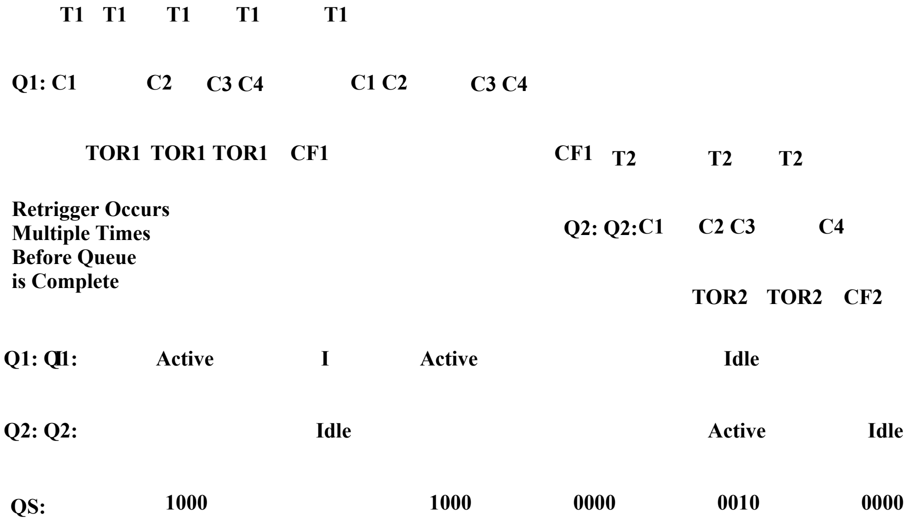


Figure 31 (b) CCW Priority Situation

In the case of Figure 31 (b), if multiple new triggers occur while the queue is running, it will occur during execution.
The rigger event sets the TOR status bit and its trigger is ignored. also,
This trigger does not interfere with the running AD conversion. Trigger detected after execution
If so, the AD conversion will be performed again.

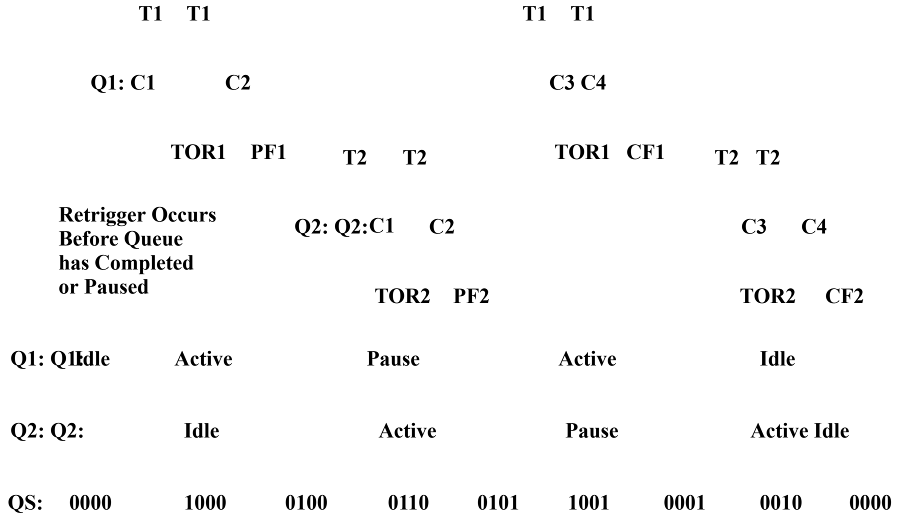

T1 T1 T1 T1
Q1: C1 C2 C3 C4
TOR1 PF1 T2 T2 TOR1 CF1 T2 T2
Retrigger Occurs
Before Queue
has Completed
or Paused
Q2: Q2:C1 C2 C3 C4
TOR2 PF2 TOR2 CF2
Q1: Q1:Idle Active Pause Active Idle
Q2: Q2: Idle Active Pause Active Idle
QS: 0000 1000 0100 0110 0101 1001 0001 0010 0000

Figure 31 (c) CCW Priority Situation

The case in Figure 31 (c) is when the pause function is used. The running trigger before the pause state is
It will be ignored as before.

T1

Q1: C1 C2 C3 C4

T2 CF1

Queue 2 is Triggered
while Queue 1
is Active

Q2: Q2: C1 C2 C3 C4

CF2

Q1: Q1: Idle Active Idle

Q2: Q2: Idle Triggered Active Idle

QS: 0000 1000 1011 0010 0000

Figure 31 (d) CCW Priority Situation

In the case of Figure 31 (d), queue 1 is terminated if queue 2 is triggered while queue 1 is running. Queue 2 processing will start later.

T1 T1

Q1: C1 C2 C3 C4

T2 T2 PF1 T2 T2 CF1

Queue 2 is Triggered
Multiple Times
while Queue 1
is Active

Q2: Q2: C1 C2 C3 C4

TOR1 PF2 TOR1 CF2

Q1: Q1: Idle Active Pause Active Idle

Q2: Q2: Idle Trig Active Pause Trig Active Idle

QS: 0000 1000 1011 0110 0101 1001 1011 0010 0000

Figure 31 (e) CCW Priority Situation

In the case of Figure 31 (e), TOR is set when multiple queue 2 triggers occur while queue 1 is running. Will be At this time, there is no effect on the execution of queue 1. Trigger is detected when the cue is in the paused state If known, the queue corresponding to that trigger will start running.

T1 T1

Q1: C1 C2 C3 C4

PF1 T2 CF1

Queue 1 Suspends
Queue 2, RES = 0

Q2: C1 **C1** **C2 C3** **C4**

CF2

Q1: Q1: Idle **Active** **Pause** **Active** **Idle**

Q2: Q2: Idle **Active** **Suspend** **Active** **I**

QS: 0000 **1000** **0100** **0110** **1010** **0010**

Figure 31 (f) CCW Priority Situation

In the case of Figure 31 (f), the execution of queue 2 is stopped when the trigger of queue 1 occurs while the execution of queue 2 is in progress. And the execution of queue 1 starts. Queue 2 is automatically restarted after queue 1 is finished.
When the RES bit = 0, the restart point resumes from the aborted trigger point.

Queue 1 Suspends
Queue 2 with Both
Queues using **T1** **T1**
Pause,
RES = 0

Q1: C1 **C2** **C3 C4**

T2 **PF1** **T2** **CF1**

Q2: C1 **C1** **C2** **C3** **C3** **C4**

PF2 **CF2**

Q1: Q1: Idle **Active** **Pause** **Active** **Idle**

Q2: Q2: Idle **Active Suspend** **Active** **Pause** **Act** **Suspend** **Active I**

QS: 0000 **0010** **1010** **0110** **0101** **0110** **1010** **0010**

Figure 31 (g) CCW Priority Situation

In the case of Figure 31 (g), queue 2 starts executing from the paused state by the second trigger.
If the trigger for queue 1 occurs at that time, the execution of queue 2 is stopped and the execution of queue 1 starts.
Then, after queue 1 ends, queue 2 automatically resumes execution from the second trigger point. RES bit
When = 0, the restart point restarts from the canceled trigger point.

Page 203

T1 **T1**

Q1: C1 **C2** **C3 C4**

PF1 **T2** **CF1**

Queue 1 Suspends
Queue 2, RES = 1 **Q2: C1** **C2 C3** **C4**

CF2

Q1: Q1: Idle **Active** **Pause** **Active** **Idle**

Q2: Q2: Idle **Active** **Suspend** **Active** **I**

QS: 0000 1000 0100 0110 1010 0010

Figure 31 (h) CCW Priority Situation

In the case of Figure 31 (h), the execution of queue 2 is stopped when the trigger of queue 1 occurs while the execution of queue 2 is in progress.
And the execution of queue 1 starts. And after queue 1 ends, queue 2 automatically resumes execution.
It will be. When the RES bit = 1, resume from the aborted queue command.

Queue 1 Suspends Queue 2 with Both Queues using Pause, RES = 1

T1 T1

Q1: C1 C2 C3 C4

T2 PF1 T2 CF1

Q2: C1 C2 C3 C4

PF2 CF2

Q1: Q1: Idle Active Pause Active Idle

Q2: Q2: Idle Active Suspend Act Pause Active Suspend Act I

QS: 0000 0010 1010 0110 0101 0110 1010 0010

Figure 31 (i) CCW Priority Situation

In the case of Figure 31 (i), queue 2 starts execution from the paused state due to the second trigger.
If the trigger for queue 1 occurs at that time, the execution of queue 2 is stopped and the execution of queue 1 starts.
Then, after queue 1 ends, execution resumes automatically from the queue command that was aborted in queue 2.
When the RES bit = 1, resume from the aborted queue command.

Page 204

Queue 1 Suspends Queue 2 with Both Queues using Pause, Queue 2 is Retriggered RES = 0

T1 T1

Q1: C1 C2 C3 C4

T2 T2 PF1 T2 T2 CF1

Q2: C1 C1 C2 C3 C3 C4

TOR2 PF2 TOR2 CF2

Q1: Q1: Idle Active Pause Active Idle

Q2: Q2: Idle Active Suspend Active Pause Act Suspend Active I

QS: 0000 0010 1010 0110 0101 0110 1010 0010

Figure 31 (j) CCW Priority Situation

In the case of Figure 31 (j), if queue 2 detects a trigger on queue 2 when it is suspended, TOR will
It will be set and the trigger will be ignored. After queue 1 is finished, queue 2 is the start of operation.
Resume from the gar point. When the RES bit = 0, the restart point resumes from the aborted trigger point.

Queue 1 Suspends

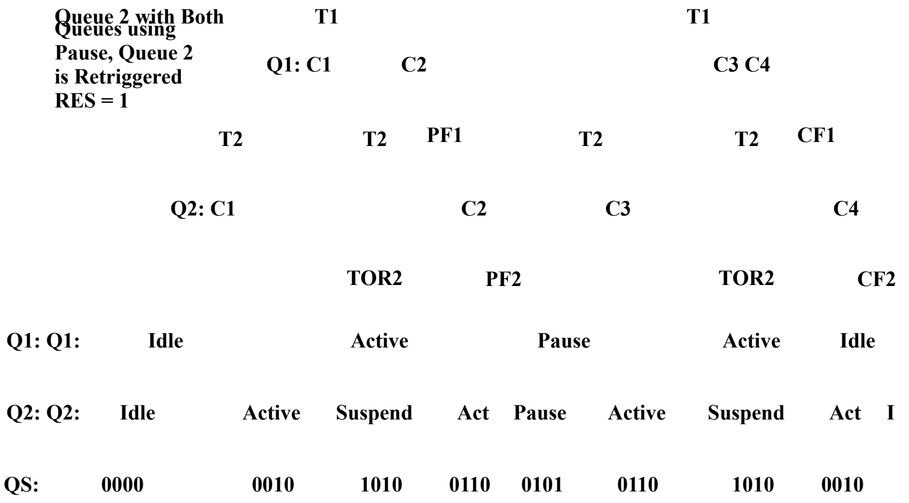


Figure 31 (k) CCW Priority Situation

In the case of Figure 31 (k), if Queue 2 detects a Queue 2 trigger when Queue 2 is Suspend, TOR will It will be set and the trigger will be ignored. Queue 2 is abandoned queue frames after queue 1 ends Resume from When the RES bit = 1, resume from the aborted queue command.

Chapter 8
Serial with RAM buffer Peripheral interface

The RAM-buffered Serial Peripheral Interface (RSPI) is a full-duration synchronous serial interface. Communicates with external peripherals and other MCUs via the computer.

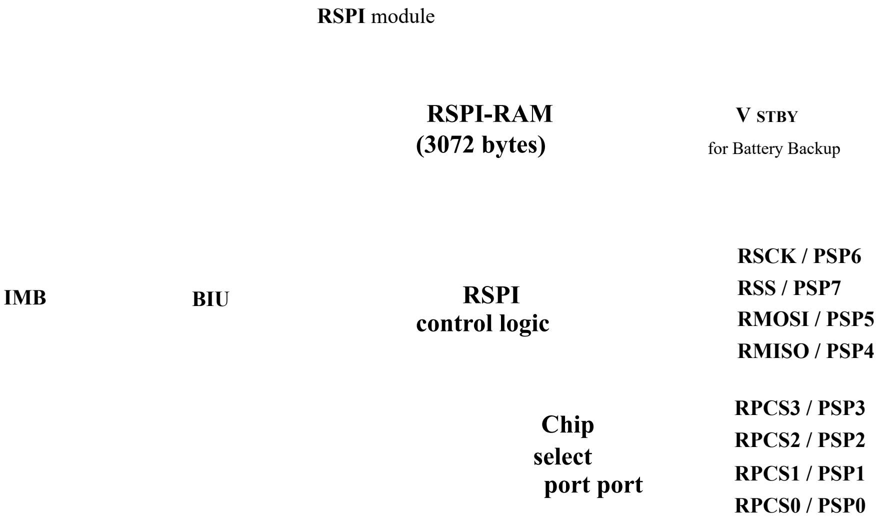


Figure 32 RSPI block diagram

8.1 Summary

RSPI has a serial communication channel with 3072 bytes of data RAM that the CPU can access at any time. There is one. RSPI is a single send / receive using two sets of send and receive RAM pointers. Provides modes and wraparound transmit / receive modes.

□ Special length

・3072 bytes RAM buffer
・Transfer mode:
    -Full double (3-wire or 4-wire) / Half-double (2-wire or 3-wire)

    -Single transfer / wraparound transfer
    -Can generate interrupts to CPU
    -Pattern generation / pattern capture function available
-Slave / master / multi-master support:
    -Slave addressing
    -Change / monitor slave RAM pointer
    -Mediation between multiple masters
• Controllable transfer parameters:
    -Baud rate (24.4kbps to 6.25Mbps)
    -Word length (8/16 bits)
    -Delay time after transmission (81.9ms to 21.0ms at 24.4kbps, 320ns to 81.9ms at 6.25Mbps)
    -2 pairs of RAM pointers (Start / End / Current)
・Advanced data transfer function:
    -Word framing check
    -Blind carbon copy (BCC) generation and checking
• Cyclic Redundancy Check (CRC) Code
・Horizontal parity code
    -Three-point sampling for serial clock and data reception
    -Master clock can be skewed by user specified value
    -Acnoridge and retry function
    -Partial RAM write protection
-PCS (Programmable Peripheral Chip Select):
    -Select up to 16 peripheral chips with 4 pins

Table 113 RSPI address map

Address Address	15 8	7 0
$ YFF600	RSPI MODULE CONFIGURATION REGISTER (RSMCR)	
$ YFF602	RSPI GLOBAL TEST (RSTEST)	
$ YFF604	RSPI INTERRUPT LEVEL (RSILR)	RSPI INTERRUPT VECTOR (RSIVR)
$ YFF606	RSPI RAM BASE ADDRESS (RSRBAR)	
$ YFF608- $ YFF60E	Reserved	
$ YFF610	RSPI PIN ASSIGNMENT (RSPAR) / RSPI DATA DIRECTION (RSDDR)	PORT SP DATA (RSPDR)
$ YFF612	RSPI CONTROL REGISTER 0 (RSCR0)	
$ YFF614	RSPI CONTROL REGISTER 1 (RSCR1)	
$ YFF616	RSPI CONTROL REGISTER 2 (RSCR2)	
$ YFF618	RSPI CONTROL REGISTER 3 (RSCR3)	
$ YFF61A	RSPI COMMAND REGISTER (RSCMD)	
$ YFF61C	RSPI INDEX REGISTER 0 (RSIX0)	
$ YFF61E	RSPI INDEX REGISTER 1 (RSIX1)	
$ YFF620	RSPI INDEX REGISTER 2 (RSIX2)	
$ YFF622	RSPI INDEX REGISTER 3 (RSIX3)	
$ YFF624	RSPI INDEX REGISTER 4 (RSIX4)	
$ YFF626	RSPI INDEX REGISTER 5 (RSIX5)	
$ YFF628	RSPI STATUS REGISTER (RSSR)	

$ YFF62A	RSPI BCC REGISTER 0 (RSBC0)
$ YFF62C	RSPI BCC REGISTER 1 (RSBC1)

Table 113 RSPI address map

Address Address	15	8	7	0
$ YFF62E	RSPI DATA SHIFT REGISTER (RSSFT)			
$ YFF630	RSPI LOCAL TEST REGISTER (RSLTST)			
$ YFF632- $ YFF636	Reserved			
$ YFF638	RPCS PIN ASSIGNMENT (RSCSPAR) / RPCS DATA DIRECTION (RSCSDDR)		RPCS PORT DATA (RSCSPDR)	
$ YFF63A- $ YFF63E	Reserved			

Y = M111. However, M is the logical state of the SLIMCR mode map (MM) bits. In this MCU
Y must be equal to $ F. If M is cleared, until a reset occurs
You cannot access the IMB module. M can only be written once after reset
vinegar.

The following table summarizes the functionality of RSPI pins when not configured for general purpose I / O. RSPI The Data Direction Register (RSDDR) and RPCS Data Direction Register (RSCSDDR) assign functions to each pin.

Table 114 Features of RSPI Pins

	Pin	Mode Mode	I / O	Pin Function
RSPI Pins	RMISO	Master	I	Serial Data Input to RSPI
		Slave	O	Serial Data Output from RSPI
	RMOSI	Master	O	Serial Data Output from RSPI
		Slave	I	Serial Data Input to RSPI
	RSCK	Master	O	Clock Output from RSPI
		Slave	I	Clock Input to RSPI
	RSS	Master	O	Frame Indication Output
		Slave	I	Slave Select
	RPCS [3: 0]	Master	O	Chip Select Output
		Slave	I / O	General I / O

8.1.1 General features

Users have many characteristics of RSPI's data transfer such as data size, baud rate, bit order, etc.
You can control the sex. You can also use RSPI to master device, slave device, or multiple devices.
It can communicate with the master device. All features of RSPI are fully functional with traditional SPI timing
For compatibility, it can also be used for communication over SPI / QSPI channels.

8.1.1.1 Master or slave operation

RSPI operates in either master mode or slave mode. In master mode
RSPI controls both the serial clock and communication operation. In addition, address slave select
In mode, the master initiates a command transaction and the slave device's RAM pointer.
You can observe and control parameters such as values and half / full / full settings.

8.1.1.2 Half-duplex and full-duplex operation

You can use both half-duplex and full-duplex transfers. Serial clock common to both transmission and reception

(SCK) is used.

8.1.1.3 bytes / word transmission word length

RSPI is between RSPI RAM and the serializer in 8-bit (byte) or 16-bit (word) increments.
You can load / store data.

8.1.1.4 Configurable CPU endian types and data Ta shift direction

RSPI supports big endian and little endian byte allocation formats
I am. The data shift direction depends on the type format of the selected endian.
Become. Most significant bit (MSB) for big endian format, Little Endian
The least significant bit (LSB) is the first in the format.

8.1.1.5 Delay time after sending programmable word

There is a pause period of 2 to 512 SCK clock cycles between each word transfer, if the period is set appropriately.
RSPI can also communicate with traditional SPI.

8.1.2 RSPI RAM and data pointers

8.1.2.1 RSPI data RAM

RSPI has 3072 bytes of data RAM. This data RAM is similar to a traditional RAM module
Can be used for. Therefore, the user is in RAM with what system data (such as stack space).
Can also be stored, and programs can also be stored. Figure 33 shows the MC68HC (9) 16Y5 / 916Y6 RSPI.
Shows the RAM address map. The CPU makes read access to the unmounted RAM area.
But the result is always uncertain. In addition, the CPU makes write access to the unmounted RAM area.
However, it does not affect RSPI.

In general, the CPU can access RSPI RAM with two system clocks. But inside the CPU and RSPI
If the circuit tries to access the RSPI RAM at the same time, the CPU will access it until the RSPI finishes accessing it.
Will be postponed. As a result, the CPU access bus bandwidth becomes the baud rate of the RSPI in transit.
It will be reduced accordingly.

The contents of RSPI RAM are not initialized at every system reset, and even during power down, Vstby Pi
It is maintained by supplying backup power to the computer.

CPU Addr. Big (Little)		IMB Addr. Offset	
000 (001)	001 (000)	$ XX000 [2]	RSBAR pointing address
002 (003)	003 (002)	$ XX002	Mounting space [1] (3072 Bytes)
BFE (BFF)	BFF (BFE)	$ XXBFE	
C00 (C01)	C01 (C00)	$ XXC00	Unimplemented space (1024 Bytes)
FFE (FFF)	FFF (FFE)	$ XXFFE	

16 bit
(Word)

Note: 1.RSPI is 4kBytes SRAM
Is possible to have
However, MC68HC (9) 16Y5 /
RSPI installed in 916Y6
Is limited to 3072Bytes
We are here.
2. $ XX is by RSRBAR
It can be decided.

Figure 33 RSPI RAM address map

8.1.2.2 Data index pointer

RSPI is two sets that point to the start and end addresses of transmit / receive data blocks in RSPI RAM space.
There is a data pointer for. Also, the current position in the transmit / receive data block in the RSPI RAM space.
There are also two current pointers to indicate.

Each register is 12 bits long and uses it to allow RSPI to access all 3072 bytes of the RSPI RAM area.
You can do it. The index register can contain any combination of address values. Tsu
In other words, the overlap between the transmit area and the receive area is fully supported. Figure 34 shows an example of overlap .

To do. The current pointer (CURT / CURR) prevents the CPU from accidentally overwriting the data currently being sent.
You can refer to it. This guarantees coherency of the data during the transaction.
increase.

Figure 34 shows an example configuration using these pointers .

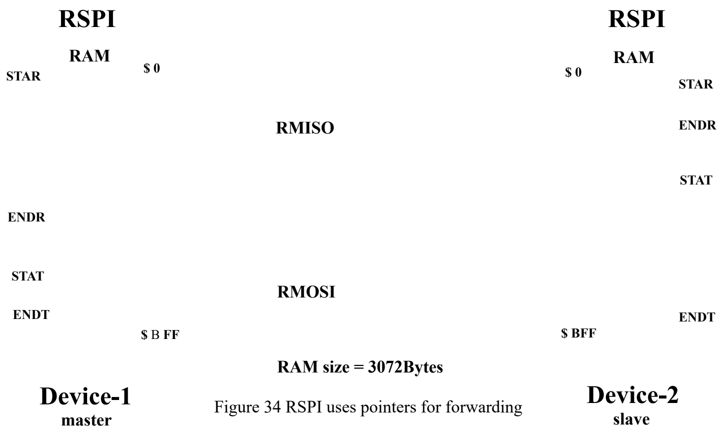


Figure 34 (a) RSPI uses pointers for transfers (without overlap)

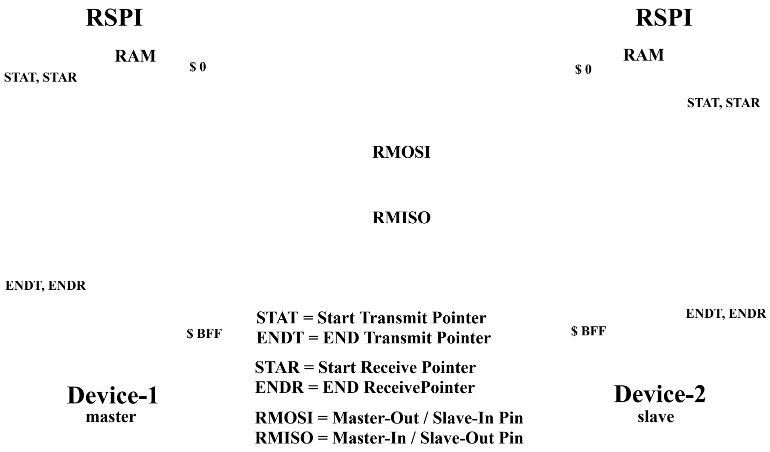

RSPI
RAM
STAT, STAR
$ 0
RMOSI
RMISO
ENDT, ENDR
$ BFF
STAT = Start Transmit Pointer
ENDT = END Transmit Pointer
STAR = Start Receive Pointer
ENDR = END ReceivePointer
RMOSI = Master-Out / Slave-In Pin
RMISO = Master-In / Slave-Out Pin
Device-1
master
RSPI
RAM
$ 0
STAT, STAR
ENDT, ENDR
$ BFF
Device-2
slave

Figure 34 (b) RSPI uses pointers for transfers (when overlapping)

8.1.2.3 Partial write protection for RSPI RAM

RSPI provides write protection for all / part of RSPI RAM. The protected area is always available to the CPU.
You can access and change it. However, the RSPI communication circuit cannot be written to the protected area and is protected.
Only area reads are allowed. System parameters such as stack pointers in the protected area
It can also be placed in. This allows other external devices to observe the data through serial communication.
can. This is useful for system debugging and more.

8.1.2.4 Wraparound transfer

Wraparound allows the user to continuously block data blocks in RSPI RAM without CPU intervention.
You can send it. This is a BCC inspection & retry option, especially for single data frame mode.
It is convenient when you use it.

With wraparound enabled, the RSPI pointer architecture sends and receives each
Allows communication with an independent ring buffer system. This allows the user to use RSPI
It is possible to realize virtual shared RAM between multiple MCUs connected by up to 4 serial communication lines.
I can do it. Figure 35 shows how to achieve this.

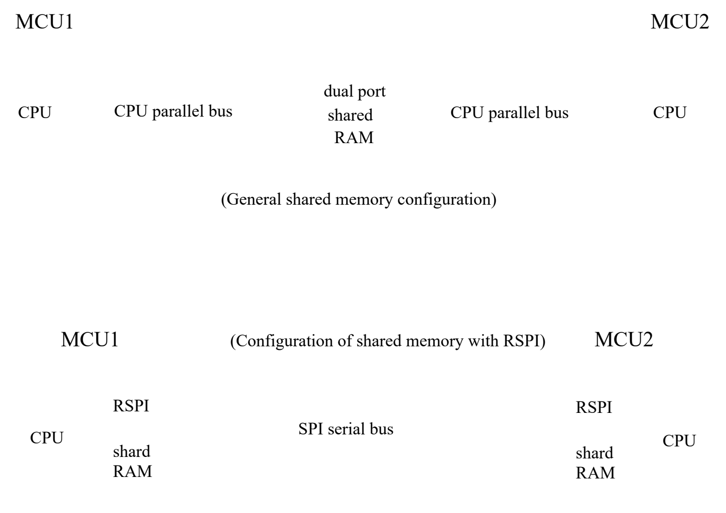


Figure 35 Virtual shared RAM with RSPI

8.1.3 Slave device selection

The master RSPI must select one or more slaves to communicate with. RSPI is a pin
There are two types of slave selection methods, slave select and address slave select.
Each selection method is explained below.

8.1.3.1 Pin Slave Select Mode

This slave selection method is fully compatible with traditional SPI / QSPI. The slave RSS port is
Once slaughtered, the slave will be selected. Select one or more slaves out of multiple slaves
To do this, each slave needs a slave select control line.

8.1.3.2 Address Slave Select Mode

This slave selection process is typically used exclusively for RSPI, but traditional SPI should also use it.
I can. The master sends a slave address to blow which slave to communicate with.
Docast The unique chip address of the slave device is the slave transmitted by the master.
If the address matches, the slave is selected. In address-slave select mode
The master can dynamically select one or more slaves. Also, this slave select mode
Now, let's use the master arbitration procedure described later in this chapter to dynamically change the master device assignment.
Can be done.

8.1.4 Multi-master support (serial bus support) Star mediation)

In addition to the traditional mode fault detection, RSPI has a new master rights automatic arbitration mechanism and protocol.
Provide. Master rights arbitration is a special case of command transfer. Use master rights arbitration protocol
By doing so, the user dynamically changes the master and minimizes software intervention.
can do.

8.1.5 Data frame mode

With RSPI, you can choose between two data frame modes, multiple and single.
to come. The main difference between the two modes is within each frame where the acknowledge and retry functions are performed.
The number of data in. The following is a brief explanation.

8.1.5.1 Multiple data frame mode

In multiple data frame mode, multiple bytes / word while RSS is asserted
Data is transferred continuously. This allows for very efficient transfers.

Receive before the BCC check is performed, even if the acknowledge and BCC check features are enabled

Note that the data is stored in RSPI RAM. That is, the received data contains an error
But I don't know until the end of the data frame.

For transfers with both BCC and acknowledgment features disabled, the data remaining in the serializer will be
It is stored in RSPI RAM regardless of the data bit size when RSS is negated. this
Therefore, the slave RSPI can receive data that is not a multiple of 8 bits.

8.1.5.2 Single data frame mode

In single data frame mode, each byte / word is checked for errors before RSPI RAM.
Will be stored in. If an error is detected, that data will be discarded. Therefore, in RSPI RAM
The data it contains is guaranteed to be error-free. In single data frame mode,
With poor, acknowledge, and retry features, you can work with multiple RSPIs without CPU intervention.
Can provide RAM.

8.1.6 Pattern generation / pattern capture function

The master RSPI transfers / receives a set of data strings with no latency between data bytes / word transfers.
It allows you to generate and capture patterns. Wraparound is enable
If so, RSPI can perform pattern generation / capture indefinitely.

8.1.7 Chip Select Port

RSPI has a 4-bit port for slave chip selection, which is equivalent to QSM's QSPI submodule.
There are (RPCS0 to RPCS3). These ports are used for communication with some traditional peripherals
Can be used. Some advanced features of RSPI, such as address / slave selection and command forwarding, are available on the PCS chip.
It cannot be used in the recto operation.

When the chip select function using PCS is enabled, the master RSPI becomes the RSPI specified in CURT.
Loads the high-order byte (D15-D8) of the data located at the RAM address into the serializer. Lower buy
The least significant 4-bit (D3-D0) is loaded into a 4-bit latch specially prepared for PCS operation.
Received byte data is always stored in the high byte of RSPI RAM.

In addition to traditional PCS operation, RSPI extends the least significant 3 bits (D7-D5) of the lower byte for PCS operation.
Used as an attribute bit. These bits are CSFIN, SSA, CONT.

CSFIN

0 = PCS operation continues.
1 = Normal data transfer with no PCS operation continues.

If CSFIN = 1, the master RSPI will send the PCS after the transfer of the currently loaded serial data is complete.
Ends the used chip select operation. At that point, the communication currently in progress has not ended yet.
If the master RSPI does not perform PCS operations and is in normal data frame mode
Continue the transfer. When CSFIN = 0, the chip select operation using PCS continues.

SSA

0 = RSS is negated (drive 1).
1 = RSS is asserted (drive 0).

If SSA = 1, RSS outputs an “L” level during PCS byte transfer. If SSA = 0, RSS is “H” level
Is output.

CONT

0 = Returns the PCS value to the state defined in RSCSPDR after the transfer is complete.
1 = Keep the chip select code asserted after the transfer is complete.

If CONT = 1, RSPI keeps the same value on the PCS port for consecutive pauses in serial data transfer.
Drive to. If CONT = 0, RSPI sets the PCS port to the value defined in RSCSPDR during that period.
Drive

Figure 36 shows an example of how to use the attribute bits.

for Dev.1
m Bytes

for Dev.2
n Bytes

for slave MCU
k Words

RPCS0

DTL

DTL

RPCS1

RSS

wrap

wrap

0

Reserved ————

3 Dev.1 Dev.1 Dev.1 Dev.2 Dev.2 Dev.2 Dev.1
Four

1: CS at DTL
0: PD at DTL
CONT

7 0000 0100 0000 0100 0110 1000 0000
8

1: RSS Assert
0 :: RSS Negate
SSA

DATA3-k DATA3-1 DATA3-k

1: CS Finish
0: CS Continue
CSFIN

DATA1-1 DATA1-m DATA2-1 DATA2-n DATA1-1

CS Attribute Bits

15

ENDT STAT

RSPI-RAM Contents

ENDT

8.1.8 Enhanced data retention function

Data reliability is important because RSPI can send and receive long rows of data at high speeds. RSPI is BCC Check code, multipoint sampling, programmable SCK clock, and more for reliability

Has several noise immunity features for.

8.1.8.1 Framing error checking

Slave RSPI goes into an unpredictable operating state with noise on the RSCK and RSS lines during communication.
May occur. This type of communication error can be classified as a "frame size error". vinegar
A frame size error in the rave breaks the operational state synchronization between the master and slave.
To handle frame size errors, the slave RSPI has a frame size check feature.
I am. Frame size checking is done on slave RSPI, except in level sensitive RSS mode.
I will. In normal edge-sensitive RSS mode communication, the slave RSPI is for each data frame.
Counts the number of RSCK pulses inside. The number of RSCK pulses in the counted data frame is the expected value
If they are not equal, the slave detects a framing error and immediately discontinues operation.

8.1.8.2 Block Inspection Code (BCC): LPC / CRC-CCITT / ITU-T

RSPI has two types of block check codes: horizontal parity check (LPC) and cyclic redundancy check (CRC).
I will. In both checks, the sending RSPI adds a 16-bit BCC code after the main data transfer.
vinegar. Therefore, the receiving RSPI compares the transmitted BCC code with the internally generated BCC code.
Check the integrity of the data. BCC code is always 16 bits long, regardless of data size.
The parity type for LPC data checks can be set in 4-bit increments.

8.1.8.3 Multipoint sampling of received data

RSPI provides a 3-point multisampling noise filter on the receive port. fill
The ports to be loaded are the slave RSCK, RSS, and RMISO lines and the master RMOSI line. fill
Ta samples each line three times and takes a majority vote. This filter has a noise width of 1/4 SCK circumference.
Please note that it is valid only for less than the period.

8.1.8.4 Master Programmable SCK Clocks queue

RSPI provides a programmable SCK skew feature to compensate for slave response delays and propagation delays.
increase. Enable multisampling of RSCK lines to perform high-speed data transactions
This slows down the response of the slave. As a result, sufficient data setup time cannot be maintained.
You may. When the master RSPI receives the delayed data, it internally delays the SCK and responds to it.
The SCK delay width can be from 0 to 3/8 SCK cycles and is user-configurable.

8.1.8.5 Acknowledgment and retry function

An acknowledgment cycle can be added to the RSPI to see if the transmitted data was received correctly.
Or, that is, it tells other devices if the received data passed the BCC check. Acnor
The system consists of 16 bits, and the received data valid acknowledge (RCVAL) and slave valid acknowledge (RCVAL).
It is divided into two parts (SLVAL). The former checks the validity of received data, and the latter is for command transfer.
Check the response of the slave. Each effective response received to provide noise immunity
The value of the Knoridge Code is determined by an 8-point majority voting method.

Since RSPI has a retry mechanism, the acknowledge is detected by the received data valid acknowledge.
If not, the master replays the failed data frame a certain number of times before interrupting communication.
You can send it.

8.1.9 Command transfer

In address-slave select mode, the master RSPI initiates command transfer and is used by RSPI.
You can take advantage of many powerful features possible. For general command transfer, use the opcode (setting the corresponding bit).
), The relevant operand word, and, if applicable, the data word (the data to be transferred).
Will be made. By issuing the appropriate command transfer, the master RSPI can do so, as shown in Figure 37.
Specify one or more slaves for subsequent communication and point to the RAM pointer of the selected slave RSPI.
It can be changed. Therefore, due to command transfer, the master RSPI is required by the master.
You can read and write data columns in any area of the RAVE RSPI RAM. CPU intervenes on the slave side
There is no need to.

8.1.9.1 Slave addressing

In address-slave select mode, the master can select the slave it wants to communicate with.
increase. The command transfer opcode contains the 4-bit slave chip address field and an ad.

It always contains 2 bits of the lessing mode field. Slave chip address fee
In Ludo, the master can select up to 16 slaves. The addressing mode field is
Used to select a combination of slaves according to the slave chip address field.

Each slave waiting for communication checks whether its chip address is selected as the master.
I will If its chip address is selected, the slave will communicate with the master on subsequent transfers.
I can trust you.

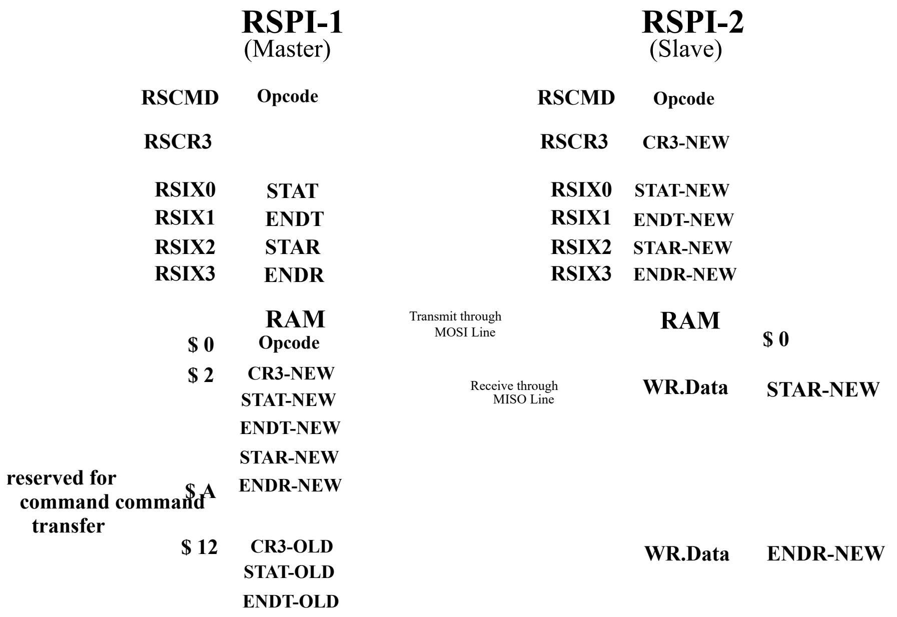
8.1.9.2 Slave data in from master RSPI Manipulating the dex pointer

The master can modify the contents of any slave's data index pointer. child
This allows the master to specify and change the location of data to and from the slave.
I can. The slave RSPI protects the master RSPI from changing the index pointer.
It also has options for. If you are using the protection option, go through the acknowledge cycle
You can then notify the master of this.

8.1.9.3 Monitoring and changing the slave status by the master RSPI Further

The master can monitor the important control registers of any slave. This conte
The role register contains master right request, master right grant, index pointer write protection, and so on.
Reve-specific chip address, full / half-duplex, wraparound enable / disable,
And the current state of the slave for RAM data protection.

Note: The master can change full / half dual and wraparound enable / day.
Only sable. Changes are made by issuing a command transfer.



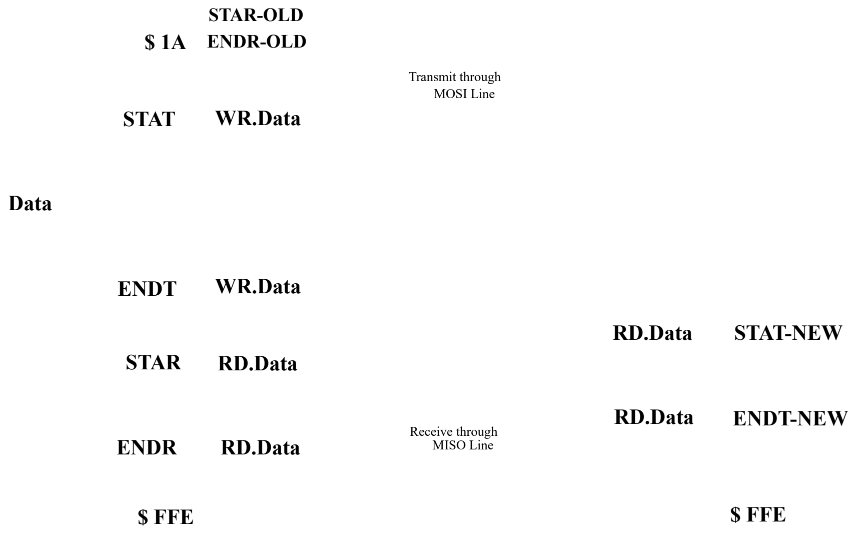


Figure 37 Command transfer data flow

Page 217

8.2 RSPI register

RSPI registers consist of RSPI global registers and RSPI local registers.

8.2.1 Global register

The RSPI global register is the RSPI module configuration register (RSMCR), RSPI.
Global test registers (RSTEST), RSPI interrupt registers (RSILR, RSIVR), and RSPI
It consists of a RAM-based address register (RSRBAR).

RSMCR—RSPI module configuration register $ YFF600

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	FRZ1	FRZ0		NOT USED			RST	SUPV		NOT USED			IARB		
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

STOP—Stop mode select (stops the clock and powers down the circuit)
0 = Normal RSPI clock operation.
1 = Stop RSPI clock operation.

FRZ1 — Freeze Enable 1
0 = Ignore the FREEZE signal of IMB.
Stop RSPI when 1 = IMB FREEZE signal is output (at the forwarding boundary).

FRZ0—Freeze enable 0
Reserved for future enhancements.

RST— reset channel
Using this bit, the user can use the state of the RSPI global register or other modules on the IMB.
You can reset the RSPI channel without affecting the state. If you write “1” to this bit,
Resets the internal state of RSPI channels other than BIU and returns the RSPI channel to the state after the master reset.
To do. All RSPI local registers are initialized when RST is asserted. This bit
Is always returned as “0”.

SUPV— Supervisor space
0 = unconstrained spatial access

1 = Supervisor spatial access

This bit has no meaning, as CPU16 only operates in supervisor mode.

IARB — Interrupt arbitration level

This 4-bit encoded field contains the interrupt arbitration number for this RSPI.
increase. This field is initialized to the non-arbitration state ($ 0) on system reset.

RSIL—RSPI interrupt level register $ YFF604

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
		NOT USED				ILRSPI					RSIVR				
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

ILRSPI—RSPI interrupt level

ILRSPI determines the interrupt level for RSPI interrupts.

Page 218

RSIVR—RSPI interrupt vector register $ YFF605

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
				ILRSPI						INTV7: 2				INTV1	INTV0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Upon system reset, RSIVR initially defaults to $ 0F, which corresponds to the uninitialized interrupt vector in the CPU exception table.
Will be set. This vector is selected until the user writes the RSIVR. RSIVR is RSPI's
Program to one of the user-defined vectors (see CPU specification for more information) during initialization
You need to

Table 115 RSPI interrupt vectors

Bit name	INTV [7: 2]	INTV1	INTV0
RSPI Vector	Programmable	0	0
	Reserved for future expansion	0	1
		1	0
		1	1

INTV7: 2—Interrupt vector [7: 2]

This field specifies the upper 6 bits of the interrupt vector.

INTV1: 0— Interrupt vector [1: 0]

Write operations to these 2 bits are invalid. Currently, these bits are set when an interrupt is generated.
It will always be “00”.

RSRBAR—RSPI RAM base address register $ YFF606

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12		NOT USED		RAMDS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

A23 to A12—RSPI RAM base address

These bits specify the address lines A23 through A12 of the base address of the RSPI RAM array.
To do. Use these bits to place the RSPI RAM array anywhere on the 4K byte boundary in the memory map.
It can also be placed in. You can write as many times as you like in test mode, but in normal operating mode
So this register can only be written by the CPU once after a system master reset.

RAMDS—Disable CPU access to the RAM array

0 = CPU access to the RSPI RAM array is enabled.

1 = CPU access to the RSPI RAM array is disabled.

RAMDS is the only bit that indicates whether the CPU can access the RSPI RAM. This bit
If is set to 1, CPU access to the RSPI RAM array is within after a system master reset.
It is disabled by the circuit. The contents of the RSPI RAM base address field is CPU

RAMDS is automatically cleared when updated by access to the RSPI RAM array.
CPU access is enabled. Once RAMDS is cleared, system master reset
Cannot be reset.

RSTEST—RSPI Global Test Register $ YFF602

RSTEST is used to test RSPI in the factory. The MCU can access this register
Only while is in factory test mode.

8.2.2 Local registers

Local registers are RSPI pin control registers (RSPAR, RSDDR), RSPI port day.
Data register (RSPDR), RSPI control register (RSCR0-3), RSPI command register (RSCMD),
RSPI index register (RSIX0-5), RSPI status register (RSSR), RSPI block check
Code register (RSBC0 ~ 1), RSPI data shift register (RSSFT), RSPI local test
Registers (RSLTST) and RSPI chip select pin control registers (RSCSPAR, RSCSDDR)
Consists of.

8.2.2.1 RSPI pin control register

The RSPI pin control registers determine the use of the pins that form the parallel port on the MCU.
increase. These pins are commonly used by serial communication subsystems, but on a pin-by-pin general purpose input / output (I / O).
O) It can also be assigned as a port.

RSPAR—RSPI pin assignment register $ YFF610

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	RSS	RSCK	RMOSI	RMISO		RSDDR						RSPDR				
RESET:																
	0	0	0	0	0	0	0	0	U	U	U	U	U	U	U	U

RSS—RSPI Slave Select
RSCK—RSPI serial clock
RMOSI—RSPI master out slave in
RMISO—RSPI master in slave out
0 = General purpose I / O port
1 = RSPI functional port

RSPAR determines whether the RSPI port is used as an RSPI functional port or a general purpose I / O port.
To decide. If the RSPAR bit is set, the corresponding port is assigned as an RSPI function pin.
You can do it. On the other hand, if the RSPAR bit is cleared, the corresponding port is a general-purpose I / O port.
Will be assigned. All RSPARs during RSBSY = 1 to prevent RSPI from behaving unpredictably.
Bits are protected from CPU write access. The contents of RSPAR can be changed when RSBSY = 0.

For communication other than pattern generation / capture operation, the RSCK bit in RSPAR must be set.
Must be. If the pattern generation / capture operation does not require the SCK clock, the RSCK pin
You can clear the allocation control bits to disable SCK output.

The RSPAR setting does not specify the full / half dual communication mode. That is, of RSPAR
The RMOSI and RMISO bits enable / disable the transmit / receive function inside RSPI.
Instead, connect / disconnect the external port of RMOSI / RMISO to the internal communication circuit. RSPAR bit charge
Instead, the TXE and RXE bits of RSCR3 define the communication mode of their respective transmit and receive circuits.
vinegar. Therefore, the RMOSI / RMISO bit setting of RSPAR is the same as the internal communication mode setting by TXE / RXE.
You have to respond.

When the acknowledgment function is enabled (ACKE = 1), set the bits in the corresponding RSPAR.
And, regardless of the TXE / RXE value, you must enable the RSPI function on both the RMOSI line and the RMISO line.
Must be.

Level Sensitivity In SS mode, the RSS bits in the RSPAR of the master RSPI are cleared.
If so, you can use the RSS port as a mode-fault input port.

RSDDR—RSPI data direction register $ YFF610

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
		RSPAR			PSP7	PSP6	PSP5	PSP4				RSPDR				
RESET:																
	0	0	0	0	0	0	0	0	U	U	U	U	U	U	U	U

PSP7 — General purpose I / O port 7
PSP6 — General purpose I / O port 6

PSP5 — General purpose I / O port 5

PSP4—General purpose I / O port 4

0 = input
1 = output

The RSDDR bit determines the data orientation of the RSPI external port assigned as a generic I / O pin. vinegar. The RSDDR bit does not affect the direction of the port to which the RSPI communication function is assigned.

RSPDRRSPI port data register $ YFF611

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RSPAR				RSDDR				EPSP7	EPSP6	EPSP5	EPSP4	PSP7	PSP6	PSP5	PSP4
RESET:															
0	0	0	0	0	0	0	0	U	U	U	U	U	U	U	U

EPSP [7: 4] — External port status bit

Reading these 4 bits always returns the actual state of the external pin. Writing to these bits Included access is invalid.

PSP [7: 4] — Port output data bits

These bits contain the data to be output to the port assigned as general purpose output. is. When these bits are read, whether the port is assigned to an output or an input Regardless of, the level that is always stored is returned. The contents of these bits are used during RSPI communication. It may be output to the corresponding port. That is, the data transmission port is used for half-duplex communication. If not, the state of the corresponding port output data bit is printed to that unused port.

Table 116 RSPI pin control

RSPAR / RSDDR / RSPDR bit	Port SP signal	RSPI pin
RSS / PSP7 / EPSP7	PSP7	RSS
RSCK / PSP6 / EPSP6	PSP6	RSCK
RMOSI / PSP5 / EPSP5	PSP5	RMOSI
RMISO / PSP4 / EPSP4	PSP4	RMISO

8.2.2.2 RSPI control register

The RSPI module consists of four RSPI control registers. Unpredictable behavior To prevent this, most control register bit changes are allowed when RSBSY = 1. It is generally allowed when RSBSY = 0. In rare cases, some control registers You may be able to change the bit regardless of RSBSY.

RSCR0—RSPI control register 0 $ YFF612

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RSE	MST	RSFIE	HMIE	MFDE	0	WOM	HALT	BAUD							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Changing other flags of RSCR at the same time in the CPU write bus cycle to set the RSE. And you can also. On the other hand, with RSE already set, RSE and the other bits of RSCR0 are simultaneously set. If you try to change it, you can clear RSE and change HALT in that bus cycle, but you can change it in other ways. You cannot change the lag.

Run RSE—RSPI

0 = Stop communication.
1 = Start communication.

The RSE bit enables or disables the RSPI's internal circuitry for both master and slave operations. Bull When RSE = 0 (and RSBSY = 0), the communication circuit inside RSPI enters the inactive state.

The user can change the contents of all local registers in RSPI. RSE is set When triggered, the master mode RSPI starts data transfer and the slave mode RSPI waits for communication. It goes into a state and responds to the data communication initiated by the master.

The master RSPI clears the RSE automatically when it finishes both transmit and receive data transfers. It will be fu. All commands, operands, and data when command transfer is in progress When the transfer is complete, the master RSE is automatically cleared. Day in wraparound mode RSE is not cleared at the wraparound point when performing a data transfer. this If so, the RSE must be cleared by the CPU.

You can also clear the RSE by a mode fault to the master RSPI.

The slave RSPI does not automatically clear the RSE when it finishes a series of data transfers. However, If a frame size error is detected in The Sensitivity SS mode, clear RSE and Slave RSPI is automatically turned off.

MST— Master / Slave Select

0 = RSPI is a slave device and only responds to serial transfers initiated by other devices.

1 = RSPI is the master and can initiate data transfer with other RSPI devices.

When MST is asserted, RSPI immediately RSCK, RMOSI, and RSS regardless of the state of the RSE bit. Start actively driving the port at the appropriate level.

If mode fault detection is enabled (MFDE = 1) with MST = 1, RSPI will have RSE. Mode fault inspection on RMISO or RSS lines, whether set or not I will go out. When a mode fault occurs on the master device, the MST bit is forced to set. B, MODF is set, and another device is requesting master rights on multiple master systems. Indicates that there is a possibility.

RSFIE—RSPI end interrupt enabled

0 = Disable RSPI interrupts.

1 = Enable RSPI interrupts.

RSFIE enables or disables CPU interrupts when RSPI asserts RSFIN. This RSPI interrupt event can be cleared by clearing RSFIN.

HMIE—Holt or mode fault interrupt enabled

0 = Disables Holt or Mode Fault interrupts.

1 = Enable Holt or Mode Fault interrupts.

HMIE occurs when the HALTA or MODF flags in the status register are asserted Enables or disables RSPI interrupts on the CPU.

MFDE—Mode Fault Detection Enabled

0 = Disables the mode fault feature.

1 = Enable the mode fault function.

MFDE is only valid on the master RSPI. MFDE enables mode fault detection for master RSPI Bull / Disable. If MFDE is cleared, the master RSPI will not request any master rights. I will see it.

WOM—RSPI pin wired OR mode

0 = The output port is driven by a normal buffer (totem pole).

1 = All RSPI output ports are driven by open-drain buffers.

WOM wires all RSPI output ports, regardless of general purpose output or serial communication output You can switch to OR mode.

HALT—Pause RSPI operation

0 = Do not pause the RSPI function.

1 = Pause the RSPI function.

When HALT is asserted, RSPI is forced into a paused state. RSPI in a paused state Can be restarted later. While paused, RSPI is open to all RSPIs, including the state of external pins. Holds the state of the function. HALT is valid for both master and slave RSPIs. HALT is the master

Page 222

It cannot be used during RSPI pattern generation / capture transfer. HALT is set regardless of the RSBSY state. You can clear it. Using this bit, the user stops RSPI operation at the appropriate byte / word boundaries. You can stop it.

BAUD—Baud rate

The master RSPI internally generates a baud rate clock (SCK) for serial communication. Zhou The wave number can be programmed by the user. In addition to the master RSPI, the slave RSPI also has a noise reduction function. Uses an internally generated SCK as the pumping clock. The clock signal is initially in BAUD Obtained from the MCU system clock using the configured 8-bit modulus counter. Next The formula determines the SCK baud rate.

SCK baud rate = system clock / (2^{2+P} × (BAUD + 1))

or,

BAUD = (system clock / (2 $^{2+P}$ x required SCK baud rate)) -1

In this equation, BAUD is 0, 1, 2, 3, 4, ..., 255, and SCK skew, noise filter.
When all the pattern generation / capture functions are disabled, P is 0 for these machines.
P is 1 when at least one of the features is enabled. Table 117 and Table 118 show the 25MHz syste [redacted]
Shows the baud rate value relative to the stem clock and the corresponding SCK frequency. Progr
The lowest of ramable SCK skew, noise filtering, or pattern generation / capture
1971 If one is enabled (SKEW> 0, NF = 1, or PG = 1), RSPI internally SCK x 8
Note that it produces a frequency of. Depending on the base clock used by the slave RSPI
Determines the internal data sampling period for noise filtering.

When RSBSY = 0, the user can change the BAUD. BAUD cannot be changed when RSBSY = 1.

Table 117 SCK Frequency Examples 1

(Other than NF = 0, SKEW = 0 and pattern generator mode)

System Clock Frequency	Required Division Ratio	Value of BAUD	Actual SCK Frequency
25 MHz	Four	0	6.25 MHz
	8	1	3.13 MHz
	12	2	2.08 MHz
	16 16	3	1.56 MHz
	32	7	781 kHz
	64	15	391 kHz
	1024	255	24.4 kHz

Table 118 SCK Frequency Examples 2

(NF = 1 or SKEW> 0 or pattern generator mode)

System Clock Frequency	Required Division Ratio	Value of BAUD	Base Clock Frequency	Actual SCK Frequency
25 MHz	8	0	25.0 MHz	3.13 MHz
	16 16	1	12.5 MHz	1.56 MHz
	twenty four	2	8.33 MHz	1.04 MHz
	32	3	6.25 MHz	781 kHz
	64	7	3.13 MHz	391 kHz
	128 128	15	1.56 MHz	195 kHz
	2048	255	97.7 kHz	12.2 kHz

RSCR1—RSPI control register 1 $ YFF614

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
SSM	SGL	SIZE	LITL	NOT USED		LVL / CS	PG	ICMD	NOT USED	BCCM		ACKE		RTRY	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSCR1 contains RSPI configuration parameters before the serial transfer starts. RSBSY is set
During that time, the CPU cannot change the contents of RSCR1.

SSM—Slave select mode

0 = Select pin-slave select mode.
1 = Select address-slave select mode. Command transfer is available.

The SSM determines the slave select mode. Pin-slave select when SSM = 0
The mode is selected. In this mode, RSPI uses the traditional slave chip select method.
You can use it. When SSM = 1, the address-slave select mode is selected. this
The mode allows you to use most of the advanced features of RSPI, such as command forwarding. Address slave
Only Edge Sensitivity SS mode is available in select mode.

SGL—Single data frame mode enable

0 = Multiple data frame mode is selected.
1 = Single data frame mode is selected.

SGL determines the data frame mode. Multiple data flare when SGL = 0
The mode is selected. In this data frame mode, the master at the start of data transfer
RSPI asserts RSS and keeps the RSS level at “L” during a series of byte / word data string transfers.
Negate RSS at the end of the transfer. Received data is stored in RSPI RAM immediately after reception and before performing BCC inspection.

Will be toured. Therefore, the data stored in RSPI RAM contains some error.
There is a possibility. When the retry function is enabled, retry the entire transfer data.
The cycle is executed. When SGL = 1, single data frame mode is selected.
In this mode, the master RSPI asserts RSS at the beginning of each byte / word data transfer and data.
Negate RSS at the end of the transfer. If BCC inspection is enabled, the received data will be BCC
After passing the check, it is stored in RSPI RAM, so only error-free data is stored in RSPI RAM.
I will be. When the retry function is enabled, the retry cycle is the byte in question.
/ Only executed when an error is detected in the word data.

SIZE—Transfer data size

0 = 8-bit data transfer
1 = 16-bit data transfer

SIZE determines the size of the transfer data. 16-bit size mode allows RSPI to be 16 bits
Data coherency can be guaranteed. When SIZE = 1, the RSPI hardware is
Click the least significant bit (LSB) of all data index pointers (STAx, ENDx, and CUTx)
I will.

LITL—Little endian mode enable

0 = Big endian memory byte allocation and MSB first shift operation
1 = Little endian memory byte allocation and LSB first shift operation

LITL sets the type of memory byte allocation and how the data serializer data shifts.
Determine the direction. In big endian mode (LITL = 0), CURT / CURR has an even address
When pointing, the high-order byte of RSPI RAM is transferred and CURT / CURR points to an odd address.
The low-order byte is transferred. RSPI shifts out / in data from MSB to LSB
vinegar. In little endian mode (LITL = 1), if CURT / CURR points to an even address
When the low byte of RSPI RAM is transferred and CURT / CURR points to an odd address, the high byte is transferred.
Will be transferred. RSPI shifts out / in data from the LSB to the MSB.

LVL / CS—Level Sensitivity SS / Chip Select Mode Enable

0 = Slave RSS input is edge sensitive / chip select port disabled
Will be done.
1 = Level sensitive / chip select ports are enabled for slave RSS inputs.

Page 224

The LVL / CS has a function as a sensitive type selection switch for RSS input and a chip selection.
Port (RPCS0 to 3) Acts as an enable / disable switch for operation.
For the first feature, LVL / CS has pin-slave select mode and multiple days.
In data frame mode (SSM = 0 and SGL = 0), neither BCC nor acknowledge function is configured.
Can be used when (BCCM =% 00 and ACKE = 0) and the user chooses the RSS input sensitivity type.
You can choose. With this setting, if LVL / CS = 1, level sensitivity is selected and
Otherwise, edge sensitivity is selected. In other mode settings, it looks like LVL / CS
Edge sensitivity is selected regardless of the state. For the second function, master RSPI
LVL / CS with pin-slave select and single data frame mode (SSM = 0 and SGL)
If you do not set BCC or acknowledge function with = 1) (BCCM =% 00 and ACKE = 0), chip select.
You can enable or disable ports (RPCS0 to 3). When LVL / CS = 1 with this setting
Is enabled for the PCS feature of the master RSPI, otherwise it is disabled. Other mo
In the configuration, the PCS function is always disabled regardless of the LVL / CS state.

PG—Pattern generation enable

0 = Disable pattern generation / capture.
1 = Enable pattern generation / capture.

PG is a pin-slave select and multiple day without BCC or acknowledge function.
It can only be used with the master RSPI (MST = 1) in data frame mode (SSM = 0 and SGL = 0). PG is
Not valid for slave RSPI. Generated when the PG feature is enabled on the master RSPI
The baud rate is half the normal baud rate and there is a latency between each byte / word data transfer.
It will not be inserted. Therefore, the master continuously generates SCK clocks. Baud rate is PG
The function defines it as follows:

SCK baud rate = system clock / (8x (BAUD + 1))

When the PG feature is enabled, the user clears the appropriate bit in RSPAR and is unused.
RSCK, RMOSI, RMISO, and RSS lines can be configured as generic ports. Pattern generation function
And the pattern capture function can be selected by setting RSPAR and TXE / RXE of RSCR3.
increase. The former function is performed when the RMOSI bit of RSPAR and the TXE bit of RSCR3 are set.
The latter function is performed when both the RSPAR RMISO bit and the RSCR3 RXE bit are set.
If both TXE and RXE are set, the pattern generation function will be executed.

ICMD— Command transfer started

0 = Data transfer is started.
1 = Command transfer is started.

ICMD should only be used with the master RSPI (MST = 1, SSM = 1) in address-slave select mode.

Can be used. ICMD does not affect RSPI in other settings.

If ICMD is cleared while the master RSE is set, the master RSPI will transfer normal data. Sending will start. On the other hand, if ICMD is set while RSE is set, the master RSPI will Start the mand transfer procedure. All slave RSPIs that respond to command transfers are pre-loaded with RSE. Both SSMs must be set and in communication standby, but ICMD must be set is not.

BCCM — Block check code mode

BCCM enables / disables the block check code (BCC) feature and determines the type of BCC. increase. The BCCM settings for the master RSPI and slave RSPI must match each other.

Table 119 BCCM Options

BCCM	Explanation
00 00	Block check code (BCC) mechanism is disabled.
01 01	LPC is enabled for BCC.
Ten	CRC (inverted) is enabled for BCC.
11 11	CRC (non-inverted) is enabled for BCC.

When BCCM =% 01, the horizontal parity check code (LPC) is selected. On the other hand, when BCCM =% 11. The non-inverted cyclic redundancy check code (CRC) is selected for, and the inverted CRC is selected when BCCM =% 10. It will be. The size of the BCC syndrome is always 16 bits regardless of the SIZE setting. LPC is selected When (BCCM =% 01), the calculated parity type (even or odd) is RSCR2's BCCP. It depends on the content. When the BCC feature is enabled, the data column of information sent is followed by 16 The BCC code of the bit is transferred. When all BCCMs are cleared, the RSPI hardware will be BCC Stops the calculation and clears BCCF.

ACKE—Acnoridge Enable

0 = Disable the acknowledge function.
1 = Enable the acknowledge function.

ACKE enables / disables the acknowledge feature for handshake communication. Shin For reliable communication, especially in address-slave select mode, this feature is available. It is recommended to navel. At least 1 of BCCM to enable the acknowledge feature The bit must be set to enable the BCC check function. Acnoridge enables When run, a 16-bit acknowledge code after the BCC code at the end of each data frame. Followed by. When the master is notified of a BCC error on the slave RSPI using the acknowledgment procedure, The master can start a retry cycle. In a similar way, the master has a BCC error on the master side. Is notified to the slave through the acknowledgment procedure, so that the slave has a retry cycle by the master. You can start preparing to get started. Also, there are selected valid slaves on the network. If not, the master RSPI detects the situation through the acknowledge procedure and immediately aborts the communication. increase.

RTRY—Retry count

The RTRY field enables / disables the retry feature and sets the retry limit. vinegar. If BCCM =% X1 or% 1X and ACKE = 1, RTRY is for both master and slave RSPIs. Determines the number of retries. RTRY is RSPI when the acknowledge feature is disabled Does not affect the behavior of. RSPI tells the user if a retry cycle is in progress There is no mechanism to do this. For example, when RTRY =% 011, up to 3 times for each data frame. You can generate retries up to. A condition has occurred that causes a retry cycle in the master RSPI If RTRY is set to zero or the limit set in RTRY is exceeded, then the mass RSPI detects a retry fault and immediately suspends operation. On the other hand, in slave RSPI, If a tri-fault is detected, the slave RSPI may generate a CPU interrupt, The operation is not interrupted.

RSCR2—RSPI control register 2 $ YFF616

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	BCCP				NF	NOT USED	SKEW		DTL							
RESET:																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RSCR2 has parameters that set delay times, noise filters, and BCC preset values. I will. These parameters must be set before starting the serial transfer. RSBSY The CPU cannot change the contents of these registers when = 1.

BCCP—BCC preset value

BCCP sets preset values for BCC syndrome in both LPC and CRC modes. BCCP

Is valid when LPC or CRC is enabled (BCCM =% 01 or BCCM =% 1X)
is. The BCCP values of the master RSPI and each slave RSPI must match.

・CRC mode

In CRC mode (BCCM =% 1X), BCCP sets the initial value of the BCC code generation register. BCCP
The most significant bit of always defaults to the most significant 4 bits of the code generation register (BCCT, BCCR).
Similarly, the lower bits of BCCP initialize the lower 4 bits of the BCC code generation register. each
Before starting the data transfer, RSPI initializes the BCC code generation register to the value defined by BCCP.

・LPC mode

In LPC mode (BCCM =% 01), BCCP is a 16-bit horizontal parity check (LPC) code code.
Sets the degree type (even / odd) in 4-bit increments. LPC when the BCCP bit is zero
The corresponding 4 bits of the code are set as even parity, and when the BCCP bit is 1, this 4
The bits are odd parity. As shown in the table below, the BCCP setting is how to shift the data.
Depends on the direction, data size, data frame mode, and start address of the data column
vinegar. Before starting data transfer, RSPI initializes BCCT to the state defined by BCCP and sets BCCR.
Clear to all zero.

Table 120 Parity configuration data bits with BCCP bits (BCCM =% 01)

BCCP Bit (RSCR2 Bit Position)				BCCP3 (15)	BCCP2 (14)	BCCP1 (13)	BCCP0 (12)
MSB First (LITL = 0)	BCC Generator Bit Position			X_{15-12}	X_{11-8}	X_{7-4}	X_{3-0}
	Data Bit Position	16 Bit Mode (SIZE = 1)		D_{15-12}	D_{11-8}	D_{7-4}	D_{3-0}
		8 Bit Mode (SIZE = 0, SG L = 0)	Even *	D_{15-12}	D_{11-8}	D_{7-4}	D_{3-0}
			Odd *	D_{7-4}	D_{3-0}	D_{15-12}	D_{11-8}
		8 Bit Mode (SIZE = 0, SGL = 1)		D_{15-12} D_{7-4}	D_{11-8} D_{3-0}	————	————
LSB First (LITL = 1)	BCC Generator Bit Position			X_{3-0}	X_{7-4}	X_{11-8}	X_{15-12}
	Data Bit Position	16 Bit Mode (SIZE = 1)		D_{15-12}	D_{11-8}	D_{7-4}	D_{3-0}
		8 Bit Mode (SIZE = 0, SG L = 0)	Even *	D_{15-12}	D_{11-8}	D_{7-4}	D_{3-0}
			Odd *	D_{7-4}	D_{3-0}	D_{15-12}	D_{11-8}
		8 Bit Mode (SIZE = 0, SGL = 1)		————	————	D_{15-12} D_{7-4}	D_{11-8} D_{3-0}

D_{15-12} D_{11-8} : High-order byte data bit
(Even addresses for big endian, odd addresses for little endian)
D_{7-4} D_{3-0} : Low byte data bit
(Odd addresses for big endian, even addresses for little endian)
Even *: The data string starts with a logical even address.
Odd *: The data string starts at a logical odd address.

NF— Noise filter enable

0 = Noise filter is disabled.
1 = Noise filter is enabled.

NF enables or disables noise filtering on all input lines
increase. When NF is set, RSPI is on the RSCK, RMOSI, and RSS (in slave mode) ports.
Or on the RMISO (in master mode) port, based on multipoint data sampling.
Activates the is filter function. The noise filter function is for each (master or
Slave) Utilizes the base clock generated by the RSPI's internal baud rate generator
To do. Therefore, when the noise filter feature is enabled, only the master RSPI
The BAUD value of the slave RSPI must also be set appropriately. Multipoint Sampurin
The frequency of the internal base clock used for the clock is eight times the SCK defined by BAUD. death
Therefore, when NF = 1, the data communication baud rate is half the normal baud rate. pose
Due to the nature of the Lucipoint noise filter, the slave side baud rate is actually generated by the master.
Must be equal to or greater than the baud rate.

SKEW—SCK Skew

SKEW is valid only on the master RSPI and has no effect on the slave RSPI. Master RSPI
When SKEW is enabled with a non-zero value, this RSPI has the same baud rate and phase.
Generates two different types of SCK clocks. That is, the external SCK and the internal SCK. For serial communication
The external SCK used is the RSCK port before the internal SCK used to time the master's data reception.
Appears in SKEW defines the skew period between these two SCKs in 1/8 SCK units. SKEW
With, the master RSPI is a slave for noise filtering and electrical propagation of data.
In high-speed communication where the data response time from is relatively delayed, the data fetch set of the RMISO port

Page 227

You can secure enough margin for to-up time.

DTL — Delay time after data transfer

The DTL determines the RSPI delay time after serial transfer of each byte / word. Lap Alaun The point wait time is the same as the data delay time defined by the DTL. BCC function or Acnor When the function is enabled, the last data, BCC code, and acknowledge code The same delay time is inserted between the files. The delay is calculated using the following equation.

Delay time after byte / word transfer = 2 * DTL * t $_{SCK}$

In this equation, DTL is (1, 2, 3, ..., 255) and t $_{SCK}$ is 1 / (SCK baud rate).

If the DTL is 0, the delay time is (2 * 256 * tSCK), which is the maximum for a 25 MHz system clock. Equal to the baud rate of 82 ∝ s.

RSCR3—RSPI control register 3 $ YFF618

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
MRQ	MGACK	PRIXT	PRIXR		CPADR			TXE	RXE	WREN	WPRTA		WPRTB		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

While RSBSY is set, the CPU may change RSCR3 bits other than MRQ and WREN. plug. MRQ, MGACK, PRIXT, PRIXR, and CPADR are in address-slave select mode (SSM). = Only works with 1). In address-slave select mode, the CR3 bit is set. Mand transfer allows the master RSPI to reference the contents of RSCR3. Command transfer OPWR If the bit is set, the master RSPI will use command transfer to select the slave RSPI. You can only change the TXE, RXE, and WREN bits on RSCR3. However, on the slave RSCR3 The other bits cannot be changed by the master RSPI with any command transfer.

MRQ—Request for master rights

0 = Do not request master rights.

1 = Request master rights.

MRQ can only be used on slave RSPIs to request master rights. MRQ is related to the state of RSBSY The CPU can be set or cleared without. If MRQ is set, the slave RSPI is the master Request master rights to RSPI. If MRQ is cleared, the operation of the slave will be affected. is not. RSPI hardware always clears the slave's MRQ when RSBSY is cleared To do.

MGACK—Master rights permission Acnoridge

0 = Master rights are not granted.

1 = Master rights are granted.

MGACK is a status bit used to arbitrate master rights. MGACK is a read-only bit (Writable in test mode). When slave RSPI hardware negates RSBSY Clear MGACK. MGACK is used for arbitration of automatic master rights. MGACK is a slave device Informs the chair that it is the only candidate device in the network to be selected as the next master candidate device. unit If MGACK of the slave RSPI is set as a result of mand transfer, the slave RSPI hardware is set. A asserts MPMT, clears RSE, MRQ, and MGACK and generates a CPU interrupt. CPU discount In the inclusive routine, the slave RSPI can acquire new master rights when the user asserts MST.

PRIXT—Protection of transmit index register

0 = Write access to the transmit slave index register by the master is allowed.

1 = The master's write access to the transmit slave index register is protected.

PRIXR—Protection of receive index register

0 = Write access to the receive slave index register by the master is allowed.

1 = The master's write access to the receive slave index register is protected.

PRIXT and PRIXR are valid for slave RSPI. The master RSPI is framed by the slave PRIXT.

Page 228

Allow or disallow changing the contents of the slave's STAT / ENDT or STAR / ENDR using ind transfer increase. The master uses command transfer with the CR3 operands on the slave's PRIXT and PRIXR.

You can see the value, but you cannot change it. Slave PRIXT and PRIXR have RSBSY = 0 Sometimes the CPU can be changed. Slave PRIXT or PRIXR before IXT or IXR operand transfer If is set, the new IXT or IXR operand will also be STAT / ENDT for the specified slave. Will not be stored in STAR / ENDR. If this happens with a command transfer with an acknowledge, the mass Data RSPI interrupts the current transfer and sets NSLVF for RSSR.

CPADR—Chip address

CPADR is the slave RSPI's chip address in address-slave select mode. Use only if you want to specify a number. CPADR is a unique chip in the serial communication network. Must have an address. Zero is an invalid value. Slave RSPI is CPADR, SLVADR, and Use ADDRM to determine if it is selected as a slave RSPI. Selected If so, the slave RSPI sets the SSEL flag to allow communication with the master device.

TXE—Send function enabled

0 = Disable data transmission.
1 = Enable data transmission.

RXE—Receive function enabled

0 = Disable data reception.
1 = Enable data reception.

TXE and RXE enable or disable the data send and receive functions, respectively. Users can take advantage of TXE and RXE to control the duality of data communication. Both TXE and RXE command transfer It does not affect the duality of the feed. TXE and RXE control the operation of the internal transmit and receive circuits, but output It does not control the operation of the force / input port. For communication with the acknowledge function, TXE and RXE are available respectively. Enables / disables the credential receive or send function. Slave RSPI TXE and RXE can change the master device using command transfer.

WREN— Rabround Enable

0 = Disable wraparound mode.
1 = Enable wraparound mode.

WREN enables or disables wraparound operation. RSPI transmit circuit or When the receiving circuit finishes data transfer at the ENDT / ENDR address, RSPI sets the STAR / CURR values to STAT / respectively. Store in CURT. At this point, if WREN = 1, RSPI transfers data from the beginning of the data column to the next. Will resume. RSPI continues such a wrapping sequence indefinitely. On the other hand, WREN = 0 In the case of, when both the transmitting circuit and the receiving circuit finish the data transfer, RSPI completes the communication and clicks RSE. Rear (master RSPI only). When the master RSPI starts data transfer with WREN = 1, WREN will Continues the data string transfer in the loop until it is gated or the RSE is negated. If the CPU negates the master WREN during data transfer (RSBSY = 1), RSPI will be both transmit and receive. Data transfer is executed until all the data columns currently being executed are transferred, and when the data transfer is completed, the transfer function To stop.

WPRTA— Write protection area

0 = RSPI is allowed write access to the RSPI RAM space below the boundary address pointed to by WPRTB. Is done.
1 = RSPI is allowed write access to the RSPI RAM space below the boundary address pointed to by WPRTB. I can't.

WPRTA sets the type of RSPI write access protection area. If WPRTA is set, RSPI protects write access to incoming data to addresses below the boundaries pointed to by WPRTB. WPRTA If is cleared, RSPI will write the received data to the address that crosses or contains the boundary. Protects crowded access. The boundaries are specified by WPRTB. RSPI gives write access to the protected area When attempted, the normal data write sequence is performed and the received data is not stored in RSPI RAM. Hmm. RSPI does not have a display flag to indicate that write access has been dropped.

WPRTB — Write protection boundary

WPRTB sets the perimeter address of the RSPI write access protection area. Boundaries are on RSPI RAM

It can be placed on a 256-byte boundary at any address. Data store access to RSPI RAM Before executing, WPRTB is compared with the upper 4 bits of the current receive pointer. The table below shows the RSPI document. Shows the relationship between the built-in protection area and the WPRTA / WPRTB state. First WPRTA / WPRTB after system reset Since the period state is all zero, the entire RSPI RAM space is in the RSPI write access protection area. Please note that.

Table 121 RSPI write protection area

WPRTB (Hex)	Boundary Address Address	WPRTA = 0		WPRTA = 1	
		RSPI write protected	RSPI write unprotected	RSPI write protected	RSPI write unprotected
0	0	0-FFF	————	————	0-FFF
1	100	100-FFF	0-FF	0-FF	100-FFF

2	200	200-FFF	0-1FF	0-1FF	200-FFF
3	300	300-FFF	0-2FF	0-2FF	300-FFF
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
D	D00	D00-FFF	0-CFF	0-CFF	D00-FFF
E	E00	E00-FFF	0-DFF	0-DFF	E00-FFF
F	F00	F00-FFF	0-EFF	0-EFF	F00-FFF

8.2.2.3 RSPI command register

RSCMD—RSPI command register $ YFF61A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED	MRL	ADDRM		SLVADR				MRADR MGR		OPWR OPRD		CR3	IXT	IXR	DT
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSCMD has an opcode for command transfer. However, in general operation, the master
Neither RSPI nor slave RSPI requires the user to configure the contents of RSCMD. This is Koman
Every time a transfer is started, the master RSPI hardware is located at the start address of the RSPI RAM.
Loads the code into RSCMD and sends the code to the slave, where the slave RSPI sends the RSCMD sent by the master.
This is to receive the code and store it in my RSCMD. Writing to RSCMD when RSBSY = 1
The inclusion is invalid. The state of the RSCMD control bit is in address-slave select mode (SSM =).
It affects the operation of 1), but does not affect the pin-slave select mode. MRL,
If one or more of the MRADRs, or MGRs, are set in the opcode, RSPI mediates master rights.
However, it does not execute the CR3, IXT, IXR, and DT operands.

MRL—Abandonment of master rights

0 = Master retains master rights.
1 = The current master relinquishes master rights.

After the transfer currently in progress is complete, the MRL clears its MST and MPMT bits to the master RSPI.
This allows the master RSPI to waive master rights. Also, MRL is
MPMT is also set for the slave RSPI for which MGACK was set by the previous operand transfer.
This will allow the slave RSPI to be configured as a new master device.
Will be.

ADDRM— Addressing mode

ADDRM sets the addressing mode for address-slave selection. With ADDRM
A properly selected slave RSPI using SLVADR can respond to a master that is performing a transfer.

Page 230

to come. Slave RSPIs that are not selected do not perform send or receive operations, but opcodes
Will receive. ADDRM also defines how to drive the slave RMISO line. ADDRM is% 00 (single)
In selection mode), the selected slave RSPI has an RMISO line in the totem pole buffer.
Drive In other modes (All, GT, or LE selection mode), open drain
The buffer is automatically enabled and the selected slave RSPI simultaneously draws its RMISO line.
Live The master RSPI also pairs the data received from the MISO line according to the ADDRM settings.
Enables or disables error checking. In slave RSPI, the CPU does the RSE
Clear ADDRM and SLVADR are automatically cleared each time the master initiates a command transfer
vinegar. This allows you to expect on the RMISO line when the user re-enables slave RSPI.
Prevents accidental data collisions.

Table 122 ADDRM options

ADDRM	Explanation
00 00	Only a slave who has a chip address number equal to the address designated by the SLVADR field is selected. (Sole select mode)
01 01	Slave who have a chip address number less or equal to the address designated by the SLVADR field are selected. (LE select mode)
Ten	Slave who have a chip address number greater than the address designated by the SLVADR field are selected. (GT select mode)
11 11	All slaves are selected. (All select mode)

SLVADR—Destination Slave Address

SLVADR is used with ADDRM in slave RSPI to give the selected slave chip address.

Specify. SLVADR can be dynamically configured using command transfer from the master device or from the CPU.
Write directly to the slave and set. In RSPI systems, all-zero chip addresses are
It is invalid. The selected slave RSPI responds to the master until a new command transfer is performed.
can do. You can also use ADDRM and SLVADR to limit the devices that participate in master rights arbitration.
I can do it.

MRADR—Collecting chip addresses of slaves requesting master rights

0 = Master does not query the address. No operand required.
1 = The master queries the slave chip address number requesting the next master right.
16-bit address information is included in the operand.

If MRADR is set for command transfer, each slave RSPI (depending on the CPU) selected by the master
MRQ is set) is decoded on the RMISO line during the MRADR operand transfer period.
You can put the chip address (CPADR) of and respond. Finally, the smallest chip a
The dress becomes the MRADR operand value received by the master RSPI. During this operand transfer
The master RSPI transmit circuit and the slave RSPI receive circuit do not work. Therefore, Single Day
During transfer of the MRADR operand in data frame mode, the master RSPI has the retry function enabled.
Even if it is, the retry will not be executed.

MGR—Authorization of master rights

0 = Master does not specify an address. No operand required.
1 = The master specifies the chip address of the next master. 16-bit address information in the operand
Information is included.

The MGR operand contains the decoded chip number to grant the master right, so the frame
If MGR is set on the transfer, all slave RSPIs waiting for communication are on the MGR operand.
Will be received. After receipt, the master rights authorized address is encoded by each slave RSPI and is individually
Compared to the CPADR value. If there is a match, the slave RSPI asserts its MGACK and matches
If not, negate. Therefore, only one slave is granted the following master rights:
BCC, acknowledge, and retry features are available for MGR operand transfers.

OPWR—Operand write enable

0 = Disables the operand write function.
1 = Enables the operand write function.

OPRD—Operand Read Enable

0 = Disables the operand read function.
1 = Enables the operand read function.

OPWR and OPRD enable / disable write / receive functions for the CR3, IXT, and IXR operands, respectively.
Disable. Opera from master to slave if OPWR is set in command transfer
The transfer is performed through the RMOSI line. If OPRD is set, from slave
Operand transfer to the star is performed through the RMISO line. Both OPWR and OPRD in the opcode
If is cleared, the operation code is followed by the transfer of CR3, IXT, and IXR operands.
plug. Similar to the TXE / RXE operation of data transfer, when the circuit is set to half-duplex communication in OPWR
It does not perform the functions related to BCC generation / checking and acknowledge response.

CR3—Access to control register 3

0 = Master is not accessing slave RSCR3. No operand required.
1 = Master is accessing RSCR3 on the destination slave. 16 for the operand
Contains RSCR3 data for bits.

Read / write operant to RSCR3 on the slave side if CR3 is set in the command transfer
Access is performed. Read and write operations are enabled by OPRD and OPWR, respectively.
Bull / Disabled. If CR3 is cleared, the following operands will have a CR3 feel.
Do not include. Even if the master performs a write operation to the slave RSCR3, the slave WREN,
Note that it only affects TXE, and RXE.

IXT—Access to transmit index register

0 = Master is not accessing slave RSIX0 / 1. No operand required.
1 = Master is accessing slave RSIX0 / 1. Two-word data of RSIX0 and RSIX1
Each is included in the operand.

IXR—Access to receive index register

0 = Master is not accessing slave RSIX3 / 4. No operand required.
1 = The master is accessing RSIX3 / 4 in the target slave. Two-word RSIX3 and RSIX4
Each data is included in the operand.

IXT or IXR indicates that the master is accessing RSIX0 / 1 or RSIX3 / 4, respectively.
vinegar. If these bits are set, the master RSPI will use the two-word data of the operands.
Transfer to the IXT or IXR operands, respectively. RSIX0 / if IXT or IXR is cleared
The 1 or RSIX3 / 4 operands are skipped on subsequent operand transfers.

DT—Data following the operand

0 = No data follow.
The 1 = operand is followed by 1 byte / word or more of data.
If the opcode is set to DT, the master RSPI follows the opcode or operand.
Transfer the data column. This data string transfer is the same as a normal data transfer without command transfer.
If the DT is cleared, the opcode / operand transfer is not followed by the data column. in general,
The user disables DT and sends the command and related data separately as two independent transfers.
can. However, the coherency between the opcode / operand transfer and the subsequent data transfer
We recommend that you use DT for command forwarding to ensure that. CR3, IXT, and IXR Opera
If the appropriate command transfer with a command is issued in advance, the master RSPI will be full / half of the transfer by DT.
You can set the dual and data location.

8.2.2.4 RSPI data index register

The RSPI data index pointer register contains STArt, END, and STArt, END, and for transmit and receive.
There are parameters for the CURrent address pointer. Each index register size
Due to the 12 bits, RSPI can access up to 4K bytes.

In the active state (RSBSY = 1), the user can change the contents of the current address pointer (CURx).
However, the contents of the start address pointer and end address pointer (STAx, ENDx) can be changed.
to come.

When inactive (RSBSY = 0), the user is on all data index pointers.
You can change the contents. If SIZE is set, RSPI is all data index points.

Page 232

Automatically clears the least significant bit of the data.

CPU read operations on the RSPI data index pointer register are bytes (8 bits).
Both) and word (16-bit) access are valid. Write to index pointer register
Only operations are valid for word (16-bit) access only. Buy to index pointer register
Write access is invalid.

RSIX0—RSPI index register 0 (sender start pointer) $ YFF61C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								STAT						
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The value of STAT is CPU write access (when RSBSY = 0) or master RSPI performing command transfer.
If changed by (RSBSY = 1), the new value will be copied to CURT at the same time. But RSBSY
CPU write access to STAT when = 1 changes STAT but not CURT.

RSIX1—RSPI index register 1 (sender end pointer) $ YFF61E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								ENDT						
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENDT can be set to any value, regardless of the value of STAT. The user is in communication operation (RSBSY = 1)
You can change the ENDT, but perform the change operation before the CURT reaches the address pointed to by the current ENDT.
Must be.

RSIX2—RSPI index register 2 (sender current pointer) $ YFF620

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								CURT						
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CURT is an address pointer used to point to the position on RAM of the data to be transferred. RSBSY =
If 0, CPU updates to STAT are also copied to CURT. But the CPU updates the CURT
But STAT does not change. When RSBSY = 1, it is invalid if the CPU writes to CURT. RSE
When asserted, the first byte / word of the data string addressed by CURT
Will be loaded. In 8-bit data size mode (SIZE = 0), for each byte of data
After sending, CURT is incremented by 1. In 16-bit data size mode (SIZE = 1)
Will increment CURT by 2 after each word data is sent.

RSIX3—RSPI index register 3 (receiver start pointer) $ YFF622

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								STAR						
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The STAR value is CPU write access (when RSBSY = 0) or the master RSPI during command transfer.
If changed by (RSBSY = 1), the new value will be copied to CURR at the same time. But RSBSY
CPU write access to STAR at = 1 modifies STAR but not CURR.

RSIX4—RSPI index register 4 (receiver end pointer) $ YFF624

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								ENDR						
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Page 233

ENDR can be set to any value, regardless of the STAR value. The user is in communication (RSBSY = 1)
You can change the ENDR, but you must change it before the CURR reaches the address pointed to by the current ENDR.
I don't.

RSIX5—RSPI index register 5 (receiver current pointer) $ YFF626

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	NOT USED								CURR						
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CURR is an address pointer that points to the position of the transferred data on RAM. If RSBSY = 0, the CPU
It will be copied to CURR as soon as you update STAR. But even if the CPU updates the CURR, the STAR changes.
Will not change. When RSBSY = 1, the CPU writes to CURR are invalid. 8-bit data size
In mode (SIZE = 0), CURR is incremented by 1 after each byte of data is sent.
In 16-bit data size mode (SIZE = 1), the CURR is 2 a after each word data is sent.
It will be clarified.

8.2.2.5 RSPI status register

RSSR—RSPI status register $ YFF628

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RSFIN	SSEL	NOT USED		MODF MPMT HALT			NOT USED	RSBSY RTYF		OPRF NSLVF NACKF BCCF				FREF SCKACT	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSSR contains RSPI status information. By asserting these bits in the register
Only the internal circuits of RSPI can be used. The CPU cannot assert a bit in RSSR,
You can negate by writing zero to the asserted bit. SSEL, MPMT, and RSBSY
Is a read-only bit, but all other bits can be read / written. One in RSSR
To clear these bits when multiple bits are set, first go to RSSR once.
There is no other way but to write zero after making the above read access. RSFIN, MODF, and
Asserting HALTA can result in CPU interrupts. SSEL, MPMT, and RSBSY
Except, most bits of RSSR are automatically cleared by asserting RSE.

RSFIN—RSPI transfer end flag

0 = RSPI has not completed the data transfer.
1 = RSPI has completed the data transfer.
RSFIN generally goes into an inactive state when RSPI finishes transferring a set of data strings, RSBSY.
Is set either when you negate or when the operation is interrupted due to an error. Lapua
RSFIN is not set per lap point during a round or retry.

SSEL—Slave select flag

0 = RSPI is not selected as the responding slave.
1 = RSPI is selected as the responding slave.
SSEL is used by slave RSPI. The master RSPI always clears the SSEL. SSEL is an access
Read-only bit that can be set. The SSEL of the active slave RSPI is automatically set or clicked.
It will be rear. This timing is synchronized with the RSS level. Pin Slave Select Mo
Level sensitivity of (SSM = 0) In SS mode, when RSS is asserted (“L” level)
Cleared when SSEL is set and RSS is negated (“H” level). Edge sensitivity
In tee SS mode, SSEL is set at the RSS falling edge and cleared at the next rising edge.
vinegar. In address-slave select mode (SSM = 1), the master is the ADDRM bit and the SLVADR bit.
Select one or more slave RSPIs by issuing properly configured command transfers.

You can choose. The selected slave RSPI asserts its SSEL when forwarding RSS = 0.
And clear when RSS = 1. The SSEL bits of the unselected slaves are also cleared.

MODF—Mode Fault Flag

0 = No mode fault occurs.
1 = A mode fault has occurred.

MODF is valid only on the master RSPI. When a mode fault occurs, the hardware Therefore, MODF is asserted. Mode fault as long as the master MFDE is set There are two possible conditions. The first condition occurs in edge-sensitive SS mode and eventually The external device pulls down the master's RMISO to “L”, while the master pulls RSS to “H” level, RSCK. To the “L” level. Slave RSPI pulls down the RMISO port under this condition There is a function to do. Another condition is the level at which RSS is assigned as a generic input port. Occurs in sensitive SS mode or PCS operation, and the RSS of the master is changed to “L” by an external device at any time. It will be pulled down. Mode faults occur even when the master RSPI is running serial communication (RSBSY). = 1) Occurs regardless of whether it is not running (RSBSY = 0). When a mode fault is detected, Star RSPI automatically performs the following operations:

1. Force the MST bit to zero to put it in slave mode.
2. If the RSE and RSBSY bits are set, force them to zero and the RSPI machine Disables the ability.
3. Set MODF and generate a CPU interrupt if HMIE = 1.

The user can set the RSPI by asserting the RSE in master mode or slave mode. It can be enabled. In either case, when RSE is asserted by the CPU, the MODF f. The lag will be cleared automatically. As long as the MPMT bit is set, the user will re-manage the RSPI. Assigned as a star device and opened master right arbitration dedicated to address slave select mode You can start.

MPMT — Granting Master Rights

0 = RSPI does not have master rights permission.
1 = RSPI has master rights permission.

MPMT is a read-only bit. Writing to MPMT is invalid. MPMT is RSPI in the network Indicates whether it can be a master device. RSE set and master RSPI enabled When this happens, the RSPI hardware asserts MPMT. When MPMT is set, RSE or MST will After being cleared, RSPI will continue to assert MPMT and the user can reassign master rights to RSPI. Indicates that. MPMT is cleared by a system reset. However, the master right is MPMT automatically sets or clears when successfully delegated from the current master to the new master To be able to do so, you must follow the correct master rights arbitration procedure.

HALTA— Holt Acnoridge

0 = RSPI is not Holt.
1 = RSPI is hollowed.

RSPI hardware asserts HALTA when it is paused in the specified way. predict To prevent impossible behavior, the user sets the HALT or HALTA bit while RSPI is paused. Except, the contents of the RSPI register and RSPI RAM must not be modified. At this point HMIE is set If so, RSPI will generate a CPU interrupt request. If you clear HALTA, the user interrupts the CPU Only the request signal can be negated. Also, when HALT is negated, HALTA is cleared and the CPU The interrupt request signal is negated.

RSBSY—RSPI busy flag

0 = RSPI is not running.
1 = RSPI is running.

RSBSY is a read-only bit. Writing to RSBSY is invalid. RSBSY is the transfer operation of RSPI Indicates whether or not. RSBSY is an RSPI hardware when RSPI is enabled by asserting RSE. A will set it automatically. RSPI continues to assert RSBSY until the data transfer is complete. Directly to RSE You can also write a zero contact to end the transfer. In this case, RSBSY is not immediately negated It may be. When RSBSY = 1, the CPU fills most of the control bits in the local register. It cannot be changed. On the other hand, when RSBSY = 0, the CPU contains the contents of all write-accessible control bits.

Can be changed.

RTYF—Retry Fault Flag

0 = No retry error has occurred.
1 = A retry error has occurred.

RTYF is only valid when the acknowledge feature is enabled. RTYF is a retry Indicates the occurrence of a fault. Retry cycle times where the retry sequence is defined in RTRY If less than a few complete, a retry fault will be detected. Master RSPI retries for If it detects an error, the master interrupts the current data transfer and clears the RSE. Meanwhile, Slay If RSPI detects a retry fault, RSFIN is set but RSE is asserted. It remains.

OPRF—operand fault flag

0 = The command operand contains no error.
1 = The command operand may contain an error.

OPRF is enabled in address-slave select mode. Master through command transfer RSPI can send CR3, IXT, or IXR operands. Master and selected sley When the RSPI starts forwarding these operands with OPWR = 1, the OPRF is first set to “H”. vinegar. If all operands are successfully forwarded, the RSPI involved negates the OPRF. normal If not forwarded to, the OPRF flag remains “H” as long as there is an error in the operand. Indicates an operand error. While OPRF is set, the master RSPI will perform "data transfer" When started, the slave RSPI disables all internal data communication operations and sets the MISO line to “H”. Respond with a bell. When this response is received, the master suspends the transfer.

NSLVF—Fault flag without slave acknowledge

0 = SLVAL Acknowledgment received.
1 = SLVAL Acknowledgment received no acknowledge.

NSLVF is the master RSPI in address-slave select mode with acknowledge operation. Only valid. In other settings, NSLVF is not set. Master RSPI is the acknowledge code If "NACK" is detected in the last part, the master sets NSLVF, otherwise NSLVF Clear. If the master RSPI sets NSLVF, the master is immediately in the middle of the currently running transfer. Decline and clear RSE. NSLVF is asserted because there are no slaves in communication standby state. Operand write sequence executed by master rejected or specified by slave If one of the errors occurs that the slave has an invalid setting.

NACKF—Fault flag without acknowledge

0 = RCVAL Acknowledgment received.
1 = RCVAL Acknowledgment received no acknowledge.

NACKF is useful when using the acknowledge function. Acnoridge is disabled If so, NSLVF is not set. Basically, the state of NACKF is that of the received acknowledge code. It reflects the result of the first part. If RSPI receives "NACK" in that part, NACKF is set. If not received, NACKF will be cleared. NACKF assert is another device in communication Indicates that at least one of the chairs has detected a BCC error. Send operation is disabled If it is temporarily suspended, NACKF evaluation is disabled and NACKF is cleared. If the retry function is enabled on the master RSPI, NACKF will cause a retry cycle. It can occur.

BCCF—BCC fault flag

0 = No BCC fault has occurred.
1 = A BCC fault has occurred.

BCCF indicates that a block check code (BCC) error was detected while receiving data. RSPI If it detects a BCC error, RSPI will use other communication devices if acknowledgment is enabled. Answer "NACK" at the beginning of the acknowledge code for the computer. Thus, the master RSPI If the retry function is enabled on, the retry cycle is performed by asserting BCCF. May occur. RSPI clears BCCF if it does not detect a BCC error. Returns "ACK". RSPI clears BCCF if data receive operation is disabled in RSPI Disables the BCC check function and responds "ACK" to the acknowledgment code.

FREF — Framing error fault flag

0 = No framing error has occurred.
1 = A framing error has occurred.

FREF reflects the result of the framing size error check in the slave RSPI. death Therefore, FREF is only valid for slave RSPI. Slave RSPI detects framing error Unless otherwise, the slave RSPI clears FREF. When a frame error is detected, a sley RSPI clears RSE and immediately interrupts its current operation. This is a flare after RSS negate Even if an error is detected, the frame error on the slave side is mastered by the acknowledge procedure. This is because RSPI is not notified.

SCKACT—SCK active flag

0 = RSCK line is not active (current master device is not performing the transfer).
1 = RSCK line is active (current master device performed the transfer).

SCKACT is for the slave to detect if the current master is active. Followed
SCKACT is the active slave regardless of whether it is selected (SSEL = 1 or 0).
Only valid for RSPI. This detection operation is performed simply by monitoring the RSCK level inversion.
Here's how to use SCKACT:

1. Write zero to SCKACT or assert RSE.
2. Wait for the right time.
3. Check the SCKACT bit.

SCKACT does not affect any behavior of the slave device.

□ RSSR summary

The following table summarizes the main functions of the status flags in RSSR. In this table, R / W is
Bits that the CPU has read / write access to, and R indicates read-only bits.

Table 123 Status flag summary

Flag	CPU access *	CPU interrupt	Valid condition	automatic clear condition
RSFIN	R / W	generate generate	any mode	RSE assertion
SSEL	R		MST = 0	not selected
MODF	R / W	generate generate	MST = 1	RSE assertion
MPMT	R		any mode	Mastership relinquished **
HALTA	R / W	generate generate	any mode	RSE assertion HALT negation
RSBSY	R		any mode	no busy
RTYF	R / W		ACKE = 1	RSE assertion
OPRF	R / W		SSM = 1	RSE assertion

NSLVF	R / W		MST = 1
			SSM = 1
			ACKE = 1
				RSE assertion

NACKF	R / W		ACKE = 1	RSE assertion
BCCF	R / W		BCCM! = 0	RSE assertion
FREF	R / W		MST = 0	RSE assertion

SCKACT	R / W		MST = 0
			RSBSY = 1
				RSE assertion

* Write access from the CPU to the RSSR is only allowed to clear the bits.

** The slave MPMT is cleared by asserting RSE in test mode.

8.2.2.6 RSPI BCC register

RSPI consists of two BCC registers. In normal operation, the BCC register is the CPU read access.
Only allow ses.

RSBC0—RSPI BCC register 0 $ YFF62A

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	BCCT															
RESET:																
	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

RSBC1—RSPI BCC register 1 $ YFF62C

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	BCCR															
RESET:																
	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

BCCT and BCCR are 16-bit BCC code generation registers for transmission and reception, respectively. child
These registers are sent or received when the BCC check function is enabled.
valid. If BCCT or BCCR is disabled, the RSPI hardware will click NACKF or BCCF, respectively.
Rear If enabled, RSPI initializes BCCT and BCCR at the start of the data frame.

vinegar. The default settings are affected by the contents of BCCP, depending on the BCC mode defined by BCCM. each
At the end of the data frame transfer, RSBC0 and RSBC1 still have the calculated BCC syndrome.
is. In general operation, the user does not need to access these registers directly.

8.2.2.7 RSPI data shift register and local Test register

RSSFT—RSPI data shift register $ YFF62E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
							DTSHFT								
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RSSFT is a 16-bit data shift register provided by RSPI as a data serializer.
It is used for both data transmission and data reception. RSSFT can be read by the CPU,
The CPU cannot write in normal operating mode. In general operation, the user will use this register.
You do not need to access directly to.

RSLTST—RSPI local test register $ YFF630

RSLTST contains control bits for factory testing.

8.2.2.8 RSPI Chip Select Pin Control register

The chip select pin control register determines how to use the PCS port. these
Port can be used as a chip select port by the master RSPI serial communication subsystem.
Each pin can also be used for general purpose input / output (I / O). After a system reset, RSCSPAR and
If you clear all the bits of RSCSDDR, all PCS ports will be assigned as general purpose input ports.
vinegar.

RSCSPA—RRSPI PCS pin assignment register $ YFF638

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
RPCS3	RPCS2	RPCS1	RPCS0		RSCSDDR						RSCSPDR				
RESET:															
0	0	0	0	0	0	0	0	U	U	U	U	U	U	U	U

0 = General purpose I / O port
1 = Chip select port
RSCSPAR provides the functionality of a master RSPI PCS port as a chip select port or general purpose I / O.
Decide which of the ports you want to set. Chip select operation enabled on master RSPI
If a bit in RSCSPAR is set while it is set, the corresponding PCS port is chip-selected.
Assigned as a functional port. Otherwise, when the RSCSPAR bit is cleared, the pair
Corresponding ports are assigned as general purpose I / O. The chip select function itself is desaved
If so, all master PCS ports should be general purpose I / O regardless of the RSCSPAR setting.
Works. All bits of RSCSPAR are RSBSY so that RSPI does not behave unpredictably.
When = 1, it is protected from CPU write access. The PCS port on the slave RSPI is
It is always assigned as a general purpose I / O port regardless of the RSCSPAR setting.

RSCSDDR—RSPI PCS data direction register $ YFF638

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	RSCSPAR			PSP3	PSP2	PSP1	PSP0				RSCSPDR				
RESET:															
0	0	0	0	0	0	0	0	U	U	U	U	U	U	U	U

0 = input
1 = output
RSCSDDR determines the direction of data flow on a PCS port defined as a generic port.
It does not affect the ports assigned as chip select.

RSCSPDR—RSPI PCS port data register $ YFF639

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	RSCSPAR				RSCSDDR			EPSP3	EPSP2	EPSP1	EPSP0	PSP3	PSP2	PSP1	PSP0

RESET:

0	0	0	0	0	0	0	0	U	U	U	U	U	U	U	U

PSP3-0—External RCS port status

Only input operation is possible with these 4 bits. Reading these bits makes the RPCS pin generic External, regardless of whether it is set to an input, output, or chip select function port Returns the current state of the RPCS pin. Write access to these bits is invalid.

PSP3-0—CS port output data bit

If the RPCS port is configured as a generic output port, strike these 4-bit corresponding bits. The value given will appear on the port. When reading these 4 bits, the value that the CPU just stored is always Will be returned.

Table 124> RSPI PCS pin control

RSCSPAR / RSCDDR / RSCSPDR bits	Port SP signal	RSPI chip select pin
RPCS3 / PSP3 / EPSP3	PSP3	RPCS3
RPCS2 / PSP2 / EPSP2	PSP2	RPCS2
RPCS1 / PSP1 / EPSP1	PSP1	RPCS1
RPCS0 / PSP0 / EPSP0	PSP0	RPCS0

8.2.2.9 Summary of RSPI features

The following table shows the slave select defined by the SSM bit and the data frame defined by the SGL bit. It summarizes the RSPI features for different combinations of modes.

Table 125 Feature Summary

Function	Mode Mode	Pin Slave Select Multi Data Frame SSM = 0, SGL = 0	Pin SS Single DF SSM = 0, SGL = 1	Addr. SS Multi DF SSM = 1, SGL = 0	Addr. SS Single DF SSM = 1, SGL = 1
Word Size		8/16 bit	←	←	←
Big / Little Endian		Big / Little	←	←	←
Full / Half Duplex		3-4 / 2-3 wire	4/3 wire	←	←
Data START & END point		Transmit / Receive	←	←	←
Wrap around		T / R Sync.	T / R Unsync.	T / R Sync.	T / R Unsync.
Programmable Delay Time		Yes Yes	←	←	←
Control Mode		Master / Slave	←	←	←
Mastership		Static	←	Dynamic	←
SS input sensitivity		Edge / Level	Edge	←	←
Chip Select w / PCS		No	Yes Yes	No	←
Pattern Generating / Capturing		Yes (Max. 2.5Mbps)	No	←	←
Fractional Bit Data		Store / Discard	Discard	Store / Discard	Discard
Wired OR output		Yes Yes	←	←	←
Handshake with ACK		Yes (String)	Yes (Word)	Yes (String)	Yes (Word)
Frame Size Check		Bits in String	Bits in Word	Bits in String	Bits in Word
BCC (16 bit)		LPC, CRC	←	←	←
Re-try		Yes Yes	←	←	←
Receive Data MultiSampling (Max. 2.5Mbps)		Yes Yes	←	←	←
SCK Skew		Yes Yes	←	←	←
Command Transfer		No	←	Yes Yes	←
Chip Addressing		No	←	Yes Yes	←
Data Indexing		No	←	Yes Yes	←
Auto Multi-Master Arbitration		No	←	Yes Yes	←
Master Mode Fault		Yes (on RMISO / RSS)	←	Yes (on MISO)	←

ACK: Acnoridge

LPC : Horizontal parity check code

CRC: Cyclic Redundancy Check Code (CRC-CCITT / ITU-T)

Word: 8-bit size or 16-bit size

Chapter 9
Multi-channel communication interface

Multi-Channel Communication Interface (MCCI) is a general purpose serial communication module
With two serial communication interfaces (SCI) and one serial peripheral interface
(SPI) is built-in. This chapter describes the normal usage of MCCI. For more information, see MCCI
See the Reference Manual (MCCIRM / AD). Figure 38 is a block diagram of the MCCI.

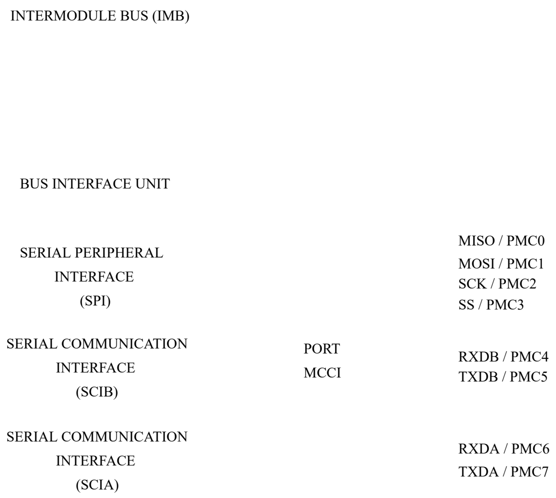


Figure 38 MCCI block diagram

9.1 Overview

MCCI's two SCI interfaces use the standard non-return-to-zero (NRZ) mark / space format.
Use to perform serial communication. SCI can operate in full or half dual mode. Individual to SCI
Transmitter and receiver enable bits, and dual data buffers for SCI
I have. Modular type baud rate generator is available from 64 to 524k baud (system clock)
Offers rates from 91.5 to 750kbo (when the system clock is 24.00MHz) (when the clock is 16.78MHz).
To do. The word length of 8 bits or 9 bits can be selected by software. Optional parity raw
The result and detection features an even or odd parity check function. With advanced error detection circuit,
Catch glitches up to 1/16 of the bit time. In addition, the wake-up function provides significant data.
You can run the CPU until you receive it.

The SPI uses a full-duplex synchronous 3-line bus for easy peripheral expansion and interprocessor communication.
Is possible. SPI is compatible with the SPI interface of other Motorola devices, but how to shift

It has additional features such as programmatic orientation.

The MCCI pin can also be configured for use as an 8-bit general purpose I / O port MC.

Table 126 MCCI Address Map

Address Address	15 8	7 0
$ YFFC00	MCCI MODULE CONFIGURATION REGISTER (MMCR)	
$ YFFC02	MCCI TEST REGISTER (MTEST)	
$ YFFC04	SCI INTERRUPT REGISTER (ILSCI)	SCI INTERRUPT VECTOR REGISTER (MIVR)
$ YFFC06	SPI INTERRUPT REGISTER (ILSPI)	NOT USED
$ YFFC08	NOT USED	PORTMC PIN ASSIGNMENT REGISTER (PMCPAR)
$ YFFC0A	NOT USED	PORTMC DATA DIRECTION REGISTER (DDRMC)
$ YFFC0C	NOT USED	PORT DATA REGISTER (PORTMC)
$ YFFC0E	NOT USED	MCCI PORT PIN STATE REGISTER (PORT MCP)
$ YFFC10 ~ $ YFFC16	NOT USED	
$ YFFC18	SCIA CONTROL REGISTER 0 (SCCR0A)	
$ YFFC1A	SCIA CONTROL REGISTER 1 (SCCR1A)	
$ YFFC1C	SCIA STATUS REGISTER (SCSRA)	
$ YFFC1E	SCIA DATA REGISTER (SCDRA)	
$ YFFC20 ~ $ YFFC26	NOT USED	
$ YFFC28	SCIB CONTROL REGISTER 0 (SCCR0B)	
$ YFFC2A	SCIB CONTROL REGISTER 1 (SCCR1B)	
$ YFFC2C	SCIB STATUS REGISTER (SCSRB)	
$ YFFC2E	SCIB DATA REGISTER (SCDRB)	
$ YFFC30 ~ $ YFFC37	NOT USED	
$ YFFC38	SPI CONTROL REGISTER (SPCR)	
$ YFFC3A	NOT USED	
$ YFFC3C	SPI STATUS REGISTER (SPSR)	
$ YFFC3E	SPI DATA REGISTER (SPDR)	

Note: Y = M111. However, M is the logical state of the SLIMCR modemap (MM) bits.
On this MCU, Y must be $ F. If M is cleared, reset
You will not be able to access the IMB module until it occurs. M is 1 after reset
You can write only once.

9.2 MCCI register

MCCI registers are MCCI global registers, MCCI pin control registers, SCI registers,
There are four types of SPI registers. SPI and SCI registers are described in the sections below.
To do. Writing to an unimplemented register bit is meaningless or invalid and reads an unimplemented bit
Always returns logical zero.

Single-chip integration module configuration register (SCIMCR)
The mode map bit of is the most significant bit of the address (ADDR23). This bit is each
It is indicated by "Y" in the Gista diagram. This bit is concatenated with the rest of the given address,
Form the absolute address of each register. CPU16 drives ADDR [23:20] into the same logical state as ADDR19
Therefore, Y must be $ F. For more information on the impact of MM state on the system, see 70.
See "Chapter 3 Streamline Low Power Integration Module" on the page.
give me.

9.2.1 MCCI Global Register

Global registers have parameters used by the SCI submodule. These paras
Meters are used by MCCI to interface with the CPU and other system modules.

MMCR — MCCI configuration register $ YFFC00

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP				NOT USED				SUPV		NOT USED			IARB		
RESET:															
0								1				0	0	0	0

STOP — Stop enable

0 = Normal MCCI clock operation.
1 = Stop the MCCI clock operation.

STOP disables the system clock on most of the modules and puts the MCCI in a low power state.
To MMCR is the only register that is guaranteed to be read when STOP is asserted.
STOP can be negated by CPU and reset.

SUPV — Supervisor / Unrestricted

0 = unconstrained access
1 = Supervisor Access

On systems with control access levels, SUPV supervises assignable MCCI registers.
Place it in either the dedicated data space or the unconstrained data space. All MCCI registers are
It will be placed in the supervisor space. CPU16 only works in supervisor mode, so SUPV
Has no meaning.

IARB — Interrupt arbitration identification number

The value in this field is used to arbitrate between concurrent interrupt requests of the same priority.
Will be done. Each module that raises an interrupt has an IARB field. Realize the mediation method
Must have a unique non-zero value for each IARB field. $ 0 IARB fee
If an interrupt request from a module with a rud value is recognized, CPU16 handles a spurious interrupt exception.
I will. All IARB fields except SCIM2 have a reset value of $ 0 (no priority) and are reset.
Eliminates interrupt processing at the time of interruption.

MTEST — MCCI test register $ YFFC02

MTEST is used in combination with the SCIM2 test feature during MCCI factory testing. MCU for MTEST
Must be accessed when is in factory test mode.

ILSCI / MIVR — SCI interrupt request level register / MCCI interrupt vector register $ YFFC04

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0		ILSCIB			ILSCIA				MIVR				1	1
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

ILSCI determines the interrupt priority level required by each SCI. SCIA and in separate fields
And there is an interrupt priority value for SCIB. Two or more modules or external circumferences using priority
Determines which interrupt to process first when side devices request interrupts at the same time.

ILSCIA, ILSCIB — SCIA, SCIB interrupt request level

ILSCIA and ILSCIB determine the SCIA and SCIB interrupt priority levels, respectively. Interrupts are recognized
To be done, if this field does not have a value between $ 1 (lowest priority) and $ 7 (highest priority)
It will not be.

MIVR — MCCI interrupt vector register

MIVR determines which vector the CPU uses to process the MCCI interrupt after it is recognized.
increase. Upon reset, MIVR is initialized to $ 0F, which corresponds to the uninitialized interrupt vector in the exception vector table.

Will be set. To handle interrupts, a user-defined vector ($ 40- $ FF) in the MIVR during MCCI initialization.
You have to program one of them.

The MCCI interrupt vectors are adjacent to each other in the exception vector table. MIVR [7: 2] has three
It is the same as the interface. MCCI supplies MIVR [1: 0] depending on the interrupt factor (SCIA is%).
00, SCIB is% 01). Writing to MIVR [1: 0] has no meaning and has no effect. MIVR [1: 0]
Returns a value of% 11.

ILSPI — SPI interrupt level register $ YFFC06

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0		ILSPI		0	0	0				RESERVED				
RESET:															

0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

ILSPI determines the priority level of interrupts requested by SPI. This is for the interrupt to be recognized
The field must have a value between $ 1 (lowest priority) and $ 7 (highest priority). ILSPI,
If ILSCIA and ILSCIB are the same, and interrupt requests occur at the same time, SPI, SCIA, and SCIB are in that order.
Will be recognized.

9.2.2 MCCI pin control register

The MCCI pin control register determines the use of the four MCU pins. These pins are siri
Used by al-subsystems, but all pins are assigned to be used as general purpose parallel ports
You can also do it.

The MCCI Data Direction Register (DDRMC) determines whether a pin is an input or an output. Bit
When reared, the corresponding pin becomes the input, and when the bit is set, it becomes the output. DDRMC is an SCI tiger
Determines the orientation of the TXD pin on the SCI only when the controller is disabled. SCI transformer
When the mitter is enabled, the TXD pin is output.

MCCI's port data register, PORTMC, latches I / O data. MCCI pin status register
PORTMCP allows you to read the pin state regardless of the configuration in the data direction.

PORTMC — MCCI port data register $ YFFC0D

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0
RESET:															
								U	U	U	U	U	U	U	U

Note: U is Unknown data.

Writes to PORTMC are stored in the internal data latch. Any bit in PORTMC is generic
When configured as an output port, the latched value is driven to the corresponding pin.
Reading PORTMC returns the value of the pin only if the pin is configured as a generic input port.
increase. Otherwise (general purpose output port), the latched value is read.

First write to PORTMC and then DDRMC to prevent indefinite data from being driven
Set.

PORTMCP — MCCI port pin status register $ YFFC0F

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0
RESET:															
								Pin status	Pin status	Pin status	Pin status	Pin status	Pin status	Pin status	Pin status

When you read PORTMCP, it always has a pin, regardless of whether the pin is configured as input or output.
Returns the status of. Writing to PORTMCP has no effect.

Page 244

PMCPAR — MCCI pin assignment register $ YFFC09

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					0	0	0	0	PMCP A3	0	PMCP A1	PMCP A0
RESET:															
								0	0	0	0	0	0	0	0

PMCPAR is an SPI server with the exception of the SCK pin (the state determined by the SPI enable bit).
Determine which SPI pins the module uses and which pins are valid for general I / O.
increase. Clearing the PMCPAR bit uses the SPI pin as a general I / O, and setting it as SPI.
And assign.

The SPI pins that PMCPAR specifies as general I / O are controlled only by DDRMC and PORTMC. SPI
Is not valid for these pins. PMCPAR for SCI submodule operations
It does not affect it.

DDRMC - port MC data direction register $ YFFC0B

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED					DDM7	DDM6	DDM5	DDM4	DDM3	DDM2	DDM1	DDM0
RESET:															
								0	0	0	0	0	0	0	0

DDRMC determines the generic I / O pin as either an input or an output. All MCCI pins are
It is configured as a general-purpose input when set. When the bit is cleared, the pin becomes an input and the bit is set.
Then it will be output.

Table 127 MCCI pin controls

PMCPAR bit	Open drain Control bit [1]	DDRMC bit	Port MC signal	MCCI pin
————	WOMC (SCCR1A bit13)	DDM7	PMC7	TXDA
————	————	DDM6	PMC6	RXDA
————	WOMC (SCCR1B bit13)	DDM5	PMC5	TXDB
————	————	DDM4	PMC4	RXDB
PMCPA3	WOMP (SPCR bit13)	DDM3	PMC3	SS
————	WOMP (SPCR bit13)	DDM2	PMC2	SCK
PMCPA1	WOMP (SPCR bit13)	DDM1	PMC1	MOSI
PMCPA0	WOMP (SPCR bit13)	DDM0	PMC0	MISO

Note 1. The open drain control bits (WOMC, WOMP) are the communication mode / general.
Effective for both output ports.

9.3 Serial Peripheral Interface

The SPI submodule communicates with external devices through a synchronous serial bus. SPI is
Not only is it fully compatible with the SPI system of other Motorola devices, but it is also enhanced.
It has been. The SPI can perform 3-wire full-duplex or 2-wire half-duplex transfers.

9.3.1 SPI pin

SPI uses four bidirectional pins. These pins are general when not needed for SPI applications.
Can be configured as an I / O for. Table 128 shows the features of SPI pins. Used as an SPI function
Sometimes the pin should have a pull-up resistor.

Table 128 SPI pin functions

Pin name	Mnemonic	mode	function
Master In Slave Out	MISO	Master	Enter serial data
		Slave	Output serial data
Master Out Slave In	MOSI	Master	Output serial data
		Slave	Enter serial data
Serial Clock	SCK	Master	Output serial synchronization clock
		Slave	Enter the serial sync clock
Slave Select	SS	Master	mode. Fault input
		Slave	Serial select input

9.3.2 SPI register

SPI's programmer's model is the MCCI global register and pin control register.
Data, SPI control register (SPCR), SPI status register (SPSR), and SPI data register.
It consists of Gista (SPDR). The CPU should read and write all SPI registers
I can. Only after initializing SPCR can SPI reliably perform the defined behavior. SPI
Is enabled when the SPE bit of SPCR is set.

SPCR — SPI control register $ YFFC38

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
SPIE	SPE	WOMP	SCBR	CPOL	CPHA	LSBF	SIZE				SPBR				
RESET:															
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

SPCR has parameters for configuring SPI. The CPU reads all control bits and
And writable, but MCCI only has read access to all bits except SPI. SPI
Writing a new value to SPCR while is enabled interferes with operation. SPI enables

Writing the same value to SPCR while it is running does not affect SPI operation.

SPIE — SPI interrupt enabled
0 = Disables SPI interrupts.
1 = Enables SPI interrupts.

SPE — SPI enabled
0 = Disable SPI. SPI pins can be used for general purpose I / 0.
1 = Enable SPI. Pins assigned by PMCPAR are controlled by SPI.

WOMP — Wired OR mode for SPI pins
0 = Output has a normal MOS driver.
1 = The pin specified for output in DDRMC has an open-drain driver.

Depending on WOMP, whether the SPI pin is used for general purpose output or SPI output, the SPI pin is wiped. Can be connected for yard OR operation. WOMP will desave even if SPI is enabled Even if it is set, it will affect the SPI pin.

MSTR — Master / Slave Mode Select
0 = SPI is a slave device and responds only to externally generated serial data.
1 = SPI is the system master and can initiate transmissions to external SPI devices.

MSTR configures SPI for master mode or slave mode operation. This bit is reset It is cleared by the CPU, but only the CPU can write.

CPOL — Clock polarity
0 = SCK inactive state is logical level 0.
1 = SCK inactive state is logical level 1.

CPOL is used to determine the inactive state of the serial clock (SCK). CPHA CPOL Used in combination with to establish the desired clock-data relationship between the master and slave devices. vinegar.

CPHA — Clock phase
0 = Data is captured at the front edge of the SCK and modified at the next edge of the SCK.
1 = Data is modified at the front edge of the SCK and captured at the next edge of the SCK.

CPHA determines at which edge of the SCK the data is modified and at which edge the data is captured. CPHA Is used in combination with CPOL to create the desired clock data relationship between master and slave devices. Establish. CPHA is set by reset.

LSBF — least significant bit first
0 = Start serial data transfer from MSB.
1 = Start serial data transfer from LSB.

SIZE — Transfer data size
0 = 8-bit data transfer
1 = 16-bit data transfer

SPBR — Serial Clock Baud Rate

SPI uses a modulus counter to generate the SCK baud rate from the MCU system clock. To do. The baud rate is selected by writing a value between 2 and 255 in the SPBR field. Writing 0 or 1 to the SPBR disables the baud rate generator.

The following formula determines the baud rate of the SCK.

$$\text{SCK Baud Rate} = \frac{\text{System clock frequency}}{2\ \text{(SPBR value)}}$$

or

$$\text{SPBR} = \frac{\text{System clock frequency}}{2\ \text{(desired SCK baud rate)}}$$

SPSR — SPI status register $ YFFC3C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
SPIF	WCOL	0	MODF	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SPSR has SPI status information. Bits in this register can only be set to SPI Hmm. To clear the status flag, read the SPSR with the flag set and then SPDR. To access.

SPIF — SPI end flag
0 = SPI has not ended.
1 = SPI is finished.

WCOL — Write collision
0 = No write collision has occurred.
1 = A write collision has occurred.

MODF — Mode Fault Flag
0 = normal operation
1 = When SPI is enabled in master mode (SS input is “L”), another SPI node
Requested to be a network SPI master.

SPDR — SPI data register $ YFFC3E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			UPPB								LOWB				
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

Writing to the SPDR initiates transmission or reception on the master device. Transmission is complete This sets the SPIF status bit on both the master and slave devices. Received Data is buffered. SPIF transfers the next data from the shift register to the buffer: Must be cleared before being released. Otherwise, an overrun will occur, causing the overrun. The lost bytes or words will be lost. Outgoing data is not buffered. Write to SPDR Then the data is sent directly into the shift register.

UPPB — Higher byte
In 16-bit transfer mode, UPPB is used to access the most significant 8 bits of data. SPDR Bit 15 is the MSB of 16-bit data.

LOWB - lower byte
In 8-bit transfer mode, the data is accessed at the LOWB address. 8-bit transfer mode The MSB is bit 7 of the SPDR. In 16-bit transfer mode, LOWB holds the least significant 8 bits of data.

9.3.3 SPI operation

SPI operates in mask mode or slave mode. In mask mode, SPI transfers data Used when calling from. Slave mode allows the external device to initiate a serial transfer to the SPI It is used when you do. Mode switching is controlled by the SPCR MSTR. In these modes The appropriate MCCI and SPI registers must be properly initialized before entering.

In master mode, write to the SPCR, set the transmit parameters, set the SPE, and set the SPI. Enable and then write data to the SPDR to get started. In slave mode, the external panel The operation proceeds in response to the assertion of the SS signal by the master. Slave operation is master Similar to mode operation.

Normally, the SPI bus performs synchronous bidirectional forwarding. Data at the SPI bus master serial clock A clock signal (SCK) for performing transfer is supplied. Clock position in CPHA and CPOL bits of SPCR You can specify four combinations of phase and polarity. The data is LSB or MSB depending on the value of the LSBF bit of SPCR. It will be transferred at the beginning. The default value for the number of bits transferred in one command is 8 bits, It can be set to 16 bits by setting the SPCR SIZE bit.

When the SPI finishes transmitting, it sets the SPIF flag and stops. The SPIE bit of SPCR is set If so, an interrupt request is generated when SPIF is set.

SPI supports multi-master operation, but has no built-in special arbitration mechanism. mode· The fault flag (MODF) indicates a request for SPI master arbitration. System software for this You must respond and mediate.

In general, the SPI bus output is open-drain unless there are multiple SPI masters in the system. Not. If necessary, set the WOMP bit of SPCR and wire OR open dray. Can provide output. An external pull-up resistor must be used for each output line. SPI pin WOMP is applied to all SPI pins, whether assigned to SPI or used as a generic I / O. Affect.

9.4 Serial communication interface

MCCI has the same independent SCI system for SCIA and SCIB. Each SCI system is a full-duplex pan

Asynchronous receiver / transmitter (UART) for. Each SCI system belongs to the M68HC11 and M68HC05 families.

It is fully compatible with the SCI system that any Motorola device has. The following explanation is SCIA
Applies to both and SCIB, but note the difference between register addresses and pin names.

9.4.1 SCI pin

Each SCI has two unidirectional transmit data pins, TXDA and TXDB, and two unidirectional receivers, RXDA and RXDB. The communication data pin is associated. Each pin can be used for associated SCI or generic I / O.

Table 129 SCI pin functions

Pin name	Mnemonic	mode	function
received data A and B	RXDA, RXDB	Receive disabled	General purpose I / O port
		Receive enable	Serial data entry as SCI
Transmission data A and B	TXDA, TXDB	Send disable	General purpose I / O port
		Send enable	Serial data output as SCI

9.4.2 SCI register

SCI's programming model includes MCCI global registers, pin control registers, and
And 8 SCI registers. Each SCI has two control registers and one stay
There is a task register and one data register.

The CPU can read or write all registers at any time. Same value for SCI cash register
Rewriting to the star does not affect the operation, but writes a different value to the SCI register during SCI operation.
And, it may interfere with its operation. To change the register value, receive and transmit
You must disable the transmitter so that the transmitter shuts down first. Register SCSR
You can clear the status flag of.

SCCR0A, SCCR0B — SCI control register 0 $ YFFC18, $ YFFC28

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED			SCBR												
RESET:															
			0	0	0	0	0	0	0	0	0	0	1	0	0

Each SCCR0 has a baud rate selection field. Beauley before SCI is enabled
Must be set. The CPU should read or write this register at any time.
I can.

SCBR — Baud rate

The SCI baud rate is programmed by writing a 13-bit value to this field. To SCBR
Writing 0 disables the baud rate generator.

The SCI receiver operates asynchronously. Internal to synchronize the receiver to the incoming data stream
I need a clock. The SCI baud rate generator predicts the baud rate frequency of incoming data.
Generates a receiver sampling clock at 16 times the frequency of. Is SCI a transition in the received waveform?
Determines the position of the bit boundary and adjusts the sampling point to an appropriate position within the bit period.
vinegar. The receiver sampling rate is always 16 times the SCI baud rate frequency, and the following equation is used:
Will be calculated.

$$\text{SCI Baud Rate} = \frac{\text{System clock frequency}}{32\ (\text{SCBR value})}$$

Where SCBR is {1, 2, 3 ..., 8191}.

SCCR1A, SCCR1B — SCI control register 1 $ YFFC1A, $ YFFC2A

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
UN- USED	LOOPS	WOMC	ILT	PT	PE	M	WAKE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Each SCCR1 has SCI configuration parameters. The CPU always reads this register or
You can write. SCI can change the RWU bit in certain situations. Generally this
Interrupts enabled on these control bits read the SCSR and then the SCDR.
(Receiver status bit) or write to SCDR (transmitter status bit)
Will be cleared.

SCCR1A / B15 — Not implemented

LOOPS — Loop mode

0 = Normal SCI operation. No looping, disable feedback path.
1 = SCI operation test. With looping and enables feedback path.

LOOPS control the feedback path for the data serial shifter. Loop mode
When enabled, the output of the SCI transmitter is fed back to the receive serial shifter.
vinegar. TXD is asserted (idle line). Transformer before SCI enters loop mode
Both the mitter and the receiver must be enabled.

WOMC — SCI Pin Wired OR Mode

When configured as 0 = output, TXD will be a normal CMOS output.
When configured as 1 = output, TXD is an open-drain output.

WOMC decides whether the TXD pin should be an open-drain output or a regular CMOS output.
This bit is only used when TXD is output. When the TXD pin is used as a general purpose input pin
WOMC is invalid. WOMC is not valid for RXD pins / PMCs (general purpose output pins).

ILT — Idle line detection type

0 = Short idle line detection (count starts at the first 1).
1 = Long idle line detection (count starts at the first 1 after the stop bit)

PT — Parity type

0 = even parity
1 = odd parity

When parity is enabled, the PT is for both the receiver and the transmitter.
Determines whether the trait is odd or even.

PE — Parity enable

0 = Disables SCI parity.
1 = Enable SCI parity.

PE enables parity or desaves for both receiver and transmitter
Decide if you want to. If the received parity bit is incorrect, SCI will issue an SCSR PF error.
Set the flag.

When PE is set, the most significant bit (MSB) of the data field is used for the parity function.
I will use it. Thereby, depending on the state of the M bits, the user data may be 7 bits or 8 bits.
It will be out of alignment. The following table shows the available options.

Table 130 Modes and Parity Options

M	PE	result
0	0	8 Data Bits
0	1	7 Data Bits, 1 Parity Bit
1	0	9 Data Bits
1	1	8 Data Bits, 1 Parity Bit

Page 250

M — Mode Select

0 = SCI frame: 1 start bit, 8 data bits, 1 stop bit (10 bits in total)
1 = SCI frame: 1 start bit, 9 data bits, 1 stop bit (11 bits in total)

WAKE — Wake up with address marks

0 = Wake up the SCI receiver with idle line detection.
1 = Wake up the SCI receiver with an address mark (the last bit is set).

TIE — transmit interrupt enabled

0 = SCI TDRE Disables interrupts.
1 = Enables SCI TDRE interrupts.

TCIE — Transmission complete interrupt enabled
0 = SCI TC Interrupts are disabled.
1 = Enables SCI TC interrupts.

RIE — Receiver interrupt enabled
0 = SCI RDRF Interrupts are disabled.
1 = Enables SCI RDRF interrupts.

ILIE — Idle line interrupt enabled
0 = SCI IDLE Disables interrupts.
1 = Enables SCI IDLE interrupts.

TE — Transmitter Enable
0 = Disable SCI transmitter (TXD pin can be used as general purpose I / O).
1 = Enable SCI transmitter (TXD pin is dedicated to SCI transmitter).

When TE is cleared, the transmitter takes control of the TXD pin until the ongoing character transfer is complete. Hold.

RE — Receiver Enable
0 = Disable SCI receiver (prohibit status bit, RXD pin used for general purpose I / O can).
1 = Enables the SCI receiver (RXD pin is for SCI only).

RWU — Receiver Wake Up
0 = Perform normal receiver operation (recognize received data).
1 = Enable wakeup mode (ignore received data until wakeup) Ru).

Setting RWU enables the wakeup feature and SCI is idle line or ad Ignore received data until waked up by one of the less marks (determined by WAKE) can do. In wakeup mode, the receiver status flag is not set and Interrupts are disabled. This bit is automatically cleared when the receiver wakes up (Return to normal mode).

SBK — Break transmission
0 = normal operation
1 = Send a break frame after the current frame is completed.

SBK can send break codes from SCI. SCI is set when SBK is set If in transit, SCI will have zero consecutive zeros after completing the transmission of the current frame until SBK is cleared. Send the frame. Toggle the SBK (from 1 to 0 within 1 frame interval) and the transmitter Sends only one or two break frames before returning to the idle line or sending data Start the trust.

SCSRA, SCSRB — SCI status register $ YFFC1C, $ YFFC2C

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED							TDRE	TC	RDRF	RAF	IDLE	OR	NF	FE	PF
RESET:															
							1	1	0	0	0	0	0	0	0

Each SCSR has a flag that indicates the operating status of the SCI. These flags are also hardware Is cleared by a special acknowledgment sequence. This sequence sets the flag It reads the SCSR and then the SCDR (writes for TDRE and TC). Long War Deleads have continuous access to both SCSR and SCDR. Do a long word lead At that time, the receive status flag bit that was set at the time of reading is cleared, but The TDRE or TC flags are not cleared.

The CPU reads the asserted status bit and then writes or reads register SCDR. If an internal SCI signal to set the status bit is sent by the time it is output, The newly set status bits are not cleared. To clear the status bit Sets this bit to read the SCSR again and write or read to the register SCDR. I have to know.

Reading any byte of the SCSR will access all 16 bits. Already with a byte The set status bit is subsequently read or written to the register SCDR. It will be cleared.

TDRE — Send data register empty flag

0 = The data to be sent to the transmission serial shifter remains in the register TDR.

1 = A new character can be written to the register TDR.

TDRE is set when a byte in register TDR is sent to the transmit serial shifter. TDRE
If is 0, it has not been sent yet, so writing to the TDR will overwrite the previous value.
vinegar. If you write to the TDR without clearing the TDRE, no new data will be sent.

TC — Transmission complete flag

0 = SCI transmitter is busy.

1 = SCI transmitter is idle.

TC is a transmitter with all data, preamble (mark idle line) in the queue,
Or set when the shift out of a break (logical 0) in the queue is completed. TC set
When this is done, read the SCSR and then write to the transmit data register (TDR) of the SCDR to interrupt.
You can clear only.

RDRF — Received data register full flag

0 = Register RDR is empty, or previously read data remains.

1 = There is new data in register RDR.

RDRF is set when the contents of the receive serial shifter are sent to register RDR. Receiving worker
If more than one error is detected, the flags NF, FE, and PF, or one of them is the same.
Set within the clock cycle.

RAF — Receiver active flag

0 = SCI receiver is idle.

1 = SCI receiver is busy.

RAF indicates whether the SCI receiver is busy. This flag is the start bit of the receiver
Set when it detects a possible bit and detects an idle line of the selected type
Will be cleared. RAF can be used to reduce collisions on systems with multiple masters.

IDLE — Idle line detection flag

0 = SCI receiver did not detect idle line condition.

1 = SCI receiver has detected an idle line condition.

IDLE is disabled when RWU for SCCR1 is set. IDLE is the SCI receiver SCCR1
Set when it detects the idle line state specified in the ILT of. When cleared, the RDRF
IDLE will not be reset until it is set. RDRF is set when a break is received, so
You can detect the next idle line.

OR — Overrun error flag

0 = RDRF is cleared before new data arrives.

1 = RDRF is not cleared until new data arrives.

The OR is ready to send a new byte from the receive serial shifter to the RDR, but the RDRF sets it.
It is set when it is left as it is. Data transfer is prohibited until the OR is cleared. Of RDR
The previous data remains valid, but the data received during the overrun state (bytes with OR set)
(Including) will be lost.

NF — Noise error flag

0 = No noise was detected in the received data.

1 = Noise occurred in the received data.

NF is the start bit, data bit, or stop bit in which the SCI receiver is valid.
It is set when it is detected. Idle line noise or noise with invalid start bits
Is not set even if is detected. Each bit is sampled 3 times. The theory that three samples are the same
If not at the logical level, use a large number of values for the received data value to set the NF. NF is a complete flare
Is received and will not be set until RDRF is set.

FE — Framing error flag

1 = A framing error or break has occurred in the received data.

0 = No framing error occurred in the received data.

FE is set when the SCI receiver finds 0 where the stop bit should occur.
FE receives the entire frame and is not set until RDRF is set. FE depends on break
It will also be set. If RXD is logical level 1 when the stop bit should occur, then the frame
You may miss a mining error.

PF — Parity error flag

1 = A parity error occurred in the received data.

0 = There was no parity error in the received data.

PF is set when the SCI receiver detects a parity error. PF is received by the entire frame
It is trusted and will not be set until RDRF is set.

SCDRA, SCDRB — SCI data register $ YFFC1E, $ YFFC2E

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
			NOT USED				R8 / T8	R7 / T7	R6 / T6	R5 / T5	R4 / T4	R3 / T3	R2 / T2	R1 / T1	R0 / T0
RESET:															
							U	U	U	U	U	U	U	U	U

Each SCDR consists of two data registers at the same address. When reading, the receiving side When a data register (RDR) is read and written, it is written to the data register (TDR) on the transmitting side. I will do it. RDR is a read-only store that stores data received on the SCI serial interface. It is a register. The data is sent to the receiving serial shifter and then transferred to RDR. TDR sends This is a write-only register that holds data. The data is first written to the TDR and then sent. It is transferred to Al Shifter, where it is sent with additional format bits added. R [0: 7] / T [0: 7] is the first 8 data bits received when the SCDR is read, or written to the SCDR. There are the first 8 data bits sent at the time. R8 / T8 has SCI configured for 9-bit operation It is used when it is.

Under the transfer conditions (with / without parity, data length, data), RDR (received data register = R [8: 0]) It affects the unreceived bits. The contents are shown in Table 131 "Extended data on the receiving side" below.

Table 131 Recipient extended data

Forwarding conditions		Sender	Receiver		
data type		data format	Depends on reception conditions Unreceived bits	Extended data (Data stored in unreceived bits)	Received data part
M	PE				
0	0	8-bit data	R8	If R7 = 1, R8 ⇐ 1 If R7 = 0, R8 ⇐ 0	R [7: 0]
0	1	7-bit data, 1-bit parity	R8, R7	R7 ⇐ Parity value If R7 = 1, R8 ⇐ 1 If R7 = 0, R8 ⇐ 0	R [6: 0]
1	0	9-bit data	————	————	R [8: 0]
1	1	8-bit data, 1-bit parity	R8	R8 ⇐ Parity value	R [7: 0]

9.4.3 SCI sender behavior

1. Set the TXD pin to the output side.
2. Set the transfer speed.
3. Set the transfer conditions. 8-bit data / 9-bit (M) data, include / omit parity (PE), If parity is included, even / odd (PT) and interrupt control (TIE, TCIE) are performed.
4. Read the status register (SCSR) and write the transmit data to the data register (SCDR). I will do it. This write causes the transfer data register empty flag (TDRE) and the transfer end flag. (TC) is cleared.
5. When transfer enable (TE) is set, one frame of idle is transferred and data transfer is started. vinegar.
6. The data transfer has started, which means that the SCDR data has been moved to the transmit serial shifter. And SCDR is empty. Therefore, the transfer data register empty flag (TDRE) Is set.
7. Write the following data to the SCDR. The TDRE flag is cleared again.
8. The data written to SCDR is moved to the transmission serial shifter until the data being transmitted is completed. I will not. When the transmission is completed, it will automatically move to the transmission serial shifter. Therefore, the sender Data can be transmitted continuously.

9.4.4 SCI receiver behavior

1. Set the RXD pin to the input side.
2. Set the transfer speed.
3. Read the received data register full flag (RDRF) in the status register (SCSR) and set the data.

Make sure that the data received in the data register (SCDR) is empty.

4. Set the transfer conditions. 8-bit data / 9-bit (M) data, include / omit parity (PE), If parity is included, even / odd (PT) and interrupt control (TIE, TCIE) are performed.
5. You can receive at any time by setting Receive Enable (RE). Receive busy flag (RAF) while receiving Is set.
6.1 The RDRF flag is set when the frame data has been received. At this time, the received serial system The lid automatically moves the SCDR data. Status bits (overrun, noise hula) If there is no problem with (framing error, parity error), read the received data from SCDR. It can be used by embedding. If there is a problem, the received data must be discarded not.
7. The RDRF is cleared when the data is read.

SCI can be received continuously. To receive data continuously, RDRF is set to notify the reception of data. At that time, please read the data promptly. If RDRF is set, from SCDR

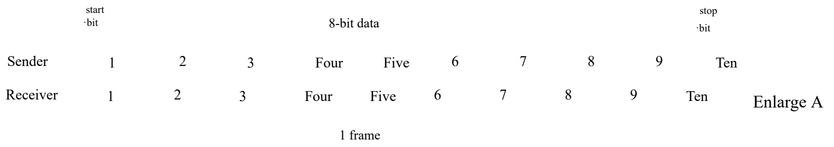
Page 254

If you do not read the data, when the next data becomes full in the receiving serial shifter, that day Data cannot be automatically transferred to SCDR. When the data is sent there again, the receiving serial shifter will be the body. It gets stuck, sets the overrun flag and stops.

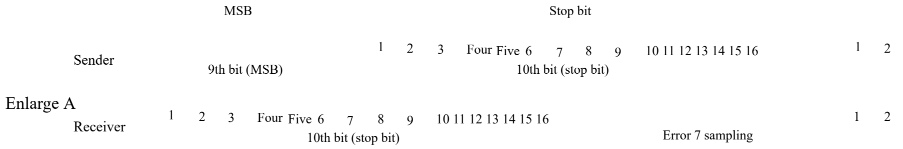
9.4.5 Outgoing / Incoming Forwarding Rate Tolerance

This SCI has a forwarding rate of +/- few percent between the sender and the receiver . Allowed using a concrete example Let's find the error. The data sampling method on the receiving side is 1-bit data from the transmitting side. Is divided into 16 and the 8th, 9th, and 10th bits are sampled, and the majority vote is performed to determine the received data. I am.

———When the transfer rate is faster on the receiving side than on the sending side



The difference shown in the above figure occurs when the transmission and reception of the transfer starts at the same time between the transmitting side and the receiving side. Will be. This example is a start bit, 8-bit data and a stop bit. This is the case when one frame is 10 bits.



The receiving side divides 1 bit into 16 and samples 8, 9, and 10 points. Because of that, this Where the three sampling points can sample the high of the first part of the stop bit It will be the communication speed at the fastest time. The margin of error, which is 8-bit length data, is calculated by the following formula. Will be.

(10x16) / (10x16-7) = 160/153 = 104.57% ---------- 8-bit data length

Therefore, the receiving side is allowed an error of + 4.57%.

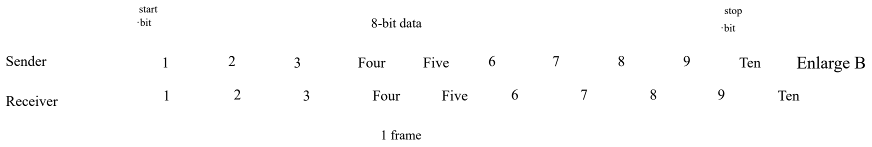
This idea can also be applied to 9-bit length data. The formula at that time is as follows.

(11x16) / (11x16-7) = 176/169 = 104.14% ---------- 9-bit data length

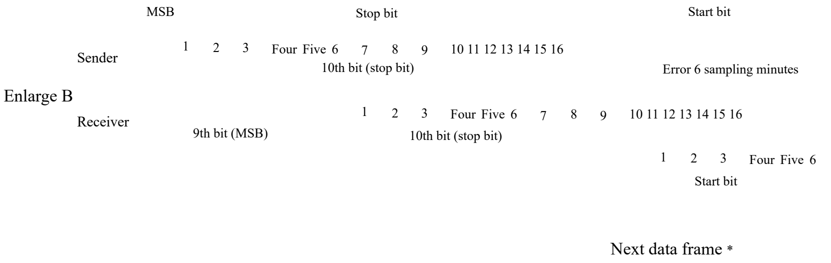
Therefore, the receiving side is allowed an error of + 4.14%.

Page 255

——When the transfer rate is slower on the receiving side than on the sending side



The difference shown in the above figure occurs when the transmission and reception of the transfer starts at the same time between the transmitting side and the receiving side. Will be. This example is a start bit, 8-bit data and a stop bit.
This is the case when one frame is 10 bits.



The receiving side divides 1 bit into 16 and samples 8, 9, and 10 points. Because of that, this
Where the three sampling points can sample the high at the end of the stop bit
However, it is the communication speed at the slowest time. The tolerance for 8-bit length data is given by the following formula.
You will be asked.

(10x16) / (10x16 + 6) = 160/166 = 96.39% ---------- 8-bit data length

Therefore, the receiving side is allowed an error of -3.61%.

This idea can also be applied to 9-bit length data. The formula at that time is as follows.

(11x16) / (11x16 + 6) = 176/182 = 96.71% ---------- 9-bit data length

Therefore, the receiving side is allowed an error of -3.29%.

Note: * The receiver will be in one frame if the three samplings of the stop bits are high.
Data reception is complete. Therefore, immediately after that, the start bit is low.
When is detected, the receiving side is automatically initialized and data reception is started.
Therefore, continuous reception is possible.

Page 256

Chapter 10
Mask ROM module
(MC68HC16Y5)

The 160K bytes (2 this 64K bytes + 32K bytes) mask ROM module (MRM) is a Motorola module. Designed for use in the entire Jura Microcontroller product line. MRM is a location It consists of a fixed control register block and a memory array. composition The information is contained in a register block. Default for an array of system address maps The reset base address is user-specified, but this array is remapped to another address. It is also possible to do. In addition to the array base address, the register block contains operating parameters, It also contains the bootstrap code and ROM verification information.

With the latest programming features, the mask program ROM module installed in the MCU Programming the actual ROM array at a later process compared to traditional ROM design It is now possible. These modules can be in program code and / or data. You can use it. See the register block address map below.

Table 132 MRM1 Control Register Address Map (64K Byte Module)

Address Addres15	8 7 0
$ YFF800	MASKED ROM MODULE CONFIGURATION REGISTER (MRM1CR)
$ YFF802	NOT USED
$ YFF804	ARRAY BASE ADDRESS REGISTER HIGH (ROM1BAH)
$ YFF806	ARRAY BASE ADDRESS REGISTER LOW (ROM1BAL)
$ YFF808	ROM SIGNATURE HIGH REGISTER (R1SIGHI)
$ YFF80A	ROM SIGNATURE LOW REGISTER (R1SIGLO)
$ YFF80C	NOT USED
$ YFF80E	NOT USED
$ YFF810	ROM BOOTSTRAP WORD 0 (ROM1BS0)
$ YFF812	ROM BOOTSTRAP WORD 1 (ROM1BS1)
$ YFF814	ROM BOOTSTRAP WORD 2 (ROM1BS2)
$ YFF816	ROM BOOTSTRAP WORD 3 (ROM1BS3)
$ YFF818 ~ $ YFF81E	NOT USED

Table 133 MRM2 Control Register Address Map (64K Byte Module)

Address Addres15	8 7 0
$ YFF820	MASKED ROM MODULE CONFIGURATION REGISTER (MRM2CR)
$ YFF822	NOT USED
$ YFF824	ARRAY BASE ADDRESS REGISTER HIGH (ROM2BAH)

$ YFF826	ARRAY BASE ADDRESS REGISTER LOW (ROM2BAL)
$ YFF828	ROM SIGNATURE HIGH REGISTER (R2SIGHI)
$ YFF82A	ROM SIGNATURE LOW REGISTER (R2SIGLO)
$ YFF82C	NOT USED
$ YFF82E	NOT USED
$ YFF830	ROM BOOTSTRAP WORD 0 (ROM2BS0)
$ YFF832	ROM BOOTSTRAP WORD 1 (ROM2BS1)
$ YFF834	ROM BOOTSTRAP WORD 2 (ROM2BS2)
$ YFF836	ROM BOOTSTRAP WORD 3 (ROM3BS3)
$ YFF838 ~ $ YFF83E	NOT USED

Table 134 MRM3 Control Register Address Map (32K Byte Module)

Address Address	15 8 7 0
$ YFF840	MASKED ROM MODULE CONFIGURATION REGISTER (MRM3CR)
$ YFF842	NOT USED
$ YFF844	ARRAY BASE ADDRESS REGISTER HIGH (ROM3BAH)
$ YFF846	ARRAY BASE ADDRESS REGISTER LOW (ROM3BAL)
$ YFF848	ROM SIGNATURE HIGH REGISTER (R3SIGHI)
$ YFF84A	ROM SIGNATURE LOW REGISTER (R3SIGLO)
$ YFF84C	NOT USED
$ YFF84E	NOT USED
$ YFF850	ROM BOOTSTRAP WORD 0 (ROM3BS0)
$ YFF852	ROM BOOTSTRAP WORD 1 (ROM3BS1)
$ YFF854	ROM BOOTSTRAP WORD 2 (ROM3BS2)
$ YFF856	ROM BOOTSTRAP WORD 3 (ROM3BS3)
$ YFF858 ~ $ YFF85E	NOT USED

Y = M111. However, M is the logical state of the SLIMCR module map (MM) bits.
It is a voice. Y must be $ F. If M is cleared, reset
You will not be able to access the IMB module until it occurs. M is once after reset
You can write to it.

10.1 Overview

The ROM array for this MCU is 160K bytes . These ROM arrays are arranged in 16-bit words
It is accessed through the intermodule bus. For bytes, words, and misaligned words
It is accessible. Access time is the number of wait states specified when programming the mask.
For bytes and aligned words, up to 2 system clocks, depending on. MRM
Responds to continuous access and has 2 bus cycles (4 system clocks) of long word access.
To run.

64K base address of the byte array 64K byte boundary, 32K base address of the byte array
The nest is 32K must be in the byte boundary. These arrays are other microcontroller modules
Do not overlap with the control register of the control register and the control register block
It should not overlap with. ROM array overruns control register of other module
If it is wrapped and mapped, the result of accessing these registers is undefined. ROM array is self
The control register, even if it overlaps and maps to your control register
Access to the data is possible, but access to the overlapping ROM arrays is ignored.

The main role of MRM is to use it as mask program memory for microcontrollers.
It can be configured to support system bootstrap at reset. CPU16 is a program
It distinguishes between spatial access and data spatial access. MRM array is dedicated to program code
Or can be used for both program code and data. Also insert a wait state
If you program MRM to do this, it will be slow external development memory without timing readjustment.

Can be migrated to a ROM array.

MRM is also in a special emulator mode that simplifies array emulation by external devices. It is possible to operate. The ROM array emulation mode is the MRMCR EMUL bit and the array enable. It is enabled by combining the two.

On the mask ROM module while emulation mode is enabled and active Each time a valid access is made to the assigned address, or bootstrap motion External module chip selection when enabled to access the bootstrap address Signal (CSE [1: 0]) is asserted. Array access is ROM-based for access to take effect Addresses that are within the range specified by the array register and are defined in the ASPC field of the MRMCR. The space condition must be met. Also, bootstrap access needs to be reset. It is invalid if it is not in the range of $ 00000 to $ 00007 immediately after being gated. CSE [1: 0] is all valid reads Asserted for access, but for write access in background debug mode Only asserted against. MRM acknowledges and responds to access to the IMB in embroidery mode not. Therefore, SLIM runs an external bus cycle. The CSE [1: 0] signal is at the falling edge of the AS It is asserted. The ROM module has the number of wait states specified in the MRMCR WAIT field. After inserting, generate an internal DTACK.

10.2 Mask ROM control block

Three 32-byte control register blocks are used to configure MRM and control ROM array functionality. Contains the registers used.

MRM1CR — Mask ROM1 module configuration register (64K bytes) $ YFF800

MRM2CR — Mask ROM2 module configuration register (64K bytes) $ YFF820

MRM3CR — Mask ROM3 module configuration register (32K bytes) $ YFF840

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	NOT USED		BOOT	LOCK	EMUL	ASPC		WAIT				NOT USED			
RESET:															
*	0	0	USER SPEC	USER SPEC	*	USER SPEC		USER SPEC		0	0	0	0	0	0

* STOP reset status = AD14. EMUL reset state = (AD10 • AD13).

STOP — Stop bit

0 = Normal ROM array operation or emulator mode (if enabled)
Make it active.

1 = Disable ROM.

The reset state of STOP is one's complement of the AD14 state at reset. When STOP is not set If so, you cannot change the ROM array base address.

BOOT — Boot ROM control bit

0 = CPU16 accesses the ROM bootstrap word address after reset.

1 = CPU16 cannot access the ROM bootstrap word address after reset.

The reset state of BOOT is specified by the user. If STOP = 1, the bootstrap function is disabled.

LOCK — lock register

0 = write lock disable-write to protected registers and fields
be able to.

1 = write lock enable-write to protected registers and fields
I can't.

The reset state of LOCK is specified by the user. If the default reset state for LOCK is zero, It can be set only once after a master reset to protect the registers after initialization. LOCK also protects the ASPC and WAIT fields, ROMBAL and ROMBAH registers. ASPC, ROMBAL, And ROMBAH is also protected by the STOP bit.

EMUL — emulator status bit

0 = Normal ROM operation

1 = MRM operates in emulator mode when STOP is cleared

The EMUL reset state is one's complement of the AD10 and AD13 states at reset. EMUL set If so, emulator mode is enabled. CSE [1: 0] is asserted at the following: If.

- When ROM array access is performed with STOP = 0.
- STOP = 0 enables bootstrap mode (BOOT = 0) and bootstrap a. If the dress is accessed.

Accessing the ROM control register does not assert CSE [1: 0] . Emule Valid access in data mode is external access.

ASPC — ROM array space field

X0 = ROM array is placed in the program / data space.
X1 = ROM array is placed in the program space.

CPU16 only operates in supervisor mode, so access is provided by the ASPC field.
Decide whether to limit it to gram space or to do it in both program space and data space.

Table 135 ASPC encoding

RASP [1: 0]	Space
X0	Program / data space
X1	Program space

WAIT — Wait state field

WAIT is inserted by MRM for access to the ROM array and ROM control registers.
Specifies the number of state states. Wait state inserted prior to internal DTACK generation
Controlling the number allows users to optimize bus speeds for individual applications. Each way
The length of the it state is one system clock cycle. Therefore, the user is a system
Slow emulation memory or development system memory without timing readjustment
You can migrate the code from to the ROM array. The reset state of WAIT is specified by the user.
The unweighted encoding (% 00) supports a 3-clock cycle bus. High
Fast termination encoding (% 11) is compatible with 2 clock cycle buses.
vinegar. Microcontroller modules usually respond at this speed, but fast termination
It is also possible to access high speed external memory using.

Page 260

Table 136 WAIT encoding

WAIT [1: 0]	Transfer cycle
00 00	3
01 01	Four
Ten	Five
11 11	2

ROM1BAH - array base address register high (64K -byte module) $ YFF804
ROM2BAH - array base address register high (64K -byte module) $ YFF824
ROM3BAH - array base address register high (32K -byte module) $ YFF844

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								ADDR twenty three	ADDR twenty two	ADDR twenty one	ADDR 20	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16
RESET:															
										USER SPECIFIED (US)					

The reset value of the shaded bit can be specified by the user, but it should be written to change the base address.
You can also. However, if the value of ADDR [23:20] does not match the value of ADDR19, CPU16 will be in the ROM array.
I can't access.

ROM1BAL - array base address low register (64K -byte module) $ YFF806
ROM2BAL - array base address low register (64K -byte module) $ YFF826

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROM2BAL - array base address low register (32K -byte module) $ YFF846

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
US	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMBAH and ROMBAL are used to specify ROM array-based registers. They are,
You can only write when STOP = 1 and LOCK = 0. This causes the array to accidentally remappin
It is prevented from being squeezed. The state of ADDR [23:20] follows the state of ADDR19, so it is from $ 080000 to $ F7FFFF.
You cannot access addresses in the range of.
A 64K byte ROM array must be mapped to a 64K byte boundary, so ROMBAL is always $ 0000.
is included.

Page 261

ROMBS1 [0: 3] — ROM1 bootstrap word (64K bytes module) $ YFF810-$ YFF816
ROMBS2 [0: 3] — ROM2 bootstrap word (64K bytes module) $ YFF830-$ YFF836
ROMBS3 [0: 3] — ROM3 bootstrap word (32K bytes module) $ YFF850-$ YFF856

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
BOOTSTRAP VECTOR															
RESET:															
PROGRAMMED VALUE															

Generally, the system CPU reset vector is in non-volatile memory and the CPU resets.
Fetched only when released from. The user uses these 4 words as a reset vector.
You can specify to use it, and you can also specify the contents of these locations. Of these words
The contents cannot be changed. In this MCU, ROMBS0 to ROMBS3 have system addresses $ 000000 to $ 000006.
It corresponds to.

10.2.1 Bootstrap operation

CPU16 is the first internal register located at IMB addresses $ 000000 to $ 000006 in the program space.
The bootstrap operation is started by fetching the period value. These are exception vector settings
The address of the bootstrap vector in the bull. RMMCR BOOT and STOP bits on reset
If is cleared, the mask ROM module will respond to bootstrap vector access.
Is set to The vector allocation is as follows:

Table 137 Bootstrap mapping

ROM bootstrap word	Vector address	Contents of the MCU reset vector
ROMBS0	$ 000000	Initial ZK, SK, and PK 1
ROMBS1	$ 000002	Initial PC
ROMBS2	$ 000004	Reset-Initial SP
ROMBS3	$ 000006	Initial IZ

Note: 1. Bit [15:12] = Reserved, Bit [11: 8] = ZK initial value, Bit [7: 4] = SK initial value,
Bit [3: 0] = PK initial value

When the address $ 000006 is read, ROM operation returns to normal operation and the module bootstruts.
It will stop responding to vector access.

Chapter 11
Flash EEPROM
(MC68HC916Y5)

Electrically erasable programmable read-only flash memory (flash EEPROM) module
It operates as 160K bytes of non-volatile high-speed access memory.
Operating system kernels, standard subroutines, etc. that need to be executed at high speed
It can be used for program code and frequently executed code.
Flash EEPROM can be used for frequently read static data or system recycle.
It can also be set to provide a bootstrap vector for the set.
The MC68HC916Y5 also has a 4K byte TPU-FLASH EEPROM. About this content
See Chapter 13, " TPU Flash EEPROM. "

11.1 Overview

The flash EEPROM module has five 32-byte cos that occupy a fixed position in the MCU address space.
It consists of a control register block and five 32K byte flash EEPROM arrays.
Five 32K byte flash EEPROM arrays can be mapped to any even 32K byte boundary. child
These arrays should reside in both the program space and the data space, or only in the program space.
Can be configured to.
The flash EEPROM array can be read as bytes, words, or long words.
increase. The module responds to IMB access in succession and has a two-bus rhino to the Arind Long Word.
Allows access with a clock (4 system clocks). Also, a wait state for each access
Slow external development memory without changing system timing as it can be programmed to insert
It is possible to migrate from.
Software for both individual control register bits with shadow bits and arrays
It can be programmed and erased by control. Program / erase voltage is supplied through an external V FPE pin
Have to. Programming is done only in bytes or already words. child
Modules only support bulk erase. Hardware interlock is flash
Stored data if the program / erase voltage to the EEPROM array is accidentally enabled.
It protects the data from being destroyed.
Since this MCU consists of multiple flash EEPROM modules, it also uses multiple reset vectors.
As you might expect, the source of the reset vector is 1 because it prohibits duplication of mappings.
Must be one.

Table 138 Flash EEPROM1 register block address map (32K byte module)

ADDRESS	15	8	7	0
$ YFF7C0	FLASH EEPROM MODULE CONFIGURATION REGISTER (FEE1MCR)			
$ YFF7C2	FLASH EEPROM TEST REGISTER (FEE1TST)			
$ YFF7C4	FLASH EEPROM BASE ADDRESS HIGH REGISTER (FEE1BAH)			
$ YFF7C6	FLASH EEPROM BASE ADDRESS LOW REGISTER (FEE1BAL)			
$ YFF7C8	FLASH EEPROM CONTROL REGISTER (FEE1CTL)			
$ YFF7CA	RESERVED			
$ YFF7CC	RESERVED			
$ YFF7CE	RESERVED			
$ YFF7D0	FLASH EEPROM BOOTSTRAP WORD 0 (FEE1BS0)			
$ YFF7D2	FLASH EEPROM BOOTSTRAP WORD 1 (FEE1BS1)			
$ YFF7D4	FLASH EEPROM BOOTSTRAP WORD 2 (FEE1BS2)			
$ YFF7D6	FLASH EEPROM BOOTSTRAP WORD 3 (FEE1BS3)			
$ YFF7D8	RESERVED			
$ YFF7DA	RESERVED			
$ YFF7DC	RESERVED			
$ YFF7DE	RESERVED			

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

Table 139 Flash EEPROM2 Register Block Address Map (32K Byte Module)

ADDRESS	15	8	7	0
$ YFF7E0	FLASH EEPROM MODULE CONFIGURATION REGISTER (FEE2MCR)			
$ YFF7E2	FLASH EEPROM TEST REGISTER (FEE2TST)			
$ YFF7E4	FLASH EEPROM BASE ADDRESS HIGH REGISTER (FEE2BAH)			
$ YFF7E6	FLASH EEPROM BASE ADDRESS LOW REGISTER (FEE2BAL)			
$ YFF7E8	FLASH EEPROM CONTROL REGISTER (FEE2CTL)			
$ YFF7EA	RESERVED			
$ YFF7EC	RESERVED			
$ YFF7EE	RESERVED			
$ YFF7F0	FLASH EEPROM BOOTSTRAP WORD 0 (FEE2BS0)			
$ YFF7F2	FLASH EEPROM BOOTSTRAP WORD 1 (FEE2BS1)			
$ YFF7F4	FLASH EEPROM BOOTSTRAP WORD 2 (FEE2BS2)			
$ YFF7F6	FLASH EEPROM BOOTSTRAP WORD 3 (FEE2BS3)			
$ YFF7F8	RESERVED			
$ YFF7FA	RESERVED			
$ YFF7FC	RESERVED			
$ YFF7FE	RESERVED			

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

Table 140 Flash EEPROM3 Register Block Address Map (32K Byte Module)

ADDRESS	15	8	7	0
$ YFF800	FLASH EEPROM MODULE CONFIGURATION REGISTER (FEE3MCR)			
$ YFF802	FLASH EEPROM TEST REGISTER (FEE3TST)			

$ YFF804	FLASH EEPROM BASE ADDRESS HIGH REGISTER (FEE3BAH)
$ YFF806	FLASH EEPROM BASE ADDRESS LOW REGISTER (FEE3BAL)
$ YFF808	FLASH EEPROM CONTROL REGISTER (FEE3CTL)
$ YFF80A	RESERVED
$ YFF80C	RESERVED
$ YFF80E	RESERVED
$ YFF810	FLASH EEPROM BOOTSTRAP WORD 0 (FEE3BS0)
$ YFF812	FLASH EEPROM BOOTSTRAP WORD 1 (FEE3BS1)
$ YFF814	FLASH EEPROM BOOTSTRAP WORD 2 (FEE3BS2)
$ YFF816	FLASH EEPROM BOOTSTRAP WORD 3 (FEE3BS3)
$ YFF818	RESERVED
$ YFF81A	RESERVED
$ YFF81C	RESERVED
$ YFF81E	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

Table 141 Flash EEPROM4 Register Block Address Map (32K Byte Module)

ADDRESS	15 8 7 0
$ YFF820	FLASH EEPROM MODULE CONFIGURATION REGISTER (FEE4MCR)
$ YFF822	FLASH EEPROM TEST REGISTER (FEE4TST)
$ YFF824	FLASH EEPROM BASE ADDRESS HIGH REGISTER (FEE4BAH)
$ YFF826	FLASH EEPROM BASE ADDRESS LOW REGISTER (FEE4BAL)
$ YFF828	FLASH EEPROM CONTROL REGISTER (FEE4CTL)
$ YFF82A	RESERVED
$ YFF82C	RESERVED
$ YFF82E	RESERVED
$ YFF830	FLASH EEPROM BOOTSTRAP WORD 0 (FEE4BS0)
$ YFF832	FLASH EEPROM BOOTSTRAP WORD 1 (FEE4BS1)
$ YFF834	FLASH EEPROM BOOTSTRAP WORD 2 (FEE4BS2)
$ YFF836	FLASH EEPROM BOOTSTRAP WORD 3 (FEE4BS3)
$ YFF838	RESERVED
$ YFF83A	RESERVED
$ YFF83C	RESERVED
$ YFF83E	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

Table 142 Flash EEPROM5 Register Block Address Map (32K Byte Module)

ADDRESS	15 8 7 0
$ YFF840	FLASH EEPROM MODULE CONFIGURATION REGISTER (FEE5MCR)
$ YFF842	FLASH EEPROM TEST REGISTER (FEE5TST)
$ YFF844	FLASH EEPROM BASE ADDRESS HIGH REGISTER (FEE5BAH)
$ YFF846	FLASH EEPROM BASE ADDRESS LOW REGISTER (FEE5BAL)
$ YFF848	FLASH EEPROM CONTROL REGISTER (FEE5CTL)
$ YFF84A	RESERVED
$ YFF84C	RESERVED
$ YFF84E	RESERVED
$ YFF850	FLASH EEPROM BOOTSTRAP WORD 0 (FEE5BS0)
$ YFF852	FLASH EEPROM BOOTSTRAP WORD 1 (FEE5BS1)
$ YFF854	FLASH EEPROM BOOTSTRAP WORD 2 (FEE5BS2)
$ YFF856	FLASH EEPROM BOOTSTRAP WORD 3 (FEE5BS3)

$ YFF858	RESERVED
$ YFF85A	RESERVED
$ YFF85C	RESERVED
$ YFF85E	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

11.2 Flash EEPROM control block

Each flash EEPROM control block has a module configuration register
Data (FEEMCR), test register (FEETST), two array-based address registers (FEEBAH)
It contains 5 registers (FEEBAL) and control register (FEECTL). Also, the controller
The four additional words in the block use the flash EEPROM as bootstrap memory.
You can store the bootstrap information when you use it.

The registers in the control block are physically located in the spare flash EEPROM line.
Some have a series of shadow registers. During the reset, the fields in the register will be charged.
The default reset information is loaded from the dow register. Shadow registers correspond
Same method as the location of a flash EEPROM array using the address of the control register
Will be programmed or erased with. Even if you program it in the shadow register, the corresponding control
No data is written to the register. New data will be controlled until the next reset
It will not be copied to the data. The contents of the shadow register are displayed every time the flash EEPROM array is erased.
Will be erased.

Configuration information from the flash EEPROM control block is separate from the array
It is placed in and controls the operation of the flash EEPROM by writing. Writable bits after reset
You can change the registers in the control block with. Write to these registers
But it doesn't affect the associated shadow registers. Some registers are FEEMCR LOCK bits
Written only when is disabled or when the FEEMCR STOP bit is set
I can't. These limits are described in the individual register description.

11.3 Flash EEPROM Array

The base address register specifies the base address of the flash EEPROM array. De
Write the reset base address of the fault to the base address shadow register
I can. The array base address must be on a 32K byte boundary for a 32K byte array
I don't. ADDR [23:20] is driven to the same logical state as ADDR19, so CPU16 is from $ 080000 to $ F7FFFF.
You cannot access addresses in the range of. Flash EEPROM array is mapped to these addresses

Page 266

If so, you will not be able to access this array without remapping the flash.

Module dedicated array overlaps with registers other than its own control register
Do not use a strong base address value. Part of the array overlaps its own register block
If you do, these registers are accessible, but you do not have access to the overlapping parts of the array.
Will be seen. However, if the array overlaps the control blocks of other modules,
If you read the overlapping registers, the result will be undefined.

11.4 Flash EEPROM register

In the register diagram below, the notation SB, which indicates the reset state, is when each control bit is reset.
Indicates that the value of the shadow bit associated with is taken.

FEE1MCR — Flash EEPROM1 module configuration register $ YFF7C0
FEE2MCR — Flash EEPROM2 module configuration register $ YFF7E0
FEE3MCR — Flash EEPROM3 module configuration register $ YFF800
FEE4MCR — Flash EEPROM4 module configuration register $ YFF820
FFE5MCR — Flash EEPROM5 module configuration register $ YFF840

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	FRZ	0	BOOT	LOCK	0	ASPC		WAIT		NOT USED					
RESET:															
SB	0	0	SB	SB	0	SB	SB	SB	SB	0	0	0	0	0	0

The data after clearing this shadow register will be $ BFFF.

STOP — Stop

0 = Normal flash EEPROM array operation.
1 = Disable the flash EEPROM.
The reset state of STOP is the inverted value of the DATA14 logical state being reset and the STOP shadow bit.

It is a logical sum. If STOP is not set, Flash EEPROM Array Base
You cannot change the address.

FRZ — Freeze mode control

0 = Disables the program / erase voltage while asserting the FREEZE signal.
1 = Enables the program / erase voltage to be applied by the ENPE bit while asserting the FREEZE signal.

BOOT — boot ROM control

0 = After reset, CPU16 will access the bootstrap word address of the flash EEPROM.
Set.
1 = After reset, CPU16 will access the bootstrap word address of the flash EEPROM.
I can't set it.

The reset state of BOOT is specified by the user. If STOP = 1, the bootstrap function is disabled.

LOCK — lock register

0 = Write lock disabled. Write to protected registers and fields
Can be done.
1 = Write lock enable. Writing to protected registers and fields
Can't.

The reset state of LOCK is specified by the user. If the default reset state for LOCK is zero, It can be set only once after a master reset so that the registers can be protected after initial setup. to come. LOCK also protects the ASPC and WAIT fields, FEEBAL and FEEBAH registers. ASPC, FEEBAL, and FEEBAH are also protected by the STOP bit.

ASPC — Flash EEPROM Array Space

Since the MCU only operates in supervisor mode, access is programmatically empty in the ASPC field. Decide whether it is limited to the interval or is done in both the program space and the data space. vinegar.

Page 267

The reset state of ASPC is user-specified.

Table 143 ASPC encoding

ASPC [1: 0]	Specified state
X0	Program and data access
X1	Program access only

WAIT — Wait state field

WAIT is inserted by FLASH EEPROM MODULE while accessing the flash EEPROM array. Specifies the number of wait states. Also, the wait state inserted before the internal DTACK is generated. By controlling the number, the user can also optimize the bus speed for a particular application. increase. The length of each wait state is one system clock cycle. This allows the user Is a slow emulation or development system without timing the system You can migrate the code from the moly to the EEPROM array. User-specified WAIT reset state To do. No-wait encoding (% 00) corresponds to the 3-clock cycle bus is. The fast end encoding (% 11) supports a 2-clock cycle bus. Microcontroller modules usually respond at this speed, but use fast termination to go out fast. It is also possible to access the internal memory.

Table 144 WAIT encoding

WAIT [1: 0]	Transfer cycle
00 00	3
01 01	Four
Ten	Five
11 11	2

FEE1TST — Flash EEPROM1 test register $ YFF7C2
FEE2TST — Flash EEPROM2 test register $ YFF7E2
FEE3TST — Flash EEPROM3 test register $ YFF802
FEE4TST — Flash EEPROM4 test register $ YFF822
FEE5TST — Flash EEPROM5 test register $ YFF842

This register is for factory testing only.

FEE1BAH — Flash EEPROM1 base address high register $ YFF7C4
FEE2BAH — Flash EEPROM2 base address high register $ YFF7E4

FEE3BAH — Flash EEPROM3 base address high register $ YFF804
FEE4BAH — Flash EEPROM4 base address high register $ YFF824

FEE5BAH — Flash EEPROM5 base address high register $ YFF844

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								ADDR twenty three	ADDR twenty two	ADDR twenty one	ADDR 20	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16
RESET:															
								SB	SB	SB	SB	SB	SB	SB	SB

The data after clearing this shadow register will be $ FFFF.

FEEBAH has the upper 16 bits of the flash EEPROM array base address. Reset
At that time, the data written in the associated shadow register is automatically copied to FEEBAH.
increase. The default value for FEEBAH is always the data in the associated shadow register. After reset
If LOCK = 0 and STOP = 1, the software writes to FEEBAH and FEEBAL and flaps.

The EEPROM array can be rearranged. ADDR [23:20] is dry to the same logical state as ADDR19
CPU16 will not be able to access addresses in the range $ 080000 to $ F7FFFF. Fluff
If the EEPROM array is mapped to these addresses, do not reset the system
And you cannot access the array.

FEE1BAL — Flash EEPROM1 base address lower register $ YFF7C6

FEE2BAL — Flash EEPROM2 base address lower register $ YFF7E6

FEE3BAL — Flash EEPROM3 base address low register $ YFF806

FEE4BAL — Flash EEPROM4 base address lower register $ YFF826

FEE5BAL — Flash EEPROM5 base address lower register $ YFF846

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
SB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The data after clearing this shadow register will be $ FFFF.

FEEBAL has the lower 16 bits of the flash EEPROM array base address. Reset
At that time, FEEBAL automatically copies the data written to the associated shadow register.
increase. The default value for FEEBAL is always the data in the associated shadow register.

FEE1CTL — Flash EEPROM1 control register $ YFF7D8

FEE2CTL — Flash EEPROM2 control register $ YFF7E8

FEE3CTL — Flash EEPROM3 control register $ YFF808

FEE4CTL — Flash EEPROM4 control register $ YFF828

FEE5CTL — Flash EEPROM5 control register $ YFF848

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	VFPE	ERAS	LAT	ENPE
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FEECTL contains the bits needed to control the programming and erasure of the flash EEPROM.
This register can only be accessed in supervisor mode.

VFPE — Program / Erase Verification

0 = normal read cycle
1 = Start the program verification circuit.

This bit activates a special program verification circuit. Programming sequence (ERAS =
During 0), VFPE, along with the LAT bit, determines that programming for a location is complete.
Used to do. A valid flash EEPROM if both VFPE and LAT are set.
When the location is read, the latched data and the data of the location in the program are bitten.
Exclusive OR is taken in units of G. If the location is fully programmed,
A value of zero is read. If a non-zero value is read, the location is completely professional
Indicates that it is not grammed.

When VFPE is cleared, a normal read of a valid flash EEPROM location is performed.
It will be done.

ERAS — Erase control

0 = Set the flash EEPROM to programming.
1 = Set the flash EEPROM to erase.

When ERAS is asserted, all locations in the array and shadow registers are set to erase at the same time. It will be.

When the LAT bit is set, ERAS is further addressed by reading. May return the value of either application (ERAS = 1) or location in the program (ERAS = 0) To decide.

The ERAS value cannot be changed when the program / erase voltage is applied (ENPE = 1).

LAT — Latch control

0 = Disable programming latch.
1 = Enable programming latch.

When LAT is cleared, the flash EEPROM address and data bus will be the IMB address bus. Connected to the data bus, the flash EEPROM is set to normal read. LAT is set When is, the flash EEPROM address and data bus are connected to a parallel internal latch and flap. The EEPROM array is set to programming or erasing.

Immediately after setting LAT, writing to a valid address in the flash EEPROM module Both the write address and the data are latched. When set to programming (ERAS = 0), the latched data is programmed for the latched address. Set to erase If yes (ERAS = 1), batch if the latched address is a valid address in the flash EEPROM It will be erased (the data to be latched is optional).

The value of LAT cannot be changed when the program / erase voltage is applied (ENPE = 1).

ENPE — Programming / Erase Enable

0 = Disable program / erase voltage.
1 = Apply the program / erase voltage.

ENPE is set only after LAT is set and data and writes to the address latch have been made. I can't. If these conditions are not met, ENPE remains cleared. ENPE is set The LAT, VFPE, and ERAS bits cannot be changed while they are in the flash EEPROM module. Attempts to read the flash EEPROM array location will be ignored.

FEE1BS [0: 3] — Flash EEPROM1 bootstrap word	$ YFF7D0-$ YFF7D6
FEE2BS [0: 3] — Flash EEPROM2 bootstrap word	$ YFF7F0-$ YFF7F6
FEE3BS [0: 3] — Flash EEPROM3 bootstrap word	$ YFF810-$ YFF816
FEE4BS [0: 3] — Flash EEPROM4 bootstrap word	$ YFF830-$ YFF836
FEE5BS [0: 3] — Flash EEPROM5 bootstrap word	$ YFF850-$ YFF856

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
BOOTSTRAP VECTOR															
RESET:															
PROGRAMMED VALUE															

The data after clearing this register will be $ FFFF.

These words can be used as a system bootstrap vector. Reset When the BOOT bit of FEEMCR is 0, the FEEBS [0: 3] of the flash EEPROM module is set. Responds to set vectors from $ 000000 to $ 000006. This response is only the first access The array supports access to $ 000000 to $ 000006 from the second time onward. When BOOT = 1, The array of flash EEPROM modules responds to the set vector. FEEBS [0: 3] is It can be read at any time in FEEMCR when STOP = 0, but the value in the word is the appropriate location. It cannot be changed without programming the option.

11.5 Flash EEPROM operation

Below are the modules for system bootstrap, normal operation, and array programming / disappearance. This section describes resetting the flash EEPROM module used earlier.

11.5.1 Reset operation

The reset initializes the control registers of all flash EEPROMs. child The bit of the register is a fixed default value or the associated flash EEPROM shadow register. The logged value is set.

If the STOP shadow bit state is zero, the FEEMCR STOP bit is cleared during reset and is displayed.

The joule responds to the range of accesses specified by FEEBAH and FEEBAL. BOOT bit cleared
If so, the module will also respond to bootstrap vector access.

When the STOP shadow bit state is 1, the FEEMCR STOP bit is set at reset.
The flash EEPROM array is disabled. The module will continue until the STOP bit is cleared.
Does not respond to array or bootstrap vector access. This will result in an external device
Is a shadow bit access or bootstracket to the flash EEPROM array address space
You can respond to access. The erase state of the shadow bit is 1. Erased mod
After the reset is released, the system will be in STOP mode.

11.5.2 Bootstrap operation

CPU16 is the initial value of the internal register located at the addresses $ 000000 to $ 000006 in the program space.
Starts the bootstrap operation by fetching. These are exception vector tables
The address of the bootstrap vector in. When FEEMCR STOP = 0 and BOOT = 0 at reset,
The rush EEPROM module is configured to respond to bootstrap vector access.
The vector allocation is as follows:

Table 145 Bootstrap mapping

EEPROM bootstrap word	Vector address	Contents of the MCU reset vector
FEEBS0	$ 000000	Initial ZK, SK, and PK 1
FEEBS1	$ 000002	Initial PC
FEEBS2	$ 000004	Reset-Initial SP
FEEBS3	$ 000006	Initial IZ

Note: 1. Bit [15:12] = Reserved, Bit [11: 8] = ZK initial value, Bit [7: 4] = SK initial value,
Bit [3: 0] = PK initial value

When the address $ 000006 is read, the flash EEPROM operation returns to normal operation and the module returns.
It will stop responding to bootstrap vector access.

11.5.3 Normal operation

The flash EEPROM module is a byte or bind word access with two system clocks.
Align long word access or in response to running IMB access
Can supply 4 system clocks for misaligned word operations. Miss Arind Lo
3-bus cycle access for words is for CPUs capable of 32-bit processing.

The module checks the function code to see the address space access type.
increase. Array access is defined by the ASPC state of FFEMCR. Flash EEPROM Mod
When set to normal operation, the array only responds to read access. No write operation
Will be seen.

11.5.4 Program / Erase operation

The unprogrammed flash EEPROM bits are in the logical 1 state. Bits state
It must be programmed to change from 1 to 0. When the bit is erased, it returns to the state of 1.
vinegar. To program / erase the flash EEPROM, write and press a series of control registers.
A write to the logging latch is required. Program the shadow register using the same procedure
I will. Only 1-byte or 1-alind words can be programmed at the same time. Fluff
The entire EEPROM array and the shadow register bits are erased at the same time.

Note: To program the array , a programming voltage must be applied to the V FPE pins.
I don't. The applied voltage must be within the range of 12.6 ≧ V FPE ≧ (SiO − 0.35V).
It will not be. Beyond this range will damage the flash EEPROM module.
There is a record.

11.5.4.1 Programming sequence

To program the flash EEPROM array, complete the following steps: Figure 39 shows programmin
It is a flowchart of the operation.

1. Raise the voltage applied to the V FPE pin to the program / erase / verification level.

2. Clear the NPP counter, which is a parameter on the software, and clear the margin flag. vinegar. This NPP counter is used to check the number of programs (up to 50 times). margin· The flag is a flag used for the pros and cons of reprogramming the programmed data. vinegar.

3. Clear the ERAS bit in FEECTL and set the LAT bit. This causes the address latch And the data latch can be programmed. Also VFPE if necessary in this step You can set it at times and start the program verification circuit.

4. Write the data to the address you want to program. This sets the module to the program Will be done.

5. Start writing to the flash by setting the ENPE bit in FEECTL. That Start the program pulse to check the write time.

6. Delay the time for one programming pulse to occur. The delay is specified by the parameter PWPP It will be.

7. Clear the ENPE bit in FEECTL. This will stop the program pulse and in the array Stops the high voltage supply to the unit.

8. Wait for the high voltage to the array to drop to the steady voltage. This waiting time (delay time) is a parameter Specified by TPR.

9. If the margin flag is not already set, increment the NPP counter and program location. Read the program and verify that it is programmed. If the VFPE bit is set, If programmed correctly, 0 will be read.

   If programmed, set the margin flag and go to "Set ENPE" in step 5. to go.

   If not programmed, the NPP counter value is the maximum number of programmed pulses NPPMAX (50 times) If you haven't reached), go to step 5 "Set ENPE".

   If NPP = NPPMAX (50 times) is reached, the program will fail. Suspension procedure To clear the LAT bit of the FEECTL and pass the program voltage applied to the V_{FPE} pin. Decrease to the normal read level (VDD level). Ends the program sequence.

   If the margin flag is set, decrement the NPP counter value by 1 until NPP = 0. Repeat "Decrease NPP Counter" in this step from "Set ENPE" in step 5. this Gives 100% program margin.

   Then read and validate the program location again.

   If it is not programmed, the program will fail. FEECTL for suspension procedure The LAT bit of is cleared, and the program voltage applied to the V_{FPE} pin is read normally. Lower to the bell (VDD level). Ends the program sequence.

   If programmed, clear the LAT bit in FEECTL and write all in step 10. Go to "End?"

10. If all writes have not been completed, then the address to program the next location. Set the flow and repeat from "Clear NPP counter / Clear margin flag" of the flow.

Page 272

To do.

If all the writing is completed, it means that the writing is completed normally. As a termination procedure The voltage applied to the V_{FPE} pin is lowered to the normal read level, and the process ends normally.

1: The margin flag is a software-defined flag. to this More margin pulses are used for write margins.

2: The ERAS bit in the FEECTL register is used when erasing. VFPE You can set the bit and start the program verification circuit vinegar.

3: If the VFPE bit is used, the VFPE bit is also cleared.

Program start

To write / erase / verification level
Increase the V_{FPE} pin voltage

1

Clear NPP counter
Click the margin flag

2

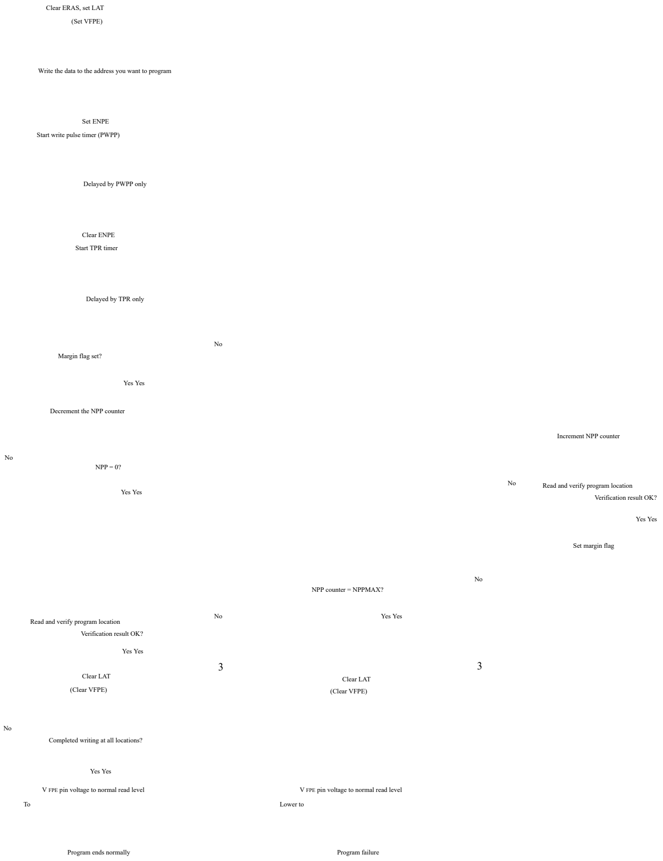


Figure 39 Flash Program Flowchart (see Table 178)

11.5.4.2 Erase operation

To erase the flash EEPROM array, complete the following steps: Figure 40 shows the flow of erasing operation . It is a chart.

1. Raise the voltage applied to the V FPE pin to the program / erase / verification level.

2. Set the erasure pulse counter NEP, which is a parameter on the software, to 1, and set the margin flag. Clear.

3. Clear the VFPE bit in FEECTL and set the ERAS and LAT bits. by this, The FLASH module is set to erase mode.

4. Write to any valid address in the array or control block (shadow) vinegar. It doesn't matter what data you write.

5. Set the ENPE bit in FEECTL. This applies an erasing voltage to the array and shadow. increase.

6. Delay by the time corresponding to one erasure pulse. The delay is specified by the parameter TEPK.

7. Clear the ENPE bit in FEECTL. This causes the array and shadow to rise inside the array. Stop the voltage supply.

8. Wait for the high voltage to the array and shadow to drop to steady-state voltage. This waiting time (during delay)) Is specified by the parameter TER.

9. If the margin flag is not already set, read the entire array and shadows and everything Verify that the location of is erased (because the shadow erase value of FEEMCR is $ BFFF)

be careful).

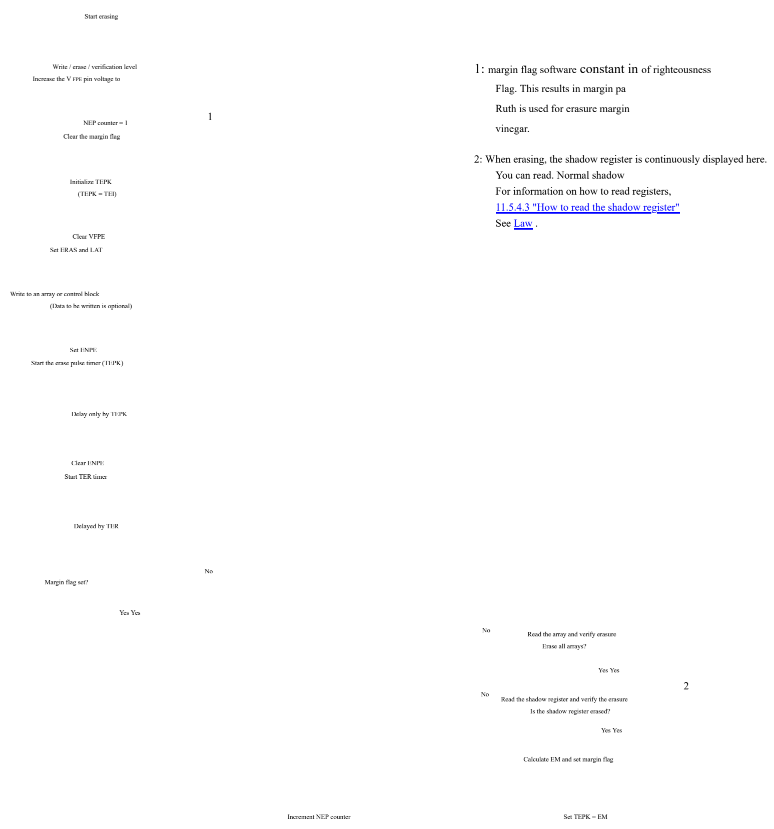
If all are erased, calculate the erase margin EM, set the margin flag and step
Go to 5 "Set ENPE".

If not all erased, increment the NEP counter by 1 and the NEP counter value is the maximum erase pulse.
If the number NEPMAX (5 times) is not exceeded, calculate a new TEPK value and set "ENPE" in step 5.
"Go to.

If the NEP counter value exceeds NEPMAX (5 times), the erase will fail. Suspension procedure
To clear the LAT and ERAS bits of the FEECTL, the erasing power applied to the V FPE pin
Reduce the pressure to the normal read level (VDD level).

If the margin flag is set, clear the LAT and ERAS bits of the FEECTL.

10. Erase ends normally. Normal reading of the erasing voltage applied to the V FPE pin as a termination procedure
Lower to the output level (VDD level).


Start erasing
Write / erase / verification level
Increase the V FPE pin voltage to
1
NEP counter = 1
Clear the margin flag
Initialize TEPK
(TEPK = TEI)
Clear VFPE
Set ERAS and LAT
Write to an array or control block
(Data to be written is optional)
Set ENPE
Start the erase pulse timer (TEPK)
Delay only by TEPK
Clear ENPE
Start TER timer
Delayed by TER
No
Margin flag set?
Yes Yes
No
Read the array and verify erasure
Erase all arrays?
Yes Yes
2
No
Read the shadow register and verify the erasure
Is the shadow register erased?
Yes Yes
Calculate EM and set margin flag
Increment NEP counter
Set TEPK = EM

1: margin flag software constant in of righteousness
Flag. This results in margin pa
Ruth is used for erasure margin
vinegar.

2: When erasing, the shadow register is continuously displayed here.
You can read. Normal shadow
For information on how to read registers,
11.5.4.3 "How to read the shadow register"
See Law .

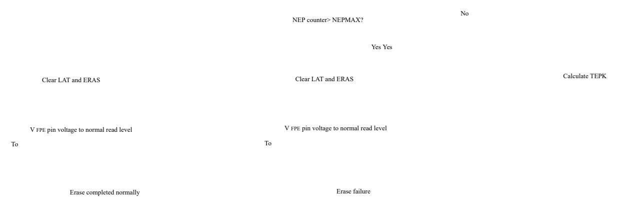


Figure 40 Flash Erasing Flowchart (see Table 178)

11.5.4.3 How to read the shadow register

Flash EEPROM module for MC68HC916 series (HD FLASH, FLASH, BE FLASH and TPU FLASH)
When it is necessary to read the data in the shadow register of the above during normal times (other than during programming or erasing).
If so, follow the steps below.

1. Clears all bits of the control register (FEECTL, HDFCTL, TFCTL, BEFCTL).
2. Set only the LAT bit of the control register.
3. Writes to the address of the shadow register to be read. Any data to write I do not mind. This operation latches the address and data.

Four. Reads the shadow location where the write was made. The read data is the shadow register It is the data of the star.

Five. Clears the LAT bit. As a result, the flash EEPROM module returns to normal operation. Allows access to the ray area and control blocks.

6. Repeat steps 2 through 5 if there are more shadow registers you want to read.

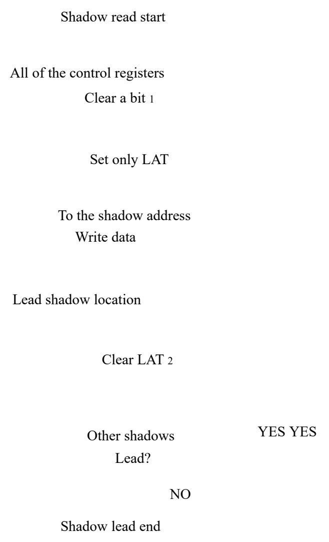


Figure 41 How to read the shadow register

Note: 1. The flash module must be enabled (the STOP bit is cleared).
Is required.
2. The address of the shadow written immediately after setting LAT is the program latch times.
Retained on the road, subsequent read access to the flash will be on the latched shadow
Read as data. Therefore, to read the next shadow, click LAT once.
Then, it is necessary to set LAT again and write the address of the shadow to be read.
I will.

11.5.5 Flash EEPROM write / erase voltage signal conditions

The minimum VDD − 0.35V voltage should be applied to the V FPE pin at all times. This condition is met
Otherwise, you may damage the FLASH module. The FLASH module is available until the V FPE is powered on.
It can also be damaged by excessive response during power off. VDD is lower than the specified minimum voltage
In the meantime , the V FPE cannot be raised to the write level. Further, while the VDD is supplied, V FPE the
It cannot be lowered below the specified minimum voltage. Figure 42 shows the operating waveforms for V FPE and VDD.

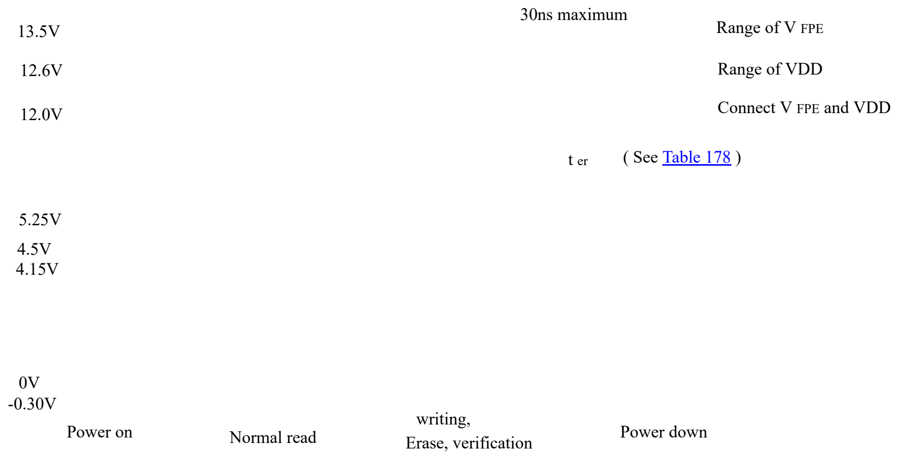
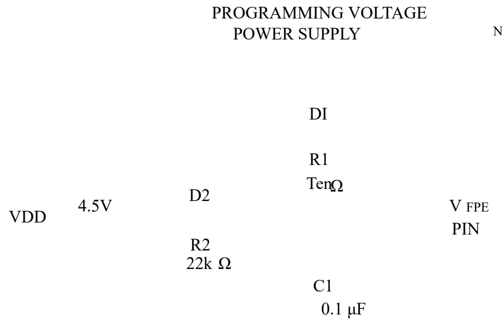


Figure 42 Write voltage waveform

Conditioning the V FPE with an external circuit is recommended. Figure 43 shows that the required voltage is maintained.
Here is a simple circuit that filters transient signals. V FPE is connected to VDD through Schottky diode D2
It will be up. Applying a programming voltage through diode D1 reverse biases D2 and
Protects VDD from excessive reverse current. D2 also FLASH in the unlikely event that the programming voltage goes to zero
Protects against damage. Adjust the programming power supply voltage to compensate for the forward bias drop at D1.
I have to make up for it. The transient signal is filtered by the charging time constants of R1 and C1, and the discharge path of C1 is defined by R2.
Offers.

This RC charge / discharge time constant can be effectively used when the power is turned on / off. When using this circuit
Low leakage current from external devices connected to V FPE pins to reduce diode voltage drop
Please hold it down.



Note: A 12V system voltage is applied to the V FPE pin.
If the RC circuit (R1 and C1) is noise reduction
Work for Therefore, this RC
The circuit must be located on the V FPE pin side
I can't. Also, there are multiple V FPE pins
If so, install the RC circuit individually.
I recommend it.

Figure 43 V FPE conditioning circuit

Page 277

Chapter 12
HD Flash EEPROM
(MC68HC916Y6)

Electrically Erasable Programmable Read-Only Flash Memory (HD Flash EEPROM) Module Acts as 160K bytes of non-volatile high-speed access memory.

Pros that need to run fast, such as operating system kernels and standard subroutines It can be used for gram code and frequently executed code.

HD flash EEPROM can be used for frequently read static data or system recycle. It can also be set to provide a bootstrap vector for the set.

The MC68HC916Y6 also has a 4K byte TPU-FLASH EEPROM. For the contents of the first See Chapter 13, "TPU Flash EEPROM" .

12.1 Overview

The HD Flash EEPROM module has three 64-bytes that occupy a fixed location in the MCU address space. Control register block and (2 64K bytes + 1 32K bytes) HD flash EEPROM It consists of an array. The 64Kbyte HD Flash EEPROM array has any even 64Kbyte boundaries. It is possible to map to the world. 32Kbyte HD Flash EEPROM array on any even 32Kbyte boundary Map is possible. These arrays can be in both program space and data space, or in program space. Can be configured to be resident only.

The HD flash EEPROM array can be read as bytes, words, or long words. increase. The module responds to IMB access in succession and has a two-bus rhino to the Arind Long Word. Allows access with a clock (4 system clocks). Also, a wait state for each access Slow external development memory without changing system timing as it can be programmed to insert It is possible to migrate from.

Software for both individual control register bits and arrays with shadow bits It can be programmed and erased by control. Program / erase voltage is supplied through an external V_{FPE} pin Must be. Programming is done only in bytes or already words. child The module supports the block erase function every 16K blocks. Hardware inter Lock if accidentally enabled program / erase voltage to HD flash EEPROM array In addition, it protects the stored data from corruption.

Since this MCU consists of multiple HD flash EEPROM modules, it also has multiple reset vectors. A number will be prepared, but since duplication of mapping is prohibited, the source of the reset vector Must be one.

Page 278

Table 146 HD Flash EEPROM1 Register Block Address Map (64K Byte Module)

ADDRESS	15	8	7	0

$ YFF780	HD FLASH EEPROM MODULE CONFIGURATION REGISTER (HDF1MCR)
$ YFF782	HD FLASH EEPROM TEST REGISTER (HDF1TST)
$ YFF784	HD FLASH EEPROM BASE ADDRESS HIGH REGISTER (HDF1BAH)
$ YFF786	HD FLASH EEPROM BASE ADDRESS LOW REGISTER (HDF1BAL)
$ YFF788	HD FLASH EEPROM CONTROL REGISTER (HDF1CTL)
$ YFF78A	RESERVED
$ YFF78C	RESERVED
$ YFF78E	RESERVED
$ YFF790	HD FLASH EEPROM BOOTSTRAP WORD 0 (HDF1BS0)
$ YFF792	HD FLASH EEPROM BOOTSTRAP WORD 1 (HDF1BS1)
$ YFF794	HD FLASH EEPROM BOOTSTRAP WORD 2 (HDF1BS2)
$ YFF796	HD FLASH EEPROM BOOTSTRAP WORD 3 (HDF1BS3)
$ YFF798	RESERVED
$ YFF79A	RESERVED
$ YFF79C	RESERVED
$ YFF79E	RESERVED
$ YFF7A0	RESERVED
$ YFF7A2	RESERVED
$ YFF7A4	RESERVED
$ YFF7A6	RESERVED
$ YFF7A8	RESERVED
$ YFF7AA	RESERVED
$ YFF7AC	RESERVED
$ YFF7AE	RESERVED
$ YFF7B0	RESERVED
$ YFF7B2	RESERVED
$ YFF7B4	RESERVED
$ YFF7B6	RESERVED
$ YFF7B8	RESERVED
$ YFF7BA	RESERVED
$ YFF7BC	RESERVED
$ YFF7BE	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

Page 279

Table 147 HD Flash EEPROM2 Register Block Address Map (64K Byte Module)

ADDRESS	15 8 7 0
$ YFF7C0	HD FLASH EEPROM MODULE CONFIGURATION REGISTER (HDF2MCR)
$ YFF7C2	HD FLASH EEPROM TEST REGISTER (HDF2TST)
$ YFF7C4	HD FLASH EEPROM BASE ADDRESS HIGH REGISTER (HDF2BAH)
$ YFF7C6	HD FLASH EEPROM BASE ADDRESS LOW REGISTER (HDF2BAL)
$ YFF7C8	HD FLASH EEPROM CONTROL REGISTER (HDF2CTL)
$ YFF7CA	RESERVED
$ YFF7CC	RESERVED
$ YFF7CE	RESERVED
$ YFF7D0	HD FLASH EEPROM BOOTSTRAP WORD 0 (HDF2BS0)
$ YFF7D2	HD FLASH EEPROM BOOTSTRAP WORD 1 (HDF2BS1)

$ YFF7D4	HD FLASH EEPROM BOOTSTRAP WORD 2 (HDF2BS2)
$ YFF7D6	HD FLASH EEPROM BOOTSTRAP WORD 3 (HDF2BS3)
$ YFF7D8	RESERVED
$ YFF7DA	RESERVED
$ YFF7DC	RESERVED
$ YFF7DE	RESERVED
$ YFF7E0	RESERVED
$ YFF7E2	RESERVED
$ YFF7E4	RESERVED
$ YFF7E6	RESERVED
$ YFF7E8	RESERVED
$ YFF7EA	RESERVED
$ YFF7EC	RESERVED
$ YFF7EE	RESERVED
$ YFF7F0	RESERVED
$ YFF7F2	RESERVED
$ YFF7F4	RESERVED
$ YFF7F6	RESERVED
$ YFF7F8	RESERVED
$ YFF7FA	RESERVED
$ YFF7FC	RESERVED
$ YFF7FE	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

Page 280

Table 148 HD Flash EEPROM3 Register Block Address Map (32K Byte Module)

ADDRESS	15	8	7	0
$ YFF800	HD FLASH EEPROM MODULE CONFIGURATION REGISTER (HDF3MCR)			
$ YFF802	HD FLASH EEPROM TEST REGISTER (HDF3TST)			
$ YFF804	HD FLASH EEPROM BASE ADDRESS HIGH REGISTER (HDF3BAH)			
$ YFF806	HD FLASH EEPROM BASE ADDRESS LOW REGISTER (HDF3BAL)			
$ YFF808	HD FLASH EEPROM CONTROL REGISTER (HDF3CTL)			
$ YFF80A	RESERVED			
$ YFF80C	RESERVED			
$ YFF80E	RESERVED			
$ YFF810	HD FLASH EEPROM BOOTSTRAP WORD 0 (HDF3BS0)			
$ YFF812	HD FLASH EEPROM BOOTSTRAP WORD 1 (HDF3BS1)			
$ YFF814	HD FLASH EEPROM BOOTSTRAP WORD 2 (HDF3BS2)			
$ YFF816	HD FLASH EEPROM BOOTSTRAP WORD 3 (HDF3BS3)			
$ YFF818	RESERVED			
$ YFF81A	RESERVED			
$ YFF81C	RESERVED			
$ YFF81E	RESERVED			
$ YFF820	RESERVED			
$ YFF822	RESERVED			
$ YFF824	RESERVED			

$ YFF826	RESERVED
$ YFF828	RESERVED
$ YFF82A	RESERVED
$ YFF82C	RESERVED
$ YFF82E	RESERVED
$ YFF830	RESERVED
$ YFF832	RESERVED
$ YFF834	RESERVED
$ YFF836	RESERVED
$ YFF838	RESERVED
$ YFF83A	RESERVED
$ YFF83C	RESERVED
$ YFF83E	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

12.2 HD Flash EEPROM Control Block

Each HD flash EEPROM control block has a module configuration check
Gista (HDFMCR), test register (HDFTST), two array-based address registers (HDFBAH)
And HDFBAL), the control register (HDFCTL) contains five registers. Also, the control
The four additional words in the roll block include HD flash EEPROM as bootstrap memory.
It can contain bootstrap information when used.

The registers in the control block are physically located in the spare HD flash EEPROM line.
Some have associated shadow registers. During the reset, the fields in the register

The default reset information is loaded from the shadow register. Shadow register is supported
Same as the location of the HD Flash EEPROM array using the address of the control register
It will be programmed or erased in the same way. Even if you program it in the shadow register, the corresponding control
No data is written to the role register. New data is controlled until the next reset
It is not copied to the register.

Configuration information from the HD Flash EEPROM control block is separate from the array
It is placed in pieces and controls the operation of the HD flash EEPROM by writing. Writable after reset
You can change the registers in the control block that have bits. In these registers
Writing does not affect the associated shadow register. Some registers are HDFMCR LOCK
When the bit is disabled or when the HDFMCR STOP bit is set.
I can't write. These limits are described in the individual register description.

12.3 HD Flash EEPROM Array

The base address register specifies the base address of the HD flash EEPROM array. De
Write the reset base address of the fault to the base address shadow register
I can. The array-based address is on a 64K-byte boundary for a 64-Kbyte array, with a 32K buy.
For array, it must be on a 32K byte boundary. ADDR [23:20] is in the same logical state as ADDR19
Because it is driven, CPU16 cannot access addresses in the range $ 080000 to $ F7FFFF. HD
If the rush EEPROM array is mapped to these addresses, the HD flash must be remapped.
You cannot access this array.

Module dedicated array overlaps with registers other than its own control register
Do not use a strong base address value. Part of the array overlaps its own register block
If you do, these registers are accessible, but you do not have access to the overlapping parts of the array.
Will be seen. However, if the array overlaps the control blocks of other modules,
If you read the overlapping registers, the result will be undefined.

12.4 HD Flash EEPROM Register

In the register diagram below, the notation SB, which indicates the reset state, is when each control bit is reset.
Indicates that the value of the shadow bit associated with is taken.

HDF1MCR — HD Flash EEPROM Module Configuration Register $ YFF780
HDF2MCR — HD Flash EEPROM Module Configuration Register $ YFF7C0
HDF3MCR — HD Flash EEPROM Module Configuration Register $ YFF800

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0	
STOP	FRZ	0	BOOT	LOCK	EMUL	ASPC		WAIT				NOT USED				
RESET:																

SB	0	0	SB	SB	0	SB	SB	SB	SB	0	0	0	0	0	0

The data after clearing this shadow register will be $ FFFF.

STOP — Stop

0 = Normal HD flash EEPROM array operation.
1 = Disables the HD flash EEPROM.

The reset state of STOP is the inverted value of the DATA14 logical state being reset and the STOP shadow bit. It is a logical sum. HD Flash EEPROM Array Base if STOP is not set
You cannot change the address.

FRZ — Freeze mode control

0 = Disables the program / erase voltage while asserting the FREEZE signal.
1 = Allows the program / erase voltage to be applied by the EHV bit while asserting the FREEZE signal.

Page 282

BOOT — boot ROM control

0 = After reset, CPU16 goes to the bootstrap word address of the HD flash EEPROM.
to access.
1 = After reset, CPU16 goes to the bootstrap word address of the HD flash EEPROM.
I can't access it.

The reset state of BOOT is specified by the user. If STOP = 1, the bootstrap function is disabled.

LOCK — lock register

0 = Write lock disabled. Write to protected registers and fields
Can be done.
1 = Write lock enable. Writing to protected registers and fields
Can't.

The reset state of LOCK is specified by the user. If the default reset state for LOCK is zero, It can be set only once after a master reset so that the registers can be protected after initial setup.
to come. LOCK also protects the ASPC and WAIT fields, HDFBAL and HDFBAH registers.
ASPC, HDFBAL, and HDFBAH are also protected by the STOP bit.

EMUL — emulation status bit

This bit is currently unavailable.

ASPC — HD Flash EEPROM Array Space

Since the MCU only operates in supervisor mode, access is programmatically empty in the ASPC field.
Decide whether it is limited to the interval or is done in both the program space and the data space.
vinegar.

The reset state of ASPC is user-specified.

Table 149 ASPC encoding

ASPC [1: 0]	Specified state
X0	Program and data access
X1	Program access only

WAIT — Wait state field

WAIT inserted by FLASH EEPROM MODULE while accessing the HD Flash EEPROM array
Specifies the number of wait states. Also, a weight stay that is inserted before the internal DTACK is generated.
By controlling the number of buses, the user can also optimize the bus speed for a particular application.
to come. The length of each wait state is one system clock cycle. This allows you
The is a slow emulation or development system without timing the system.
You can migrate the code from memory to the EEPROM array. The reset state of WAIT is fingered by the user
Set. No-wait encoding (% 00) supports 3 clock cycle buses
I am. The fast end encoding (% 11) supports a 2-clock cycle bus.
Microcontroller modules usually respond at this speed, but use fast termination to go out fast.
It is also possible to access the internal memory.

Table 150 WAIT encoding

WAIT [1: 0]	Transfer cycle
00 00	3
01 01	Four

Ten Five

11 11 2

HDF1TST — HD Flash EEPROM Test Register $ YFF782
HDF2TST — HD Flash EEPROM Test Register $ YFF7C2
HDF3TST — HD Flash EEPROM Test Register $ YFF802

This register is for factory testing only.

HDF1BAH — HD Flash EEPROM Base Address Higher Register $ YFF784
HDF2BAH — HD Flash EEPROM Base Address Higher Register $ YFF7C4
HDF3BAH — HD Flash EEPROM Base Address Higher Register $ YFF804

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								ADDR twenty three	ADDR twenty two	ADDR twenty one	ADDR 20	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16
RESET:															
								SB	SB	SB	SB	SB	SB	SB	SB

The data after clearing this shadow register will be $ FFFF.

HDFBAH has the upper 16 bits of the HD Flash EEPROM array base address. Ri
At set time, HDFBAH automatically copies the data written to the associated shadow register.
Will be done. The default value for HDFBAH is always the data in the associated shadow register. Reset
After that, if LOCK = 0 and STOP = 1, the software writes to HDFBAH and HDFBAL and HD
You can relocate the rush EEPROM array. ADDR [23:20] is in the same logical state as ADDR19
Because it is live, CPU16 cannot access addresses in the range $ 080000 to $ F7FFFF. HD
If the rush EEPROM array is mapped to these addresses, reset the system first.
Without it, you will not be able to access the array.

HDF1BAL — HD Flash EEPROM Base Address Low Register $ YFF786
HDF2BAL — HD Flash EEPROM Base Address Low Register $ YFF7C6

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The data after clearing this shadow register will be $ FFFF.

HD flash EEPROM array-based add because it maps to any even 64K byte boundary
The lower 16 bits of the less are not used.

HDF3BAL — HD Flash EEPROM Base Address Low Register $ YFF806

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
SB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The data after clearing this shadow register will be $ FFFF.

HD flash EEPROM array-based add because it maps to any even 32K byte boundary
The lower 15 bits of the less are not used.

HDF1CTL — HD Flash EEPROM Control Register $ YFF788
HDF2CTL — HD Flash EEPROM Control Register $ YFF7C8
HDF3CTL — HD Flash EEPROM Control Register $ YFF808

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	VPE	PE	LAT	EHV
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The HDFCTL contains the bits needed to program and control the erasure of the HD Flash EEPROM. increase. This register can only be accessed in supervisor mode.

VPE — Program / Erase Verification

0 = normal read cycle
1 = Start the program verification circuit.

This bit activates a special program verification circuit. Set LAT (bit 1) and this bit When you read the location on the flash, the inverted value of the latched data and the data on the real array Read the logical product of.

When the VPE bit is set and the data is read, the threshold value rises (it is shifted in a severe direction). Therefore, it should be used with VPE = 0 to keep the cell characteristics compatible with other flash modules. It is recommended.

The values read by the VPE and PE settings when LAT is set are shown below.

Table 151 Flash read value when the verification circuit is started

Data in latch		0	0	1	1
Current cell state		0	1	0	1
VPE	PE				
0	0	0	1	0	1
0	1	0	1	0	1
1	0	0	1	0	0
1	1	0	1	0	1

Note: When VPE = 1 and PE = 0, the array is in the programmed state (0) and the latched data is 1.
Cannot be verified (latch and cell values do not match, but the read result is 0
Therefore, you need to be careful. However, erase it before programming.
If you follow the rules, you will not be in this state.

PE — Erase control

0 = Set HD Flash EEPROM to programming.
1 = Set HD Flash EEPROM to Erase.

When PE is asserted, the location of the array and the shadow bits of the control block Is set to erase.

When the LAT bit is set, the PE is further addressed by the read location. It also determines whether the value of the option (PE = 1) or the location in the program (PE = 0) is returned. When the program / erase voltage is applied (EHV = 1), the PE value cannot be changed. In addition, this flash memory can select the block at the time of erasing.

Table 152 Block selection when erasing (64KB block)

Selection block	Starting address	Write data after setting the LAT bit								
		D8	D7	D6	D5	D4	D3	D2	D1	D0
BLOCK # 0 (16KB)	Base_Address *	X	X	X	X	X	X	X	X	1
BLOCK # 1 (16KB)	Base_Address + $ 4000	X	X	X	X	X	X	X	1	X
BLOCK # 2 (16KB)	Base_Address + $ 8000	X	X	X	X	X	X	1	X	X
BLOCK # 3 (16KB)	Base_Address + $ C000	X	X	X	X	X	1	X	X	X
Shadow Registers		1	X	X	X	X	X	X	X	X

Note: * Base_Address is for flash arrays defined by HDFBAH and HDFBAL.
Means the start address.

LAT — Latch control

0 = Disable programming latch.
1 = Enable programming latch.

When LAT is cleared, the HD Flash EEPROM address and data bus is the IMB address bus. And connected to the data bus, the HD Flash EEPROM is set to normal read. LAT is set When is, the HD flash EEPROM address and data bus are connected to a parallel internal latch. The HD Flash EEPROM array is set to programming or erasing.

Immediately after setting LAT, write to a valid address in the HD flash EEPROM module. And both the write address and the data are latched. When set to programming (PE = 0), the latched data is programmed for the latched address. Set to erase If yes (PE = 1), finger if the latched address is a valid address in the HD Flash EEPROM. The specified block is erased. The erase block is specified by the data to be latched.

The LAT value cannot be changed when the program / erase voltage is applied (EHV = 1).

EHV — Programming / Erase Enable

0 = Disable program / erase voltage.
1 = Apply the program / erase voltage.

EHV is set only after LAT is set and data and writes to the address latch are made. I can't. If these conditions are not met, the EHV will remain cleared. EHV set The LAT, VPE, and PE bits cannot be changed while they are in the HD flash EEPROM module. Attempts to read the HD Flash EEPROM array location will be ignored.

HDF1BS [0: 3] — HD Flash EEPROM Bootstrap Word $ YFF790-$ YFF796
HDF2BS [0: 3] — HD Flash EEPROM Bootstrap Word $ YFF7D0-$ YFF7D6
HDF3BS [0: 3] — HD Flash EEPROM Bootstrap Word $ YFF810-$ YFF816

15 14 13 12 11 11 Ten 9 8 7 6 Five Four 3 2 1 0

BOOTSTRAP VECTOR

RESET:

PROGRAMMED VALUE

The data after clearing this register will be $ FFFF.

These words can be used as a system bootstrap vector. Reset When HDFMCR STOP = 0 and BOOT bit = 0, the HD Flash EEPROM module HDFBS [0: 3] responds to the reset vector from $ 000000 to $ 000006. This response is the first time The array will support the second and subsequent access to $ 000000 to $ 000006. STOP = 0 And when BOOT = 1, the array of HD flash EEPROM modules for the reset vector Will respond. HDFBS [0: 3] can be read at any time when STOP = 0 in HDFMCR, The value in the word cannot be changed without programming the appropriate location.

12.5 HD Flash EEPROM operation

Below are the modules for system bootstrap, normal operation, and array programming / disappearance. Describes the reset of the HD Flash EEPROM module used earlier.

12.5.1 Reset operation

The reset initializes the control registers of all HD flash EEPROMs. Bits in this register have a fixed default value or the associated HD Flash EEPROM shadow register. The programmed value is set in the data.

If the STOP shadow bit state is zero, the HDFMCR STOP bit is cleared during reset and is displayed. The joule responds to the range of access specified by HDFBAH and HDFBAL. BOOT bit cleared If so, the module will also respond to bootstrap vector access.

When the STOP shadow bit state is 1, the HDFMCR STOP bit is set at reset. The HD Flash EEPROM array is disabled. The module will be cleared until the STOP bit is cleared.

Does not respond to array or bootstrap vector access. This will result in an external device The chair has shadow bit access or boot to the HD Flash EEPROM array address space. You can respond to strap access. The erase state of the shadow bit is 1. Erase The module will be in STOP mode after the reset is released.

12.5.2 Bootstrap operation

CPU16 is the initial value of the internal register located at the addresses $ 000000 to $ 000006 in the program space. Starts the bootstrap operation by fetching. These are exception vector tables The address of the bootstrap vector in. HD when STOP = 0 and BOOT = 0 of HDFMCR at reset The flash EEPROM module is configured to respond to bootstrap vector access. vinegar. The vector allocation is as follows:

Table 153 Bootstrap mapping

EEPROM bootstrap

word	Vector address	Contents of the MCU reset vector
HDFBS0	$ 000000	Initial ZK, SK, and PK [1]
HDFBS1	$ 000002	Initial PC
HDFBS2	$ 000004	Reset-Initial SP
HDFBS3	$ 000006	Initial IZ

Note: 1. Bit [15:12] = Reserved, Bit [11: 8] = ZK initial value, Bit [7: 4] = SK initial value,
Bit [3: 0] = PK initial value

When the address $ 000006 is read, the HD flash EEPROM operation returns to normal operation and the module It will stop responding to bootstrap vector access.

12.5.3 Normal operation

The HD Flash EEPROM module has two system clocks in bytes or already word a. Execute access and respond to continuous IMB access to Align Long Word Access Or it can supply 4 system clocks for misaligned word operations. Miss India 3-bus cycle access for longwords is for CPUs capable of 32-bit processing.

The module checks the function code to see the address space access type. vinegar. Array access is defined by the ASPC state of FFEMCR. HD Flash EEPROM Mod When set to normal operation, the array only responds to read access. No write operation Will be seen.

12.5.4 Program / Erase operation

The unprogrammed HD Flash EEPROM bits are in the logical 1 state. Bit is shape The state must be programmed to change from 1 to 0. Erasing a bit returns it to the state of 1. I will. To program / erase the HD flash EEPROM, write to a set of control registers It is necessary to write to the programming latch. Use the same procedure to push the shadow register Loggram Only 1-byte or 1-alind words can be programmed at the same time.

Note: To program the array , a programming voltage must be applied to the V_{FPE} pins. I don't. The applied voltage must be within the range of $12.6 \geqq V_{FPE} \geqq (SiO - 0.35V)$. It will not be. Beyond this range will damage the HD Flash EEPROM module. There is a risk.

12.5.4.1 Programming sequence

To program the HD Flash EEPROM array, complete the following steps: Figure 44 shows the program It is a flowchart of the operation.

1. Raise the voltage applied to the V_{FPE} pin to the program / erase / verification level.

2. Clear the NPP counter, which is a parameter on the software, and clear the margin flag. vinegar. This NPP counter is used to check the number of programs (up to 50 times). margin· The flag is a flag used for the pros and cons of reprogramming the programmed data. vinegar.

3. Clear the PE bit in HDFCTL and set the LAT bit. This causes the address latch And the data latch can be programmed. Also VPE at the same time if necessary in this step You can set it to and start the program verification circuit.

4. Write the data to the address you want to program. This sets the module to the program Will be done.

5. Start writing to the HD flash by setting the EHV bit in HDFCTL. That Start the program pulse to check the write time.

6. Delay the time for one programming pulse to occur. The delay is specified by the parameter PWPP It will be.

7. Clear the EHV bit in HDFCTL. This will stop the program pulse and in the array Stops the high voltage supply to the unit.

8. Wait for the high voltage to the array to drop to the steady voltage. This waiting time (delay time) is a parameter Specified by TPR.

9. If the margin flag is not already set, increment the NPP counter and program location. Read the program and verify that it is programmed. If the VPE bit is set,

If programmed correctly, 0 will be read.

If programmed, set the margin flag and go to "Set EHV" in step 5.
to go.

If not programmed, the NPP counter value is the maximum number of programmed pulses NPPMAX (50 times)
If you haven't reached), go to step 5 "Set EHV".

If NPP = NPPMAX (50 times) is reached, the program will fail. Suspension procedure
To clear the LAT bit of HDFCTL and pass the program voltage applied to the V FPE pin.
Decrease to the normal read level (VDD level). Ends the program sequence.

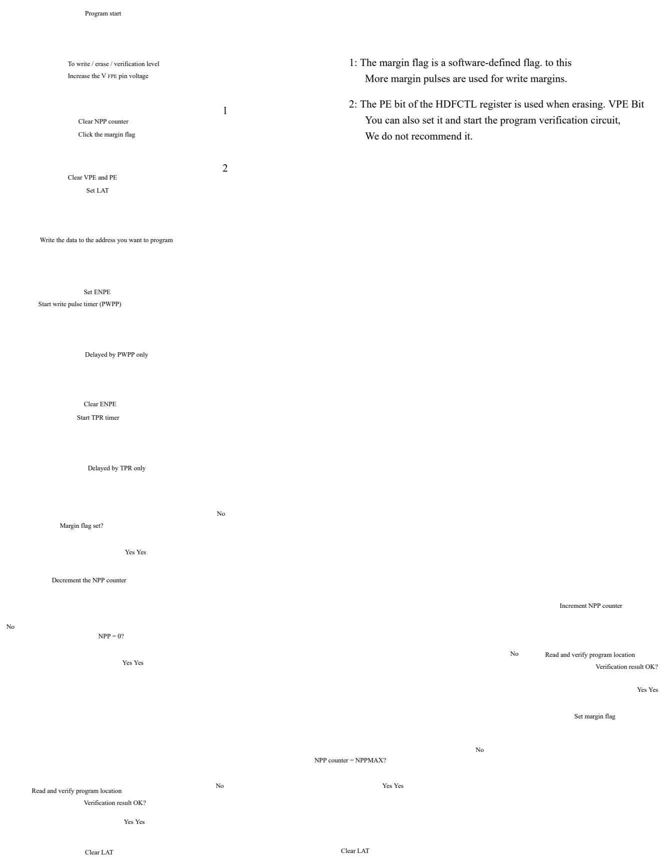
If the margin flag is set, decrement the NPP counter value by 1 until NPP = 0.
Repeat "Decrease NPP Counter" in this step from "Set EHV" in step 5. this
Gives 100% program margin.
Then read and validate the program location again.

If it is not programmed, the program will fail. HDFCTL for suspension procedure
Clears the LAT bit of and reads the program voltage applied to the V FPE pin at the normal read level.
Decrease to the control level. Ends the program sequence.

If programmed, clear the LAT bit in HDFCTL and write all in step 10.
Go to "End?"

10. If all writes have not been completed, then the address to program the next location.
Set the flow and repeat from "Clear NPP counter / Clear margin flag" of the flow.

If all the writing is completed, it means that the writing is completed normally. V FPE as a termination procedure
The voltage applied to the pin is lowered to the normal read level, and the process ends normally.



1: The margin flag is a software-defined flag. to this
More margin pulses are used for write margins.

2: The PE bit of the HDFCTL register is used when erasing. VPE Bit
You can also set it and start the program verification circuit,
We do not recommend it.

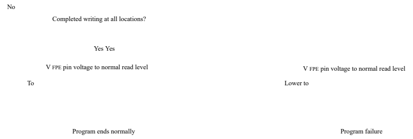

No
Completed writing at all locations?
Yes Yes
V FPE pin voltage to normal read level
To
V FPE pin voltage to normal read level
Lower to
Program ends normally
Program failure

Figure 44 HD Flash Program Flowchart (see Table 180)

12.5.4.2 Erase operation

To clear the HD flash EEPROM array, perform the following steps . Figure 45 shows the flow of erasing operation . It is a chart.

1. Raise the voltage applied to the V FPE pin to the program / erase / verification level.

2. Set the erase pulse counter NEP, which is a parameter on the software, to 1, and set the margin flag. Clear.

3. Clear the VPE bit in HDFCTL and set the PE and LAT bits. This allows FLASH The module is set to erase mode.

4. Write to any valid address in the array or control block (shadow) vinegar. Select the erase block according to the address to write.

5. Set the EHV bit of HDFCTL. This applies an erasing voltage to the array or shadow. increase.

6. Delay by the time corresponding to one erasure pulse. The delay is specified by the parameter TEPK.

7. Clear the EHV bit in HDFCTL. This causes the height of the array or shadow inside the array Stop the voltage supply.

8. Wait for the high voltage to the array or shadow to drop to steady-state voltage. This waiting time (during delay)) Is specified by the parameter TER.

9. If the margin flag is not already set, read the entire array and shadows and everything Verify that the location of is erased (since the HDFMCR shadow erase value is $ FFFF) be careful).

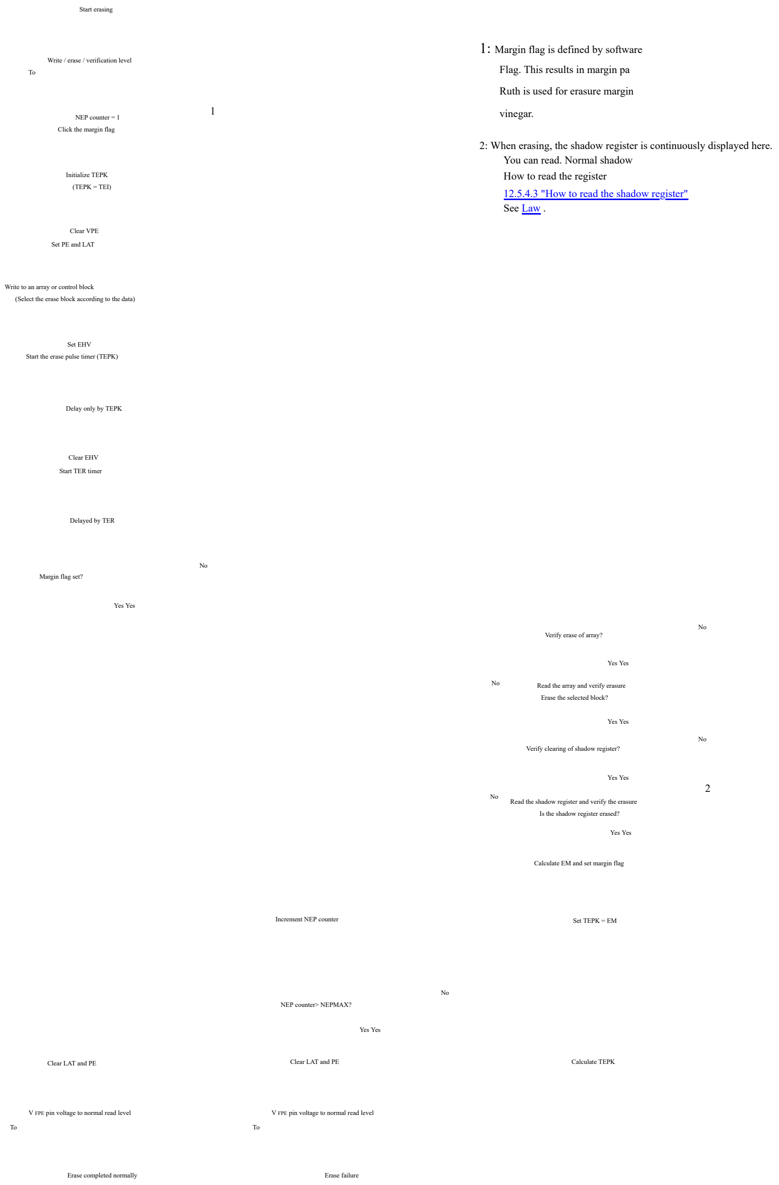
   If all are erased, calculate the erase margin EM, set the margin flag and step Go to 5 "Set EHV".

   If not all erased, increment the NEP counter by 1 and the NEP counter value is the maximum erase pulse. If the number NEPMAX (5 times) is not exceeded, calculate a new TEPK value and "set EHV" in step 5. I will go to.

   If the NEP counter value exceeds NEPMAX (5 times), the erase will fail. Suspension procedure Clear the LAT and PE bits of the HDFCTL and the erasing voltage applied to the V FPE pin. To the normal read level (VDD level).

   If the margin flag is set, clears the LAT and PE bits of HDFCTL.

10. Erase ends normally. Normal reading of the erasing voltage applied to the V FPE pin as a termination procedure Lower to the output level (VDD level).


Start erasing
Write / erase / verification level
To
1
NEP counter = 1
Click the margin flag
Initialize TEPK
(TEPK = TEI)
Clear VPE
Set PE and LAT
Write to an array or control block
(Select the erase block according to the data)
Set EHV
Start the erase pulse timer (TEPK)
Delay only by TEPK
Clear EHV
Start TER timer
Delayed by TER
No
Margin flag set?
Yes Yes
No
Verify erase of array?
Yes Yes
No
Read the array and verify erasure
Erase the selected block?
Yes Yes
No
Verify clearing of shadow register?
Yes Yes
2
No
Read the shadow register and verify the erasure
Is the shadow register erased?
Yes Yes
Calculate EM and set margin flag
Increment NEP counter
Set TEPK = EM
No
NEP counter> NEPMAX?
Yes Yes
Clear LAT and PE
Clear LAT and PE
Calculate TEPK
V FPE pin voltage to normal read level
To
V FPE pin voltage to normal read level
To
Erase completed normally
Erase failure

1: Margin flag is defined by software
Flag. This results in margin pa
Ruth is used for erasure margin
vinegar.

2: When erasing, the shadow register is continuously displayed here.
You can read. Normal shadow
How to read the register
12.5.4.3 "How to read the shadow register"
See Law .

Figure 45 HD Flash Erase Flowchart (see Table 180)

12.5.4.3 How to read the shadow register

Flash EEPROM module for MC68HC916 series (HD FLASH, FLASH, BE FLASH and TPU FLASH)
When it is necessary to read the data in the shadow register of the above during normal times (other than during programming or erasing).
If so, follow the steps below.

1. Clears all bits of the control register (FEECTL, HDFCTL, TFCTL, BEFCTL).
2. Set only the LAT bit of the control register.
3. Writes to the address of the shadow register to be read. Any data to write
   I do not mind. This operation latches the address and data.

Four. Reads the shadow location where the write was made. The read data is the shadow register
It is the data of the star.

Five. Clears the LAT bit. As a result, the flash EEPROM module returns to normal operation.
Allows access to the ray area and control blocks.

6. Repeat steps 2 through 5 if there are more shadow registers you want to read.

Shadow read start

All of the control registers
Clear a bit 1

Set only LAT

To the shadow address
Write data

Lead shadow location

Clear LAT 2

Other shadows
Lead?

YES YES

NO

Shadow lead end

Figure 46 How to read the shadow register

Note: 1. The flash module must be enabled (the STOP bit is cleared).
Is required.

2. The address of the shadow written immediately after setting LAT is the program latch times.
Retained on the road, subsequent read access to the flash will be on the latched shadow
Read as data. Therefore, to read the next shadow, click LAT once.
Then, it is necessary to set LAT again and write the address of the shadow to be read.
I will.

12.5.5 HD Flash EEPROM Write / Erase Voltage Signal

conditions

The minimum VDD - 0.35V voltage should be applied to the V FPE pin at all times. This condition is met
Otherwise, you may damage the FLASH module. The FLASH module is available until the V FPE is powered on.
It can also be damaged by excessive response during power off. VDD is lower than the specified minimum voltage
In the meantime , the V FPE cannot be raised to the write level. Further, while the VDD is supplied, V FPE the
It cannot be lowered below the specified minimum voltage. Figure 47 shows the operating waveforms for V FPE and VDD.

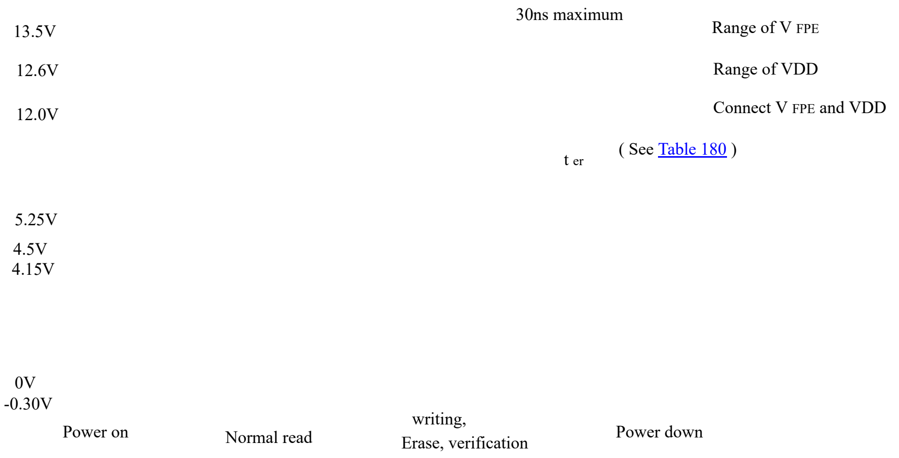
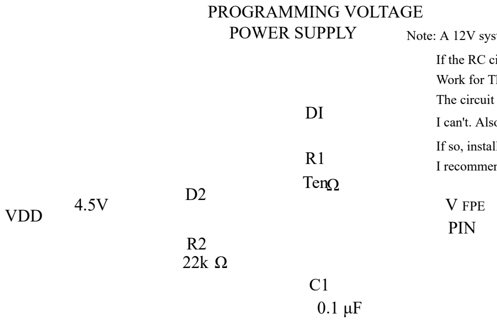


Figure 47 Write voltage waveform

Conditioning the V FPE with an external circuit is recommended. Figure 48 shows that the required voltage is maintained.
Here is a simple circuit that filters transient signals. V FPE is connected to VDD through Schottky diode D2
It will be up. Applying a programming voltage through diode D1 reverse biases D2 and
Protects VDD from excessive reverse current. D2 also FLASH in the unlikely event that the programming voltage goes to zero
Protects against damage. Adjust the programming power supply voltage to compensate for the forward bias drop at D1.
I have to make up for it. The transient signal is filtered by the charging time constants of R1 and C1, and the discharge path of C1 is defined by R2.
Offers.

This RC charge / discharge time constant can be effectively used when the power is turned on / off. When using this circuit
Low leakage current from external devices connected to V FPE pins to reduce diode voltage drop
Please hold it down.



Note: A 12V system voltage is applied to the V FPE pin.
If the RC circuit (R1 and C1) is noise reduction
Work for Therefore, this RC
The circuit must be located on the V FPE pin side
I can't. Also, there are multiple V FPE pins
If so, install the RC circuit individually.
I recommend it.

Figure 48 V FPE conditioning circuit

Chapter 13
TPU flash EEPROM

The TPU flash memory (block erasable flash EEPROM) module is a 4K byte non-volatile
It operates as a high-speed access memory. Also programmable non-volatile microcode to TPU2

It can also be used as a storage space. This module serves as a TPU2 microcode development environment (TPU mode).
A flash module that can be erased in 1K block units. TPU2 to CPU program array (IMB)
When used as a mode), the TPU flash module emulates TPU microcode storage.
It cannot be used as a flash EEPROM (BEFLASH) that can be erased in general block units.
I can do it.

When in TPU mode, the CPU cannot write or erase data in the array. Control register
Ta can not be written only with leads. To enter TPU mode, TPU2 emulation
Set the mode bit or the TPU mode shadow bit in the TFMCR register of this module.
Is possible by programming to 1.

If it is not in TPU mode, it is fast and real, such as the operating system kernel and standard subroutines.
It can be used for plum code that needs to be lined or for code that runs frequently.

The TPU flash EEPROM can be used for frequently read static data or system recycle.
It can also be set to provide a bootstrap vector for the set.

13.1 Overview

The TPU Flash EEPROM (TPUFLASH EEPROM) module occupies a fixed position in the MCU address space.
Consists of a 32-byte control register block and a 4K-byte TPUFLASH EEPROM array
is. The 4K byte TPUFLASH EEPROM array can be mapped to any even 4K byte boundary. these
Arrays can reside in both program space and data space, or only in program space
You can do it.

The TPU flash EEPROM array can be read as bytes, words, or long words.
to come. The module responds to IMB access in succession and has a two-bus service to the Arind Long Word.
Enables access with an Icle (4 system clocks).

Software for both individual control register bits and arrays with shadow bits
It can be programmed and erased by control. Program / erase voltage is supplied through an external V FPE pin
Must be. Programming is done only in bytes or already words. child
Module supports bulk erase and block erase modes with minimum program / erase life
Is 100 cycles. Hardware interlock is programmed into the TPU flash EEPROM array
/ Protects stored data from corruption if erase voltage is accidentally enabled
vinegar.

The TPUFLASH array is enabled by a combination of DATA12 and the reset shadow bit.
/ Disabled. However, when single chip mode is selected, the logic of the STOP shadow bit
Only controls the enable / disable of the TPUFLASH array.

Since this MCU consists of multiple flash EEPROM modules, it also uses multiple reset vectors.
As you might expect, the source of the reset vector is 1 because it prohibits duplication of mappings.
Must be one.

When programming / erasing the TPU Flash EEPROM module, V FPE 1 pin is used, and other functions.
Note that the V FPE 2 pin is used for the rush EEPROM module .

Page 294

Table 154 TPU Flash EEPROM Register Block Address Map

ADDRESS	15 8 7 0
$ YFF860	TPU FLASH EEPROM MODULE CONFIGURATION REGISTER (TFMCR)
$ YFF862	TPU FLASH EEPROM TEST REGISTER (TFTST)
$ YFF864	TPU FLASH EEPROM BASE ADDRESS HIGH REGISTER (TFBAH)
$ YFF866	TPU FLASH EEPROM BASE ADDRESS LOW REGISTER (TFBAL)
$ YFF868	FLASH EEPROM CONTROL REGISTER (TFCTL)
$ YFF86A	RESERVED
$ YFF86C	RESERVED
$ YFF86E	RESERVED
$ YFF870	TPU FLASH EEPROM BOOTSTRAP WORD 0 (TFBS0)
$ YFF872	TPU FLASH EEPROM BOOTSTRAP WORD 1 (TFBS1)
$ YFF874	TPU FLASH EEPROM BOOTSTRAP WORD 2 (TFBS2)
$ YFF876	TPU FLASH EEPROM BOOTSTRAP WORD 3 (TFBS3)
$ YFF878	RESERVED
$ YFF87A	RESERVED
$ YFF87C	RESERVED
$ YFF87E	RESERVED

Y = M111. However, M indicates the logical state of the SLIMCR mode map (MM) bits.

13.2 TPU Flash EEPROM Control Block

Module configuration register in the TPU flash EEPROM control block
Star (TFMCR), test register (TFTST), two array-based address registers (TFBAH)
It contains 5 registers (TFBAL) and control register (TFCTL). Also, the controller
The four additional words in Le Block use the TPU flash EEPROM as bootstrap memory.
It can store bootstrap information when you use it.

The registers in the control block are physically located in the spare TPU flash EEPROM line.
Some have associated shadow registers. During the reset, the fields in the register
The default reset information is loaded from the shadow register. Shadow register is supported
The location of the TPU flash EEPROM array and the location of the control register
It will be programmed or erased in the same way. Even if it is programmed in the shadow register, the corresponding controller
No data is written to the trawl register. New data will be controlled until the next reset
It is not copied to the register. The contents of the shadow register are the TPU flash EEPROM array.
It will be erased each time it is erased.

Configuration information from the control block of the TPU flash EEPROM is separate from the array
It is placed in pieces and controls the operation of the TPU flash EEPROM by writing. Writable after reset
You can change the registers in the control block that have bits. In these registers
Writing does not affect the associated shadow register. Some registers are TFMCR LOCK
Only when the bit is disabled or when the TFMCR STOP bit is set
I can't write. These limits are described in the individual register description.

13.3 TPU Flash EEPROM Array

The base address register specifies the base address of the TPU flash EEPROM array.
Write the default reset base address to the base address shadow register
Can be done. Array-based addresses must be on 4K byte boundaries for 4K byte arrays
I don't. ADDR [23:20] is driven to the same logical state as ADDR19, so CPU16 is from $ 080000 to $ F7FFFF.
You cannot access addresses in the range of. TPU flash EEPROM array is mapped to these addresses
If so, this array cannot be accessed without remapping the TPU flash.

Page 295

Module dedicated array overlaps with registers other than its own control register
Do not use a strong base address value. Part of the array overlaps its own register block
If you do, these registers are accessible, but you do not have access to the overlapping parts of the array.
Will be seen. However, if the array overlaps the control blocks of other modules,
If you read the overlapping registers, the result will be undefined.

13.4 TPU Flash EEPROM Register

In the register diagram below, the notation SB, which indicates the reset state, is when each control bit is reset.
Indicates that the value of the shadow bit associated with is taken.

TFMCR — TPU Flash EEPROM Module Configuration Register $ YFF860

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP	FRZ	0	BOOT	LOCK	0	ASPC		0	0	0	TME	TPU	BUSY	0	0
RESET:															
SB	0	0	SB	SB	0	SB	SB	0	0	0	SB	0	0	0	0

The data after clearing this shadow register will be $ BFFF.

STOP — Stop

0 = Normal TPU flash EEPROM array operation.
1 = Disables the TPU flash EEPROM.

The reset state of STOP is the inverted value of the DATA12 logical state being reset and the STOP shadow bit.
It is a logical sum. If STOP is not set, the TPU Flash EEPROM Array Base
You cannot change the address.

FRZ — Freeze mode control

0 = Disables the program / erase voltage while asserting the FREEZE signal.
1 = Enables program / erase voltage to be applied with ENPE bits while asserting the FREEZE signal.

BOOT — boot ROM control

0 = After reset, CPU16 goes to the bootstrap word address of the TPU flash EEPROM.
to access.
1 = After reset, CPU16 goes to the bootstrap word address of the TPU flash EEPROM.
I can't access it.

The reset state of BOOT is specified by the user. If STOP = 1, the bootstrap function is disabled.

LOCK — lock register

0 = Write lock disabled. Write to protected registers and fields
Can be done.

1 = Write lock enable. Writing to protected registers and fields
Can't.

The reset state of LOCK is specified by the user. If the default reset state for LOCK is zero, It can be set only once after a master reset so that the registers can be protected after initial setup. to come. LOCK also protects ASPC and WAIT fields, TFBAL and TFBAH registers. ASPC, TFBAL and TFBAH are also protected by the STOP bit.

ASPC — TPU Flash EEPROM Array Space

Since the MCU only operates in supervisor mode, access is programmatically empty in the ASPC field. Decide whether it is limited to the interval or is done in both the program space and the data space. vinegar.

The reset state of ASPC is user-specified.

Table 155 ASPC encoding

ASPC [1: 0]	Specified state
X0	Program and data access
X1	Program access only

TME — TPU mode enable shadow bit

0 = Automatically set the EMU bit of the TPU2MCR after the TPU flash EEPROM has reset.
Read / write from the CPU to the TPU flash is disabled.

1 = TPU flash EEPROM works normally. TPU mode when the EMU bit of TPU2MCR is set
to go into.

When TME is programmed to zero, the TPU flash EEPROM function is similar to TPUROM. This bit is not the bit in the TFMCR register, but the bit in the shadow register that corresponds to the TFMCR. vinegar. This bit cannot be read in the normal way. To read this bit, the user has a shadow register You must follow the same procedure as when reading the data.

TPU — TPU status bit

0 = TPU flash EEPROM module is in IMB mode.

1 = TPU flash EEPROM module is in TPU mode.

BUSY — BUSY status bit

This bit allows the user to TPU flash when the TPU flash EEPROM module is in TPU mode. This is a warning flag when you try to program in, and you should not perform such an operation. To indicate this, the TPU flash module sets this bit. This will result in the TPU You say that the rush module is "BUSY" while supplying microcode to TPU2 Shown in The.

0 = Is TPU Flash EEPROM accessible to microcode? Required for microcode access
Unnecessary state.

1 = TPU flash EEPROM is in TPU mode and cannot be used. This is the TFCTL register
Occurs when the LAT bit is set and TPU2 is requesting data from the TPU flash.

TFTST — TPU Flash EEPROM Test Register $ YFF862

This register is for factory testing only.

TFBAH — TPU Flash EEPROM Base Address Higher Register $ YFF864

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								ADDR twenty three	ADDR twenty two	ADDR twenty one	ADDR 20	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16
RESET:															
								SB	SB	SB	SB	SB	SB	SB	SB

The data after clearing this shadow register will be $ FFFF.

The TFBAH has the upper 16 bits of the TPU flash EEPROM array base address. Ri At set time, TFBAH automatically copies the data written to the associated shadow register. It will be. The default value for TFBAH is always the data in the associated shadow register. After reset If LOCK = 0 and STOP = 1, the software writes to TFBAH and TFBAL and the TPU flag. The EEPROM array can be rearranged. ADDR [23:20] is dry to the same logical state as ADDR19 CPU16 will not be able to access addresses in the range $ 080000 to $ F7FFFF. TPU fluff If the EEPROM array is mapped to these addresses, do not reset the system And you cannot access the array.

TFBAL — TPU Flash EEPROM Base Address Low Register $ YFF866

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR 15	ADDR 14	ADDR 13	ADDR 12	0	0	0	0	0	0	0	0	0	0	0	0
RESET:															
SB	SB	SB	SB	0	0	0	0	0	0	0	0	0	0	0	0

The data after clearing this shadow register will be $ FFFF.

TFBAL has the lower 16 bits of the TPU flash EEPROM array base address. Ri

At set time, TFBAL automatically copies the data written to the associated shadow register. It will be. The default value for TFBAL is always the data in the associated shadow register.

TFCTL — TPU Flash EEPROM Control Register $ YFF868

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	VFPE	ERAS	LAT	ENPE
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TFCTL contains the bits needed to control the programming and erasure of the TPU flash EEPROM. increase. This register can only be accessed in supervisor mode.

VFPE — Program / Erase Verification

0 = normal read cycle
1 = Start the program verification circuit.

This bit activates a special program verification circuit. Programming sequence (ERAS = During 0), VFPE, along with the LAT bit, determines that programming for a location is complete. Used to do. A valid TPU flash if both VFPE and LAT are set When reading the EEPROM location, the latched data and the data of the location in the program Exclusive OR is taken bitwise with. The location is fully programmed If so, a zero value is read. If a non-zero value is read, the location is complete Indicates that it is not fully programmed.

Normal read of valid TPU flash EEPROM location when VFPE is cleared Will be done.

ERAS — Erase control

0 = Set the TPU flash EEPROM to programming.
1 = Set the TPU flash EEPROM to erase.

When ERAS is asserted, one array block (block erase) of the selected array or all locations of the array The effect and shadow register (batch erase) are set to erase.

When the LAT bit is set, the array's address is latched regardless of the ERAS value. If so, any access to any array will return data for that latched address. Access to any array or shadow if the shadow register address is latched Even so, you can read the data of the latched shadow.

The ERAS value cannot be changed when the program / erase voltage is applied (ENPE = 1).

LAT — Latch control

0 = Disable programming latch.
1 = Enable programming latch.

When LAT is cleared, the TPU flash EEPROM address and data bus is the IMB address bus. And connected to the data bus, the TPU flash EEPROM is set to normal read. LAT is set When is, the TPU flash EEPROM address and data bus are connected to a parallel internal latch. The TPU flash EEPROM array is set to programming or erasing.

Immediately after setting LAT, write to a valid address in the TPU flash EEPROM module. Both the write address and the data will be latched. If set to programming (ERAS = 0), the latched data is programmed for the latched address. Set for erasure If set (ERAS = 1), the latched address is a valid address for the TPU flash EEPROM. If so, batch erase or block erase is performed. (The data to be latched is arbitrary.)

The value of LAT cannot be changed when the program / erase voltage is applied (ENPE = 1).

ENPE — Programming / Erase Enable

0 = Disable program / erase voltage.
1 = Apply the program / erase voltage.

ENPE is set only after LAT is set and data and writes to the address latch have been made. I can't. If these conditions are not met, ENPE remains cleared. ENPE is set The LAT, VFPE, and ERAS bits cannot be changed while they are in the TPU Flash EEPROM module. Any attempt to read the TPU Flash EEPROM array location will be ignored.

TFBS [0: 3] — TPU Flash EEPROM Bootstrap Word $ YFF870-$ YFF876

	15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
	BOOTSTRAP VECTOR															
RESET:																
	PROGRAMMED VALUE															

The data after clearing this register will be $ FFFF.

These words can be used as a system bootstrap vector. Reset
When the BOOT bit of TFMCR is 0 at the time of operation, the TFBS [0: 3] of the TPU flash EEPROM module is displayed.
Responds to set vectors from $ 000000 to $ 000006. This response is only the first access
The array supports access to $ 000000 to $ 000006 from the second time onward. When BOOT = 1,
The array of TPU flash EEPROM modules responds to the set vector. TFBS [0: 3] is
It can be read at any time when STOP = 0 in TFMCR, but the value in the word is the appropriate location.
It cannot be changed without programming.

13.5 TPU Flash EEPROM operation

Below are the modules for system bootstrap, normal operation, and array programming /
Describes resetting the TPU flash EEPROM module used for erasing.

13.5.1 Reset operation

The reset initializes the control registers of all TPU flash EEPROMs.
Bits in this register have a fixed default value or the associated TPU flash EEPROM shadow register.
The programmed value is set in the data.

If the STOP shadow bit state is zero, the TFMCR STOP bit is cleared during reset and is displayed.
The joule responds to the range of accesses specified by TFBAH and TFBAL. The BOOT bit is cleared
The module will then also respond to bootstrap vector access.

When the STOP shadow bit state is 1, the TFMCR STOP bit is set at reset and the TPU is set.
The flash EEPROM array is disabled. The module will continue until the STOP bit is cleared.
Does not respond to array or bootstrap vector access. This will result in an external device
For access to the TPU flash EEPROM array address space or bootstrap access
You can respond. The erase state of the shadow bit is 1. Erased modules are reset
After the switch is released, it will be in STOP mode.

13.5.2 Bootstrap operation

CPU16 is the initial value of the internal register located at the addresses $ 000000 to $ 000006 in the program space.
Starts the bootstrap operation by fetching. These are exception vector tables
The address of the bootstrap vector in. When TFMCR STOP = 0 and BOOT = 0 at reset, TPU
The flash EEPROM module is configured to respond to bootstrap vector access.
vinegar. The vector allocation is as follows:

Table 156 Bootstrap mapping

EEPROM bootstrap word	Vector address	Contents of the MCU reset vector
TFBS0	$ 000000	Initial ZK, SK, and PK 1
TFBS1	$ 000002	Initial PC
TFBS2	$ 000004	Reset-Initial SP
TFBS3	$ 000006	Initial IZ

Note: 1. Bit [15:12] = Reserved, Bit [11: 8] = ZK initial value, Bit [7: 4] = SK initial value,
Bit [3: 0] = PK initial value

When the address $ 000006 is read, the TPU flash EEPROM operation returns to normal operation and the module

Will stop responding to bootstrap vector access.

13.5.3 Normal operation

The TPU flash EEPROM module has two system clocks in bytes or already word. Execute access and align long word access in response to continuous IMB access Or you can supply 4 system clocks for misaligned word operations. Miss India 3-bus cycle access for longwords is for CPUs capable of 32-bit processing.

The module checks the function code to see the address space access type. vinegar. Array access is defined by the ASPC state of FFEMCR. TPU Flash EEPROM Mod When set to normal operation, the array only responds to read access. No write operation Will be seen.

13.5.4 TPU mode operation

TPU flash provides microcode data to TPU2 by setting a bit of TPU2. You can pay. It also automatically enters TPU mode after being reset by the shadow bit (TME). It is also possible.

13.5.5 Program / Erase operation

The unprogrammed TPU flash EEPROM bits are in the logical 1 state. Bit is shape The state must be programmed to change from 1 to 0. Erasing a bit returns it to the state of 1. I will. To program / erase the TPU flash EEPROM, write to a set of control registers. Requires entry and writing to the programming latch. Use the same procedure to create a shadow register Program Only 1 byte or 1 aligned word can be programmed at the same time vinegar. Erasing is done in bulk or in blocks. In block erase mode, the selected block in the array Only the lock is erased. In bulk erase mode, the entire TPU flash EEPROM array and shadows The gista bits are erased at the same time. Completely erase the TPU flash EEPROM before the final data value Must be programmed.

Note: To program the array , a programming voltage must be applied to the V_{FPE} pins. I don't. The applied voltage must be within the range of $12.6 \geqq V_{FPE} \geqq$ (SiO − 0.35V). It will not be. Beyond this range will damage the TPU Flash EEPROM module. There is a risk of

To specify batch erase or block erase at the time of erase, refer to "TPU Flash Block Configuration" in Table 157 below. please refer to.

Table 157 TPU flash block configuration

Erase mode	A23-A9	A8-A6	A5	A4	A3	A2	A1	A0	block
Block erase	base· dress	X	1	0	0	0	X	X	Block 0 (1KByte)
		X	1	0	0	1	X	X	Block 1 (1KByte)
		X	1	0	1	0	X	X	Block 2 (1KByte)
		X	1	0	1	1	X	X	Block 3 (1KByte)
Batch erase		X	0	X	X	X	X	X	With all the blocks Shadow register

The block number divides the 4k bytes of the array into 1k bytes, and blocks 0,1,2,3 from the lower address side. Is assigned to.

13.5.5.1 Programming sequence

To program the TPU Flash EEPROM array, complete the following steps: Figure 49 shows the program It is a flowchart of the mining operation.

1. Raise the voltage applied to the V_{FPE} pin to the program / erase / verification level.

2. Clear the NPP counter, which is a parameter on the software, and clear the margin flag. vinegar. This NPP counter is used to check the number of programs (up to 50 times). margin· The flag is a flag used for the pros and cons of reprogramming the programmed data. vinegar.

3. Clear the ERAS bit in TFCTL and set the LAT bit. This causes the address latch

And the data latch can be programmed. Also VFPE if necessary in this step
You can set it at times and start the program verification circuit.

4. Write the data to the address you want to program. This sets the module to the program
Will be done.

5. Start writing to the TPU flash by setting the ENPE bit in TFCTL. That
Start the program pulse to check the write time.

6. Delay the time for one programming pulse to occur. The delay is specified by the parameter PWPP
It will be.

7. Clear the ENPE bit in TFCTL. This will stop the program pulse and in the array
Stops the high voltage supply to the unit.

8. Wait for the high voltage to the array to drop to the steady voltage. This waiting time (delay time) is a parameter
Specified by TPR.

9. If the margin flag is not already set, increment the NPP counter and program location.
Read the program and verify that it is programmed. If the VFPE bit is set,
If programmed correctly, 0 will be read.

If programmed, set the margin flag and go to "Set ENPE" in step 5.
to go.

If not programmed, the NPP counter value is the maximum number of programmed pulses NPPMAX (50 times)
If you haven't reached), go to step 5 "Set ENPE".

If NPP = NPPMAX (50 times) is reached, the program will fail. Suspension procedure
To clear the LAT bit of the TFCTL, the program voltage applied to the V FPE pin is normally applied.
Decrease to the read level (VDD level) of. Ends the program sequence.

If the margin flag is set, decrement the NPP counter value by 1 until NPP = 0.
Repeat "Decrease NPP Counter" in this step from "Set ENPE" in step 5. this
Gives 100% program margin.

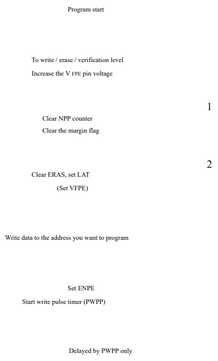
Then read and validate the program location again.

If it is not programmed, the program will fail. TFCTL for suspension procedure
Clears the LAT bit of and reads the program voltage applied to the V FPE pin at the normal read level.
Decrease to the control level. Ends the program sequence.

If programmed, clear the LAT bit in TFCTL and write all in step 10.
Go to "End?"

10. If all writes have not been completed, then the address to program the next location.
Set the flow and repeat from "Clear NPP counter / Clear margin flag" of the flow.

If all the writing is completed, it means that the writing is completed normally. V FPE as a termination procedure
The voltage applied to the pin is lowered to the normal read level, and the process ends normally.


Program start
To write / erase / verification level
Increase the V FPE pin voltage
1
Clear NPP counter
Clear the margin flag
2
Clear ERAS, set LAT
(Set VFPE)
Write data to the address you want to program
Set ENPE
Start write pulse timer (PWPP)
Delayed by PWPP only

1: The margin flag is a software-defined flag. to this
More margin pulses are used for write margins.

2: The ERAS bit in the FEECTL register is used when erasing. VFPE
You can set the bit and start the program verification circuit.

3: If the VFPE bit is used, the VFPE bit is also cleared.

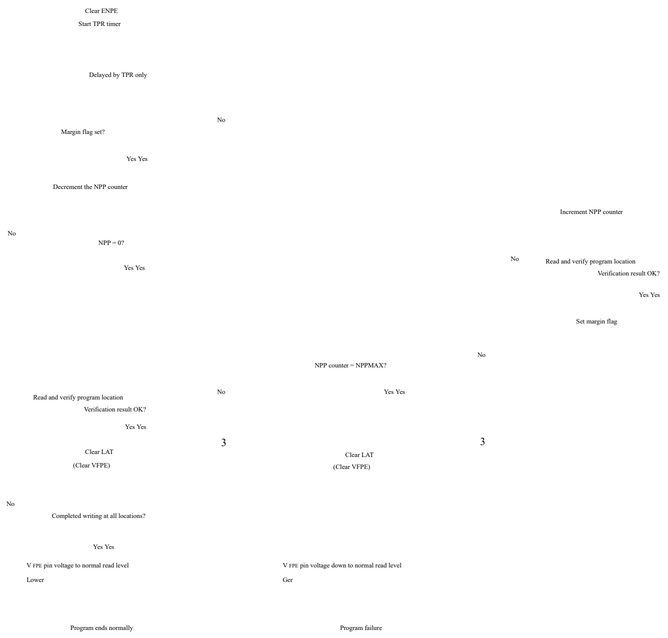


Figure 49 Flowchart of TPU Flash Program (see Table 112)

13.5.5.2 Erase operation

To erase the TPU Flash EEPROM array, complete the following steps: Figure 50 shows the flow of erasing operation . It is a chart.

1. Raise the voltage applied to the V FPE pin to the program / erase / verification level.

2. Set the erasure pulse counter NEP, which is a parameter on the software, to 1, and set the margin flag. Clear.

3. Clear the VFPE bit in TFCTL and set the ERAS and LAT bits. By this, The joule is set to erase mode.

4. Write to the valid address of the array. It doesn't matter what data you write. Add to write Les decides which block to erase.

5. Set the ENPE bit in TFCTL. This applies an erasing voltage to the array and shadow. increase.

6. Delay by the time corresponding to one erasure pulse. The delay is specified by the parameter TEPK.

7. Clear the ENPE bit in TFCTL. This causes the array and shadow to rise inside the array. Stop the voltage supply.

8. Wait for the high voltage to the array and shadow to drop to steady-state voltage. This waiting time (during delay)) Is specified by the parameter TER.

9. Clear the ERAS and LAT bits.

10. If the margin flag is not already set, erase selected block (block erase) or Reads the entire array (batch erase) and verifies that all locations have been erased vinegar. In the case of batch erasure, shadow verification is also performed. For details, see Figure 51 “Reading the shadow register” . How to do it. Note that the TFMCR shadow erase value is $ BFFF.

    If all the locations are erased, the erase margin EM is calculated and the margin flag is set. Set and go to step 3, "Set of ERAS and LAT bits".

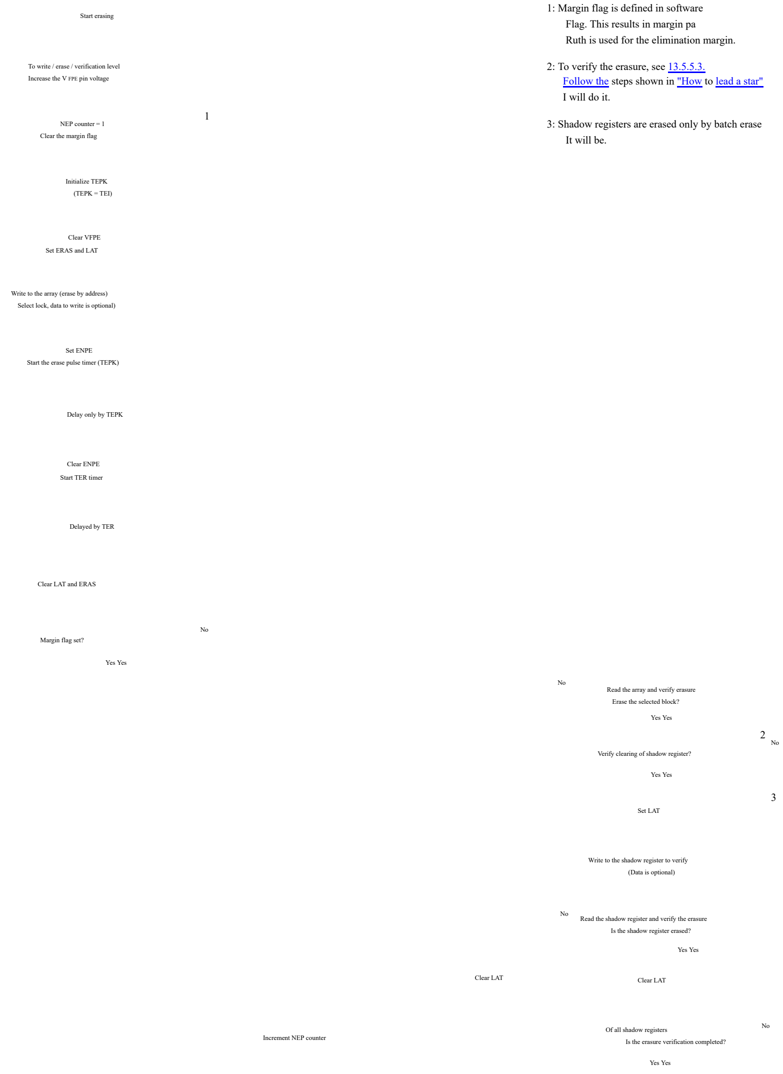
If not all erased, increment the NEP counter by 1 and the NEP counter value is the maximum erase pulse.
If the number NEPMAX (5 times) is not exceeded, calculate a new TEPK value and use the "ERAS bit in step 3".
Go to the LAT bit set ".

If the NEP counter value exceeds NEPMAX (5 times), the erase will fail. Suspension procedure
Clears the LAT and ERAS bits of the TFCTL and the erasing voltage applied to the V FPE pin.
To the normal read level (VDD level).

If the margin flag is set, go to step 11 for successful completion.

11. The erasure ends normally. Normal reading of the erasing voltage applied to the V FPE pin as a termination procedure
Lower to the output level (VDD level).

Page 303


Start erasing
To write / erase / verification level
Increase the V FPE pin voltage
1
NEP counter = 1
Clear the margin flag
Initialize TEPK
(TEPK = TEI)
Clear VFPE
Set ERAS and LAT
Write to the array (erase by address)
Select lock, data to write is optional)
Set ENPE
Start the erase pulse timer (TEPK)
Delay only by TEPK
Clear ENPE
Start TER timer
Delayed by TER
Clear LAT and ERAS
No
Margin flag set?
Yes Yes
No
Read the array and verify erasure
Erase the selected block?
Yes Yes
2
No
Verify clearing of shadow register?
Yes Yes
3
Set LAT
Write to the shadow register to verify
(Data is optional)
No
Read the shadow register and verify the erasure
Is the shadow register erased?
Yes Yes
Clear LAT
Clear LAT
Increment NEP counter
Of all shadow registers
Is the erasure verification completed?
No
Yes Yes

1: Margin flag is defined in software
Flag. This results in margin pa
Ruth is used for the elimination margin.

2: To verify the erasure, see 13.5.5.3.
Follow the steps shown in "How to lead a star"
I will do it.

3: Shadow registers are erased only by batch erase
It will be.

No

NEP counter> NEPMAX?

Calculate EM and set margin flag

Yes Yes

V FPE pin voltage to normal read level
Lower

V FPE pin voltage to normal read level
Lower

Calculate TEPK

Set TEPK = EM

Erase completed normally

Erase failure

Figure 50 TPU Flash Erase Flowchart (see Table 112)

13.5.5.3 How to read the shadow register

Flash EEPROM module for MC68HC916 series (HD FLASH, FLASH, BE FLASH and TPU FLASH)
When it is necessary to read the data in the shadow register of the above during normal times (other than during programming or erasing).
If so, follow the steps below.

1. Clears all bits of the control register (FEECTL, HDFCTL, TFCTL, BEFCTL).
2. Set only the LAT bit of the control register.
3. Writes to the address of the shadow register to be read. Any data to write
   I do not mind. This operation latches the address and data.

Four. Reads the shadow location where the write was made. The read data is the shadow register
   It is the data of the star.

Five. Clears the LAT bit. As a result, the flash EEPROM module returns to normal operation.
   Allows access to the ray area and control blocks.

6. Repeat steps 2 through 5 if there are more shadow registers you want to read.

Shadow read start

All of the control registers
Clear a bit 1

Set only LAT

To the shadow address
Write data

Lead shadow location

Clear LAT 2

Other shadows
Lead?

YES YES

NO

Shadow lead end

Figure 51 How to read the shadow register

Note: 1. The flash module must be enabled (the STOP bit is cleared).
Is required.

2. The shadow address written immediately after setting LAT is sent to the program latch circuit.
Retained and subsequent read access to the flash is the data in the latched shadow
Read as. Therefore, to lead the next shadow, clear LAT once and then again.
You need to set the degree LAT to write the address of the shadow to read.

13.5.6 TPU Flash EEPROM Write / Erase Voltage Signal conditions

The minimum VDD − 0.35V voltage should be applied to the V FPE pin at all times. This condition is met
Otherwise, you may damage the FLASH module. The FLASH module is available until the V FPE is powered on.
It can also be damaged by excessive response during power off. VDD is lower than the specified minimum voltage
In the meantime , the V FPE cannot be raised to the write level. Further, while the VDD is supplied, V FPE the
It cannot be lowered below the specified minimum voltage. Figure 52 shows the operating waveforms for V FPE and VDD.

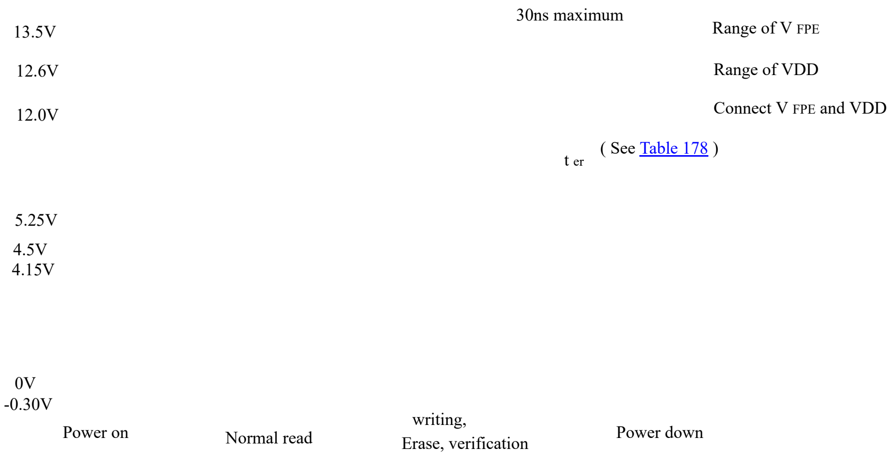
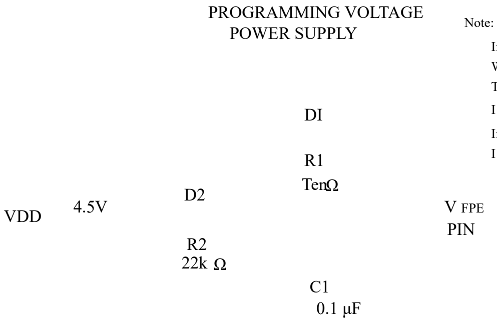


Figure 52 Write voltage waveform

Conditioning the V FPE with an external circuit is recommended. Figure 53 shows that the required voltage is maintained.
Here is a simple circuit that filters transient signals. V FPE is connected to VDD through Schottky diode D2
It will be up. Applying a programming voltage through diode D1 reverse biases D2 and
Protects VDD from excessive reverse current. D2 also FLASH in the unlikely event that the programming voltage goes to zero
Protects against damage. Adjust the programming power supply voltage to compensate for the forward bias drop at D1.
I have to make up for it. The transient signal is filtered by the charging time constants of R1 and C1, and the discharge path of C1 is defined by R2.
Offers.

This RC charge / discharge time constant can be effectively used when the power is turned on / off. When using this circuit
Low leakage current from external devices connected to V FPE pins to reduce diode voltage drop
Please hold it down.



Note: A 12V system voltage is applied to the V FPE pin.
If the RC circuit (R1 and C1) is noise reduction
Work for Therefore, this RC
The circuit must be located on the V FPE pin side
I can't. Also, there are multiple V FPE pins
If so, install the RC circuit individually.
I recommend it.

Figure 53 V FPE conditioning circuit

Chapter 14
Standby RAM module

The MC68HC (9) 16Y5 / 916Y6 is a 7K byte standby RAM module (4K byte standby SRAM and RSPI). The module's 3K byte standby SRAM) is installed. 4K byte SRAM module is system You can map to any 4K byte boundary in the memory map. Is this SRAM a V_{DD} during normal operation ? Will be powered. Power from standby power supply (V_{STBY}) even if V_{DD} disappears in standby mode It is supplied so you will not lose any data. Automatically V_{STBY} when V_{DD} drops below the specified level It will switch to. The SRAM on the 4K side has a V_{STBY} disappearance flag. This disappearance flag is only on 4KSRAM It exists, but it is the status of both 4KSRAM and RSPI 3KSRAM arrays.

14.1 Overview

SRAM is in normal mode when powered by the V_{DD} . This array is bytes, words, and Is accessible in longword. Byte access or already word (upper byte is Access (at even addresses) requires only one bus cycle or two system clocks. Hmm. Two bus cycles are required for long word or missed word access.

In standby mode, the RAM contents are preserved even if the V_{DD} runs out. SRAM contents V_{STBY} by Will be maintained. The circuit in the SRAM module retains the data and goes to the higher of V_{DD} or $V_{ST BY}$. It will switch. Access to the array is not guaranteed when the SRAM is powered by V_{STBY} .

In reset mode, the CPU completes the currently running bus cycle before the reset is performed. Can be If a synchronous reset occurs while a byte or word SRAM access is in progress, Access is complete. When a reset occurs during the first word access of a long word operation If so, only the first word access will be completed. Second word access for long word operations If a reset occurs in the meantime, the entire access is complete. Read from RAM or The data written to RAM can be corrupted in the event of an asynchronous reset.

Test mode works in conjunction with SLIM 's test capabilities. Test mode is MCU factory test Used in

Writing to the STOP bit of the SRAMMCR puts the SRAM module in stop mode. RAM array Is disabled and external logic can decode the SRAM address if desired, but the data is lost. All are retained. If the V_{DD} drops below V_{STBY} during stop mode , the internal circuitry will be on standby. Switch to V_{STBY} as in mode . Stop mode is canceled by clearing the STOP bit.

14.2 SRAM register block

The SRAM control register occupies an 8-byte block. This block contains SRAM Joule configuration register (SRAMMCR), SRAM test register (SRAMTST), and And four control registers in the RAM array-based address register (SRAMBAH, SRAMBAL) there is. The rest of the register block has unimplemented register locations.

Table 158 SRAM address map

ADDRESS	15	8	7	0
$ YFFB00	SRAM MODULE CONFIGURATION REGISTER (SRAMMCR)			
$ YFFB02	SRAM TEST REGISTER (SRAMTST)			
$ YFFB04	SRAM BASE ADDRESS REGISTER HIGH (SRAMBAH)			
$ YFFB06	SRAM BASE ADDRESS REGISTER LOW (SRAMBAL)			

14.3 SRAM registers

The SRAM registers are described below.

SRAMMCR — SRAM module configuration register $ YFFB00

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
STOP			PDS	RLCK		RASP1	RASP0				NOT USED				
RESET:															
1			U	0		1	1								

The bit of the module configuration register is that the RAM module consumes less power. Determines whether to go into top mode or normal mode and also indicates a standby RAM power failure

To do. In addition, it determines the address space in which the array resides. Whenever you read an unimplemented bit Zero is returned. Writing to unimplemented bits has no effect.

STOP — Stop control bit

0 = RAM array operates normally.

1 = RAM array enters low power stop mode.

This bit controls whether the RAM array goes into low power mode or operates normally. Ri
The set state is 1, and it is configured in the low power consumption mode. In stop mode, the contents of the RAM array
Is retained, but the CPU cannot read or write.

PDS - standby power status bit

0 = There is a loss of standby power supply.

1 = No loss of standby power supply.

The SRAM array has a standby power supply (V STBY) when the microcontroller V DD is turned off.
It can be powered from. PDS indicates that the voltage of V STBY is below the reference level. and
However, if the PDS changes from 1 to 0 in standby mode, it is considered that V STBY has lost power.
However, the data in the array at that time is not guaranteed. The PDS that became 0 due to the loss of power is
You cannot write to 1 in software. You can only write after a reset
increase.

The PDS is set in software after a reset, goes into standby mode, and then returns to its standby mode.
You will check the status of the bits.

This flag is only present on 4KSRAM, not on the 3KSRAM side of RSPI, but in standby state.
The confirmation of is also possible with the PDS flag of 4KSRAM.

RLCKS — RAM-based address lock

You can write to the SRAM base address register until RLCK is set. RLCK
Can only be set to IMB and will only be reset by a master reset.

0 = SRAM-based address register can be written from the IMB.

1 = Writes to the SRAM base address register are locked.

RASP [1: 0] — RAM array spatial field

X0 = RAM array is placed in the program / data space.

X1 = RAM array is placed in the program space.

These bits are micros that support user mode of operation and supervisor mode of operation separately.
Controls access to the controller's SRAM array. CPU16 is only in supervisor mode
RASP1 is invalid because it does not work.

Table 159 RASP encoding

RASP [1: 0]	Space
X0	Program / data space
X1	Program space

SRAMTST — SRAM test register $ YFFB02

SRAM TST is used for factory testing of SRAM modules.

SRAMBAH — SRAM base address, higher $ YFFB04

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
NOT USED								ADDR twenty three	ADDR twenty two	ADDR twenty one	ADDR 20	ADDR 19 19	ADDR 18 18	ADDR 17 17	ADDR 16 16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRAMBAL — SRAM base address, lower level $ YFFB06

15	14	13	12	11 11	Ten	9	8	7	6	Five	Four	3	2	1	0
ADDR 15	ADDR 14	ADDR 13	ADDR 12	NOT USED											
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The array-based address register is an array-based address in the system's memory map.
Specify the space. For SRAMBAH / SRAMBAL, the SRAM is in STOP mode (STOP = 1) and the lock is
It can only be written when the bit is not set (RCLK = 0). Once RLCK bit
Once set, writing to SRAMBAH and SRAMBAL is invalid, which causes
It prevents the array from being accidentally remapped. An array of 2K bytes of SRAM is called 2K bytes.

ADDR [10: 0] is fixed and cannot be written due to array boundary constraints.
Hmm. ADDR [11: 0] is rigid because a 4K byte SRAM array has an array boundary constraint of 4K bytes.
It is fixed and cannot be written.

Module arrays overlap with registers other than their own control registers
Do not use the base address value. Part of the array overruns its own register block
When wrapping, these registers are accessible, but access to the overlapping part of the array
Is ignored. However, the array overlaps the control blocks of other modules
If so, reading the overlapping registers will result in undefined results.

Page 309

Chapter 15
Boot ROM

15.1 Purpose of boot ROM code

The purpose of the SLIM boot ROM code is to provide an easy way to download and execute programs, new
Make it easy to test MC68HC (9) 16Y5 / 916Y6 and prototypes in new applications
is.

15.2 size

The size of the boot ROM is 512 bytes.
Currently, the boot ROM code starts at $ 0012 (overlapping any system interrupt vector).
I won't do it).

15.3 Special length

15.3.1 Initial settings

15.3.1.1 CPU register

The EK, X, Y, and Z registers are set to the following values by the boot ROM code:

- EK = $ F (SLIM, MCCI, and SRAM control registers
  Specify the address)
- XK, X = $ 000000 (address the base page and SRAM)
- ・YK, Y = $ 000000 (unused)
- ・ZK, Z = $ 000000 (Specify the address of SRAM)

The stack pointer is set to access the on-chip SRAM. This is SRAM enable
This means that interrupts that occurred before being interrupted may terminate abnormally. The space above the stack
Reserved for the following boot ROM code variables:

・Warmstart 3 bytes Warm ・ Jump destination address at reboot	
Baudrate Baud rate to reload on a 2-byte warm reboot	
・Buffer	Buffer used to read 1-byte S records
・Len	Variable used to read a 1-byte S record

15.3.1.2 On-chip SRAM

The boot ROM defaults to the MC68HC (9) 16Y5 / 916Y6 on-chip SRAM to the base address $ 001000.
To enable it. The boot ROM code assumes a SRAM size of 4K bytes.

15.3.1.3 SLIM MCR

The boot ROM does not enable show cycle (SHEN) and does not change the value of SLIM MCR.

15.3.1.4 SLIM SYNCR (VCO / PLL)

The boot ROM does not change the SLIM SYNCR register settings. After reset, the boot ROM code
Run the “autobaud” routine to automatically set the SCIB baud rate for MCCI (page 309).
" Refers to the communication with the AutoBaud / host"). However, the automatic baud rate setting function (auto baud function) is available.
It is affected by the system clock frequency set by SYNCR in two ways: That is, Sith
The system clock frequency is the maximum baud rate that can be matched exactly with the accuracy with which a particular baud rate is matched.
To decide.

15.3.1.5 port

The boot ROM does not change any SLIM port registers. The boot ROM has port F at startup.
The IRQ [6: 1] will be masked by the HC16 CPU because it is set to bell-sensitive interrupts.
Since IPQ7 is NMI, it is prohibited for MC68HC (9) 16Y5 / 916Y6 to exit the reset while keeping it “L”.
vinegar. At this time, the boot ROM cannot handle this situation (currently an interrupt handle is defined).
Is not).

15.3.1.6 SLIM SYPCR

The boot ROM does not change any SLIM system protection registers. Software watchdog
It is assumed to exit the reset in the disabled state. Boot ROM is a security mode
Do not lock or unlock. If MC68HC (9) 16Y5 / 916Y6 is locked, user
Downloads the program, unlocks security mode, and then reverts to the boot ROM code
Or (if possible) you need to reset the device without locking the security mode.

15.3.1.7 Chip Select

The boot ROM does not change any of SLIM's chip select registers.

15.3.2 Warm reboot

The boot ROM is the MCCI's PORTMC after setting the HC16 CPU's registers (EK, X, Y) and enabling SRAM.
Check the pin state of bit 6 of. This bit is used as a generic input pin.
If PORTMC [6] == 0 (indicates a warm start), the boot ROM is specified by the variable Warmstart.
Jump to location (assuming it is preset with the'S'and'G' commands
increase). If PORTMC [6] == 1 (indicates that it is not a warm start), the boot ROM will be autoboard.
Performs a function and waits for'S'and'G' commands from the host.

15.3.3 Autobo / Communication with host

After reset, the boot ROM is MC68HC (9) 16Y5 / 916Y6 MCCI SCIB without parity, 8 data bits, 1 Initializes to the stop bit. Then set the baud rate to maximum and set the TXDB and RXDB pins. Enable and send a break. This causes the host to have a boot ROM character of $ 7F (DEL). Know that you are waiting (corresponding to the key). The host responds $ 7F and the boot ROM is the host's Beauley To calculate. After receiving $ 7F, the boot ROM turns off the break character and autos End the bow process. When you exit the autobo process, the boot ROM will send a confirmation message (Send'Ready!>') To inform the host that the command is available.

15.3.4 Command loop state machine and error process

When the autobo process ends, the boot ROM can process commands sent by the host. It will be in a state. After the autobo process, the boot ROM becomes a simple state machine and commands To process. There are four commands that the host can accept:'G',' g',' S', and's'. In general, boot Characters received by the ROM as the correct command are echoed back to the host. All illegal characters '? It will be echoed back with'. However, there are limits to the errors that the boot ROM code can detect.

The'S'or's' command indicates “read S-record”. Boot ROM is'S'or's' Interprets the following bytes as an S-record type indicator. Boot ROM is Motorola's S record Expected to read format (S1 or S2). Other formats (S0, S3, S7, S8, or S9) are ignored vinegar. If the S-record type is not what you expected,'? 'Echo back. Anticipate If so, the type indicator bytes are echoed back to the host and control is in boot ROM. Moves to the specified processing subroutine.

Example S224001000274C 35
S224001020F500 ..... 7B

The'G'or'g' command indicates "go to specified address". Boot ROM is'G'or'g' Reads the following hexadecimal character and interprets it as an address. Each character is echoed as read There can be a space between the'G'or'g' and the address. The address is a carriage Ends with a return or newline (the last 5 hexadecimal digits entered are interpreted, other digits are ignored vinegar). The control flow immediately moves to the specified address.

Example G01000 (boot program changes CPU PC to $ 01000)

15.3.5 Connection example

The following is an example of host connection.

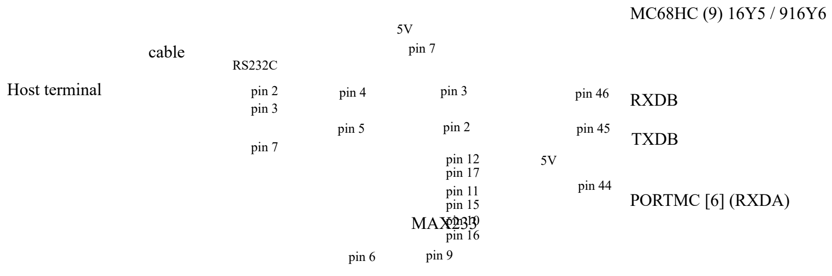


Figure 54 Connection example

15.3.6 Mapping

The boot ROM code mapping is as follows:

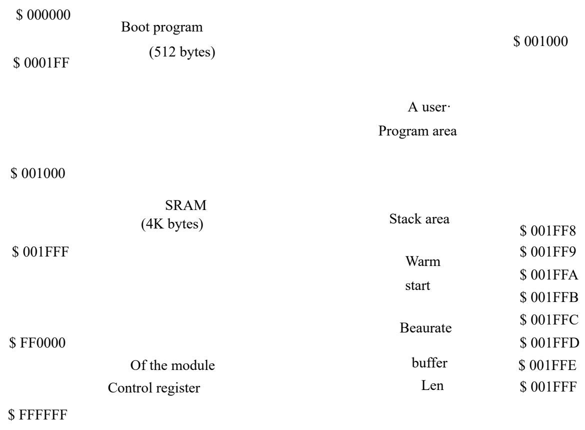


Figure 55 Boot ROM Code Mapping

The user must be careful that the address does not conflict with other modules. In the boot ROM From $ 0 to $ 1FF will be allocated. SRAM is assigned from $ 01000 to $ 01FFF. Boot ROM code Do reserves $ 01FF9 to $ 01FFF for user program downloads. In the stack area Requires a few bytes from $ 01FF8.

15.3.7 Easy boot operation

15.3.7.1 Connection

The board settings are shown in Figure 54, “Example Connection ”.

15.3.7.2 Baud rate setting

PORTMC [6] will be pulled up. The user sends the'DEL' character ($ 7F) from the host terminal. SCIB The baud rate is calculated and written to the SCIB baud rate register for a warm start. The same value is put into the SRAM's $ 01FFC and $ 01FFD (baud rate). Once the baud rate is set, The boot ROM code returns'Ready!>' To the host terminal.

15.3.7.3 Program Download

When the'Ready!>'Prompt appears, the user can access the MC68HC (9) 16Y5 / 916Y6 SRAM from the host terminal. You can send the code. For example, if you are using Kermit software, a simple send You can send S-records with a message command.

15.3.7.4 Program counter settings

After loading the S-record, change the PC (CPU program counter) to S-record from the host terminal. You can run the user program specified in the code. Use the'G'command to change the PC Yes. For example, if a user sends'G01000' to the SCIB of MC68HC (9) 16Y5 / 916Y6, the boot ROM

The code changes the PC to $ 01000. For warm start, the same PC value is SRAM $ 01FF9,
You can put it in $ 01FFA and $ 01FFB (warm start). Then the user program starts.

15.3.7.5 Warm start

The user can restart the user program with an external reset at power-on. PORTMC [6]
When is pulled down, the Baudrate stored in the SRAM becomes the SCIB Baud Rate.
The PC value (Warmstart) that was transferred to the register and stored in SRAM is sent to the program counter of the CPU.
Will be transferred.

Next, the user program starts.

Appendix A
Electrical characteristics

This appendix provides a specification table of electrical characteristics and a reference timing diagram. Specifications table for each timing diagram
The numbers corresponding to the parameters detailed in are listed. Applicable to this related timing specification table
It also contains notes. MC68HC (9) 16Y5 / 916Y6 did not use certain option pins
Therefore, these pins are treated as "no connection".

Page 315

Table 160 Maximum rating

Num	Rating	Symbol	Value Value.	Unit
1	Supply Voltage [1, 2]	V_{DD}	– 0.3 to + 6.5	V
2	Input Voltage [1, 2, 3, 5]	V_{in}	– 0.3 to + 6.5	V
3	Instantaneous Maximum Current Single pin limit (applies to all pins) [1, 5, 6, 7]	I_D	twenty five	mA
Four	Operating Maximum Current Digital Input Disruptive Current [4, 5, 6, 7] $V_{NEGCLAMP} \approx -0.3$ V $V_{POSCLAMP} \approx V_{DD} + 0.3$ V	I_{ID}	– 500 to +500	∝ A
Five	Operating Temperature Range MC68HC16Y5 C Suffix MC68HC916Y5 C Suffix MC68HC916Y6 C Suffix Operating Temperature Range	T_A	T_L to T_H – 40 to + 85	° C

	MC68HC16Y5 VSuffix MC68HC916Y5 VSuffix MC68HC916Y6 VSuffix		T_L to T_H – 40 to + 105	
6	Storage Temperature Range	T_{stg}	– 55 to + 150	° C

NOTES: NOTES:

1. Permanent damage can occur if maximum ratings are exceeded. Exposure to voltages or currents in ex- Device modules may not operate normally while be- ing exposed to electrical extremes.
2. Although sections of the device contain circuits to protect against damage from high static voltages or electrical fields, take normal precautions to avoid exposure to voltages higher than maximum-rated factors.
3. All functional non-supply pins are internally clamped to V_{SS} . All functional pins except EXTAL and XFC are internally clamped to V_{DD} .
4. Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values for positive and negative clamp voltages, then use the larger of the two values.
5. Power supply must maintain regulation within operating V_{DD} range during operating and operating maximum current condition.
6. This parameter is periodically sampled rather than 100% tested.
7. Total input current for all digital input-only and all digital input / output pins must not exceed 10mA. Exceeding this limit can cause disruption of normal operation.

Page 316

Table 161 Thermal characteristics

Num	Characteristic		Symbol	Value Value.	Unit
1	Thermal Resistance Plastic 160-Pin Surface Mount	MC68HC916Y5 MC68HC916Y6	Θ_{JA}	26	° C / W
		MC68HC16Y5		37 37	

The average chip-junction temperature (TJ) in C can be obtained from:

$$T_J = T_A + (P_D \cdot \Theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, ° C

Θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, ° C / W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{DD} \cdot V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273 ° C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \cdot (T_A + 273 ° C) + \Theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving Equations (1) And (2) Iteratively For Any Value Of T_A .

Page 317

Table 162 Clock Control Timing

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ To T_H)

Num	Characteristic	Symbol	Min	Max	Unit
1	PLL Reference Frequency Range	f_{ref}	3.2 3.2	8.0 8.0	MHz
	System Frequency (MC68HC916Y5 / 916Y6)				
	Fast On-Chip PLL System Frequency		3.2 3.2	25.0	MHz
	External Clock Operation		3.2 3.2	25.0	
2		f_{sys}			
	System Frequency (MC68HC16Y5)				
	Fast On-Chip PLL System Frequency		(f_{ref}) / 128	25.0	MHz
	External Clock Operation		dc	25.0	
3	PLL Startup Time [1, 2, 3, 4]	t_{spll}	— —	50	ms
Four	PLL Lock Time [1, 3, 4, 5]	t_{lpll}	— —	20	ms
Five	Limp Mode Clock Frequency [6]	f_{limp}	0.1	f_{sys} max / 2	MHz
	CLKOUT Stability [1, 3, 4, 7]				
6	Short term (5 ∝ s interval)	C_{stab}	– 0.5	0.5	%%
	Long term (500 ∝ s interval)		– 0.05	0.05	

NOTES: NOTES:

1. This parameter is periodically sampled rather than 100% tested.
2. Parameter measured from the time V_{DD} and V_{DDSYN} are valid to release.
3. Assumes that a low-leakage external filter capacitor with value of 0.1 ∝ F is attached to the XFC pin. Total external resistance from the XFC pin due to external leakage sources must be greater than 15M Ω to guarantee this specification.
4. Proper layout procedures must be followed to achieve the specifications.
5. Assumes that stable V_{DDSYN} is applied, and that the crystal oscillator is stable. Lock time is measured from the V Time DD And V_{DDSYN} Valid To Release. This Specification Also Applies To The Period Required For PLL Lock After changing the W and Y frequency control bits in the synthesizer control register (SYNCR) while the PLL is running, and to the period required for the clock to lock after LPSTOP.
6. Determined by the initial control voltage applied to the on-chip VCO.
7. Stability is the average deviation from the programmed frequency measured over the specified interval at maximum f_{sys} . Measurements are made with the device powered by filtered supplies and clocked by a stable external clock signal. Noise injected into the PLL circuitry via V_{DDSYN} and V_{SS} and variation in crystal oscillator frequency increase the C_{stab} percentage for a given interval.

Page 318

Table 163 DC characteristics

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
0	Supply Voltage	V_{DD}	4.5 4.5 (V_{DD} min)	5.25 (V_{DD} max)	V
1	Input High Voltage	V_{IH}	0.7 (V_{DD})	V_{DD} + 0.3	V
2	Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.2 (V_{DD})	V
3	Input Hysteresis [1,8]	V_{HYS}	0.5	— —	V
Four	Input Leakage Current [2] V_{in} = V_{DD} or V_{SS} All input-only pins except ADC pins	I_{IN}	–2.5	2.5	∝ A
Five	High Impedance (Off-State) Leakage Current [2] V_{in} = V_{DD} or V_{SS} All input / output pins	I_{OZ}	–2.5	2.5	∝ A
6	CMOS Output High Voltage [2,3] I_{OH} = –10.0 ∝ AGroup 1, 2 pins	V_{OH}	V_{DD} – 0.2	— —	V
7	CMOS Output Low Voltage [2] I_{OL} = 10.0 ∝ A Group 1, 2 pins	V_{OL}	— —	0.2 0.2	V
8	Output High Voltage [2,3] I_{OH} = –0.8 mA Group 1, 2 pins	V_{OH}	V_{DD} – 0.8	— —	V
9	Output Low Voltage [2] I_{OL} = 1.6 mA Group 1 pins I_{OL} = 5.3 mA Group 2 pins I_{OL} = 12 mA Group 3 pins	V_{OL}	 — — — — — —	 0.4 0.4 0.4 0.4 0.4 0.4	V
Ten	Mode Select Pull-up Current [3] V_{in} = V_{IL} V_{in} = V_{IH}	I_{MSP}	 — — –15	 –120 — —	∝ A
	MC68HC16Y5 V_{DD} Supply Current Run [6] LPSTOP, crystal, VCO Off (STSLIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	 — — — — — —	 230 230 Five Ten	 mA mA mA
11 11	MC68HC916Y5 V_{DD} Supply Current Run [6] LPSTOP, crystal, VCO Off (STSLIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	 — — — — — —	 300 Five Ten	 mA mA mA
	MC68HC916Y6 V_{DD} Supply Current Run [6] LPSTOP, crystal, VCO Off (STSLIM = 0) LPSTOP, external clock input frequency = maximum f_{sys}	I_{DD}	 — — — — — —	 270 Five Ten	 mA mA mA
12	Clock Synthesizer Operating Voltage	V_{DDSYN}	4.5 4.5	5.25	V
13	V_{DDSYN} Supply Current [5] VCO on, maximum f_{sys} : Fast reference mode (6MHz) LPSTOP, VCO off (STSLIM = 0), 6MHz crystal V_{DD} powered down, 6MHz crystal	I_{DDSYN}	 — — — — — —	 3 1 3	 mA mA mA
14	RAM Standby Voltage [9] Normal mode [10] (V_{DD} min ≤ V_{DD} ≤ V_{DD} max) Standby mode [11] (V_{DD} = 0V) Power down status negation (PDS flag)	V_{SB}	 0.0 0.0 3.0 3.0 — —	 V_{DD} V_{DD} max 3.5 3.5	V
15	RAM Standby Current Normal RAM operation: V_{DD} > V_{SB} – 0.2 V Transient condition: V_{SB} – 0.2 V ≥ V_{DD} ≥ V_{SS} + 0.5 V Standby operation: V_{DD} < V_{SS} + 0.5 V	I_{SB}	 — — — — — —	 50 2 200	 ∝ A mA ∝ A

Page 319

Table 163 DC characteristics (continued)

(4.5Vdc ≤ (V_{DD} and V_{DDSYN}) ≤ 5.25Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
16 16	MC68HC16Y5 Power Dissipation [6]	P_D	— —	1210	mW
	MC68HC916Y5 Power Dissipation [6]	P_D	— —	1575	mW
	MC68HC916Y6 Power Dissipation [6]	P_D	— —	1420	mW
17 17	Capacitance Input [2, 8] All input-only pins except ADC pins All input / output pins	C_{IN}	— — — —	Ten 20	pF
18 18	Load Capacitance [2]	C_L			pF
	Group 1 I / O pins, half load		— —	45 45	
	Group 1 I / O pins, full load		— —	90	
	Group 2 I / O pins, half load		— —	50	
	Group 2 I / O pins, full load		— —	100	
	Group 3 I / O Pins		— —	100	
	Group 4 I / O Pins		— —	130	

NOTES: NOTES:

1.Parameter applies to the following pins:

Port A / B: A [15: 8] / AD [15: 8] / PA [7: 0], A [7: 0] / AD [7: 0] / PB [7: 0]
Port C: A [23] / ECLK / PC [7], A [18:16] / PC [2: 0]
Port D: CSA / PD7, CSB / PD6, CSC / PD5, DTACK / PD4, FREEZE / QOUT / FASTREF / CSE1 / PD3,
BKPT / DSCLK / CSE0 / PD2, IPIPE1 / DSI / PD1, IPIPE0 / DSO / PD0
Port E: FC [1: 0] / PE [6: 5], CLKOUT / PE4, SIZE / PE3, AS / PE2, DS / RD / PE1, R / W / WR / PE0
Port F: IRQ [7: 2] / PF [7: 2], IRQ [X] / BERR / PF1, ALE / PF0
Port G / H: D [15: 8] / PG [7: 0], D [7: 0] / PH [7: 0]
Port QA: PQA [7: 5] / AN [59:57], PQA4 / AN56 / ETRIG2, PQA3 / AN55 / ETRIG1, PQA [2: 0] / AN [54:52] / MA [2: 0]
Port QB: PQB [7: 4] / AN [51:48], PQB3 / AN3 / ANZ, PQB2 / AN2 / ANY, PQB1 / AN1 / ANX, PQB0 / AN0 / ANW
Port CT: CTD22, CTD20, CTD [18:14], CTD [8: 3]
Port CS: ACS [2: 0] / PCS [2: 0]
Port MC: TXDA / PMC7, RXDA / PMC6, TXDB / PMC5, RXDB / PMC4, SS / PMC3, SCK / PMC2, MOSI / PMC1, MISO / PMC0
Port SP: RSS / PSP7, RSCK / PSP6, RMOSI / PSP5, RMISO / PSP4, RPCS [3: 0] / PSP [3: 0]
Other: T2CLK, TP [15: 0], CTM2C, EBR / TSC, RESET

2. Input / Output Pins:

Group 1 (Full and Half Load Capability):
Port E: CLKOUT / PE4
Group 2 (Full and Half Load Capability):
Port SP: RSS / PSP7, RSCK / PSP6, RMOSI / PSP5, RMISO / PSP4, RPCS [3: 0] / PSP [3: 0]
Group 3 (Full Load Capability):
Port A / B: A [15: 8] / AD [15: 8] / PA [7: 0], A [7: 0] / AD [7: 0] / PB [7: 0]
Port C: A [23] / ECLK / PC [7], A [18:16] / PC [2: 0]
Port D: CSA / PD7, CSB / PD6, CSC / PD5, DTACK / PD4, FREEZE / QOUT / FASTREF / CSE1 / PD3,
BKPT / DSCLK / CSE0 / PD2, IPIPE1 / DSI / PD1, IPIPE0 / DSO / PD0
Port E: FC [1: 0] / PE [6: 5], SIZE / PE3, AS / PE2, DS / RD / PE1, R / W / WR / PE0
Port F: IRQ [7: 2] / PF [7: 2], IRQ [X] / BERR / PF1, ALE / PF0
Port G / H: D [15: 8] / PG [7: 0], D [7: 0] / PH [7: 0]
Port QA: PQA [7: 5] / AN [59:57], PQA4 / AN56 / ETRIG2, PQA3 / AN55 / ETRIG1, PQA [2: 0] / AN [54:52] / MA [2: 0]
Port CT: CTD22, CTD20, CTD [18:14], CTD [8: 3]
Port MC: TXDA / PMC7, RXDA / PMC6, TXDB / PMC5, RXDB / PMC4, SS / PMC3, SCK / PMC2, MOSI / PMC1, MISO / PMC0
Port SP: RPCS [3: 0] / PSP [3: 0]
Other: TP [15: 0]
Group 4 (Open Drain): RESET

Page 320

Output Pins:

Group 3 (Full Load Capability):

Port CS: ACS [2: 0] / PCS [2: 0]

Other: CPWM [13: 9]

3. Use of an active pulldown device is recommended.
5. Total operating current is the sum of the appropriate $I_{DD,}$ I_{DDSYN} and I_{SB} values, plus $I_{DDA.}$
6. Current measured with system clock frequency of 25.0MHz, all modules active.
7. Power dissipation measured with system clock frequency of 25.0 MHz, all modules active. Power dissipation measured with system clock frequency of 25.0 MHz, all modules active.
   can be calculated using the following expression:

   $$P_D = \text{Maximum } V_{DD} (I_{DD} + I_{DDSYN} + I_{SB}) + \text{Maximum } V_{DDA} (I_{DDA})$$

   I_{DD} includes supply currents for all device modules powered by V_{DD} pins.
8. This parameter is periodically sampled rather than 100% tested.
9. V_{DD} , V_{DD} min, V_{DD} max are defined as described below:
   V_{DD} : a value of supply voltage actually applied to VDD pins
   V_{DD} min: the lowest value of supply voltage within operating condition range (see No.0).
   V_{DD} max: the highest value of supply voltage within operating condition range (see No.0).
10. In normal mode, RAM is readable / writable.
11. In standby mode, RAM array is protected.

Page 321

--SLIM AC characteristics

Table 164 AC Timing

$(4.5\text{Vdc} \le (V_{DD}$ and $V)_{DDSYN}) \le 5.25\text{Vdc}, V_{SS} = 0$ Vdc, $T_A = T_L$ to $T_H)$[1]

Num	Characteristic	Symbol	Min	Max	Unit
F2	Frequency of Operation, Fast Reference (6 MHz crystal [2] , ROM version)	f_{FST}	0.46	25.0	MHz
	Frequency of Operation, Fast Reference (6 MHz crystal [2] , FLASH version)	f_{FST}	3.2 3.2	25.0	MHz
1	Clock Period	t_{cyc}	40	— —	ns

1A	ECLK Period	t_{Ecyc}	8t cyc	16t cyc	ns
1B	External Clock Input Period [3]	t_{Xcyc}	40	— —	ns
twenty three	Clock Pulse Width	t_{CW}	0.5 t_{cyc} – 5	— —	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	4t cyc	8t cyc	ns
2B, 3B	External Clock Input High / Low Time [3]	t_{XCHL}	0.5t cyc	— —	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	— —	Five	ns
4A, 5A	Rise and Fall Time (All outputs except CLKOUT)	t_{rf}	— —	8	ns
4B, 5B	External Clock Input Rise and Fall Time	t_{XCrf}	— —	Five	ns
6	Clock High to ADDR, FC, SIZE, R / W Valid	t_{CHAV}	0	28 28	ns
7	Clock High to ADDR, Data, FC, SIZE, R / W High Impedance	t_{CHAZx}	0	0.5t cyc	ns
8	Clock High to ADDR, FC, SIZE Invalid [4]	t_{CHAZn}	0	— —	ns
9	Clock Low to AS, DS, CS, RD, WR Asserted	t_{CLSA}	2	twenty two	ns
9A	AS to DS or CS Asserted (Read) [5]	t_{STSA}	-15	15	ns
9B	AS Asserted to CS Asserted (STRB = 0) DS Asserted to CS Asserted (STRB = 1)	t_{ADCS}	-15	15	ns
11 11	ADDR, FC, SIZE Valid to AS, CS, DS RD, WR Asserted	t_{AVSA}	Ten	— —	ns
12	Clock Low to AS, DS, CS RD, WR Negated	t_{CLSN}	2	twenty two	ns
13	AS, DS, CS, RD, WR Negated to ADDR, FC, SIZE Invalid (Address Hold) [4]	t_{SNAI}	Ten	— —	ns
14	AS, CS (STRB = 0) Width Asserted [8]	t_{SWA}	(0.66 + ws) t cyc	— —	ns
14A	DS, CS (STRB = 1) RD, WR Width Asserted (Read or Write) [8,19]	t_{SWAW}	(ws – 0.5) t cyc	— —	ns
15	AS, DS, CS, RD, WR Width Negated [9]	t_{SN}	0.5t cyc	— —	ns
16 16	Clock High to AS, DS, R / W, RD, WR, SIZE, ALE High Impedance	t_{CHSZ}	— —	0.5t cyc	ns
17 17	AS, DS, CS Negated to R / W Invalid	t_{SNRN}	Ten	— —	ns
18 18	Clock High to R / W High (Read)	t_{CHRH}	0	twenty two	ns
20	Clock High to R / W Low (Write)	t_{CHRL}	0	twenty two	ns
twenty one	R / W High to AS, CS, RD, WR Asserted (Read)	t_{RAAA}	Ten	— —	ns
twenty two	R / W Low to DS, CS, RD, WR Asserted (Write)	t_{RASA}	15	— —	ns
twenty three	Clock High to Data Out Valid (Write)	t_{CHDO}	— —	0.5t cyc	ns

Table 164 AC Timing (continued)

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
twenty four	Data Out Valid to Negating Edge of AS, CS, RD, WR (Write)	t_{CHDO}	Ten	— —	ns
twenty five	DS, CS, WR Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	Five	— —	ns
26	Data Out Valid to DS, CS Asserted (Write) [10]	t_{CHDO}	Ten	— —	ns
27	Data In Valid to Clock Low (Data Setup)	t_{DICL}	Ten	— —	ns
27A	Late BERR Asserted to Clock Low (Setup Time) [11]	t_{BELCL}	15	— —	ns
27B	Late BERR Hold Time [11]	t_{LBEH}	15	— —	ns
28 28	AS, DS Negated to DTACK, BERR Negated	t_{SNDN}	0	0.5t cyc	ns
29	DS, CS Negated to Data In Invalid (Data In Hold) [12]	t_{SNDI}	0	— —	ns
29A	DS, CS, RD Negated to Data In High Impedance (Read, IDLE = 0) [12]	t_{SHDI0}	— —	1t cyc	ns
29B	DS, CS, RD Negated to Data In High Impedance (Read, IDLE = 1) [12]	t_{SHDI1}	— —	2t cyc	ns
30	CLKOUT Low to Data In Invalid (Data In Hold) [12]	t_{CLDI}	15	— —	ns

30A	CLKOUT Low to Data In High Impedance (IDLE = 0) [12]	t_{CLDH0}	Ten	$1t_{cyc}$	ns
30B	CLKOUT Low to Data In High Impedance (IDLE = 1) [12]	t_{CLDH1}	15	$2\ t_{cyc}$	ns
31	DTACK Asserted to Data In Valid [11]	t_{DADI}	— —	0.8t cyc	ns
32	AS, CS Asserted to DTACK Asserted. [8, 20]	t_{SADTI}	(ws – 1) t_{cyc}	(ws – 0.5) t cyc – 5	ns
46	R / W Width Asserted (Write or Read Cycle) [8]	t_{RWA}	(1.5 + ws) t cyc	— —	ns
47A	Input Setup Time EBR, DTACK, BERR [11]	t_{AIST}	15	— —	ns
47B	Input Hold Time EBR, DTACK, BERR	t_{AIHT}	15	— —	ns
48	DTACK Asserted to BERR or late BERR Asserted [11, 13]	t_{DABA}	— —	0.5t cyc	ns
49	IPIPE0 High to RESET High	t_{IHRH}	0.5t cyc	— —	ns
50	IPIPE0 High to Mode Select Impedance	t_{IHMZ}	— —	5t cyc	ns
51	RESET High to IPIPE0 Valid [14]	t_{RHIV}	— —	0.5tcyc	ns
53	Data Out Hold from Clock High	t_{DOCH}	0	— —	ns
54	Clock High to Data Out High Impedance	t_{CHDH}	— —	0.5t cyc	ns
55 55	R / W Asserted to Data Bus Impedance Change	t_{RADC}	0.5t cyc	— —	ns
56	RESET Pulse Width	t_{HRPW}	512t cyc	— —	ns
66 66	ADDR, FC, SIZE, R / W Valid to ALE Low (Setup) [17]	t_{AVAL}	Ten	— —	ns
69	Clock Low to Data Out High Impedance (Show Cycle), Only if the Next Cycle is a Read Cycle	t_{CHDOZ}	Five	0.5 t_{cyc}	ns
70A	Clock High to Beginning Data Bus Driven (Show Cycle)	t_{SCHDD}	0	0.5t cyc	ns
71 71	Data Setup Time to Clock Low (Show Cycle)	t_{SCLDS}	-3	— —	ns

Page 323

Table 164 AC Timing (continued)

(4.5Vdc ≤ (V DD and V) DDSYN) ≤ 5.25Vdc, V SS = 0 Vdc, T A = T L to T H) [1]

Num	Characteristic	Symbol	Min	Max	Unit
72	Data Hold from Clock Low (Show Cycle)	t_{SCLDH}	Five	— —	ns
73	BKPT Input Setup Time	t_{BKST}	Five	— —	ns
74 74	BKPT Input Hold Time	t_{BKHT}	15	— —	ns
75	Mode Select Setup Time	t_{MSS}	20t cyc	— —	ns
76 76	Mode Select Hold Time	t_{MSH}	0	— —	ns
77 77	RESET Assertion Time [14]	t_{RSTA}	4t cyc	— —	ns
78 78	RESET Rise Time [15, 16]	t_{RSTR}	— —	190t cyc	ns
79 79	RESET Low and IPIPE0 Low to Mode Select Drive	t_{RILM}	0	— —	ns
80	Data-In Valid to CLKOUT Low (Data Setup, Read) [17]	t DIOL	Ten	— —	ns
81	Clock High to Address, FC, SIZE, R / W Valid [17]	t CHAVM	0	37 37	ns
82	Clock High to ALE Asserted [17]	t CHAA	0	twenty five	ns
83	ALE Asserted to AddressValid [17]	t AAAV	-Ten	Ten	ns
84 84	Clock Low to ALE Negated [17]	t CLAN	0	20	ns
85	ALE Negated to AD [15: 0] Invalid [17]	t ANAI	Five	— —	ns
86	ALE Negated to AD [15: 0] High Impedance (Read) [17]	t ANAZ	Five	0.5 t cyc	ns
87	Clock High to AD [15: 0] Invalid [17]	t CHAI	0	— —	ns
88	Clock High to Address High Impedance (Read) [17]	t CHAZ	0	0.5t cyc	ns
89	AD [15: 0] High Impedance to DS, RD Asserted (Read) [17]	t AZDA	Ten	— —	ns
90	ALE Negated to DS, RD, WR Asserted [17]	t ANDA	0.5t cyc	— —	ns
91	Minimum ALE Assertion Period [17]	t MALA	0.5 t cyc – 10	— —	ns

92	DS, RD Asserted to Data-In Valid (Read) [8, 17]	t DSDV	1	(0.33 + ws-1) t cyc	ns
93	Data-Out Valid to DS, WR Asserted (Write, Setup, SHEN = 0) [17]	t DVDA	Five	— —	ns
93A	Data-Out Valid to DS, WR Asserted (Write, Setup, SHEN = 1) [17]	t DVDAI	-Ten	Ten	ns
94	CS, DS, WR Negated to Data-Out Invalid (Write, Hold) [17]	t DNDOI	8	— —	ns
95	Clock High to Data-Out Valid (Write, SHEN = 0) [17]	t CHDV0	— —	30	ns
95A	Clock Low to Data-Out Valid (Write, SHEN = 1) [17]	t CLDV1	— —	30	ns
96	Clock High to Data-Out Invalid (Write, Hold)	t CHDI	2	— —	ns
97	DS, CS, RD Negated to Data-In Invalid (Data-In Hold, Read) [17]	t DNDII	0	— —	ns
97A	DS, CS, RD Negated to Data-In High Impedance (Read, IDLE = 0) [17]	t DNDZ	0	0.25t cyc	ns
97B	DS, CS, RD Negated to Data-In High Impedance (Read, IDLE = 1) [17]	t DNDZI	0	1.25t cyc	ns
98	DS Negated to Data-Out Valid (Show MUX, followed by another bus cycle; see timing diagram) [17]	t DNDV	— —	2t cyc	ns

Table 164 AC Timing (continued)

$(4.5\text{Vdc} \le (V_{DD} \text{ and } V)_{DDSYN}) \le 5.25\text{Vdc}, V_{SS} = 0 \text{ Vdc}, T_A = T_L \text{ to } T_H)$ [1]

Num	Characteristic	Symbol	Min	Max	Unit
98A	DS Negated to Data-Out Valid (Show MUX, followed by an idle bus) cycle; see timing diagram) [17]	t DNDVI	20	— —	ns
99	CLKOUT Low to Data-In Invalid (Data-In Hold, Read) [17]	t OLDI	Ten	— —	ns
99A	CLKOUT Low to Data-In High Impedance (Read, IDLE = 0) [17]	t OLDZ	— —	0.5t cyc +10	ns
99B	CLKOUT Low to Data-In High Impedance (Read, IDLE = 1) [17]	t OLDZI	— —	1.75t cyc	ns
100	CLKOUT High to Phase 1 Asserted [18]	t CHP1A	3	0.5t cyc +10	ns
101	CLKOUT High to Phase 2 Asserted [18]	t CHP2A	3	0.5t cyc +10	ns
102	Phase 1 Valid to AS or DS Asserted [18]	t P1VSA	Ten	— —	ns
103	Phase 2 Valid to AS or DS Negated [18]	t P2VSN	Ten	— —	ns
103A	Phase 2 Valid to CLKOUT High [18]	t P2VSN	Ten	— —	ns
104	AS or DS Asserted to Phase 1 Negated [18]	t SAP1N	Ten	— —	ns
105	AS or DS Negated to Phase 2 Negated [18]	t SNP2N	Ten	— —	ns
105A	CLKOUT High to Phase 2 Negated [18]	t SNP2N	Ten	— —	ns

NOTES: NOTES:

1.All AC timing is shown with respect to 20% and 70% V_{DD} levels unless otherwise noted.

2.Minimum system clock frequency is crystal frequency divided by 128, subject to specified limits.

3.Minimum external clock high and low times are based on a 50% duty cycle. The minimum Allow t_{Xcyc} period is reduced when the duty cycle of the external clock varies. The relationship between external clock input duty cycle and minimum t_{Xcyc} is expressed:

Minimum t_{Xcyc} period = minimum t_{XCHL} / (50% – external clock input duty cycle tolerance).

To achieve maximum operating frequency (f_{sys}) while using an external clock input, adjust clock input duty cyclt to obtain a 50% duty cycle on CLKOUT.

4.For non-multiplexed operation, applies to ADDR [23: 0], FC [1: 0], SIZE, and R / W; for multiplexed operation, applies only to ADDR [23:16], FC [1: 0], SIZE, and R / W.

5.Specification 9A is the worst-case skew between AS and DS or CS. The amount of skew depends on the relative loading of these signals. When loads are kept within specified limits, skew will not cause AS and DS to fall outside the limits shown in specification 9.

8.Number of wait states = ws. When fast termination is used (2-clock bus) ws = 0. For 3-clock bus, ws = 1.

9. If multiple chip selects are used, CS width negated (specification 15) applies to the time from the negation of a heavily loaded chip The CS width negated specification between multiple chip selects does not. apply to chip selects being used for synchronous ECLK cycles.

10. Specification 26 is valid only for 3-clock or more bus cycles. Specification 26 does not apply to 2-clock bus cycles.

11. If the setup time (specification 47A) requirements are satisfied, the DTACK low to data setup time (specification 31) and DTACK low to BERR low setup time (specification 48) can be ignored. The data must only satisfy the data-in to clock low setup time (specification 27) for the following clock cycle. BERR must satisfy only the late BERR low to clock low setup time (specification) 27A) and the late BERR hold time (specification 27B) for the following clock cycle.

12. These hold times are specified with respect to DS or CS on asynchronous reads or CLKOUT. The user is free to choose either hold time.
13. In the absence of DTACK, BERR is an asynchronous input using the asynchronous setup time (specification 47A or 27A).
14. After external RESET negation is detected, a short transition period (approximately 2 tcyc) elapses, then the SLIM drives RESET low for 512 tcyc. During the transition period IPIPE0 may be High or Low.
15. External assertion of the RESET input can overlap internally-generated resets. To insure that an external reset is recognized in all cases, RESET must be asserted for at least 750 CLKOUT cycles.
16. External logic should pull RESET high during this period. If RESET is not High by the end of this period, the SLIM waits an additional 180 tcyc.

17. Multiplexed timing parameters are for 3-clock and greater bus cycle operation, with the exception of Emulation Mode bus cycles.
18. Applies only to devices that include the CPU16. Eight pipeline states are multiplexed into IPIPE [1: 0]. The multiplexed signals have two phases.
19. When ws = 0, min value = 0.5 tcyc.
20. The WS is 1 up to.

Table 165 Baggrand debug mode timing

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H) [1]

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	— —	ns
B1	DSI Input Hold Time	t_{DSIH}	Ten	— —	ns
B2	DSCLK Setup Time	t_{DSCSU}	18 18	— —	ns
B3	DSCLK Hold Time	t_{DSCH}	Ten	— —	ns
B4	DSO Delay Time	t_{DSOD}	— —	twenty five	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	$2t_{cyc}$	— —	ns
B6	CLKOUT High to FREEZE Asserted / Negated	t_{FRZAN}	— —	50	ns
B7	CLKOUT High to IPIPE1 / DSI High Impedance	t_{IFZ}	— —	50	ns
B8	CLKOUT High to IPIPE1 / DSI High Impedance Change (Drive)	t_{IF}	— —	50	ns
B9	DSCLK Low Time	t_{DSCLO}	$1t_{cyc}$	— —	ns
B10	IPIPE1 / DSI High Impedance to FREEZE Asserted	t_{DSIZFA}	16 16	— —	ns
B11	FREEZE Negated to IPIPE1 / DSI High Impedance Change (Drive)	t_{FNDSID}	16 16	— —	ns

NOTES: NOTES:

1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.

Page 326

Table 166 ECLK Bus Timing

(4.5Vdc ≤ (V $_{DD}$ and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
E1	ECLK Low to Address Valid[2]	t_{EAD}	——	$1t_{cyc}$	ns
E2	ECLK Low to Address Hold	t_{EAH}	Ten	——	ns
E3	ECLK Low to CS Valid (CS Delay)	t_{ECSD}	——	$3.5t_{cyc}$	ns
E4	ECLK Low to CS Hold	t_{ECSH}	15	——	ns
E5	CS Negated Width	t_{ECSN}	$0.5t_{cyc}$	——	ns
E6	Read Data Setup Time	t_{EDSR}	$0.75t_{cyc}$	——	ns
E7	Read Data Hold Time	t_{EDHR}	15	——	ns
E8	ECLK Low to Data High Impedance	t_{EDHZ}	——	$1t_{cyc}$	ns
E9	CS Negated to Data Hold (Read)	t_{ECDH}	0	——	ns
E10	CS Negated to Data High Impedance (Read)	t_{ECDZ}	——	$1t_{cyc}$	ns
E11	ECLK Low to Data Valid (Write)	t_{EDDW}	——	$2t_{cyc}$	ns
E12	ECLK Low to Data Hold (Write)	t_{EDHW}	Five	——	ns
E13	CS Negated to Data Hold (Write)	t_{ECHW}	0	——	ns
E14	Address Access Time (Read)[3]	t_{EACC}	$6t_{cyc}$	——	ns
E15	Chip-Select Access Time (Read)[4]	t_{EACS}	$4.5t_{cyc}$	——	ns
E16	Address Setup Time	t_{EAS}	$0.5t_{cyc}$	——	ns

NOTES: NOTES:

1. All AC timing is shown with respect to V_{IH} / V_{IL} levels unless otherwise noted.
2. When previous bus cycle is not an ECLK cycle, the address may be valid before ECLK goes low.
3. Address access time = t_{Ecyc} – t_{EAD} – t_{EDSR} .
4. Chip-select access time = t_{Ecyc} – t_{ECSD} – t_{EDSR} .

Page 327

--ACS AC characteristics

Table 167 ACS Timing

$(4.5\text{Vdc} \leq (V_{DD} \text{ and } V_{DDSYN}) \leq 5.25\text{Vdc}, V_{SS} = 0 \text{ Vdc}, T_A = T_L \text{ to } T_H)$

Num	Characteristic	Symbol	Min	Max	Unit
1	Clock Low to ACS asrt	t_{CLSA}	2.0 2.0	25.0	ns
2	Clock Low to ACS neg	t_{CLSN}	2.0 2.0	23.0	ns

--SPI AC characteristics

Table 168 SPI Timing

$(4.5\text{Vdc} \leq (V_{DD} \text{ and } V_{DDSYN}) \leq 5.25\text{Vdc}, V_{SS} = 0 \text{ Vdc}, T_A = T_L \text{ to } T_H)^1$

Num	Parameter	Symbol	Min	Max	Unit
0	Operating Frequency				
	Master	f_{op}	DC	1/4	System Clock Frequency
	Slave		DC	1/4	System Clock Frequency

1	Cycle Time				
	Master	t_{qcyc}	Four	510 510	t_{cyc}
	Slave		Four	— —	t_{cyc}
2	Enable Lead Time				
	Master	t_{lead}	2	— —	t_{cyc}
	Slave		2	— —	t_{cyc}
3	Enable Lag Time				
	Master (CPHA = 1)	t_{lag}	— —	1/2	SCK
	Master (CPHA = 0)		2	— —	t_{cyc}
	Slave		2	— —	t_{cyc}
Four	Clock (SCK) High or Low Time				
	Master	t_{sw}	T_{Cyc} - 60	255 t_{cyc}	ns
	Slave[2]		T_{Cyc} - N	— —	ns
Five	Sequential Transfer Delay				
	Master	t_{td}	— —	— —	t_{cyc}
	Slave (Does Not Require Deselect)		13	— —	t_{cyc}
6	Data Setup Time (Inputs)				
	Master	t_{su}	30	— —	ns
	Slave		20	— —	ns
7	Data Hold Time (Inputs)				
	Master	t_{hi}	0	— —	ns
	Slave		20	— —	ns
8	Slave Access Time	t_{a}	— —	1	t_{cyc}
9	Slave MISO Disable Time	t_{dis}	— —	2	t_{cyc}
10	Data Valid (after SCK Edge)				
	Master	t_{v}	— —	50	ns
	Slave		— —	50	ns
11	Data Hold Time (Outputs)				
	Master	$t_{ho ho}$	0	— —	ns
	Slave		0	— —	ns
12	Rise Time				
	Input	t_{ri}	— —	2	∝ s
	Output Output	t_{ro}	— —	30	ns
13	Fall Time				
	Input	t_{ri}	— —	2	∝ s
	Output Output	t_{ro}	— —	30	ns

NOTE: NOTE:

1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. For high time, n = External SCK rise, for low time, n = External SCK fall time.

Page 329

--RSPI AC characteristics

Table 169 RSPI Timing (Master)

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_{A} = T_{L} to T_{H})

Num	Function	Symbol	Skew Off		Skew On		Unit
			Min	Max	Min	Max	
0	System Operation Frequency	f_{sys}	DC	twenty five	DC	twenty five	MHz
1	RSE Setup Time	t_{rsu}	---	12 * t_{cyc} + 25	---	12 * t_{cyc} + 25	ns
2	RSE Hold Time	t_{rho}	0	---	0	---	ns
3	RSS Fall Time	t_{sf}	---	30	---	30	ns
Four	RSS Rise Time	t_{rf}	---	30	---	30	ns
Five	Slave Enable Lead Time	t_{lead}	* T 2 $_{Cyc}$ - 25	1024 * t_{cyc}	* T 2 $_{Cyc}$ - 25	1024 * t_{cyc}	ns
6[1]	Master Cycle Time	t_{sck}	Four	2048	Four	2048	t_{cyc}
7[1]	Slave Enable Lag Time	t_{lag}	Four	2048	Four	2048	t_{cyc}

8	RSCK Low Time	t_{slow}	* T 2 Cyc - 30	1024 * T Cyc - 30	* T 2 Cyc - 30	* T 1024 Cyc - 30	ns
9	RSCK High Time	t_{shigh}	* T 2 Cyc - 30	1024 * T Cyc - 30	* T 2 Cyc - 30	* T 1024 Cyc - 30	ns
Ten	RSCK Fall Time	t_f	---	30	---	30	ns
11 11	RSCK Rise Time	tr_r	---	30	---	30	ns
12 [1,2]	Command Enable Setup Time	t_{csu}	2	1024	2	1024	t_{cyc}
13 [2]	Command Enable Hold Time	t_{cho}	2 * t_{cyc}	1024 * t_{cyc}	2 * t_{cyc}	1024 * t_{cyc}	ns
14	Input Data Setup Time	t_{isu}	30	---	30 --t_{sck} * SKEW / 8	---	ns
15	Input Data Hold Time	t_{iho}	0	---	t_{sck} * SKEW / 8	---	ns
17 17	Output Data Valid Time	t_v	---	50	---	50	ns
18 18	Output Data Hold Time	$t_{ho\ ho}$	---	50	---	50	ns
19 19	RSFIN Assertion Time	t_{fin}	---	10 * t_{cyc}	---	10 * t_{cyc} + t_{sck} * 3/8 ns	
27	MST bit Setup Time	t_{msu}	0	---	0	---	ns
28 28	Output Port Enable Lead Time	t_{olead}	---	35	---	35	ns
29	Output Port Enable Lag Time	t_{olag}	---	35	---	35	ns
30	RSE Clear Lead Time	t_{rlead}	---	10 * t_{cyc}	---	10 * t_{cyc} + t_{sck} * 3/8 ns	

NOTES: NOTES:

1. t_{cyc} is the system clock period of Master RSPI. (= 1 / f_{sys})
2. Valid only for Address Slave-Select Mode (SSM = 1)

Table 170 RSPI Timing (Slave)

(4.5Vdc ≤ (V DD and V) $_{DDSYN}$) ≤ 5.25Vdc, V $_{SS}$ = 0 Vdc, T $_A$ = T $_L$ to T $_H$)

Num	Function	Symbol	Skew Off		Skew On		Unit
			Min	Max	Min	Max	
0	System Operation Frequency	f_{sys}	DC	twenty five	DC	twenty five	MHz
1 [1]	RSE Setup Time	t_{rsu}	12	---	12	---	t_{cyc}
2 [2]	RSE Hold Time	t_{rho}	0	---	t_{baud} * 2/8	---	ns
3	RSS Fall Time	t_{sf}	---	1	---	1	ns
Four	RSS Rise Time	t_{rf}	---	1	---	1	ns
Five	Slave Enable Lead Time	t_{lead}	* T 2 Cyc - 25	---	* T 2 Cyc - 25	---	ns
6	Master Cycle Time	t_{sck}	Four	---	Four	---	t_{cyc}
7	Slave Enable Lag Time	t_{lag}	0	---	0	---	t_{cyc}
8	RSCK Low Time	t_{slow}	50	---	50	---	ns
9	RSCK High Time	t_{shigh}	50	---	50	---	ns
Ten	RSCK Fall Time	t_f	---	1	---	1	∝ s
11 11	RSCK Rise Time	tr_r	---	1	---	1	∝ s
12 [1]	Command Enable Setup Time	t_{csu}	30	---	30	---	t_{cyc}
13 [3]	Command Enable Hold Time	t_{cho}	0	---	0	---	ns
14	Input Data Setup Time	t_{isu}	30	---	30	---	ns
15	Input Data Hold Time	t_{iho}	0	---	0	---	ns
16 [2]	Slave Select Lag Time	t_{slag}	---	twenty five	---	25 + t_{baud} * 2/8	ns

17 [2]	Output Data Valid Time	t_{v}	---	twenty five	---	25 + t_{baud} * 2/8	ns
18 [2]	Output Data Hold Time	$t_{\text{ho ho}}$	---	twenty five	---	25 + t_{baud} * 2/8	ns
19 [2]	RSFIN Assertion Time	t_{fin}	---	10 * t_{cyc}	---	10 * t_{cyc} + t_{sck} * 3 / 8	ns
20 [2]	RSS Noise (High)	t_{sn}	---	20	---	20 + t_{baud} * 2/8	ns
21 [2]	RSS Noise (Low)	t_{sa}	---	20	---	20 + t_{baud} * 2/8	ns
22 [2]	Valid RSS Negation	t_{sv}	20	---	20 + t_{baud} * 2/8	---	ns
23 [2]	MODF Window Lead Time	t_{mflead}	twenty five	---	20 + t_{baud} * 2/8	---	ns
twenty four	MODF Window Lag Time	t_{mflag}	0	---	0	---	ns
25 [1]	RSCK Don't Care Period	t_{sd}	---	Ten	---	Ten	t_{cyc}
26 [1]	Wait Time	t_{dtl}	Ten	---	Ten	---	t_{cyc}

NOTES: NOTES:

1.t_{cyc} is the system clock period of Slave RSPI. (= 1 / f_{sys})

2. t_{baud} is the cycle time of slave baud rate generator.

3.Valid only for Address Slave-Select Mode (SSM = 1)

Page 331

--QADC DC characteristics / AC timing

Table 171 QADC maximum rating

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply, with reference to V_{SSA}	V_{DDA}	– 0.3	6.5 6.5	V
2	Internal Digital Supply, with reference to V_{SSI}	V_{DDI}	– 0.3	6.5 6.5	V
3	Reference Supply, with reference to V_{RL}	V_{RH}	– 0.3	6.5 6.5	V
Four	V_{SS} Differential Voltage	V_{SSI} – V_{SSA}	– 0.1	0.1	V
Five	V_{DD} Differential Voltage	V_{DDI} – V_{DDA}	– 6.5	6.5 6.5	V
6	V_{REF} Differential Voltage	V_{RH} – V_{RL}	– 6.5	6.5 6.5	V
7	V_{RH} to V_{DDA} Differential Voltage	V_{RH} – V_{DDA}	– 6.5	6.5 6.5	V
8	V_{RL} to V_{SSA} Differential Voltage	V_{RL} – V_{SSA}	– 6.5	6.5 6.5	V
9	Disruptive Input Current [1, 2, 3, 4, 5, 6, 7] $V_{NEGCLAMP}$ = – 0.3 V $V_{POSCLAMP}$ = 8 V	I_{NA}	– 500	500	m A
Ten	Positive Overvoltage Current Coupling Ratio [1 , 5 , 6 , 8] PQA PQB	K_{P}	2000 2000	— —	— —
11 11	Negative Overvoltage Current Coupling Ratio [1 , 5 , 6 , 8] PQA PQB	K_{N}	125 500	— —	— —
12	Maximum Input Current [3 , 4 , 6] $V_{NEGCLAMP}$ = – 0.3 V $V_{POSCLAMP}$ = 8 V	I MA	- twenty five	twenty five	mA

NOTES: NOTES:

1.Below disruptive current conditions, the channel being stressed has conversion values of $ 3FF for analog inputs greater than V_{RH} and $ 000 for values less than V_{RL} . This assumes that V_{RH} V_{DDA} and V_{RL} V_{SSA} due to the presence of the sample amplifier. Other channels are not affected by non-disruptive conditions.

2.Input signals with large slew rates or high frequency noise components cannot be converted accurately. These signals also affect the conversion accuracy of other channels.

3.Exceeding limit may cause conversion error on stressed channels and on unstressed channels. Transitions within the limit do not affect device reliability or cause permanent damage.

4.Input must be current limited to the value specified. To determine the value of the required current-limiting resistor, calculate resistance values using positive and negative clamp values, then use the larger of the calculated values.

5.This parameter is periodically sampled rather 100% tested.

6.Condition applies to one pin at a time.

7.Determination of actual maximum disruptive input current, which can affect operation, is related to external system component values.

8.Current coupling is the ratio of the current induced from overvoltage (positive or negative, through an external series coupling resistor), divided by the current induced on adjacent pins. A voltage drop may occur across the external source impedances of the adjacent pins, impacting conversions on these adjacent pins.

Page 332

Table 172 QADC DC electrical characteristics (in operation)

(V_{SSI} and V_{SSA} = 0Vdc, F_{QCLK} = 2.1 MHz, $T_A = T_L$ to T_H)

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply [1]	V_{DDA}	4.5 4.5	5.25	V
2	Internal Digital Supply [1]	V_{DDI}	4.5 4.5	5.25	V
3	V_{SS} Differential Voltage	$V_{SSI} - V_{SSA}$	– 1.0	1.0 1.0	mV
Four	V_{DD} Differential Voltage	$V_{DDI} - V_{DDA}$	– 1.0	1.0 1.0	V
Five	Reference Voltage Low [2]	V_{RL}	V_{SSA}	— —	V
6	Reference Voltage High [2]	V_{RH}	— —	V_{DDA}	V
7	V_{REF} Differential Voltage [3]	$V_{RH} - V_{RL}$	4.5 4.5	5.25	V
8	Mid-Analog Supply Voltage	V_{DDA} / 2	2.25	2.625	V
9	Input Voltage	V_{INDC}	V_{SSA}	V_{DDA}	V
Ten	Input High Voltage, PQA and PQB	V_{IH}	0.7 (V_{DDA})	V_{DDA} +0.3	V
11 11	Input Low Voltage, PQA and PQB	V_{IL}	V_{SSA} – 0.3	0.2 (V_{DDA})	V
12	Input Hysteresis [4]	V_{HYS}	0.5	— —	V
13	Output Low Voltage, PQA [5] I_{OL} = 5.3 mA I_{OL} = 10.0 A	V_{OL}	— — — —	0.4 0.4 0.2 0.2	V
14	Analog Supply Current Normal Operation [6] Low-Power Stop	I_{DDA}	— — — —	15.0 30.0	mA ∝ A
15	Reference Supply Current	I_{REF}	— —	150	∝ A
16 16	Load Capacitance, PQA	C_L	— —	90	pF
17 17	Input Current, Channel Off [7] PQA PQB	$I_{OFF\ OFF}$	— — — —	250 150	nA
18 18	Total Input Capacitance [8] PQA Not Sampling PQA Sampling PQB Not Sampling PQB Sampling	C_{IN}	— — — — — — — —	15 20 Ten 15	pF

NOTES: NOTES:

1.Refers to operation over full temperature and frequency range.

2.To obtain full-scale, full-range results, V_{SSA} V_{RL} V_{INDC} V_{RH} V_{DDA} .

3.Accuracy tested and guaranteed at $V_{RH} - V_{RL}$ = 4.5V-5.25V

4.Parameter applies to the following pins:

Port A: PQA [7: 0] / AN [59:58] / ETRIG [2: 1]
Port B: PQB [7: 0] / AN [3: 0] / AN [51:48] / AN [Z: W]

5.Open drain only.

6.Current measured at maximum system clock frequency with QADC active.

7.Maximum leakage occurs at maximum operating temperature. Current decreases by approximately one-half for each 10 C decrease from maximum temperature.

8.This parameter is periodically sampled rather than 100% tested.

Table 173 QADC AC electrical characteristics (in operation)

(4.5Vdc ≤ (V DDI and VDDA) ≤ 5.25Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Parameter	Symbol	Min	Max	Unit
1	QADC Clock (QCLK) Frequency [1]	F_{QCLK}	0.5	2.1 2.1	MHz
2	QADC Clock Duty Cycle [2, 3] High Phase Time ($T_{PSL} \leq T_{PSH}$)	T_{PSH}	500	— —	ns
3	Conversion Cycles [4]	CC	18 18	32	QCLK cy- cles
Four	Conversion Time [2, 4, 5] F_{QCLK} = 1 MHz [6] Min = CCW / IST =% 00 Max = CCW / IST =% 11 F_{QCLK} = 2.083 MHz [1, 7] Min = CCW / IST =% 00 Max = CCW / IST =% 11	T_{CONV}	18.0 8.64	32 17.28	∝ s
Five	Stop Mode Recovery Time	T_{SR}	— —	Ten	∝ s

NOTES: NOTES:

[1] Conversion characteristics vary with F_{QCLK} rate. Reduced conversion accuracy occurs at max F_{QCLK} rate.

[2] Duty cycle must be as close as possible to 75% to achieve optimum performance.

[3] Minimum applies to 1.0 MHz operation.

[4] Assumes that short input sample time has been selected (IST = 0).

[5] Assumes that f_{SYS} = 25 MHz.

[6] Assumes F_{QCLK} = 1 MHz, with clock prescaler values of:
QACR0: PSH =% 10001, PSA =% 1, PSL =% 110)
CCW: BYP =% 0

[7] Assumes F_{QCLK} = 2.083 MHz, with clock prescaler values of:
QACR0: PSH =% 01000, PSA =% 1, PSL =% 010)
CCW: BYP =% 0

Page 334

Table 174 QADC conversion characteristics (in operation)

(4.5Vdc ≤ (V_{DD} and VDDA) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H

0.5 MHz ≤ F_{QCLK} ≤ 2.1 MHz, 2 clock input sample time))

Num	Parameter	Symbol	Min	Typ	Max	Unit
1	Resolution [1]	1 Count	——	Five	——	mV
2	Differential nonlinearity [2]	DNL	——	——	0.5	Counts
3	Integral nonlinearity	INL	——	——	1.5 1.5	Counts
Four	Absolute error [2, 3, 4]	AE				Counts
	F_{QCLK} = 1 MHz [5]					
	PQA		——	——	± 2.0	
	PQB		——	——	± 2.0	
	F_{QCLK} = 2.083 MHz [6]					
	PQA		——	——	± 2.5	
	PQB		——	——	± 2.5	
Five	Source impedance at input [7]	R_S	——	20	——	k Ω

NOTES: NOTES:

1.At V_{RH} – V_{RL} = 5.12 V, one count = 5 mV.

2.This parameter is periodically sampled rather than 100% tested.

3.Absolute error includes 1/2 count (2.5 mV) of inherent quantization error and circuit (differential, integral, and offset) error. Specification assumes that adequate low-pass filtering is present on analog input pins — capacitive filter with 0.01 Fto 0.1 Fcapacitor between analog input and analog ground, typical source isolation impedance of 20 K.

4.Assumes f_{sys} = 25 MHz.

5.Assumes clock prescaler values of:
QACR0: PSH =% 10001, PSA =% 1, PSL =% 110)
CCW: BYP =% 0

6.Assumes clock prescaler values of:
QACR0: PSH =% 01000, PSA =% 1, PSL =% 010)
CCW: BYP =% 0

7.Maximum source impedance is application-dependent. Error resulting from pin leakage depends on junction leakage into the pin and on leakage due to charge-sharing with internal capacitance.

Error from junction leakage is a function of external source impedance and input leakage current. In the following expression, expected error in result value due to junction leakage is expressed in voltage (V_{errj}):

$$V_{errj} = R_S X I_{OFF}$$

where I_{OFF} is a function of operating temperature. (See Table Table 172, Note 7).

Charge-sharing leakage is a function of input source impedance, conversion rate, change in voltage between Error levels are best determined empir-

ically. In general, continuous conversion of the same channel may not be compatible with high source impedance.

Page 335

--CTM AC characteristics

Table 175 MCSM Timing

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L To T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequecy	f_{CLK}	0	25.0	MHz

Input pin frequecy [1]	f_{PCNTR}	0	f_{CLK} / 4	MHz
Input pin low time [1]	t_{PINL}	$2.0 / f_{CLK}$	— —	∝ s
Input pin high time [1]	t_{PINH}	$2.0 / f_{CLK}$	— —	∝ s
Clock pin to counter increment	t_{PINC}	$4.5 / f_{CLK}$	$6.5 / f_{CLK}$	∝ s
Clock pin to new TBB value	t_{PTBB}	$5.0 / f_{CLK}$	$7.0 / f_{CLK}$	∝ s
Clock pin to COF set ($ FFFF)	t_{PCOF}	$4.5 / f_{CLK}$	$6.5 / f_{CLK}$	∝ s
Load pin to new counter value	t_{PLOAD}	$2.5 / f_{CLK}$	$3.5 / f_{CLK}$	∝ s
Pin to IN bit delay	t_{PINB}	$1.5 / f_{CLK}$	$2.5 / f_{CLK}$	∝ s
Flag to IMB interrupt request	t_{FIRQ}	$1.0 / f_{CLK}$	$1.0 / f_{CLK}$	∝ s
Counter resolution [2]	tf_{CRES}	— —	$2.0 / f_{CLK}$	∝ s

Note: 1. Value applies when using external clock.

2. Value applies when using internal clock. Minimum counter resolution depends on prescaler division ration selection.

Table 176 DASM Timing

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L To T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequecy	f_{CLK}	0	25.0	MHz
Input capture mode: (IPLM, IPM, IC):				
Input pin low time	t_{PINL}	$2.0 / f_{CLK}$	— —	∝ s
Input pin high time	t_{PINH}	$2.0 / f_{CLK}$	— —	∝ s
Input capture resolution [1]	t_{RESCA}	— —	$2.0 / f_{CLK}$	∝ s
Pin to input capture delay	t_{PCAPT}	$2.5 / f_{CLK}$	$4.5 / f_{CLK}$	∝ s
Pin to FLAG set	t_{PFLAG}	$2.5 / f_{CLK}$	$4.5 / f_{CLK}$	∝ s
Pin to IN bit delay	t_{PCAPT}	$1.5 / f_{CLK}$	$2.5 / f_{CLK}$	∝ s
Output compare mode: (OC, OPWM):				
OCT output pulse	t_{OCT}	$2.0 / f_{CLK}$	— —	∝ s
Compare resolution [1]	t_{RESCM}	— —	$2.0 / f_{CLK}$	∝ s
TBB change to FLAG set	t_{FLAG}	$1.5 / f_{CLK}$	$1.5 / f_{CLK}$	∝ s
TBB change to pin change [2]	t_{CPIN}	$1.5 / f_{CLK}$	$1.5 / f_{CLK}$	∝ s
Flag to IMB interrupt request [2]	t_{FIRQ}	$1.0 / f_{CLK}$	$1.0 / f_{CLK}$	∝ s

Note: 1. Minimum resolution depends on counter and prescaler divide ratio selection.

2. Time given from when new value is stable on time base bus.

Table 177 PWMSM Timing

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L To T_H)

Characteristic	Symbol	Min	Max	Unit
System operating frequecy	f_{CLK}	0	25.0	MHz
PWMSM output resolution [1]	t_{PWMR}	— —	$2.0 / f_{CLK}$	∝ s
PWMSM output pulse [2]	t_{PWMO}	$2.0 / f_{CLK}$	— —	∝ s
CPSM enable to output set	t_{PWMP} [3]	$3.5 / f_{CLK}$		∝ s
	t_{PWMP} [Four]	$6.5 / f_{CLK}$		∝ s
PWM enable to output set	t_{PWMP} [Five]	$3.5 / f_{CLK}$	$4.5 / f_{CLK}$	∝ s
	t_{PWMP} [6]	$5.5 / f_{CLK}$	$6.5 / f_{CLK}$	∝ s
Flag to IMB interrupt request [2]	t_{FIRQ}	$1.0 / f_{CLK}$	$1.0 / f_{CLK}$	∝ s

Note: 1. Minimum resolution depends on counter and prescaler divide ratio selection.

2. Excluding the case where the output in always zero.

3. With PWMSM enabled before enabling CPSM and DIV23 = 0.

4. With PWMSM enabled before enabling CPSM and DIV23 = 1.

5. With CPSM enabled before enabling PWMSM and DIV23 = 0.

6. With CPSM enabled before enabling PWMSM and DIV23 = 1.

Page 337

--Flash EEPROM Module / TPU Flash EEPROM Module characteristics

Table 178 Flash EEPROM Module / TPU Flash EEPROM Module Characteristics

$(4.5\text{Vdc} \le (V_{DD} \text{ and } V)_{DDSYN}) \le 5.25\text{Vdc}, V_{SS} = 0 \text{ Vdc}, T_A = T_L$ to $T_H)$[1]

Num	Characteristic	Symbol	Min	Max	Unit
	Program / Erase Supply Voltage [2]				
1	Read Operation	V_{FPE}	$V_{DD} – 0.35$	5.25	V
	Program / Erase / Verify Operation		12	12.6	
	Program / Erase / Verify Supply Current [3]				
	Read Operation		——	15	∝ A
	Program / Erase / Verify Operation				
2	Verify (ENPE = 0)	I_{FPE}	——	50	∝ A
	Program Byte (ENPE = 1)		——	15	mA
	Program Word (ENPE = 1)		——	30	mA
	Erase (ENPE = 1)		——	Four	mA
3	Program Recovery Time [4]	t_{pr}	Ten	——	∝ s
Four	Program Pulse Width	pw_{pp}	20	twenty five	∝ s
Five	Number of Program Pulses [5]	n_{pp}	1	50	——
6	Program Margin [6]	p_m	100	——	%%
7	Number of Erase Pulses [5]	n_{ep}	1	Five	——
8	Erase Pulse Time (= $t_{ei} \cdot n_{ep}$)	t_{epk}	90	550	ms
9	Amount to Increment t_{epk}	t_{ei}	90	110	ms
Ten	Erase Margin = $\sum_{k=1}^{n_{ep}} t_{ei} \cdot k$	e_m	90	1650	ms
11	Erase Recovery Time	t_{er}	1	——	ms
12	Low-Power Stop Recovery Time , [7]	t_{sb}	1	——	∝ s

NOTES: NOTES:

1. All AC timing is shown with respect to 20% and 70% V_{DD} levels unless otherwise noted. For The Program As / Erase Operation, T_A Must Be In The Range Of 0 ° C To 70 ° C.
2. V_{FPE} must not be raised to programming voltage while V_{DD} is below specified minimum value. V_{FPE} must not be reduced below minimum specified value while V_{DD} is applied.
3. Current parameters apply to each individual EEPROM module.
4. Minimum software delay from the end of the write cycle that clears ENPE bit to the read of the flash array
5. Without margin.
6. At 100% margin, the number of margin pulses required is the same as the number of pulses used to program the byte or word.
7. Minimum software delay from the end of the write cycle that clears the STOP bit to the read of the flash array.

Table 179 Flash EEPROM / TPU Flash EEPROM life

Num	Parameter	Symbol	Value Value.	Unit
1	Program-Erase Endurance [1]	e_{pe}	100	cyc

NOTES: NOTES:

1. Number of program-erase cycles (1 to 0, 0 to 1) per bit.

--HD Flash EEPROM Module characteristics

Table 180 HD Flash EEPROM Module Characteristics

(4.5Vdc ≤ (V_{DD} and V) $_{DDSYN}$) ≤ 5.25Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H)[1]

Num	Characteristic	Symbol	Min	Max	Unit
1	Program / Erase Supply Voltage [2] Read Operation Program / Erase / Verify Operation	V_{FPE}	V_{DD} – 0.35 12	5.25 12.6	V
2	Program / Erase / Verify Supply Current [3] Read Operation Program / Erase / Verify Operation Verify (EHV = 0) Program Byte (EHV = 1) Program Word (EHV = 1) Erase (EHV = 1)	I_{FPE}	—— —— —— —— ——	15 50 15 30 Four	∝ A ∝ A mA mA mA
3	Program Recovery Time [4]	t_{pr}	Ten	——	∝ s
Four	Program Pulse Width	pw_{pp}	20	twenty five	∝ s
Five	Number of Program Pulses [5]	n_{pp}	1	50	——
6	Program Margin [6]	p_m	100	——	%%
7	Number of Erase Pulses [5]	n_{ep}	1	Five	——
8	Erase Pulse Time (= $t_{ei} \cdot n_{ep}$)	t_{epk}	90	550	ms
9	Amount to Increment t_{epk}	t_{ei}	90	110	ms
Ten	Erase Margin = $\sum_{k=1}^{n_{ep}} t_{ei} \cdot k$	e_m	90	1650	ms
11	Erase Recovery Time	t_{er}	1	——	ms
12	Low-Power Stop Recovery Time , [7]	t_{sb}	1	——	∝ s

NOTES: NOTES:

1. All AC timing is shown with respect to 20% and 70% V_{DD} levels unless otherwise noted. For The Program As / Erase Operation, T_A Must Be In The Range Of 0 ° C To 70 ° C.
2. V_{FPE} must not be raised to programming voltage while V_{DD} is below specified minimum value. V_{FPE} must not be reduced below minimum specified value while V_{DD} is applied.
3. Current parameters apply to each individual EEPROM module.
4. Minimum software delay from the end of the write cycle that clears EHV bit to the read of the flash array
5. Without margin.
6. At 100% margin, the number of margin pulses required is the same as the number of pulses used to program the byte or word.
7. Minimum software delay from the end of the write cycle that clears the STOP bit to the read of the

flash array.

Table 181 HD Flash EEPROM Life

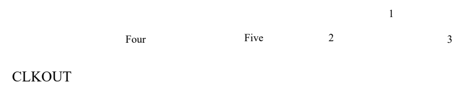
Num	Parameter	Symbol	Value Value.	Unit
1	Program-Erase Endurance [1]	e_{pe}	100	cyc

NOTES: NOTES:

1. Number of program-erase cycles (1 to 0, 0 to 1) per bit.

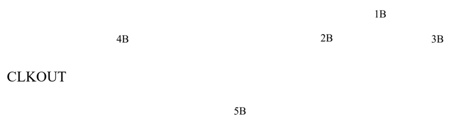
Timing diagram

--SLIM timing



TIMING SHOWN WITH RESPECT TO 20% AND 70% OF V_{DD} .

Figure 56 Timing diagram for CLKOUT output (see Table 164)



TIMING SHOWN WITH RESPECT TO 20% AND 70% OF V_{DD} . PULSE WIDTH SHOWN WITH RESPECT TO 50% V_{DD} .

Figure 57 Timing diagram of external clock input (see Table 164)

TIMING SHOWN WITH RESPECT TO 20% AND 70% OF V_{DD} .

Figure 58 Timing diagram of ECLK output (see Table 164)

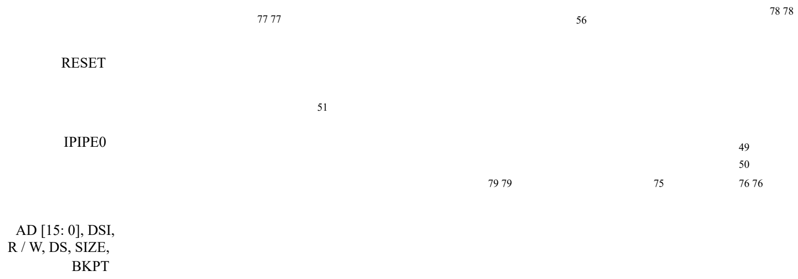


Figure 59 Reset and Mode Select Timing Diagram (see Table 164)

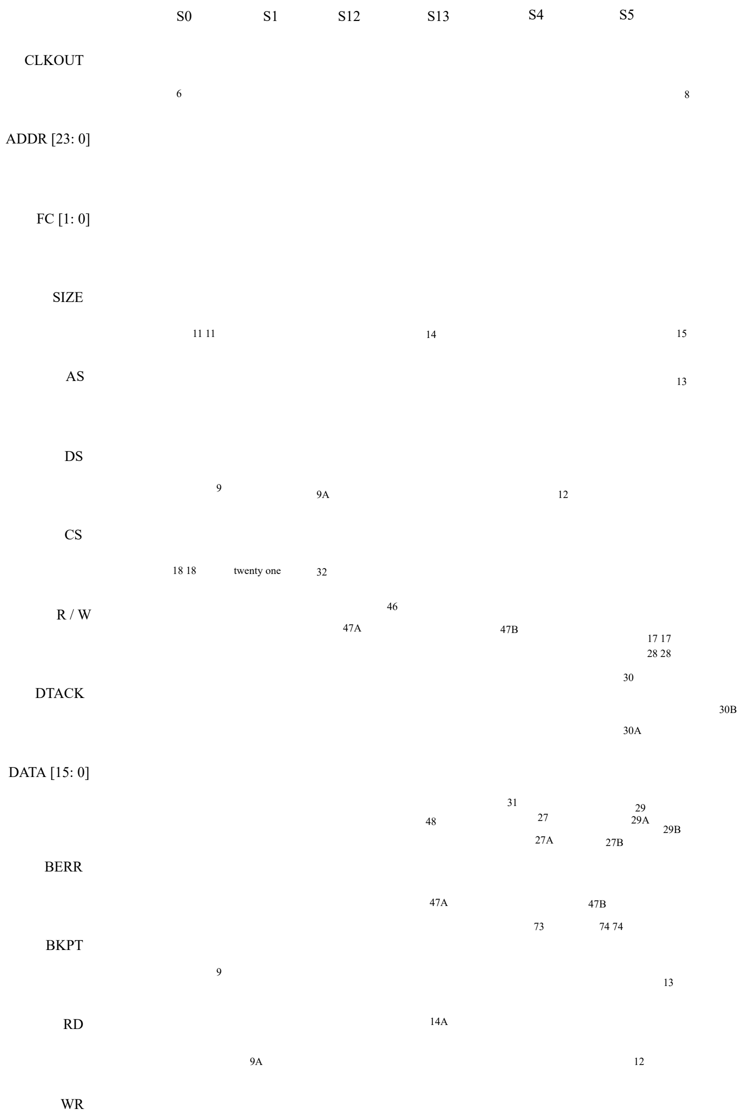
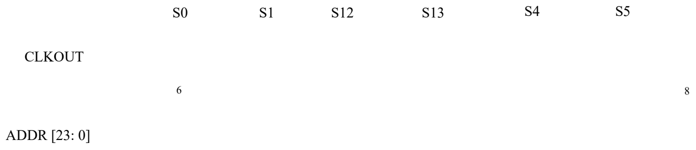


Figure 60 Non-multiplexed read cycle (see Table 164)



FC [1: 0]

SIZE

11 11 14 15

AS

13

DS

9 9 12

CS

20 twenty two 14A

17 17

R / W

46

32 47A 28 28

47B

DTACK

55 55

DATA [15: 0]

twenty three 26 47A twenty five 54

48 53

47B

27A 27B

BERR

73 74 74

BKPT

RD

12

9

13

WR

14A

Figure 61 Non-multiplexed write cycle (see Table 164)

S0 S1 S4 S5

CLKOUT

6 8

ADDR [23: 0]

FC [1: 0]

SIZE

11 11 14 15

AS

13

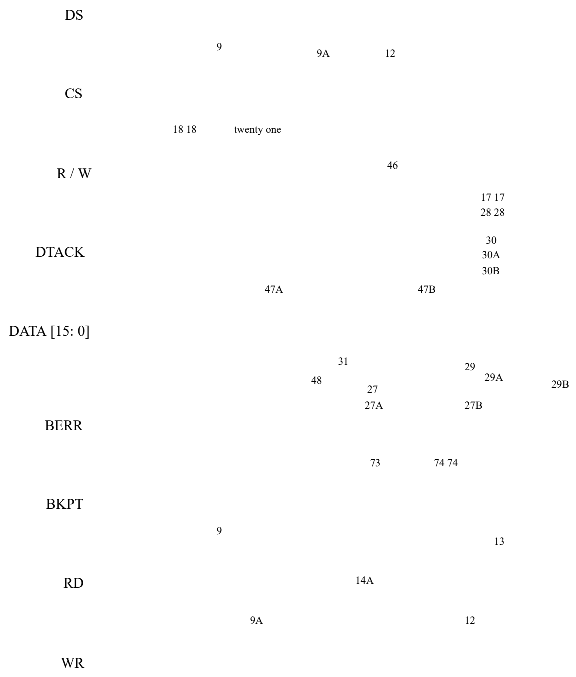
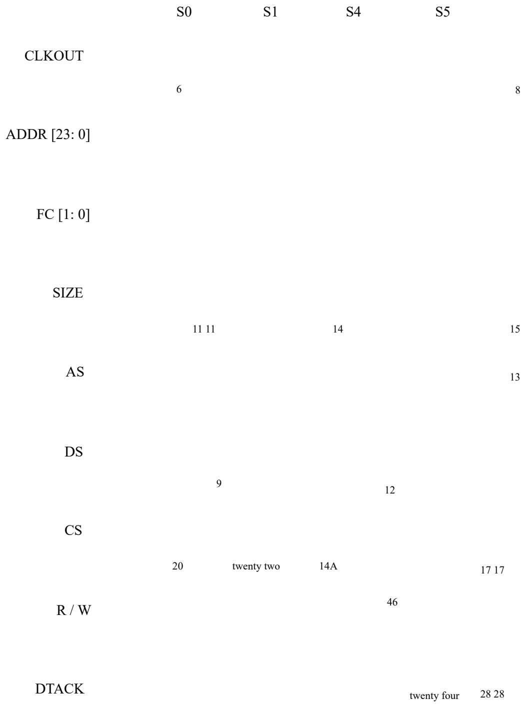


Figure 62 Demultiplexed 2 clock read cycle (see Table 164)



47A 47B

DATA [15: 0]

55 55
twenty three twenty five
48 53
27A 27B 54

BERR

73 74 74

BKPT

RD

9
12 13

14A

WR

Figure 63 Demultiplexed 2 clock write cycle (see Table 164)

S0 S1 S12 S13 S4 S5 S0 S1

CLKOUT

87
88
81 86 80 97 97B

AD [15: 0] ADDRESS DATA ADDRESS

83 84 84 97A

85 99
99A
ADDR [23:16],
FC [1: 0], SIZE 99B

66 66 13
82 8
12
ALE 91

14 15

AS

9
11 11 89 14A

DS 9

9
9B 90
9B

CS 9

18 18 twenty one 17 17

46
R / W 92

47A 31 47B
28 28
DTACK

32
28 28
27A 27B
BERR
47A 47B
73 74 74

BKPT
48
89 14A

RD

9

Figure 64 Multiplexed Read Cycle Timing Diagram (see Table 164)

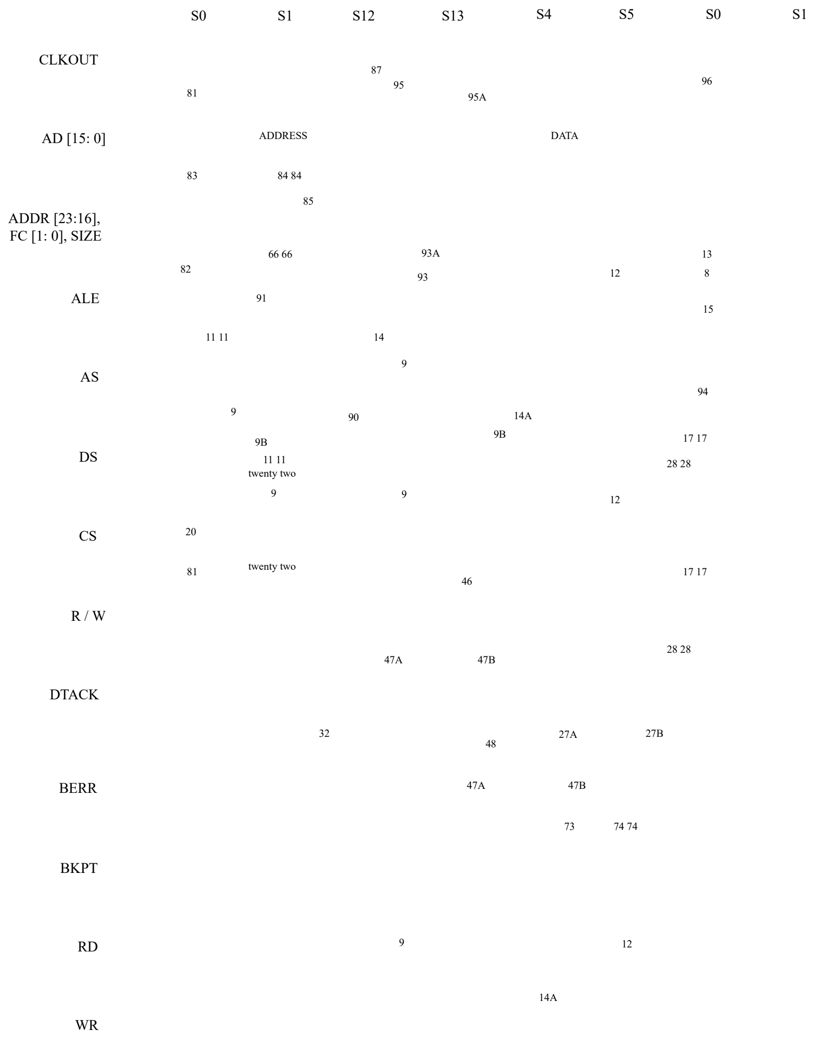
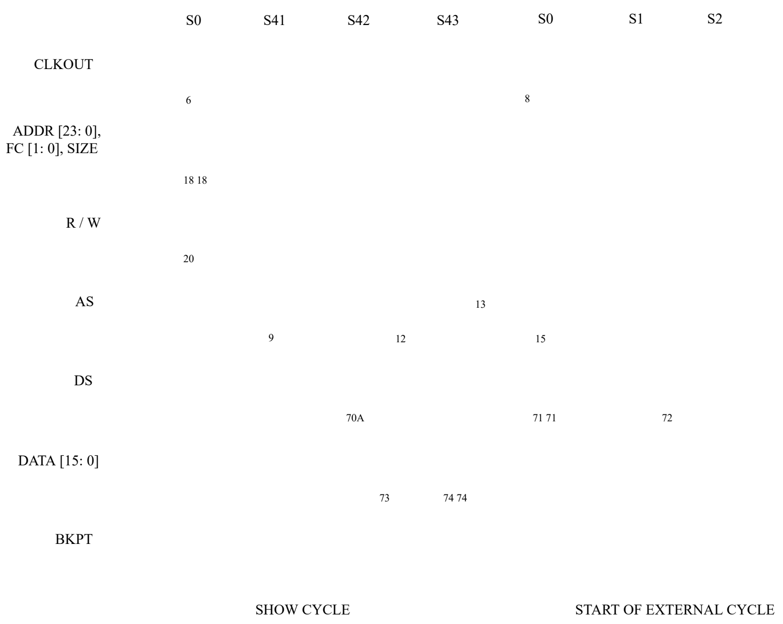


Figure 65 Timing diagram of the multiplexed write cycle (see Table 164).

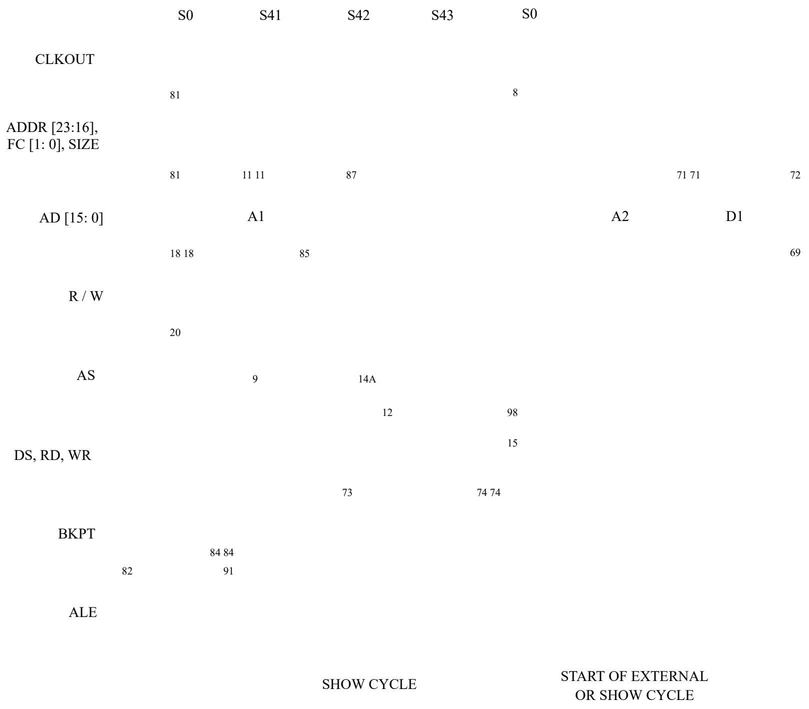


TIMING FOR LOGIC ANALYZER TRIGGER POINTS:

LATCH ADDR, FC, SIZE, R / W, AS, CS, ETC. ON FALLING EDGE OF DS.

LATCH DATA ON FALLING EDGE OF CLKOUT AFTER DS GOES HIGH.

Figure 66 Non-multiplexed 2 clock show cycle (see Table 164)

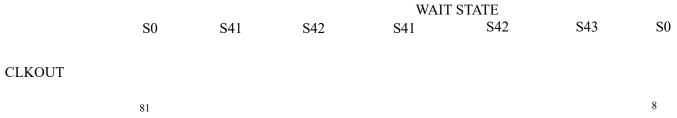


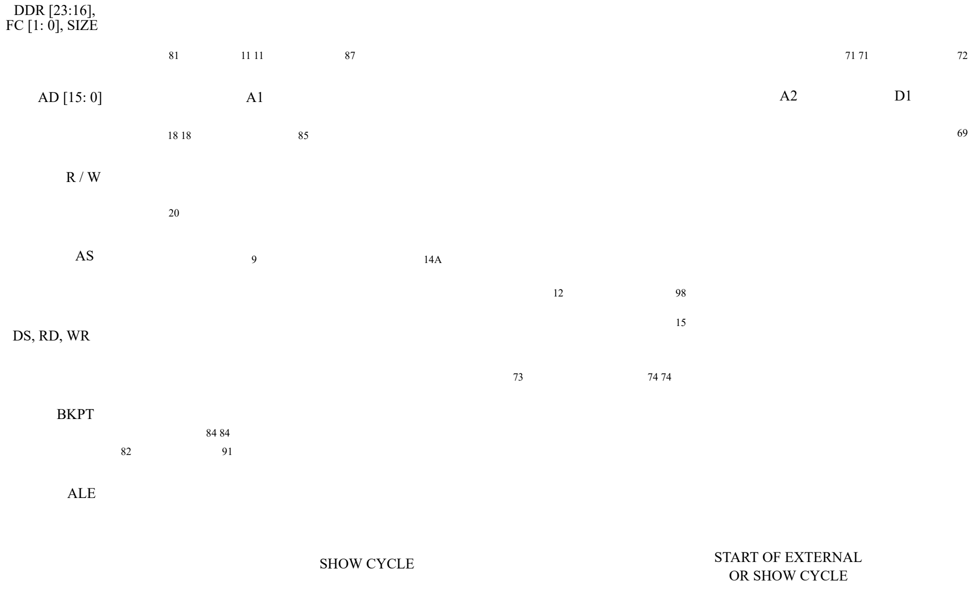
TIMING FOR LOGIC ANALYZER TRIGGER POINTS:

LATCH ADDR, SIZE, R / W, AS, CS, ETC. ON FALLING EDGE OF ALE OR DS.

LATCH DATA ON FALLING EDGE OF CLKOUT (SECOND FALLING EDGE AFTER DS GOES HIGH).

Figure 67 Multiplexed show cycle and next timing diagram (see Table 164)



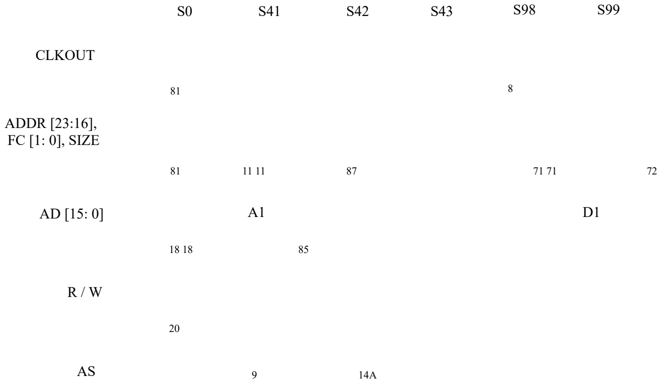


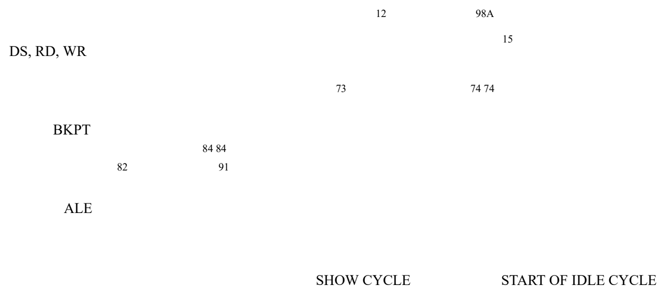
TIMING FOR LOGIC ANALYZER TRIGGER POINTS:

LATCH ADDR, SIZE, R / W, AS, CS, ETC. ON FALLING EDGE OF ALE OR DS.

LATCH DATA ON FALLING EDGE OF CLKOUT (SECOND FALLING EDGE AFTER DS GOES HIGH).

Figure 68 Timing diagram of a multiplexed show cycle with internal weight states (see Table 164).



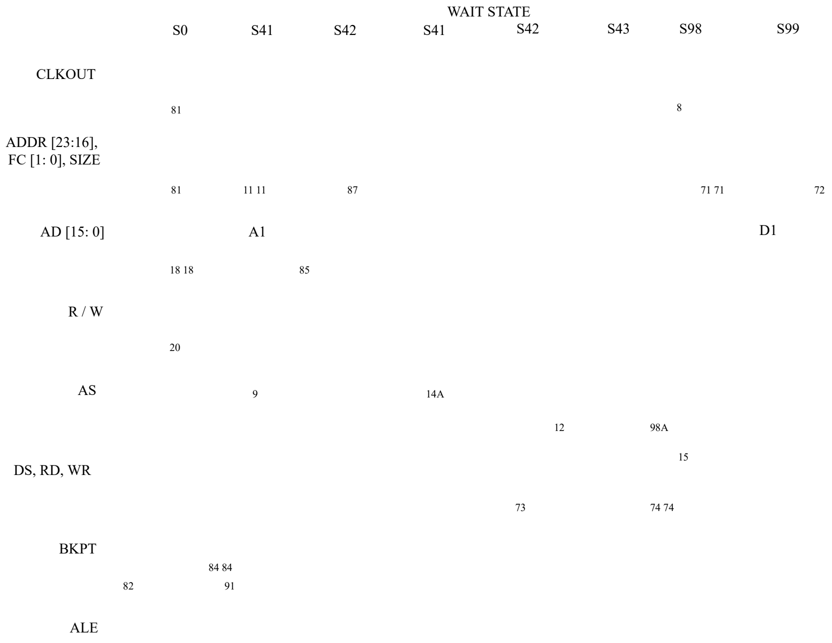


TIMING FOR LOGIC ANALYZER TRIGGER POINTS:

LATCH ADDR, SIZE, R / W, AS, CS, ETC. ON FALLING EDGE OF ALE OR DS.

LATCH DATA ON FALLING EDGE OF CLKOUT AFTER DS GOES HIGH.

Figure 69 Timing diagram of the multiplexed show cycle and the next idle cycle (see Table 164).



SHOW CYCLE START OF IDLE CYCLE

TIMING FOR LOGIC ANALYZER TRIGGER POINTS:

LATCH ADDR, SIZE, R / W, AS, CS, ETC. ON FALLING EDGE OF ALE OR DS.

LATCH DATA ON FALLING EDGE OF CLKOUT AFTER DS GOES HIGH.

Figure 70 Multiplexed with internal weight state by idle cycle
Show cycle timing diagram (see Table 164)

S0 S1 S4 S5

CLKOUT

ADDR [23: 0]

FC [1: 0]

SIZE

AS 16 16

DS

ALE

R / W

16 16

DATA [15: 0]

RD

WR 47A 7

EBR

Figure 71 EBR Timing-Active Bus Case Timing Diagram (see Table 164)

S0 S1

CLKOUT

ADDR [23: 0]

FC [1: 0]

SIZE

DATA [15: 0]

AS

DS

ALE

R / W

RD

WR

47A 47A

EBR

Figure 72 EBR Timing-Idle Bus Case Timing Diagram (see Table 164)

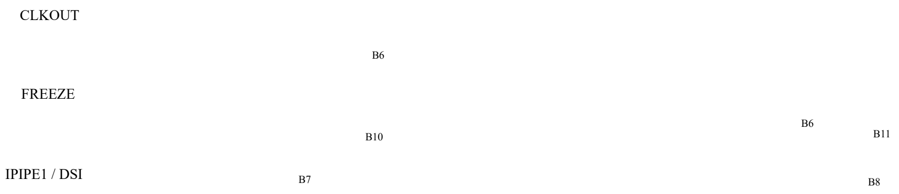


Figure 73 Background debug mode (assert / negate FREEZE)
Timing diagram (see Table 164)

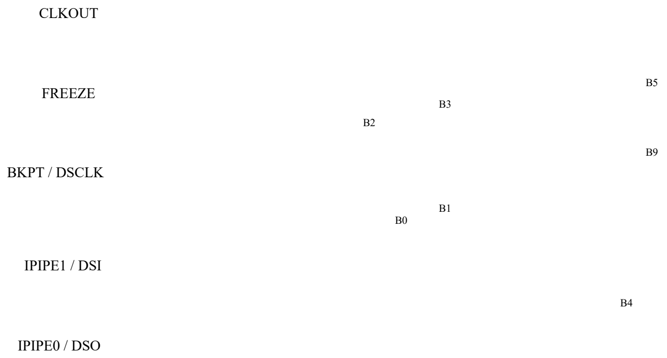


Figure 74 Background debug mode (serial communication)
Timing diagram (see Table 164)

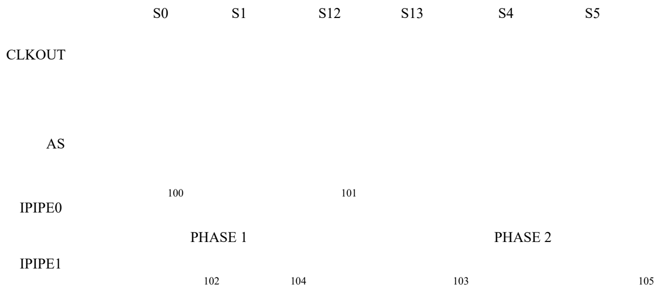


Figure 75 CPU16 pin (read / write cycle) timing diagram (see Table 164)

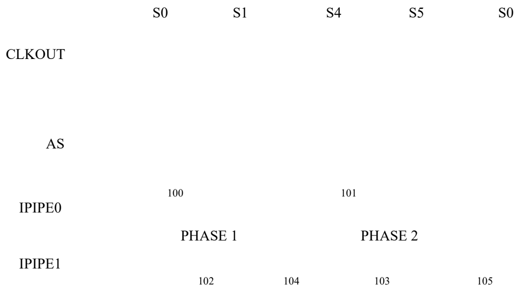
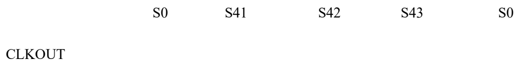


Figure 76 Timing diagram for CPU 16-pin (2-read / write cycle) (see Table 164).



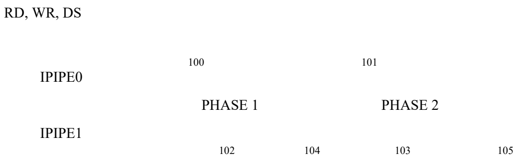


Figure 77 Timing diagram for CPU 16-pin (2 clock show cycles) (see Table 164).

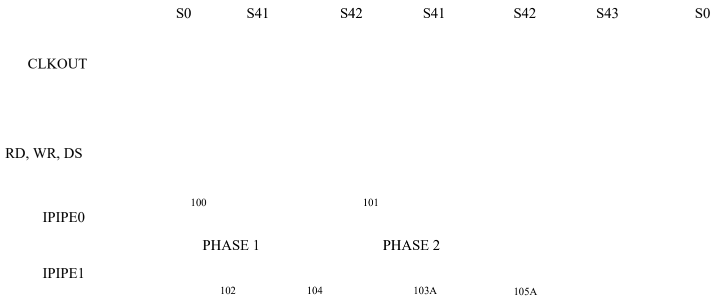
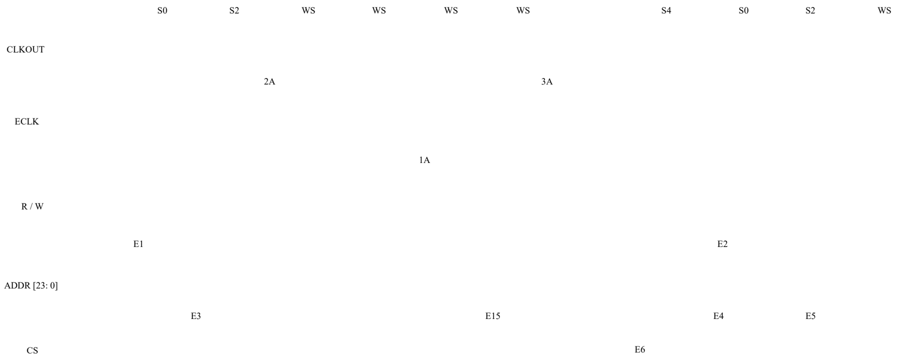


Figure 78 Timing diagram for CPU 16-pin (3 clock show cycles) (see Table 164).



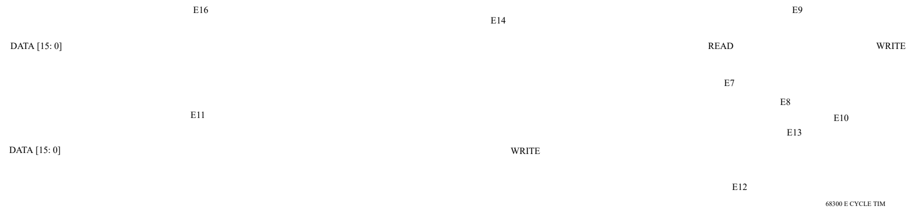
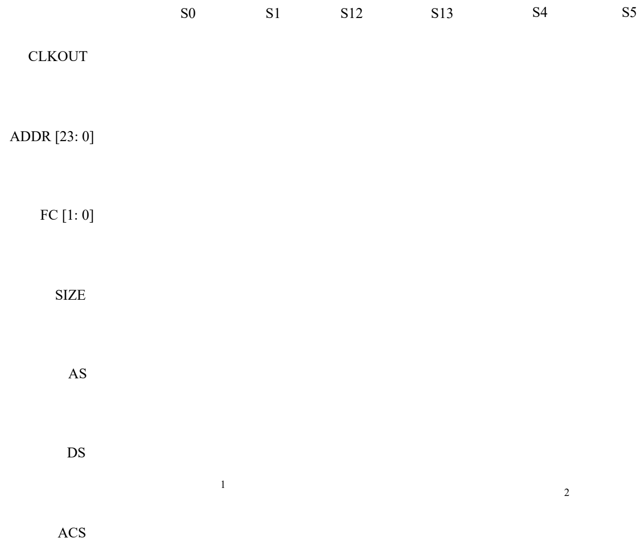


Figure 79 ECLK timing diagram (see Table 164)

--ACS timing



R / W

Figure 80 ACS non-multiplexed master read cycle (see Table 167).

Page 359

S0 S1 S12 S13 S4 S5

CLKOUT

ADDR [23: 0]

FC [1: 0]

SIZE

AS

DS

1 1 2

ACS

R / W

Figure 81 ACS non-multiplexed master write cycle (see Table 167).

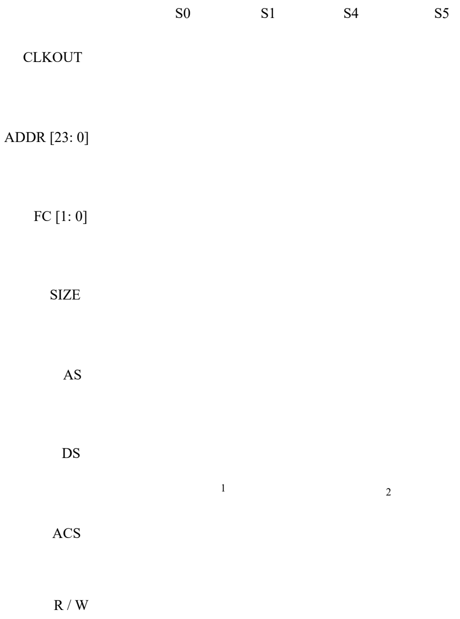


Figure 82 ACS non-multiplexed master 2-clock read cycle (see Table 167).

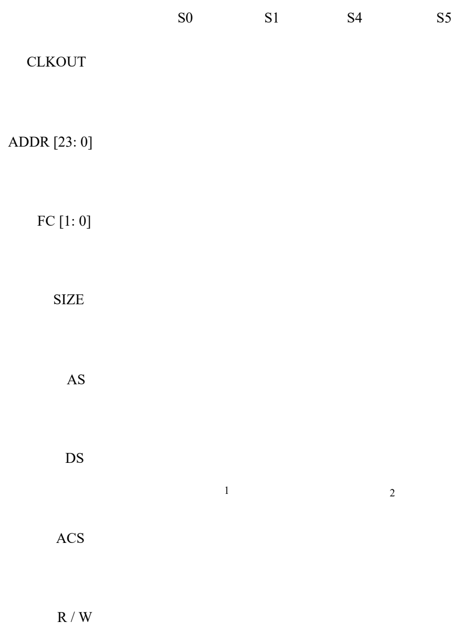


Figure 83 ACS non-multiplexed master 2-clock write cycle (see Table 167).

--SPI (MCCI) timing

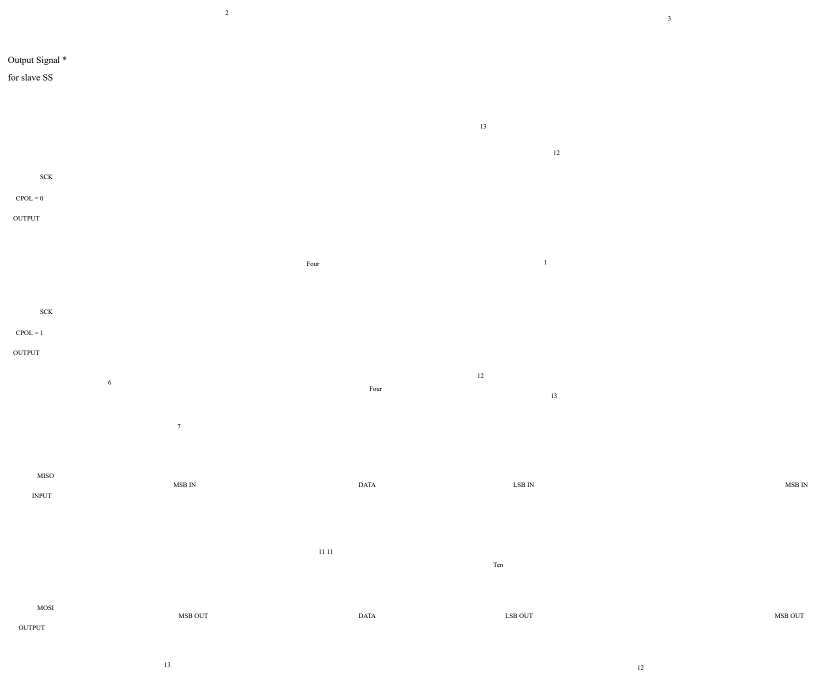
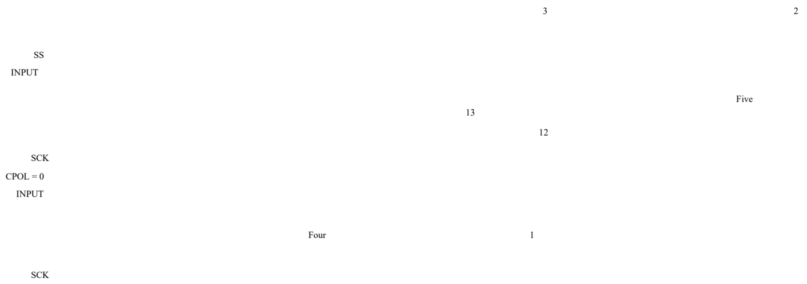


Figure 84 SPI Timing Master CPHA = 0 (see Table 168)

Note *: This signal must be generated on a general purpose port in master mode.

Figure 85 SPI Timing Master CPHA = 1 (see Table 168)

Note *: This signal must be generated on the general-purpose port in master mode, but if the communication destination is single, Communication is possible even if this signal is not used.



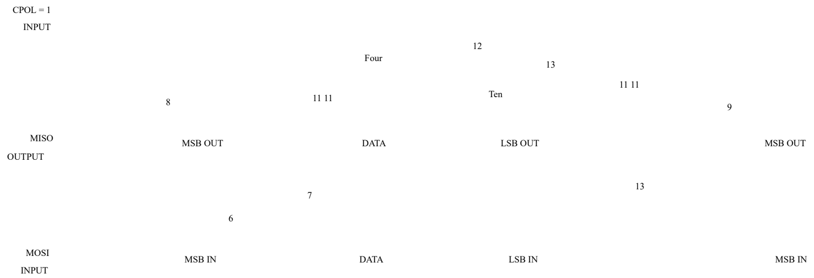


Figure 86 SPI Timing Slave CPHA = 0 (see Table 168)

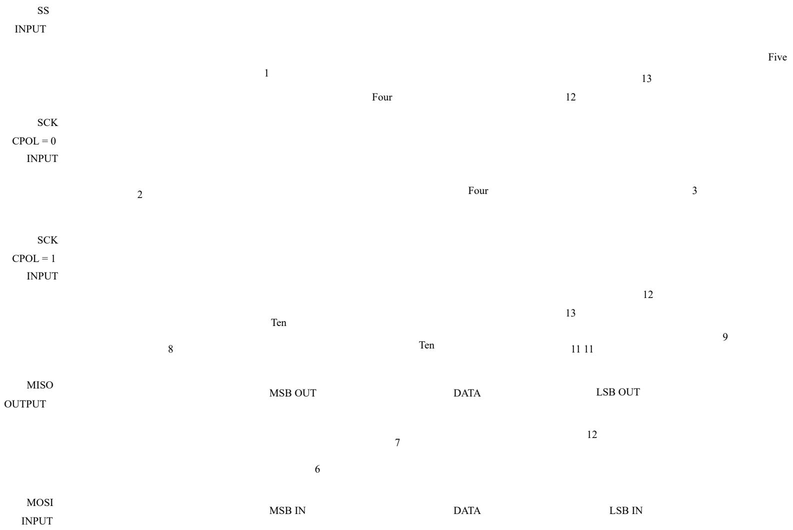


Figure 87 SPI Timing Slave CPHA = 1 (see Table 168)

--RSPI AC timing

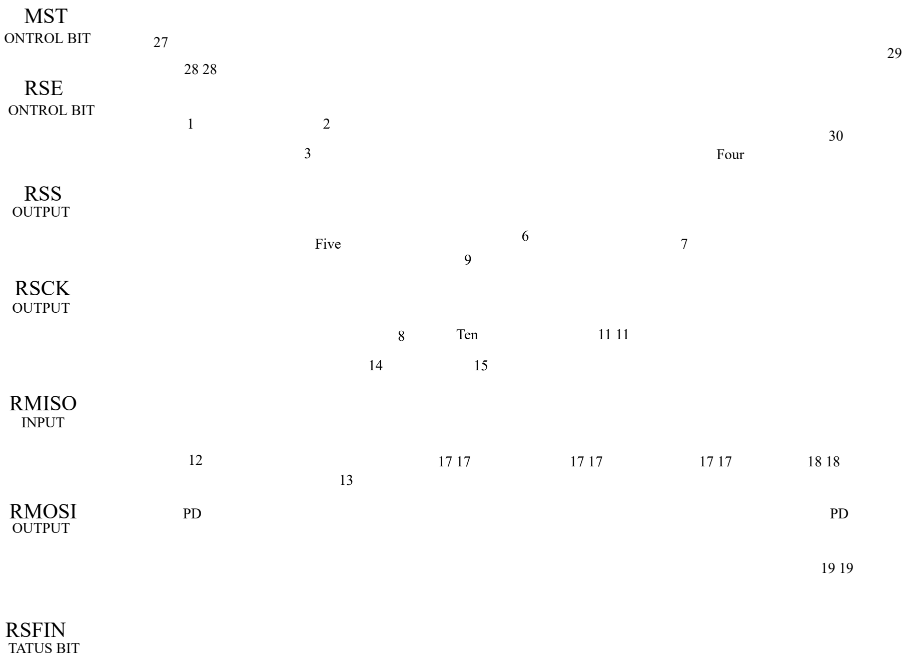


Figure 88 RSPI AC Timing (Master Mode) (see Table 169)
Address Slave Select Mode (SSM = 1)

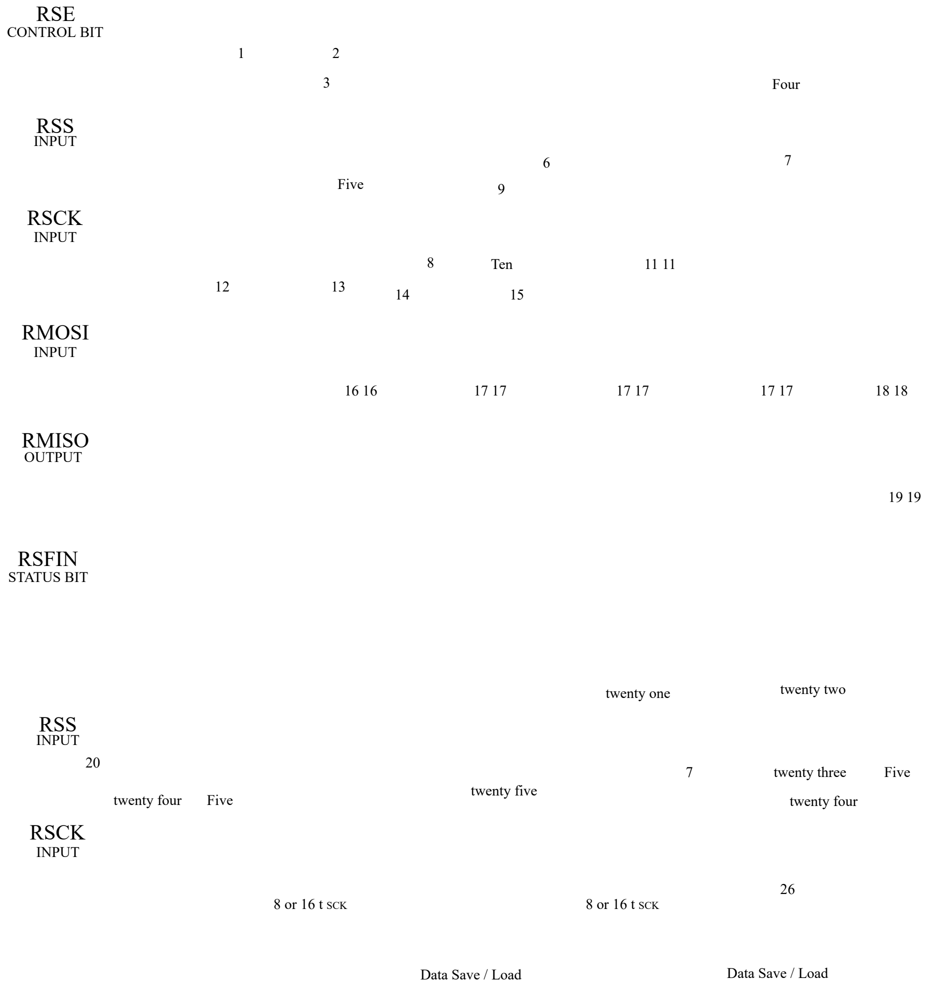


Figure 89 RSPI AC Timing (Slave Mode) (see Table 170)
Address Slave Select Mode (SSM = 1)

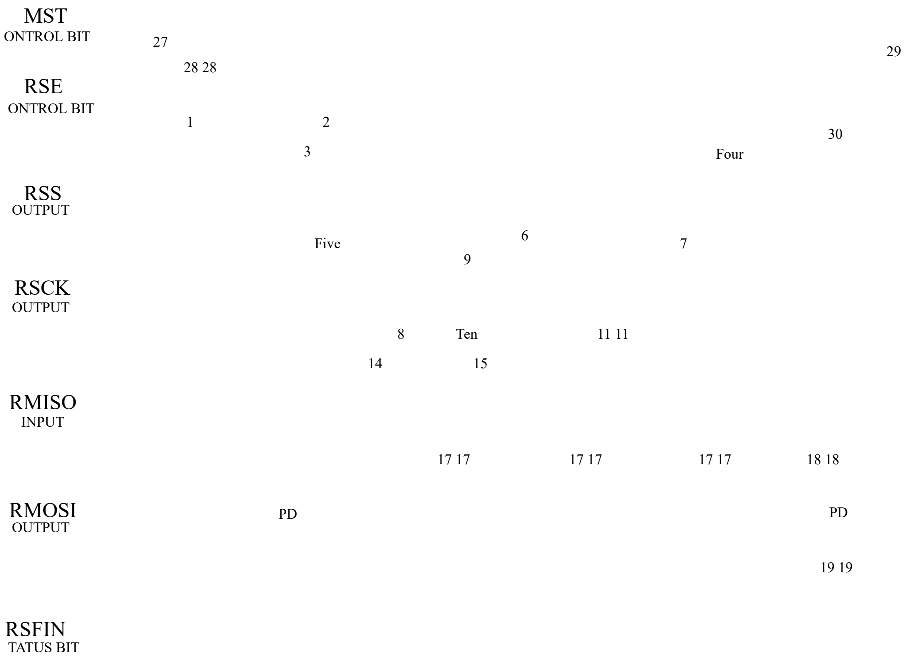


Figure 90 RSPI AC Timing (Master Mode) (see Table 169)
Address Slave Select Mode (SSM = 0)

Page 369

RSE
CONTROL BIT

1 2

3 Four

RSS
INPUT

6 7

Five 9

RSCK
INPUT

8 Ten 11 11

12 13 14 15

RMOSI
INPUT

16 16 17 17 17 17 17 17 18 18

RMISO
OUTPUT

PD

19 19

RSFIN
STATUS BIT

Figure 91 RSPI AC Timing (Slave Mode) (see Table 170)
Address Slave Select Mode (SSM = 0)

Page 370

--CTM timing

If there is no write to the counter register:

Clock

Counter register $ 1025 $ 1026 $ 1027 $ 1028

Time base bus $ 1025 $ 1026 $ 1027 $ 1028

If there is a write to the counter register:

Clock

Counter register $ 1025 $ 1026 $ 5000 $ 5001 $ 5002

Time base bus $ 1025 $ 1026 $ 5001 $ 5002

write $ 5000 to counter register here

Clock

Counter register $ 1025 $ 5000 $ 5001 $ 5002

Time base bus $ 1025 $ 5000 $ 5001 $ 5002

write $ 5000 to counter register here

Figure 92 MCSM Timebase Timing Diagram (see Table 175)

t_{PTBB}

t_{PCOF}

t_{PINC}

Clock

Figure 93 MCSM clock pin to counter timing diagram (see Table 175).

Figure 94 Timing diagram from the MCSM LOAD pin to the counter (see Table 175).

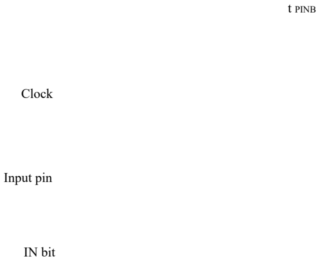


Figure 95 Timing diagram from MCSM pin to IN bit (see Table 175)

Figure 96 Timing diagram from the COF bit of MCSM to the interrupt request (see Table 175).

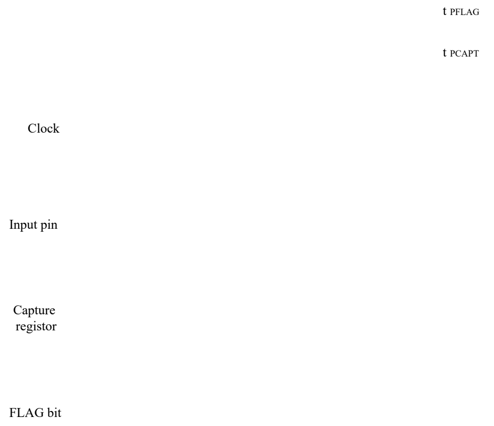


Figure 97 DASM Input Capture Timing Diagram (see Table 176)

Input pin

IN bit

Figure 98 Timing diagram from DASM pin to IN bit (see Table 176)

t_{CPIN}

t_{CFLAG}

Clock

Input pin

FLAG bit

Output pin

Figure 99 DASM Output Compare Timing Diagram (see Table 176)

t_{FIRQ}

Clock

FLAG bit

SMB IRQx

IMB IRQx

Figure 100 Timing diagram from the FLAG bit of DASM to the interrupt request (see Table 176).

Figure 101 PWMSM minimum output pulse timing diagram (see Table 177)

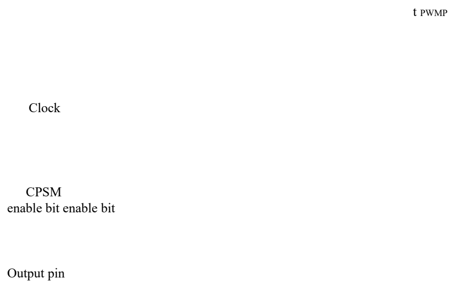


Figure 102 Set of PWM outputs from CPSM enable for PWMSM (see Table 177).

t PWME

Clock

PWMSM
enable bit enable bit

Output pin

Figure 103 Timing diagram from enabling PWMSM to setting the output (see Table 177).

t FIRQ

Clock

FLAG bit

SMB IRQx

IMB IRQx

Figure 104 Timing diagram from the FLAG bit of PWMSM to the interrupt request (see Table 177).

--QADC conversion accuracy diagram

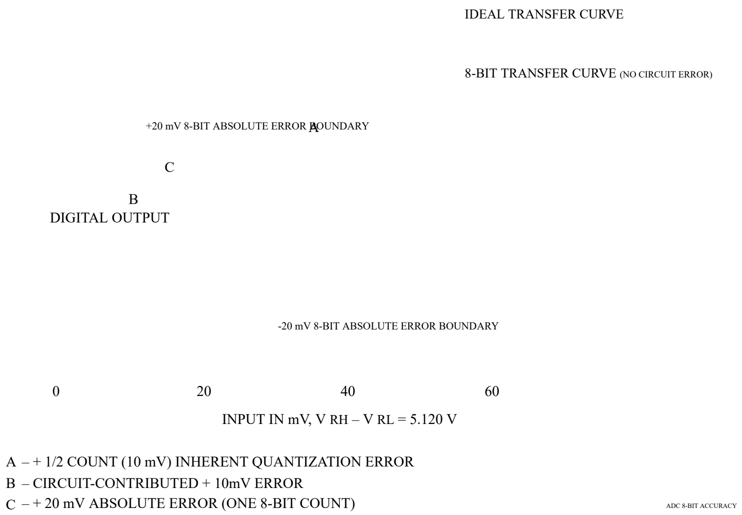


Figure 105 8-Bit QADC conversion accuracy (see Table 174)

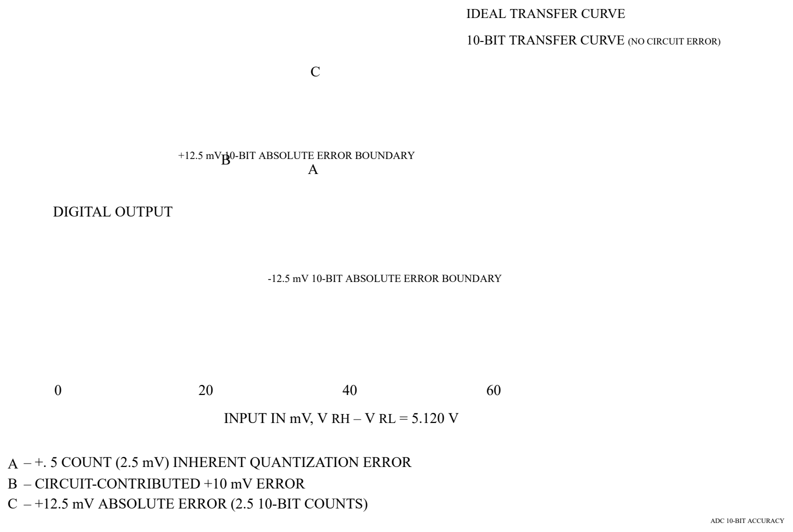
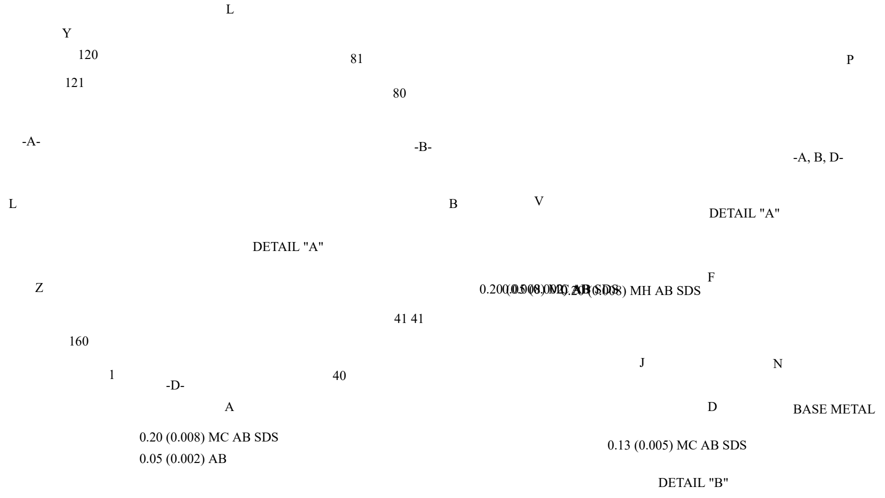
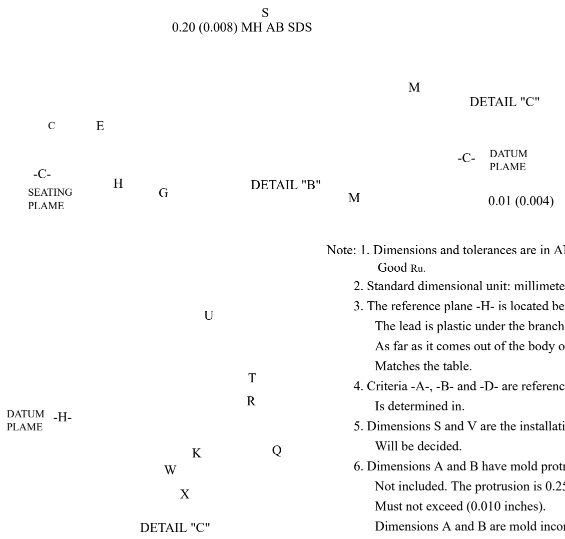


Figure 106 10-Bit QADC conversion accuracy (see Table 174)

Appendix B
Package outline dimensions





Note: 1. Dimensions and tolerances are in ANSI Y14.5M, 1982
Good Ru.
2. Standard dimensional unit: millimeter
3. The reference plane -H- is located below the lead
The lead is plastic under the branch line
As far as it comes out of the body of the package
Matches the table.
4. Criteria -A-, -B- and -D- are reference planes -H-
Is determined in.
5. Dimensions S and V are the installation surface -C- odor
Will be decided.
6. Dimensions A and B have mold protrusions
Not included. The protrusion is 0.25 mm on one side
Must not exceed (0.010 inches).
Dimensions A and B are mold inconsistencies
Not included and determined by the reference plane -H-
Is done.
7. Dimension D does not include damper protrusions. Da
The protrusions of the number are in total under the maximum material conditions.
Dimension D is 0.08 mm (0.003 inch)
Do not cross.

Size	Millimeter		inch	
	minimum	maximum	minimum	maximum
A	27.90	28.10	1.098	1.106
B	27.90	28.10	1.098	1.106
C	3.35	3.85	0.132	0.152
D	0.22	0.38	0.009	0.015
E	3.20	3.50	0.126	0.138
F	0.22	0.33	0.009	0.013
G	0.650BSC		0.0256BSC	
H	0.25	0.35	0.010	0.012
J	0.11	0.23	0.004	0.009
K	0.70	0.90	0.028	0.035
L	25.35REF		0.998REF	
M	5 °	16 °	5 °	16 °
N	0.11	0.19	0.004	0.007
P	0.325BSC		0.013BSC	
Q	0 °	7 °	0 °	7 °
R	0.13	0.30	0.005	0.012
S	31.00	31.40	1.220	1.236
T	0.13	———	0.005	———
U	0 °	———	0 °	———
V	31.00	31.40	1.220	1.236
W	0.40	———	0.016	———
X	1.60REF		0.050REF	
Y	1.33REF		0.052REF	
Z	1.33REF		0.052REF	

CASE684A-01

Figure 107 160-pin QFP package dimensions

We have the right to make changes to the products described in this document without notice to improve reliability, functionality or design.
I have a limit. We assume no liability for the application or use of the products and circuits described herein.
Nor does it grant a license based on our patent rights or the rights of third parties. Made by our company
The item is intended as a component of a system intended to be surgically transplanted into the human body, or as any other life-sustaining component.
There will be situations where the application or the inconvenience of our products may cause harm or even death to the human body.
It is not designed, intended or licensed for use in such applications. By any chance the purchaser is this
If you purchase or use our products for such intent or unauthorized use, you may purchase us and us.
Injuries related to its use, directly or indirectly, to its officers, employees, subsidiaries, affiliates and agents
And all allegations of death (even with allegations that we were careless in designing and manufacturing parts)
We will cover all claims, costs, damages, and considerable attorney's fees resulting from (if any) and will not cause any damage.
I will do it.

"Motorola" and Is a registered trademark of Motorola. We want to give everyone equal employment opportunities
It is a company that is making an effort.