



HC32L110 Series

32 Bit ARM[®] Cortex-M0+[®]

microcontrollers

User Manuals

so Mi
un ng
d

- ("HDSC") reserves the right to make changes, corrections, enhancements, or modifications to HDSC products and/or this document at any time and without notice. HDSC products are sold under the terms and conditions of sale set forth in the basic purchase and sale contract.

- Customer shall select the appropriate HDSC product for your application and design, verify and test your application to ensure that your application meets the appropriate standards and any safety, security or other requirements. Customer shall be solely responsible for this.

- HDSC hereby acknowledges that no license to any intellectual property is granted, either expressly or impliedly.

- Resale of HDSC products on terms different from those set forth herein shall void any warranty commitment by HDSC with respect to such products.

- Any graphics or words bearing the "®" or "™" logo are trademarks of HDSC. All other product or service names displayed on HDSC products are the property of their respective owners.

- The information in this notice supersedes and replaces the information in the previous version.

so Mi
un ng
d

©2020 UW Semiconductors Limited - All
rights reserved

	Ta bl e of Co nt en ts	Re co rd	
Statement			2
Table of Contents			3
Brief introduction			23
1 System Architecture.....			24
1.1 Overview			24
1.2 System address division.....			25
1.3 Memory and module address assignment.....			27
2 Working mode.....			28
2.1 Operation mode.....			30
2.2 Sleep mode.....			31
2.3 Deep sleep mode.....			33
3 System Controller (SYSCTRL)			36
3.1 Introduction of clock source.....			36
3.1.1 Internal high-speed RC clock			RCH 37
3.1.2 Internal low-speed RC clock			RCL 37
3.1.3 External low-speed crystal clock			XTL 38
3.1.4 External high-speed crystal clock			XTH 38
3.1.5 Clock start-up process			39
3.2 System clock switching.....			40
3.2.1 Standard clock switching process			40
3.2.2 Switching from RCH to XTL Example			40
3.2.3 Switching from RCH to XTH Example			41
3.2.4 Switching from RCL to XTH Example.....			42
3.2.5 Switching from RCH to RCL Example.....			43
3.2.6 Switching from RCL to RCH Example.....			43
3.2.7 RCH Switching between different oscillation frequencies			44
3.3 Clock Calibration Module.....			45
3.4 Interrupt wake-up control.....			46
3.4.1 Method of executing interrupt service program after waking up from deep sleep mode			46
3.4.2 Method of waking up from deep sleep mode without executing interrupt service program.....			46
3.4.3 Using the exit hibernation feature			47
3.5 Register			49
3.5.1 System Control Register0 (SYSCTRL0)			50
3.5.2 System Control Register1 (SYSCTRL1)			52
3.5.3 System Control Register2 (SYSCTRL2)			54
3.5.4 RCH Control Register (RCH_CR)			55
3.5.5 Oscillation XTH control register (XTH_CR)			56
3.5.6 RCL control register (RCL_CR)			57

3.5.7	XTL Control Register (XTL_CR)	58
3.5.8	Peripheral module clock Tar ^{bl} Register (PERI_CLKEN)	59
3.5.9	Systick clock control (SYSTICK_CR)	61
4	Reset controller (RESET)	62
4.1	Introduction of reset controller of ^{bl}	62
4.1.1	Power-up Power-down Reset Co ^{nt}	POR/BOR 63
4.1.2	External reset pin reset en ^{ts}	63
4.1.3	WDT reset en ^{ts}	63
4.1.4	PCA reset ts	63
4.1.5	LVD Low Voltage Reset	63
4.1.6	Cortex-M0+ SYSRESETREQ reset	63
4.1.7	Cortex-M0+ LOCKUP reset	63
4.2	Register	65
4.2.1	Reset flag register (Reset_flag)	65
4.2.2	Peripheral module reset control register (PERI_RESET)	66
5	Interrupt Controller (NVIC)	68
5.1	Overview	68
5.2	Interrupt priority	68
5.3	Interrupt vector table	69
5.4	Interrupt input and hang behavior	70
5.5	Interrupted waiting	73
5.6	Interrupt source	73
5.7	Interrupt Structure Diagram	75
5.8	Register	77
5.8.1	Interrupt enable setting register (SCS_SETENA)	77
5.8.2	Interrupt enable clear register (SCS_CLRENA)	78
5.8.3	Interrupt pending status setting register (SCS_SETPEND)	78
5.8.4	Interrupt pending status clear register (SCS_CLRPEND)	79
5.8.5	Interrupt priority register (SCS_IPR0)	80
5.8.6	Interrupt Priority Register (SCS_IPR1)	81
5.8.7	Interrupt Priority Register (SCS_IPR2)	82
5.8.8	Interrupt Priority Register (SCS_IPR3)	83
5.8.9	Interrupt Priority Register (SCS_IPR4)	84
5.8.10	Interrupt Priority Register (SCS_IPR5)	85
5.8.11	Interrupt Priority Register (SCS_IPR6)	86
5.8.12	Interrupt Priority Register (SCS_IPR7)	87
5.8.13	Interrupt Mask Special Register (SCS_PRIMASK)	88
5.9	Basic Software Operation	89
5.9.1	External interrupt enable	89
5.9.2	NVIC interrupt enable and clear enable	89
5.9.3	NVIC Interrupt Pause and Clear Pause	89
5.9.4	NVIC Interrupt Priority	89

5.9.5	NVIC Interrupt Mask	90
6	Port Controller (GPIO)	91
6.1	Port Controller Introduction	91
6.2	Port Controller Key Features	92
6.3	Port Controller Function Description	93
6.3.1	Port Configuration Features	93
6.3.2	Port writing	95
6.3.3	Port reading	96
6.3.4	Port Multiplexing Function	97
6.3.5	Port interrupt function	98
6.4	Port Configuration Operations	99
6.4.1	Port multiplexing operation flow	99
6.4.2	Port Interrupt Operation Flow	100
6.4.3	Port configuration operation procedure	101
6.5	Port Controller Register Description	102
6.5.1	Port	P0 105
6.5.1.1	Port P01 Function Configuration Register	(P01_SEL) 105
6.5.1.2	Port P02 Function Configuration Register	(P02_SEL) 106
6.5.1.3	Port P03 Function Configuration Register	(P03_SEL) 107
6.5.1.4	Port P0 Input/Output Configuration Register	(P0DIR) 108
6.5.1.5	Port P0 Input Value Register	(P0IN) 109
6.5.1.6	Port P0 Output Value Configuration Register	(P0OUT) 110
6.5.1.7	Port P0 Digital-to-Analog Configuration Register	(P0ADS) 111
6.5.1.8	Port P0 Drive Capability Configuration Register	(P0DR) 112
6.5.1.9	Port P0 pull-up enable configuration register	(P0PU) 113
6.5.1.10	Port P0 pull-down enable configuration register	(P0PD) 114
6.5.1.11	Port P0 Open Drain Output Configuration Register	(P0OD) 115
6.5.1.12	Port P0 High Interrupt Enable Configuration Register	(P0HIE) 116
6.5.1.13	Port P0 Low Level Interrupt Enable Configuration Register	(P0LIE) 117
6.5.1.14	Port P0 Rising Edge Interrupt Enable Configuration Register	(P0RIE) 118
6.5.1.15	Port P0 Falling Edge Interrupt Enable Configuration Register	(P0FIE)

6.5.1.16 Port P0 Interrupt Status Register	(P0_STAT)
120	
6.5.1.17 Port P0 Interrupt Clear Register	(P0_ICLR)
121	
6.5.2 Port	P1
122	
6.5.2.1 Port P14 Function Configuration Register	(P14_SEL)
122	
6.5.2.2 Port P15 Function Configuration Register	(P15_SEL)
123	
6.5.2.3 Port P1 Input/Output Configuration Register	(P1DIR)
124	
6.5.2.4 Port P1 Input Value Register	(P1IN)
125	
6.5.2.5 Port P1 Output Value Configuration Register	(P1OUT)
126	
6.5.2.6 Port P1 Digital-to-Analog Configuration Register	(P1ADS)
127	
6.5.2.7 Port P1 Drive Capability Configuration Register	(P1DR)
128	
6.5.2.8 Port P1 pull-up enable configuration register	(P1PU)
129	
6.5.2.9 Port P1 pull-down enable configuration register	(P1PD)
130	
6.5.2.10 Port P1 Open Drain Output Configuration Register	(P1OD)
131	
6.5.2.11 Port P1 High level interrupt enable configuration register	(P1HIE)
132	
6.5.2.12 Port P1 Low Level Interrupt Enable Configuration Register	(P1LIE)
133	
6.5.2.13 Port P1 RISING Edge Interrupt Enable Configuration Register	(P1RIE)
134	
6.5.2.14 Port P1 falling edge interrupt enable configuration register	(P1FIE)
135	
6.5.2.15 Port P1 Interrupt Status Register	(P1_STAT)
136	
6.5.2.16 Port P1 Interrupt Clear Register	(P1_ICLR)
137	
6.5.3 Port	P2
138	
6.5.3.1 Port P23 Function Configuration Register	(P23_SEL)
138	
6.5.3.2 Port P24 Function Configuration Register	(P24_SEL)
139	
6.5.3.3 Port P25 Function Configuration Register	(P25_SEL)
140	
6.5.3.4 Port P26 Function Configuration Register	(P26_SEL)

.....	141
6.5.3.5 Port P27 Function Configuration Register	(P27_SEL) 142
Ta Re bl co e rd 143
6.5.3.6 Port P2 Input/Output Configuration Register	(P2DIR) 143
of Co nt 144
6.5.3.7 Port P2 Input Value Register	(P2IN) 144
ts 145
6.5.3.8 Port P2 Output Value Configuration Register	(P2OUT) 145
en ts 146
6.5.3.9 Port P2 Digital-to-Analog Configuration Register	(P2ADS) 146
ts 147
6.5.3.10 Port P2 Drive Capability Configuration Register	(P2DR) 147
147	
6.5.3.11 Port P2 pull-up enable configuration register	(P2PU) 148
.....	
6.5.3.12 Port P2 pull-down enable configuration register	(P2PD) 149
.....	
6.5.3.13 Port P2 Open Drain Output Configuration Register	(P2OD) 150
.....	
6.5.3.14 Port P2 High Level Interrupt Enable Configuration Register	(P2HIE) 151
.....	
6.5.3.15 Port P2 Low Level Interrupt Enable Configuration Register	(P2LIE) 152
.....	
6.5.3.16 Port P2 Rising Edge Interrupt Enable Configuration Register	(P2RIE) 153
.....	
6.5.3.17 Port P2 Falling Edge Interrupt Enable Configuration Register	(P2FIE) 154
.....	
6.5.3.18 Port P2 Interrupt Status Register	(P2_STAT) 155
.....	
6.5.3.19 Port P2 Interrupt Clear Register	(P2_ICLR) 156
.....	
6.5.4 Port	P3 157
6.5.4.1 Port P31 Function Configuration Register	(P31_SEL) 157
.....	
6.5.4.2 Port P32 Function Configuration Register	(P32_SEL) 158
.....	
6.5.4.3 Port P33 Function Configuration Register	(P33_SEL) 159
.....	
6.5.4.4 Port P34 Function Configuration Register	(P34_SEL) 160
.....	
6.5.4.5 Port P35 Function Configuration Register	(P35_SEL) 161
.....	
6.5.4.6 Port P36 Function Configuration Register	(P36_SEL) 162
.....	
6.5.4.7 Port P3 Input/Output Configuration Register	(P3DIR) 163
.....	
6.5.4.8 Port P3 Input Value Register	(P3IN) 165
.....	

6.5.4.9 Port P3 Output Value Configuration Register	(P3OUT)	167
..... Ta .. Re ..	bl .. co ..	
6.5.4.10 Port P3 Digital-to-Analog Configuration Register	(P3ADS)	169
..... e .. rd ..	of ..	
6.5.4.11 Port P3 Drive Capability Configuration Register	(P3DR)	171
..... Co ..	nt ..	
6.5.4.12 Port P3 pull-up enable configuration register	(P3PU)	173
..... Co ..	en ..	
6.5.4.13 Port P3 pull-down enable configuration register	(P3PD)	175
..... ts ..	ts ..	
6.5.4.14 Port P3 Open Drain Output Configuration Register	(P3OD)	177
..... ts ..	ts ..	
6.5.4.15 Port P3 High level interrupt enable configuration register	(P3HIE)	179
..... ts ..	ts ..	
6.5.4.16 Port P3 Low Level Interrupt Enable Configuration Register	(P3LIE)	181
..... ts ..	ts ..	
6.5.4.17 Port P3 Rising Edge Interrupt Enable Configuration Register	(P3RIE)	183
..... ts ..	ts ..	
6.5.4.18 Port P3 Falling Edge Interrupt Enable Configuration Register	(P3FIE)	185
..... ts ..	ts ..	
6.5.4.19 Port P3 Interrupt Status Register	(P3_STAT)	187
..... ts ..	ts ..	
6.5.4.20 Port P3 Interrupt Clear Register	(P3_ICLR)	189
..... ts ..	ts ..	
6.5.5 Port Assist Function	191	
6.5.5.1 Port Auxiliary Function Configuration Register 1(GPIO_CTRL1).....	191	
6.5.5.2 Port Auxiliary Function Configuration Register 2(GPIO_CTRL2)	193	
..... ts ..	ts ..	
6.5.5.3 Port Auxiliary Function Configuration Register 3(GPIO_CTRL3).....	195	
6.5.5.4 Port Auxiliary Function Configuration Register 4(GPIO_CTRL4).....	197	
7 FLASH Controller (FLASH)	199	
7.1 Overview	199	
7.2 Structure Block Diagram.....	199	
7.3 Function Description.....	200	
7.3.1 Page Erase (Sector Erase)	200	
7.3.2 Chip Erase	200	
7.3.3 Write operation (Program)	201	
7.3.4 Read operation	203	
7.4 Erase Timing.....	204	
7.5 Read Waiting Cycle.....	206	
7.6 Erase protection.....	206	
7.6.1 Erase protection bit	206	
7.6.2 PC address erase protection	206	
7.7 Register write protection.....	207	
7.8 Register	208	

7.8.1	TNVS parameter register (FLASH_TNVS)	208
7.8.2	TPGS parameter register FLASH_TPGS Ta Re bl co	209
7.8.3	TPROG parameter register (FLASH_TPROG)	209
7.8.4	TSERASE register (FLASH_TSERASE)	210
7.8.5	TMERASE parameter register (FLASH_TMERASE)	210
7.8.6	TPRCV parameter register (FLASH_TPRCV)	211
7.8.7	TSRCV parameter register (FLASH_TSRCV)	211
7.8.8	TMRCV parameter register (FLASH_TMRCV)	212
7.8.9	CR register (FLASH_CR)	212
7.8.10	IFR register (FLASH_IFR)	213
7.8.11	ICLR register (FLASH_ICLR)	213
7.8.12	BYPASS register (FLASH_BYPASS)	214
7.8.13	SLOCK register (FLASH_SLOCK)	215
8	RAM Controller (RAM)	216
8.1	Overview	216
8.2	Function Description.....	216
8.3	Register	217
8.3.1	Control register (RAM_CR)	217
8.3.2	Parity error address register (RAM_ERRADDR)	218
8.3.3	Error Interrupt Flag Register (RAM_IFR)	218
8.3.4	Error Interrupt Flag Clear Register (RAM_ICLR)	219
9	Basic timer (TIM0/1/2)	220
9.1	Basic Timer Introduction.....	220
9.2	Base Timer Function Description	221
9.2.1	Counting function.....	223
9.2.2	Timing function.....	223
9.2.3	Buzzer Features.....	224
9.3	Base Timer Interconnection	225
9.3.1	GATE Interconnection	225
9.3.2	Toggle Output Interconnect.....	225
9.4	Base Timer Register Description	226
9.4.1	16 Bit Mode Reload Register (TIMx_ARR)	226
9.4.2	16 Bit Mode Count Register (TIMx_CNT)	227
9.4.3	32 Bit Mode Count Register (TIMx_CNT32)	227
9.4.4	Control register (TIMx_CR)	228
9.4.5	Interrupt Flag Register (TIMx_IFR)	229
9.4.6	Interrupt Flag Clear Register (TIMx_ICLR)	229
10	Low Power Timer (LPTIM)	230
10.1	LPTimer Introduction	230
10.2	LPTimer Function Description	231
10.2.1	Counting function.....	232
10.2.2	Timing function.....	232

10.3	LPTimer Interconnect	233
10.3.1	GATE Interconnection Ta Re	233
10.3.2	EXT Interconnection bl co	233
10.3.3	Toggle Output Interconnect e rd	233
10.4	LPTimer Register Description of Co nt en ts	234
10.4.1	Counter Count Value Register (LPTIM_CNT)	235
10.4.2	Reload Register (LPTIM_ARR)	235
10.4.3	Control register (LPTIM_CR)	236
10.4.4	Interrupt Flag Register (LPTIM_IFR)	237
10.4.5	Interrupt Flag Clear Register (LPTIM_ICLR)	237
11	Programmable Counting Array (PCA)	238
11.1	PCA Introduction.....	238
11.2	PCA Functional Description.....	239
11.2.1	PCA Timer/Counter	239
11.2.2	PCA capture function	241
11.2.3	PCA comparison function	243
	11.2.3.1 16 Bit Software Counting Mode	243
	11.2.3.2 High-speed output mode.....	244
	11.2.3.3 WDT function for PCA Module 4.....	245
	11.2.3.4 PCA 8-bit pulse width modulation function	247
11.3	Interconnection and control of PCA modules with other modules	249
11.3.1	ECI Interconnection.....	249
11.3.2	PCACAP0	249
11.3.3	PCACAP1	249
	11.3.4 PCACAP	[4:2]
		249
11.4	PCA Register Description.....	250
11.4.1	Control register (PCA_CCON)	251
11.4.2	Mode register (PCA_CMOD)	252
11.4.3	Counting register (PCA_CNT)	253
11.4.4	Interrupt Clear Register (PCA_ICLR)	253
11.4.5	Compare capture mode registers (PCA_CCAPM0~4)	254
11.4.6	Compare capture data register high 8 bits (PCA_CCAP0~4H)	255
11.4.7	Compare capture data register low 8 bits (PCA_CCAP0~4L)	255
11.4.8	Compare capture bit16 registers (PCA_CCAP0~4)	256
11.4.9	Compare high-speed output flag register (PCA_CCAPO)	256
12	Advanced timer (TIM4/5/6)	257
12.1	Advanced Timer Introduction.....	257
12.2	Advanced Timer Function Description.....	259
12.2.1	Basic movements	259
	12.2.1.1 Basic waveform mode.....	259
	12.2.1.2 Compare Outputs.....	260

12.2.1.3 Capture input.....	261
12.2.2 Clock source selection..... Ta Re	262
12.2.3 Counting direction..... bl co	263
12.2.3.1 Sawtooth wave counting direction..... e rd	263
12.2.3.2 Triangular wave counting direction..... of	263
12.2.4 Digital filtering..... Co	264
12.2.5 Software Synchronization..... nt	265
12.2.5.1 Software Synchronization Launch..... en	265
12.2.5.2 Software synchronization stop..... ts	265
12.2.5.3 Software synchronization zeroing..... ts	265
12.2.6 Hardware Synchronization.....	267
12.2.6.1 Hardware Synchronized Start.....	267
12.2.6.2 Hardware synchronization stop.....	267
12.2.6.3 Hardware Synchronous Zeroing.....	267
12.2.6.4 Hardware synchronized capture input	267
12.2.6.5 Hardware synchronized counting.....	267
12.2.7 Cache function.....	269
12.2.7.1 Cache transfer time point.....	270
12.2.7.2 Generic cycle reference value cache transfer time point	270
12.2.7.3 Generic comparison of reference value cache transfer time points	270
12.2.7.4 Capture input value cache transfer time point.....	270
12.2.7.5 Cache transfer during clear action.....	270
12.2.8 Universal PWM Output	271
12.2.8.1 PWM Spread Spectrum Output	271
12.2.8.2 Independent PWM output	271
12.2.8.3 Complementary PWM Outputs	271
12.2.8.3.1 Software Setting GCMBR Complementary PWM Output.....	272
12.2.8.3.2 Hardware Set GCMBR Complementary PWM Output	273
12.2.8.4 Multi-phase PWM output	274
12.2.9 Orthogonal coding count	276
12.2.9.1 Position counting mode.....	276
12.2.9.1.1 Basic Count.....	276
12.2.9.1.2 Phase difference counting.....	277
12.2.9.1.3 Direction Counting.....	278
12.2.9.2 Public Rotation Mode.....	278
12.2.9.2.1 Z-phase count.....	278
12.2.9.2.2 Position Overflow Count.....	279
12.2.9.2.3 Mixed counting.....	279
12.2.9.2.4 Z-phase action shield.....	280
12.2.10 Cycle interval response	282
12.2.11 Protection mechanism.....	283
12.2.12 Interrupt description	284

12.2.12.1 Counting comparison match interrupts.....	284
12.2.12.2 Counting cycle matching interrupts.....	284
12.2.12.3 Dead time error interrupt.....	284
12.2.13 Brake Protection.....	285
12.2.13.1 Port Brakes vs. Software Brakes.....	285
12.2.13.2 Low power mode automatic brake.....	285
12.2.13.3 Output level same high and low brake.....	286
12.2.13.4 VC Brakes.....	286
12.2.14 Internal Interconnection.....	287
12.2.14.1 Interrupt trigger output.....	287
12.2.14.2 AOS Trigger.....	287
12.2.14.3 Port Trigger	TRIGA-TRIGD
	288
12.2.14.4 Compare output VC with Advanced Timer interconnection	289
12.2.14.5 UART interconnects with Advanced Timer.....	289
12.3 Register Description.....	290
12.3.1 General purpose count reference value register	(TIMx_CNTER)
	292
12.3.2 General Periodic Reference Value Register (TIMx_PERAR)	292
12.3.3 General Cycle Buffer Register (TIMx_PERBR)	293
12.3.4 General Comparison Reference Value Register (TIMx_GCMAR-GCMDR)	293
12.3.5 Dead Time Reference Value Register (TIMx_DTUAR- DTDAR)	294
12.3.6 General Control Register (TIMx_GCONR)	295
12.3.7 Interrupt Control Register (TIMx_ICONR)	297
12.3.8 Port Control Register (TIMx_PCONR).....	298
12.3.9 Cache Control Register (TIMx_BCONR)	301
12.3.10 Deadband Control Register (TIMx_DCONR)	302
12.3.11 Filter Control Register (TIMx_FCONR)	303
12.3.12 Valid Period Register (TIMx_VPERR)	305
12.3.13 Status Flag Register (TIMx_STFLR)	306
12.3.14 Hardware start event selection register (TIMx_HSTAR)	308
12.3.15 Hardware Stop Event Select Register (TIMx_HSTPR)	310
12.3.16 Hardware Clear Event Select Register (TIMx_HCELR)	312
12.3.17 Hardware Capture A Event Select Register (TIMx_HCPAR)	314
12.3.18 Hardware Capture B Event Select Register (TIMx_HCPBR)	316
12.3.19 Hardware recursive event selection register (TIMx_HCUPR)	318
12.3.20 Hardware decrement event selection register (TIMx_HCDOR)	320
12.3.21 Software Synchronization Start Register (TIMx_SSTAR)	322
12.3.22 Software Synchronization Stop Register (TIMx_SSTPR)	323
12.3.23 Software Synchronous Clear Register (TIMx_SCLRR)	324
12.3.24 Interrupt Flag Register (TIMx_IFR)	325
12.3.25 Interrupt Flag Clear Register (TIMx_ICLR)	327

12.3.26	Spread Spectrum and Interrupt Trigger Selection (TIMx_CR)	328
12.3.27	AOS Selection Control Register (TIMx_AOSSR)	329
12.3.28	AOS Select Control Register Flag Clear (TIMx_AOSCL)	330
12.3.29	Port Brake Control Register (TIMx_PTAKS)	331
12.3.30	Port Trigger Control Register (TIMx_TTRIG)	332
12.3.31	AOS Trigger Control Register (TIMx_ITRIG)	333
12.3.32	Port Brake Polarity Control Register (TIMx_PTAKP)	334
13	Real Time Clock (RTC)	335
13.1	Real Time Clock Introduction.....	335
13.2	Real Time Clock Function Description.....	336
13.2.1	Power-up settings.....	336
13.2.2	RTC Counting Start Setting.....	336
13.2.3	System low-power mode switching.....	336
13.2.4	Read out the count register.....	337
13.2.5	Write to count register.....	337
13.2.6	Alarm settings.....	338
13.2.7	13.2.71Hzoutput	338
13.2.8	Clock error compensation.....	339
13.3	RTC interrupt.....	341
13.3.1	RTC alarm interrupt	341
13.3.2	RTC Cycle Break	341
13.4	RTC Register Description.....	342
13.4.1	Control register0 (RTC_CR0)	343
13.4.2	Control register1 (RTC_CR1)	345
13.4.3	Second Count Register (RTC_SEC)	347
13.4.4	Sub-count register (RTC_MIN)	347
13.4.5	Time Count Register (RTC_HOUR)	348
13.4.6	Day Count Register (RTC_DAY)	350
13.4.7	Week Count Register (RTC_WEEK)	351
13.4.8	Monthly Count Register (RTC_MON)	352
13.4.9	Year Count Register (RTC_YEAR)	352
13.4.10	Sub-alarm register (RTC_ALMMIN)	353
13.4.11	Time Alarm Clock Register (RTC_ALMHOUR)	353
13.4.12	Weekly alarm clock register (RTC_ALM WEEK)	354
13.4.13	Clock error compensation register (RTC_COMPEN)	355
14	Watchdog Timer (WDT)	357
14.1	WDT Introduction	357
14.2	WDT Function Description.....	358
14.2.1	Interrupt generation after WDT overflow.....	358
14.2.2	Reset generated after WDT overflow	358
14.3	WDT Register Description.....	359
14.3.1	WDT Clear Control Register (WDT_RST)	359

14.3.2	WDT_CON register	360
15	Universal Synchronous Asynchronous Transceiver (UART)	361
15.1	Overview	361
15.2	Structure Block Diagram.....	361
15.3	Main Features.....	362
15.4	Function Description.....	363
15.4.1	Working mode.....	363
15.4.1.1	Mode0~Mode3 Function Comparison.....	363
15.4.1.2	Mode0 (synchronous mode, half duplex)	363
15.4.1.3	Mode1 (asynchronous mode, full duplex)	365
15.4.1.4	Mode2 (asynchronous mode, full duplex)	365
15.4.1.5	Mode3 (asynchronous mode, full duplex)	366
15.4.2	Baud rate generation.....	368
15.4.2.1	Mode1/Mode3 baud rate setting example.....	369
15.5	Frame error detection.....	373
15.6	Multi-machine communication.....	374
15.7	Automatic address recognition.....	375
15.7.1	Given address.....	375
15.7.1.1	Broadcast Address.....	375
15.7.1.2	Examples	375
15.8	Sender-sender cache.....	376
15.8.1	Receiving Cache	376
15.8.2	Send cache	376
15.9	Register	377
15.9.1	Data register (UARTx_SBUF)	377
15.9.2	Control register (UARTx_SCON)	378
15.9.3	Address register (UARTx_SADDR)	379
15.9.4	Address Mask Register (UARTx_SADEN)	379
15.9.5	Flag Bit Register (UARTx_ISR)	380
15.9.6	Flag Bit Clear Register (UARTx_ICR)	381
16	Low Power Synchronous Asynchronous Transceiver (LPUART)	382
16.1	Overview	382
16.2	Structure Block Diagram.....	383
16.3	Main Features.....	384
16.4	Function Description.....	385
16.4.1	Configuration clock and transmission clock	385
16.4.2	Working mode	385
16.4.2.1	Mode0~Mode3 Function Comparison.....	386
16.4.2.2	Mode0 (synchronous mode, half-duplex) data sending and receiving instructions	387
16.4.2.3	Mode1 (asynchronous mode, full duplex) data sending and receiving instructions	388

16.4.2.4 Mode2 (asynchronous mode, full duplex) data sending and receiving instructions	Ta Re	389
16.4.2.5 Mode3 (asynchronous mode, full duplex) data sending and receiving instructions	bl co	390
16.4.3 Baud rate generation.....	e rd	391
16.5 Frame error detection.....	of	391
16.6 Multi-machine communication.....	Co	392
16.7 Automatic address recognition.....	nt	393
16.7.1 Given address.....	en	393
16.7.2 Broadcast Address.....	ts	393
16.7.3 Examples.....		393
16.8 Sender-sender cache.....		394
16.8.1 Receiving Cache		394
16.8.2 Send cache.....		394
16.9 Register		395
16.9.1 Data register (LPUART_SBUF)		395
16.9.2 Control register (LPUART_SCON)		396
16.9.3 Address register (LPUART_SADDR)		398
16.9.4 Address Mask Register (LPUART_SADEN)		398
16.9.5 Interrupt Flag Bit Register (LPUART_ISR)		399
16.9.6 Interrupt Flag Bit Clear Register (LPUART_ICR)		400
17 I2C bus (I2C)		401
17.1 Introduction.....		401
17.2 Main Features.....		401
17.3 Protocol Description.....		401
17.3.1 Data transfer on the I2C bus.....		402
17.3.2 Answer on the I2C bus.....		403
17.3.3 Arbitration on the I2C bus		404
17.4 Function Description.....		406
17.4.1 Serial Clock Generator.....		407
17.4.2 Input Filter.....		407
17.4.3 Address Comparator.....		407
17.4.4 Answer flag bit.....		408
17.4.5 Interrupt generators		408
17.4.6 Working mode.....		408
17.4.7 Status code representation.....		415
17.5 Programming Examples.....		417
17.5.1 Host sending example.....		417
17.5.2 Host reception example.....		418
17.5.3 Slave reception example		419
17.5.4 Slave sending example		420
17.6 Register Description.....		421

17.6.1	I2C baud rate counter enable register	(I2C_TMRUN)
	421	Ta Re
17.6.2	I2C baud rate counter configuration register	(I2C_TM)
	422	bl co
17.6.3	I2C configuration register	(I2C_CR)
	423	e rd
17.6.4	I2C data register	(I2C_DATA)
	425	Co
17.6.5	I2C address register	(I2C_ADDR)
	426	nt en
17.6.6	I2C Status Register	(I2C_STAT)
	427	ts
18	Serial Peripheral Interface (SPI)	428
18.1	SPI Introduction.....	428
18.2	SPI Key Features.....	428
18.3	SPI Function Description.....	429
18.3.1	SPI Host Mode.....	429
18.3.2	SPI Slave Mode.....	429
18.3.3	SPI Data Frame Format	431
18.3.4	SPI Status Flags and Interrupts	432
18.3.5	SPI Multi-Computer System Configuration Instructions.....	432
18.3.6	SPI Pin Configuration Description	434
18.4	SPI Programming Example.....	435
18.4.1	SPI Host Send Example	435
18.4.2	SPI host reception example	435
18.4.3	SPI Slave Send Example	436
18.4.4	SPI Slave Receive Example	436
18.6	SPI Register Description.....	438
18.6.1	SPI Configuration Register	(SPI_CR)
	439	
18.6.2	SPI Chip Select Configuration Register	(SPI_SSN)
	441	
18.6.3	SPI Status Register	(SPI_STAT)
	442	
18.6.4	SPI data register	(SPI_DATA)
	443	
19	Clock Calibration Module (CLKTRIM)	444
19.1	CLK_TRIM Introduction.....	444
19.2	CLK_TRIM Main Features.....	444
19.3	CLK_TRIM Function Description.....	445
19.3.1	CLK_TRIM Calibration Mode.....	445
19.3.1.1	Operation process.....	445
19.3.2	CLK_TRIM Monitoring Mode	446
19.3.2.1	Operation process.....	446

19.4	CLK_TRIM Register Description.....	447
19.4.1	Configuration register Ta.....Re.....(CLKTRIM_CR) 448	
19.4.2	Reference counter initial value configuration register (CLKTRIM_REFCON) 449	
19.4.3	Reference counter value register(CLKTRIM_REFCNT) 449	
19.4.4	Calibration counter value register(CLKTRIM_CALCNT) 450	
19.4.5	Interrupt Flag Bit Register(CLKTRIM_IFR) 451	
19.4.6	Interrupt Flag Bit Clear Register(CLKTRIM_ICLR) 452	
19.4.7	Calibration counter overflow value configuration register (CLKTRIM_CALCON) 453	
20	Cyclic Redundancy Check (CRC)	454
20.1	Overview	454
20.2	Main Features.....	454
20.3	Function Description.....	454
20.3.1	Working mode.....	454
20.3.2	Coding method.....	454
20.3.3	Write Bit Width.....	454
20.4	Programming Examples.....	455
20.4.1	CRC-16 encoding mode.....	455
20.4.2	CRC-16 test pattern.....	455
20.5	Register Description.....	456
20.5.1	Register List.....	456
20.5.2	Result register (CRC_RESULT)	457
20.5.3	Data register (CRC_DATA)	457
21	Analog to Digital Converters (ADC)	458
21.1	Module Introduction.....	458
21.2	ADC Block Diagram.....	458
21.3	Conversion Timing and Conversion Speed.....	459
21.4	Single conversion mode.....	460
21.5	Continuous conversion mode.....	462
21.6	Continuous conversion accumulation mode.....	464
21.7	Comparison of ADC conversion results	466
21.8	ADC Interrupt	467
21.9	Ambient temperature measurement using temperature sensors.....	467
21.10	ADC Module Register	469
21.10.1	ADC Configuration Register0 (ADC_CR0)	470
21.10.2	ADC Configuration Register1 (ADC_CR1)	472
21.10.3	ADC Configuration Register2 (ADC_CR2)	475
21.10.4	ADC channel Conversion0 result (ADC_result0)	477

21.10.5	ADC channel Conversion1 result (ADC_result1)	477
21.10.6	ADC channel Conversion2 results (ADC_result2)	478
21.10.7	ADC channel Conversion3 results (ADC_result3)	478
21.10.8	ADC channel Conversion4 results (ADC_result4)	479
21.10.9	ADC channel Conversion5 results (ADC_result5)	479
21.10.10	ADC channel Conversion6 results (ADC_result6)	480
21.10.11	ADC channel Conversion7 results (ADC_result7)	480
21.10.12	ADC channel Conversion8 results (ADC_result8)	481
21.10.13	ADC conversion result accumulation value (ADC_result_acc)	481
21.10.14	ADC Compare Upper Threshold (ADC_HT)	482
21.10.15	ADC Compare Lower Threshold (ADC_LT)	482
21.10.16	ADC Interrupt Flag Register (ADC_IFR)	483
21.10.17	ADC Interrupt Clear Register (ADC_ICLR)	484
21.10.18	ADC result (ADC_result)	484
22	Analog Comparator (VC)	485
22.1	Analog Voltage Comparator VC Introduction	485
22.2	Voltage comparator frame diagram.....	486
22.3	Set-up/response time.....	486
22.4	Filtering time.....	487
22.5	Hysteresis function.....	487
22.6	VC Register.....	488
22.6.1	VC Configuration Register (VC_CR)	489
22.6.2	VC0 Configuration Register (VC0_CR)	491
22.6.3	VC1 Configuration Register (VC1_CR)	493
22.6.4	VC0 Output Configuration Register (VC0_OUT_CFG)	495
22.6.5	VC1 Output Configuration Register (VC1_OUT_CFG)	497
22.6.6	VC Interrupt Register (VC_IFR)	499
23	Low Voltage Detector (LVD)	500
23.1	LVD Introduction	500
23.2	LVD Block Diagram	500
23.3	Digital filtering.....	501
23.4	Hysteresis function.....	501
23.5	Configuration Example.....	502
23.5.1	LVD configured for low voltage reset	502
23.5.2	LVD configured for voltage change interrupt	502
23.6	LVD Register	503
23.6.1	LVD Configuration Register (LVD_CR)	503
23.6.2	LVD Interrupt Register (LVD_IFR)	505
24	Simulate other registers.....	506
24.1	BGR Configuration Register (BGR_CR)	506
25	SWD Debugging Interface.....	507
25.1	SWD Debugging Add-On.....	507

25.2	ARM® Reference Documentation.....	508
25.3	Debug port pins.....	509
25.3.1	SWD Port Pins	509
25.3.2	SW-DP Pin Assignment.....	509
25.3.3	Internal pull-up on SWD pin	509
25.4	SWD Port.....	510
25.4.1	Introduction to SWD Protocol	510
25.4.2	SWD Protocol Sequence	510
25.4.3	SW-DP state machine (reset, idle state, ID code)	511
25.4.4	DP and AP read/write access	511
25.4.5	SW-DP Register	512
25.4.6	SW-AP register	513
25.5	Kernel debugging.....	514
25.6	BPU (Breakpoint Unit)	514
25.6.1	BPU Function	514
25.7	DWT (Data Watch Point)	515
25.7.1	DWT Function.....	515
25.7.2	DWT Program Counter Sampling Register	515
25.8	MCU Debugging	Gathering (DBG)
	516	
25.8.1	Debugging support for low-power modes	516
25.8.2	Debugging support for timers, watchdogs	516
25.9	Debug mode module working state control (DEBUG_ACTIVE)	517
26	Electronic signature of devices.....	519
26.1	Product Unique Identifier (UID) register (80bits)	519
26.2	Product Model Register.....	520
26.3	FLASH Capacity Register.....	520
26.4	RAM Capacity Register.....	521
26.5	Number of Pins Register.....	521
27	Appendix	ASysTickTimer
	522	
27.1	SysTick Timer Introduction.....	522
27.2	Set	SysTick
	522	
27.3	SysTick register	523
27.3.1	SysTick Control and Status Register (CTRL)	523
27.3.2	SysTick Reload Register (LOAD)	523
27.3.3	SysTick Current Value Register (VAL)	524
28	Appendix B	Documentation Conventions
	525	
28.1	List of abbreviations related to registers.....	525
28.2	Glossary	525
Imprint & Contact	526

Table of Contents

Table 2-1	Diagram of runnable modules in operation mode	30
Table 2-2	Diagram of runnable modules in hibernation mode	32
Table 2-3	Diagram of runnable modules in deep hibernation mode	34
Table 5-1	Cortex-M0+Processor Exception List.....	68
Table 5-2	Correspondence between external interrupts and NVIC interrupt inputs.....	74
Table 6-1Port Status Truth Table	94
Table 6-2Port Multiplexing Table	97
Table 7-1	FLASH erase time parameters at different frequencies	204
Table 9-1	BaseTimer Register List	226
Table 10-1	List of LPTimerRegisters.....	234
Table 11-1	PCAComparison Capture Function Module Settings	248
Table 11-2	PCARegister List	250
Table 12-1	AdvancedTimer Basic Features.....	257
Table 12-2	AdvancedTimer Port List.....	257
Table 12-3	AOSSource Selection.....	288
Table 12-4Port Trigger Selection	288
Table 12-5	AdvancedTimer Register List.....	291
Table 13-1	Basic characteristics of the	RTC 335
Table 13-2	RTCRegister List	342
Table 14-1	WDTRegister List.....	359
Table 15-1	Mode0/1/2/3Data Structure	363
Table 16-1	Mode0/1/2/3Data Structure	386
Table 17-1	I2Cclock signal baud rate	407
Table 17-2	I2CStatus Code Expressions	416
Table 17-3Register List	421
Table 18-1	SPIPin Configuration Description Table	434
Table 18-2	SPIRegister List	438
Table 18-3Host Mode Baud Rate Selection	440
Table 19-1Register List	447
Table 21-1	ADCRegisters	469
Table 22-1	VCRegisters	488

Figure Catalog

Figure 1-1	System architecture diagram	24
Figure 1-2	Address area division diagram	26
Figure 2-1	Block diagram of control mode	28
Figure 3-1	Block diagram of clock control module	37
Figure 3-2	Crystal clock start-up diagram	39
Figure 3-3	Clock switching diagram	42
Figure 3-4	Clock calibration schematic	45
Figure 4-1	Schematic diagram of reset source	62
Figure 5-1	uses only the high two priority registers	68
Figure 5-2	Interrupt vector table	69
Figure 5-3	Interrupt Activation and Suspend Status	70
Figure 5-4	Interrupt pending state is cleared and then re-acknowledged	71
Figure 5-5 Interrupt ... request held high at interrupt exit will cause interrupt processing to be executed again		71
Figure 5-6 An interrupt hang generated during interrupt processing can also be acknowledged		72
Figure 5-7	Interrupt Structure Diagram	75
Figure 6-1	Port Circuit Schematic	95
Figure 6-2Variation of	AHBbus port with system clock	95
Figure 6-3	Read Port Pin Data Synchronization Diagram	96
Figure 7-1	Memory Sector Segmentation	
	199	
Figure 9-1BaseTimer Block Diagram.....		220
Figure 9-2TimerMode 1 Block Diagram.....		222
Figure 9-3TimerMode 2 Block Diagram.....		222
Figure 9-4	Mode 1 Timing Diagram	
	223	
Figure 9-5	Mode 2 Timing Diagram (Prescaler set to 2)	
	223	
Figure 10-1LPTimerblock diagram.....		230
Figure 10-2LPTimermode		1231
Figure 10-3LPTimermode		2232
Figure 11-1PCAoverall block diagram.....		238
Figure 11-2PCACounter block diagram.....		240
Figure 11-3PCAcapture function block diagram.....		242
Figure 11-4PCAComparison Function Block Diagram.....		244
Figure 11-5PCAWDT Function Block Diagram.....		245
Figure 11-6PCAPWM Function Block Diagram.....		247
Figure 11-7PCAPWM Output Waveform.....		248
Figure 12-1AdvancedTimer Block Diagram.....		258
Figure 12-2	Sawtooth waveform (recursive counting)	
	259	
Figure 12-3	Triangular waveform	
	259	

Figure 12-4	Comparison Output Action
260	
Figure 12-5	Capturing Input Actions
261	
Figure 12-6	captures the filtering function of the input port
264	

Figure 12-7	Software synchronization action
265	
Figure 12-8	Hardware synchronization action
268	
Figure 12-9	Comparison of output timings in single cache mode
269	
Figure 12-10PWMspread spectrum output diagram	271
Figure 12-11CHAOutput PWM Wave.....	271
Figure 12-12	Software Setting GCMBR Complementary PWM Output in Delta Wave A Mode
272	
Figure 12-13	Hardware Setting GCMBR Complementary PWM Output in Delta B Mode (Symmetric Deadband)
273	
Figure 12-14Phase PWM Wave	274
Figure 12-15 Triangular wave A mode with dead time three-phase complementary PWM wave output	
275	
Figure 12-16	Basic counting action in position mode
276	
Fig. 12-17	Phase difference counting action setting in position mode (1X)
277	
Fig. 12-18	Phase difference counting action setting in position mode (2 times)
277	
Figure 12-19	Phase difference counting action setting in position mode (4 times)
277	
Figure 12-20	Direction counting action in position mode
278	
Fig. 12-21	Z-phase counting action during the rotation mode
278	
Fig. 12-22	Position counter output counting action in revolution mode
279	
Fig. 12-23	Mixed counting action of Z-phase counting and position counter output during revolution mode
279	
Figure 12-24	Rotation counting mode - mixed counting Z-phase shield action example
1280	
Fig. 12-25	Rotation counting mode - mixed counting Z-phase shield action example
2281	
Figure 12-26	Cycle interval valid request signal action
282	
Figure 12-27	Schematic diagram of port brakes and software brakes
285	
Fig. 12-28	Schematic diagram of the same high and low brake output
286	
Figure 12-29VCBrake Control Schematic	286
Figure 12-30Timer4/5/6Interrupt Selection.....	287
Figure 13-1RTCBlock Diagram.....	335
Figure 14-1WDToverall block diagram.....	357
Figure 15-1Block diagram of	UARTstructure
361	
Figure 15-2 Mode0Sending Data	364
Figure 15-3 Mode0Receiving Data	364

Figure 15-4Mode1Sending Data	365
Figure 15-5Mode1Receiving Data	365
Figure 15-6Mode2Sending Data	366
Figure 15-7Mode2Receiving Data	366
Figure 16-1Block diagram of LPUARTstructure	
383	
Figure 16-2 Mode0Sending Data	387
Figure 16-3 Mode0Receiving Data	387
Figure 16-4Mode1Sending Data	388
Figure 16-5Mode1Receiving Data	388
Figure 16-6Mode2Sending Data	389
Figure 16-7Mode2Receiving Data	389
Figure 17-1I2CTransfer Protocol.....	402
Figure 17-2STARTand STOP Conditions	402

Figure 17-3	I2Cbus upstream transmission.....	403
Figure 17-4	Answer Signal onI2CBus	404
Figure 17-5	Arbitration on theI2CBus	405
Figure 17-6	I2CFunction Module Diagram.....	406
Figure 17-7Master transmit mode data synchronization diagram	409
Figure 17-8	I2CHost Transmit Status Diagram.....	409
Figure 17-9Master Receive Mode Data Synchronization Diagram	410
Figure 17-10	I2CHost Receive Status Diagram.....	411
Figure 17-11Data synchronization diagram from receive mode	412
Figure 17-12Slave Receive Status Diagram	412
Figure 17-13Data synchronization diagram from transmit mode	413
Figure 17-14	I2Cslave transmit status diagram.....	413
Figure 17-15	I2CBroadcast Call Status Diagram.....	414
Figure 18-1Slave reception diagram	430
Figure 18-2Slave transmitting diagram	430
Figure 18-3Host Mode Frame Format	431
Figure 18-4Data frame format when the slave CPHA is 0	431
Figure 18-5Data frame format when the slave CHPA is 1	432
Figure 18-6	Schematic ofSPIMulti-Master/Multi-Slave System	433
Figure 21-1	ADC Schematic Block Diagram	458
Figure 21-2	ADC Conversion Timing Diagram	459
Figure 21-3	ADC Continuous Conversion Process Example.....	462
Figure 21-4	ADC Continuous Conversion Accumulation Process Example.....	464
Figure 22-1	VCFramework Diagram	486
Figure 22-2	VCFilter Response Time	487
Figure 22-3	VCHysteresis Function	487
Figure 23-1	LVDBlock Diagram	500
Figure 23-2	LVDFilter Output	501
Figure 23-3	LVDHysteresis Response.....	501

Brief introduction

The HC32L110 series is an ultra-low power, low pin count, wide voltage range MCU designed to extend the battery life of portable measurement systems. It is characterized by high integration, high interference immunity, high reliability and ultra-low power consumption. The core of this product adopts Cortex-M0+ kernel with mature Keil & IAR debug development software, supporting C language and assembly language, assembly instructions.

Ultra Low Power MCU Typical Applications

- Sensor applications, IoT applications
- Intelligent Transportation, Smart City, Smart Home
- Intelligent sensor applications such as fire sensors, smart door locks, wireless monitoring, etc.
- A variety of battery-powered and power-critical portable devices, etc.

About this brochure

This manual introduces the functions, operations, and usage of the chip. For the specifications of the chip, please refer to the corresponding "Data Sheet".

1 System Architecture

1.1 Overview

This product system consists of the following components.

- 1 AHB Bus Master.
 - Cortex-M0+
- 4 AHB Bus Slaves.
 - FLASH Memory
 - SRAM memory
 - AHB0, AHB to APB Bridge, contains all APB interface peripherals
 - AHB1, containing all AHB interface peripherals

The entire system bus structure is implemented using multi-level AHB-lite bus interconnections. The following figure shows.

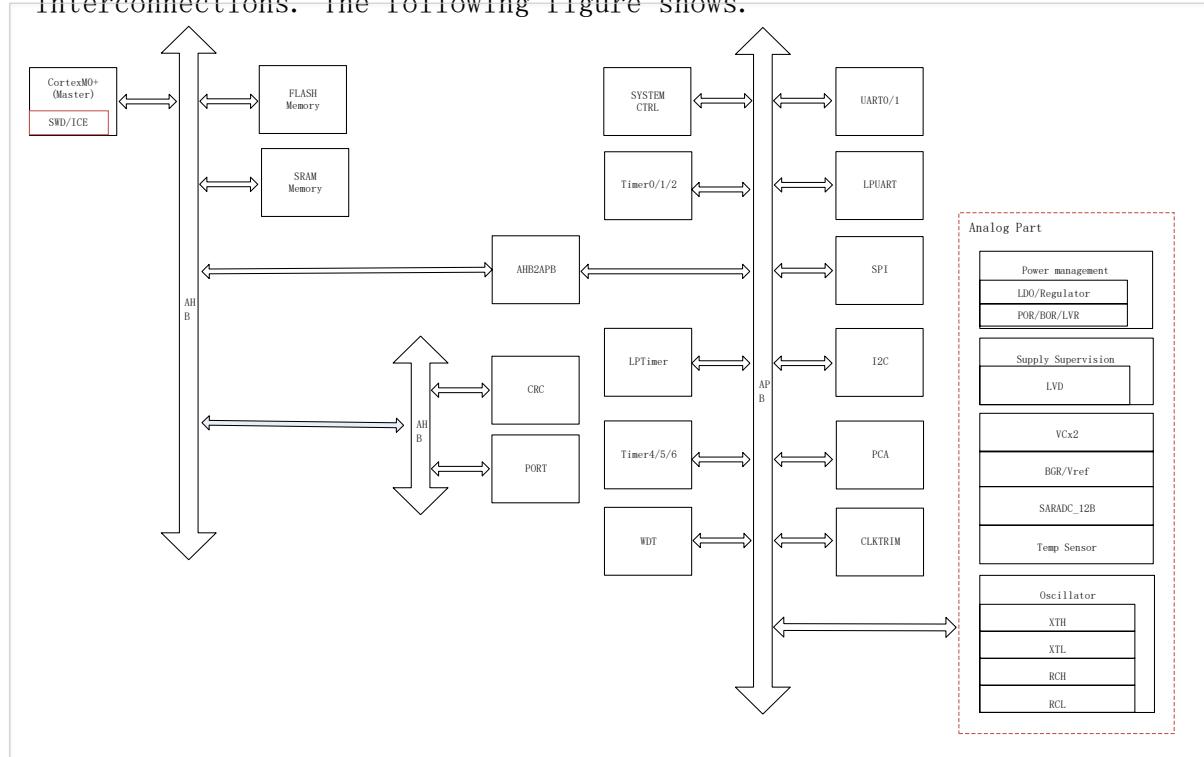
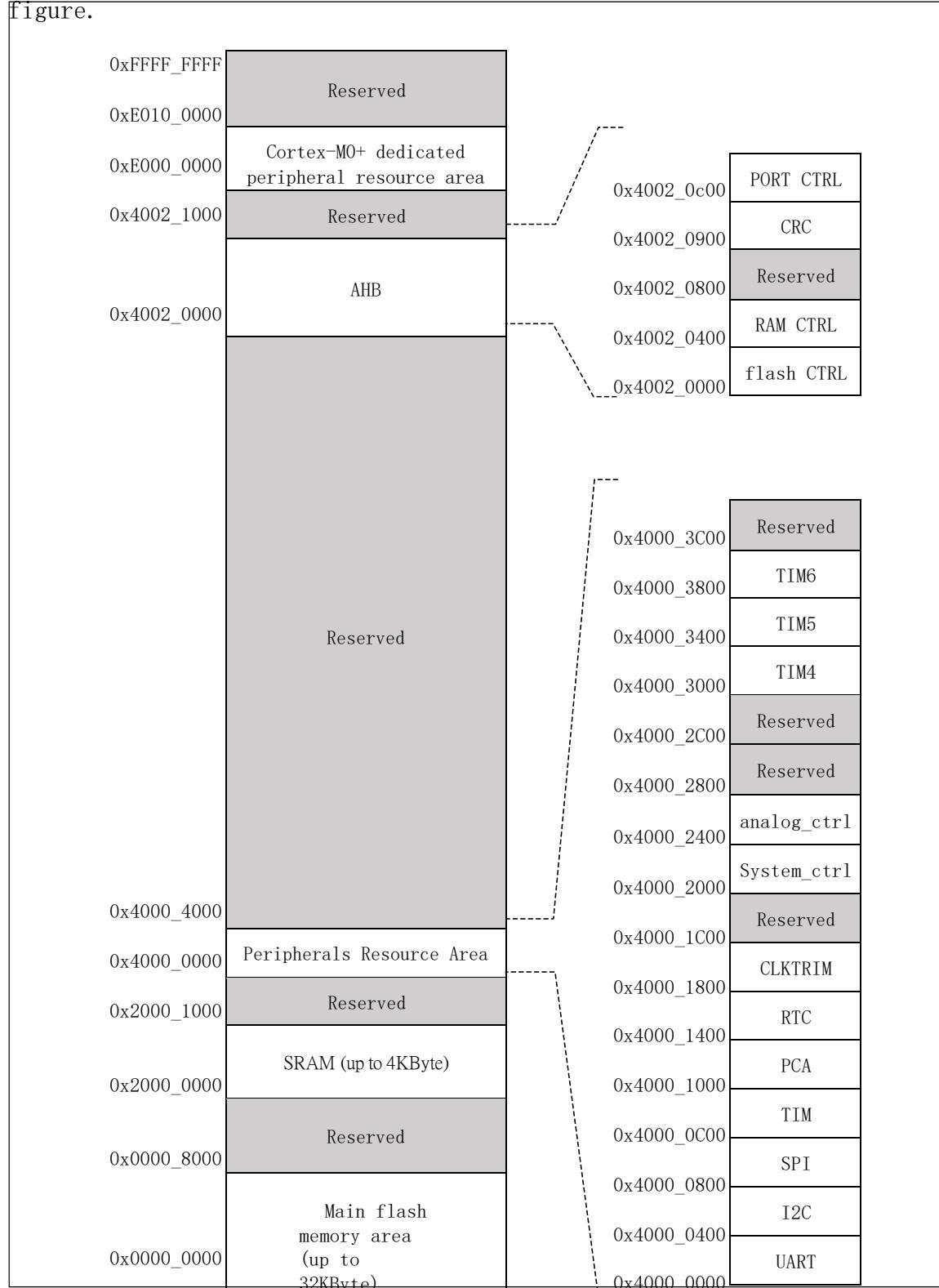


Figure 1-1 System architecture diagram

1.2 System address division

The address area division of the whole HC32L110 system is shown in the following figure.



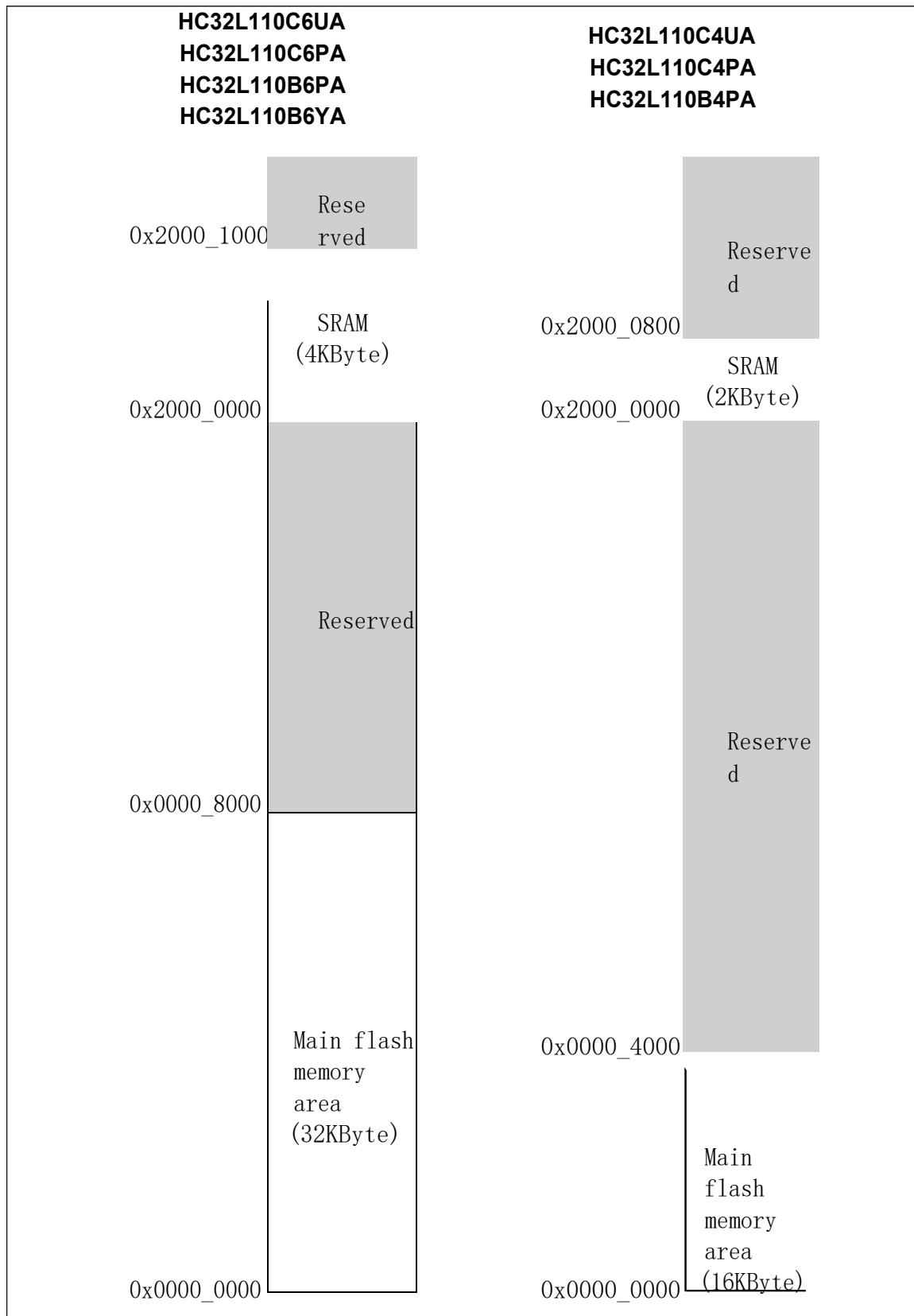


Figure 1-2 Address area division diagram

1.3 Memory and module address assignment

Boundary Address	Size	Memory Area	Description
0x0000_0000 - 0x0000_7FFF	32kByte	FLASH Memory	
0x0000_8000 - 0x0010_0BFF	-	Reserved	
0x0010_0C00 - 0x0010_0C3B	60Byte	Trim Data	
0x0010_0C3C - 0x0010_0E6F	-	Reserved	
0x0010_0E70 - 0x0010_0E7F	16Byte	UID	
0x0010_0E80 - 0x1FFF_FFFF	-	Reserved	
0x2000_0000 - 0x2000_0FFF	4kByte	SRAM Memory	
0x2000_1000 - 0x3FFF_FFFF	-	Reserved	
0x4000_0000 - 0x4000_00FF	256Byte	UART0	
0x4000_0100 - 0x4000_01FF	256Byte	UART1	
0x4000_0200 - 0x4000_02FF	256Byte	LPUART	
0x4000_0300 - 0x4000_03FF	-	Reserved	
0x4000_0400 - 0x4000_07FF	1kByte	I2C	
0x4000_0800 - 0x4000_0BFF	1kByte	SPI	
0x4000_0C00 - 0x4000_0FFF	1kByte	Timer0/1/2/WDT/LPTimer	
0x4000_1000 - 0x4000_13FF	1kByte	PCA	
0x4000_1400 - 0x4000_17FF	1kByte	RTC	
0x4000_1800 - 0x4000_1BFF	1kByte	CLKTRIM	
0x4000_1C00 - 0x4000_1FFF	-	Reserved	
0x4000_2000 - 0x4000_23FF	1kByte	SYSTEMCTRL	
0x4000_2400 - 0x4000_27FF	1kByte	ANALOGCTRL	
0x4000_2800 - 0x4000_2FFF	-	Reserved	
0x4000_3000 - 0x4000_33FF	1kByte	Timer4	
0x4000_3400 - 0x4000_37FF	1kByte	Timer5	
0x4000_3800 - 0x4000_3BFF	1kByte	Timer6	
0x4000_3C00 - 0x4001_FFFF	-	Reserved	
0x4002_0000 - 0x4002_03FF	1kByte	FLASH CTRL	
0x4002_0400 - 0x4002_07FF	1kByte	RAM CTRL	
0x4002_0800 - 0x4002_08FF	256Byte	Reserved	
0x4002_0900 - 0x4002_0BFF	768Byte	CRC	
0x4002_0C00 - 0x4002_0FFF	1kByte	PORT CTRL	

2 Working mode

The power management module is responsible for managing the switching between the various operating modes of the product and controlling the operating status of each function module in each operating mode. The operating voltage (VCC) of this product is ~1.8~5.5V.

This product has the following operating modes.

- (1) Operation mode: CPU operation, peripheral function module operation.
- (2) Hibernation mode: CPU stops running and peripheral function modules run.
- (3) Deep hibernation mode: CPU stops running and high-speed clock stops running.

From the run mode, other low-power modes can be accessed by executing software programs. From various other low-power modes, it is possible to return to run mode by interrupt triggering.

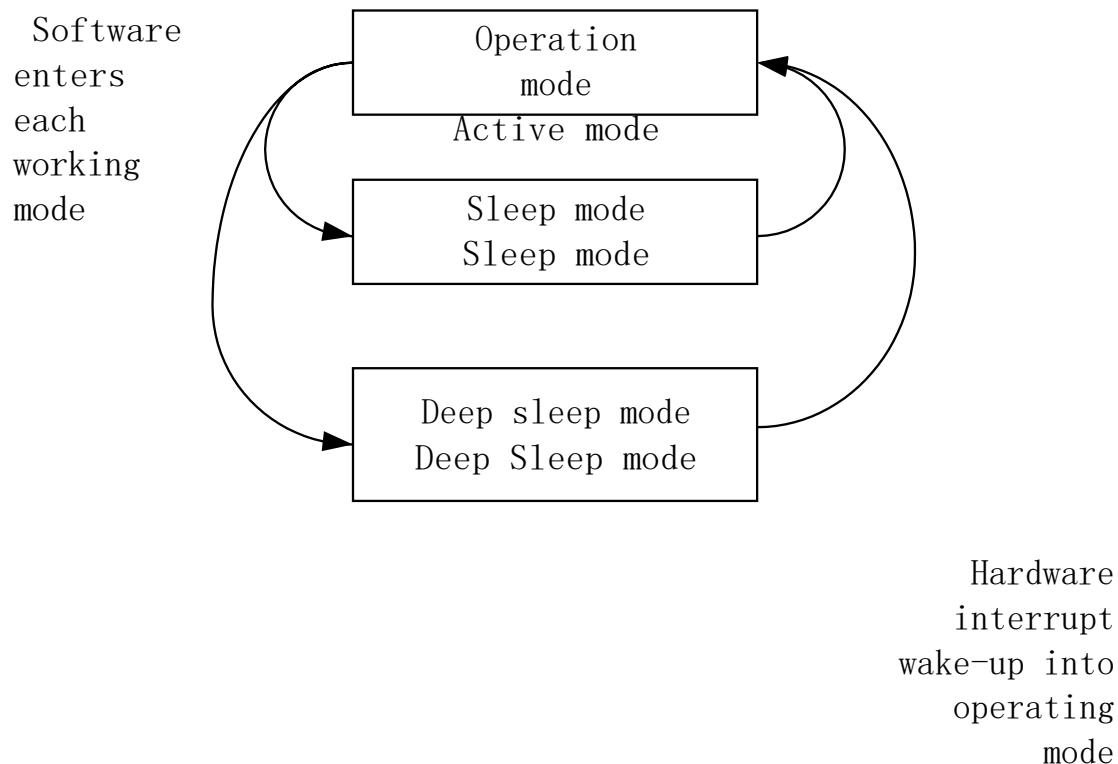


Figure 2-1 Block diagram of control mode

In each mode, the CPU can respond to all interrupt types.

	Interruption source	Operation mode	Sleep mode	Deep sleep mode	
[0]	GPIO_P0	✓	✓	✓	
[1]	GPIO_P1	✓	✓	✓	
[2]	GPIO_P2	✓	✓	✓	
[3]	GPIO_P3	✓	✓	✓	
[6]	UART0	✓	✓		
[7]	UART1	✓	✓		
[8]	LPUART	✓	✓	✓	
[10]	SPI	✓	✓		
[12]	I2C	✓	✓		

[14]	Timer0	✓	✓	
[15]	Timer1	✓	✓	
[16]	Timer2	✓	✓	
[17]	LPTimer	✓	✓	✓
[18]	Timer4	✓	✓	
[19]	Timer5	✓	✓	
[20]	Timer6	✓	✓	
[21]	PCA	✓	✓	
[22]	WDT	✓	✓	✓
[23]	RTC	✓	✓	✓
[24]	ADC	✓	✓	
[26]	VC0	✓	✓	✓
[27]	VC1	✓	✓	✓
[28]	LVD	✓	✓	✓
[30]	RAM FLASH	✓	✓	
[31]	CLKTRIM	✓	✓	✓

In each mode, this product responds to all reset types.

	Reset Source	Operation mode	Sleep mode	Deep sleep mode
[0]	Power-on Power-off Reset POR	✓	✓	✓
[1]	External Reset Pin Reset	✓	✓	✓
[2]	LVD reset	✓	✓	✓
[3]	WDT reset	✓	✓	✓
[4]	PCA reset	✓	✓	
[5]	Cortex-M0+ LOCKUP Hardware Reset	✓		
[6]	Cortex-M0+ SYSRESETREQ Software Reset	✓		

2.1 Operation mode

This product operates in the Modes (Active Mode)

The microcontroller MCU is running after the system is reset at power-on, or after waking up from each low-power state. Various low-power modes can be utilized to save energy when the CPU does not need to continue running, such as when waiting for an

Active Mode		
Corex M0+ lowest available wake-up sources.	SWD, UART0-1, SPI, I2C, CRC, XTL, RCL, RTC, WDT OSC	XTH up time, RCH, ADC, RESET, POR/BOR, LVD, VC, CLKTRIM, WDT
FLASH	UART0-1	RCH
RAM	SPI	ADC
TIM0-2	I2C	RESET
TIM4-6	CRC	POR/BOR
PCA	XTL	LVD
LPUART	RCL	VC
LPTIM	RTC	CLKTRIM
GPIO	WDT OSC	WDT

Table 2-1 Diagram of runnable modules in operation mode

Several ways to reduce chip power consumption in operating mode.

- 1) In operation mode, the speed of any of the system clocks (HCLK, PCLK) can be reduced by programming the prescaler (SYSCLK0.AHB_CLK_DIV, SYSCLK0.APB_CLK_DIV). The prescaler can also be used to reduce the clock of the peripheral before entering the sleep mode.
- 2) In run mode, turn off the clock that does not use peripherals (PERI_CLKx) to reduce power consumption.
- 3) Turn off clocks not using peripherals in run mode (PERI_CLKx) to reduce power consumption and put the system into sleep mode to reduce power consumption even more, and turn off clocks not using peripherals before executing WFI instructions

- 4) Low-power mode is used instead of sleep mode because the wake-up time of this product is extremely short (~4uS), can also meet the demand of real-time response of the system.

2.2 Sleep mode

This product sleep mode (Sleep Mode)

The WFI command allows you to enter hibernation mode, where the CPU stops running, but the clock module, system clock, NVIC interrupt processing and peripheral function modules are still working.

The system enters the hibernation state without changing the port state. Change the state of the IO to the state under hibernation as needed before entering hibernation.

How to enter hibernation mode.

Hibernate state is entered by executing the WFI instruction. According to the Cortex™-M0+ system control register

The value of the SLEEPONEXIT bit, with two options for selecting the hibernation mode entry mechanism.

SLEEP-NOW: If the SLEEPONEXIT bit is cleared, the microcontroller immediately enters sleep mode when the WFI or WFE is executed.

SLEEP-ON-EXIT: If the SLEEPONEXIT bit is set and the system exits from the lowest priority interrupt handler, the microcontroller immediately enters hibernation mode.

How to exit hibernation mode.

If the WFI instruction is executed to enter hibernate mode, any peripheral interrupt responded to by a high priority nested vector interrupt controller can wake up the system from hibernate mode.

Notes on use.

- **SLEEP-ON-EXIT** The position1 , after the execution of the interrupt automatically into sleep, the program does not need to write wfi().
- **SLEEP-ON-EXIT** This bit is cleared0, main() executes wfi() and then goes to sleeping, after the interrupt is triggered and the interrupt program returns to main(), the WFI instruction is executed and then goes to sleeping, waiting for the subsequent interrupt to be triggered.
- The SLEEP-ON-EXIT bit does not affect the execution of the wfi() instruction.

sLEEP-ON-EXIT = 0: main() execution

After wfi(), the interrupt is triggered and the interrupt program returns to main(), then continues on to the next step.

- If sleep is entered in an interrupt, only interrupts with higher priority than this interrupt can be woken up, and the higher priority is executed first, then the lower priority; interrupts with priority lower than or equal to this interrupt cannot be woken up.

Sleep Mode		
Cortex-M0+	SWD	XTH
FLASH	UART0-1	RCH
RAM	SPI	ADC
TIM0-2	I2C	RESET
TIM4-6	CRC	POR/BOR
PCA	XTL	LVD
LPUART	RCL	VC
LPTIM	RTC	CLKTRIM
GPIO	WDT OSC	WDT

Modules in gray are not working in
the current state.

Table 2-2 Diagram of runnable modules in hibernation mode

2.3 Deep sleep mode

This product Deep Sleep Mode (Deep Sleep Mode)

In deep sleep mode, the CPU stops running, the high-speed clock is turned off, the low-speed clock can be configured to run or not, some low-power peripheral modules can be configured to allow or not, and NVIC interrupt processing can still work.

- The system enters deep hibernation mode from the high-speed clock, the high-speed clock is automatically turned off, and the low-speed clock remains in the state before entering deep hibernation.
- The system enters deep hibernation mode from the low-speed clock, which remains running and enters hibernation mode as the low-speed clock does not automatically shut down. Only the ARM Cortex-M0+ does not run, all other modules run.
- When the system clock is switched, all clocks are not automatically turned off, and the corresponding clocks need to be turned off and on according to the power consumption and system requirements software.
- The system enters deep hibernation without changing the port state. Change the state of the IO to the state under hibernation as needed before entering hibernation. Unused IO pins and IO pins that are not pinout of the package need to be set as input and enabled for pull-up.

How to enter deep hibernation mode.

The SLEEPDEEP bit in the Cortex-M0+ System Control Register is first set to enter the hibernation state by executing the WFI instruction. Depending on the value of the SLEEPONEXIT bit in the CortexTM-M0+ System Control Register, two options are available for selecting the deep hibernation mode entry mechanism.

SLEEP-NOW: If the SLEEPONEXIT bit is cleared, the microcontroller immediately enters sleep mode when the WFI or WFE is executed.

SLEEP-ON-EXIT: If the SLEEPONEXIT bit is set and the system exits from the HC32L110 Series User's Manual Rev2.31

lowest priority interrupt handler, the microcontroller immediately enters hibernation mode.

How to exit deep sleep mode.

If the WFI instruction is executed to enter sleep mode, any of the peripheral interrupts that are responded to by the nested vector interrupt controller

(Peripheral module interrupts that can run under Deep Sleep) can wake up the system from hibernation mode. For wake-up settings, refer to the **3.4Wake-on-Inside Control WIC**.

Deep Sleep Mode		
Cortex-M0+	SWD	XTH
FLASH	UART0-1	RCH
RAM	SPI	ADC
TIM0-2	I2C	RESET
TIM4-6	CRC	POR/BOR
PCA	XTL	LVD
LPUART	RCL	VC
LPTIM	RTC	CLKTRIM
GPIO	WDT OSC	WDT

Modules in gray are not working in
the current state.

Table 2-3 Diagram of runnable modules in deep hibernation mode

System control register (**Cortex-M0+ kernel system control register**) Address: **0xE000ED10**

Reset value: **0x0000 0000**

position	Marker	Function Description	Reading and writing
31:5	RESERVED	Reserved	
4	SEVONPEND	When set to1, each new interrupt hang will generate an event if WFE hibernation is used, which can be used to wake up the processor	RW
3	RESERVED	Reserved	
2	SLEEPDEEP	When set to1, WFI is executed to enter deep hibernation and this product enters Deep sleep mode When set to0, WFI is executed to enter sleep and the product enters sleep/Idle Mode	RW
1	SLEEPONEXIT	When set to1, the processor automatically enters sleep mode (WFI) when exiting exception handling and returning to the program thread When set to0, this feature is automatically disabled	RW
0	RESERVED	Reserved	

After entering deep hibernation, there are two options for the system clock after waking up, the default is to use the clock that entered deep hibernation, after configuring register **SYSCTRL0.wakeup_byRCH** to use the internal high speed clock RCH after waking up regardless of the clock before entering deep hibernation. if you use external crystal oscillation this setting can speed up waking up the system.

3 System Controller (SYSCTRL)

3.1 Introduction of clock source

The clock control module mainly controls the system clock and peripheral clocks, and can be configured with different clock sources as system clocks, different system clock dividers, and peripheral clocks can be enabled or disabled. In addition, to ensure the accuracy of the oscillator, the internal clocks are calibrated.

This product supports the following four different clock sources as system clocks.

- Internal high-speed RC clock RCH (output frequency of 4~24MHz)
- Internal low-speed RC clock RCL (38.4K and 32.768K configurable)
- External high-speed crystal clock XTH
- External low-speed crystal clock XTL

Caution.

- When switching the clock source of the system clock, please follow the procedure strictly, as described in the chapter3.2.
- XTL can input 32.768KHz clock signal directly from P14 without crystal.
- XTH can input 4~32MHz clock signal directly from P01 without crystal.

This product also contains the following two auxiliary clocks.

- Internal low-speed 10K clock; for watchdog and CLKTRIM module use only.
- Internal 150K clock: for LVD and VC modules only. The following diagram shows the clock architecture of this product.

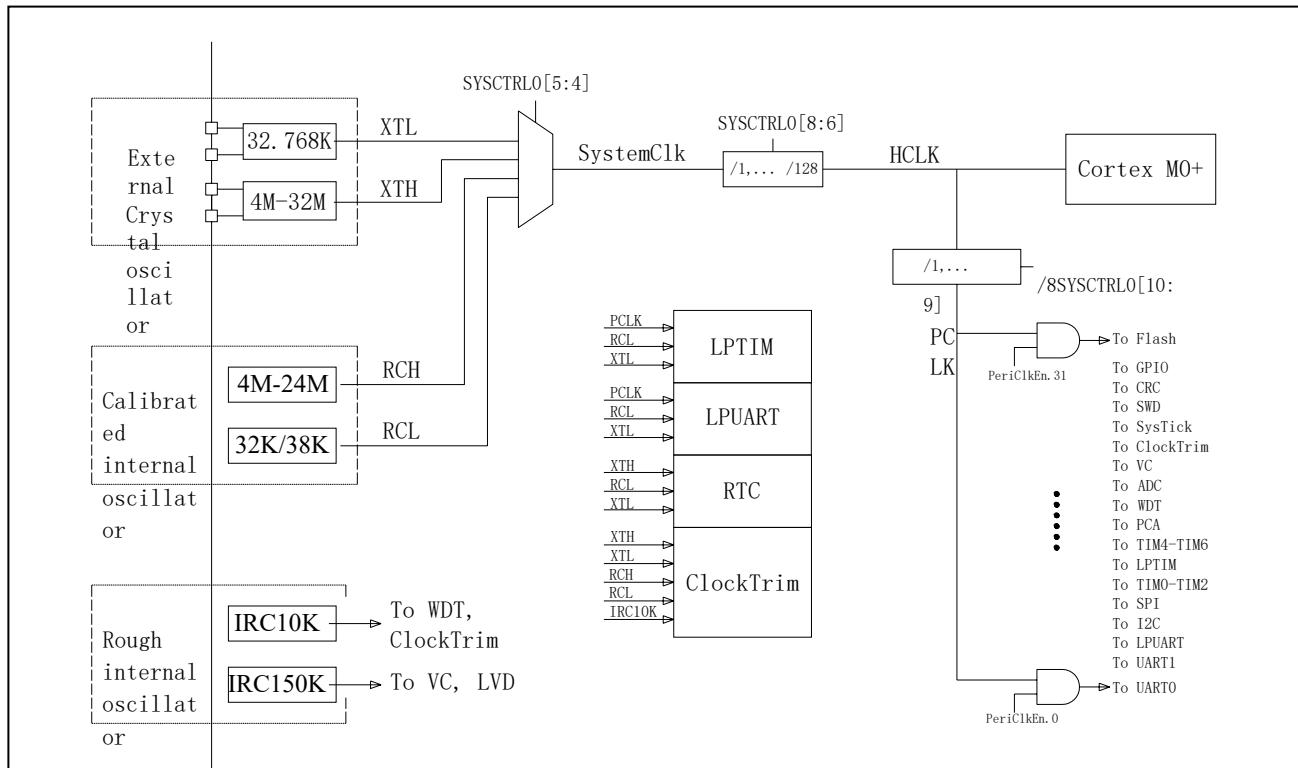


Figure 3-1 Block diagram of clock control module

3.1.1 Internal high-speed RC clock RCH

The default clock source after power-on or reset is the internal high-speed clock with a frequency of 4MHz; when the system enters Deep Sleep, this high-speed clock will be automatically turned off.

The output frequency of RCH can be adjusted by changing the value of register RCH_CR[10:0]. Each time the register value is increased, the output frequency of RCH is increased by about 0.2%, with a total adjustment range of 4 to 24 MHz.

The factory pre-tuned 5 frequencies 4MHz, 8MHz, 16MHz, 22.12MHz, 24MHz; if you need other frequencies, please adjust the value of this register manually.

Changing the RCH output frequency requires following a specific change in timing, as detailed in the System Clock Switching section.

The internal high-speed clock takes only 4us from start-up to stabilization, and it is recommended to switch the system clock to RCH before entering deep sleep mode in order to respond quickly to interrupts in deep sleep mode.

3.1.2 Internal low-speed RC clock RCL

The internal low-speed clock can be selected by register RCL_CR[9:0], and the available frequencies are 38.4KHz, 32.768KHz. when the system enters Deep Sleep, this low-speed clock will not be turned off automatically, and the ultra-low-power peripheral module can select RCL as its clock.

3.1.3 External low-speed crystal clock XTL

The external low-speed crystal clock requires an external 32.768KHz low-power crystal with ultra-high accuracy and ultra-low power consumption. This low-speed clock is not automatically turned off when the system goes into Deep Sleep. Peripheral modules operating in ultra-low power mode can choose XTL as their clock.

XTL can also be used to input a 32.768KHz clock signal directly from the P14 pin without a crystal. To input the clock signal from P14, configure P14 pin as GPIO input; set SYSCTRL1. EXTL_EN to1; set SYSCTRL0. XTL_EN to1.

Caution.

- The crystal and its mating devices are subject to the requirements for the low-speed external clock XTL in the electrical characteristics of the data sheet.

3.1.4 External high-speed crystal clock XTH

The external high speed crystal clock requires an external high speed crystal of MHz 4~32MHz. When the system enters Deep Sleep, this high-speed clock will be automatically turned off.

XTH can also be connected without crystal and input 4MHz ~ 32MHz clock signal directly from P01 pin. To input the clock signal from P01, configure P01 pin as GPIO input, set SYSCTRL1. EXTH_EN to1; set SYSCTRL0.

Caution.

- The crystals and their mating devices are subject to the requirements for the high-speed external clock XTH in the electrical characteristics of the data sheet.

3.1.5 Clock start-up process

The above four clock sources all require start-up stabilization time, and the following figure illustrates the start-up stabilization process of the clock with an external XTH as an example.

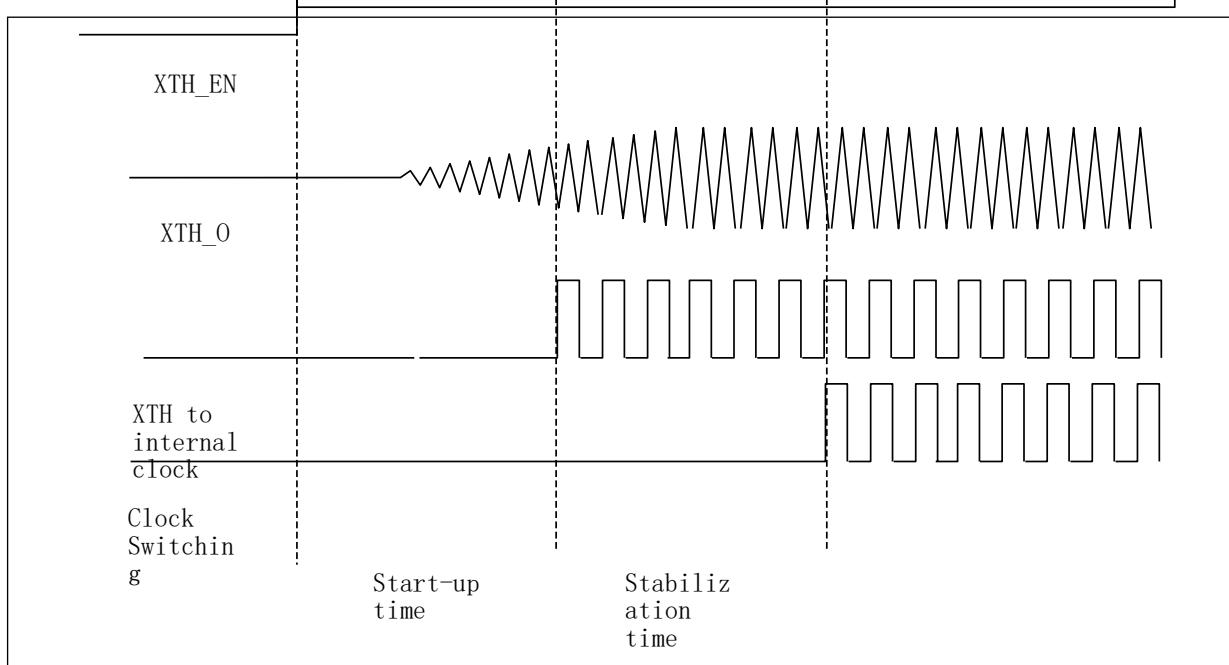


Figure 3-2 Crystal clock start-up diagram

3.2 System clock switching

The clock source of the system clock can be switched between RCH, RCL, XTH, and XTL via SYSCTRL0[5:4]. The clock switching operation must be performed according to the standard clock switching procedure, otherwise an exception may occur. If the frequency of the new clock source is greater than 24MHz, FLASH_CR.WAIT needs to be set to 1.

Note: To rewrite FLASH_CR.WAIT, you need to write 0x5A5A, 0xA5A5 to FLASH_BYPASS register and then assign the value to FLASH_CR.WAIT, see FLASH controller chapter.

3.2.1 Standard clock switching process

The operation process is as follows.

Step1: If the new clock source requires an external pin, set the pin to the appropriate mode.

Note: Analog pin is required when connecting to external crystal; GPIO input and enable external clock input is required when connecting to external clock input.

Step2: Configure the oscillation

parameters of the new clock

source. **Step3:** Enable the

oscillator of the new clock

source.

Step4: According to the higher frequency of the current clock source and the new clock source, configure the Flash controller chapter flow FLASH_CR.WAIT.

Step5: Wait for the new clock source to output a stable frequency.

Step6: Configure SYSCTRL0.clk_sw4_sel to select the source of the system clock as the new clock source. **Step7:** Configure FLASH_CR.WAIT according to the frequency of the new clock source as per the Flash Controller chapter flow. **Step8:** Turn off the clock source that is no longer in use.

3.2.2 Switching from RCH to XTL Example

The operation process is as follows.

Step1: Set P1ADS.P1ADS4 and P1ADS.P1ADS4 as 1, and configure P14/P15 pins as analog ports.

Step2: According to the crystal characteristics, configure XTL_CR.Driver and XTL_CR.Startup.

Step3: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register in turn to enable register rewrite. step3: Set SYSCTRL0.

Step6: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register in order to enable register rewriting.

Step7: Set SYSCTRL0.clk_sw4_sel to3 switch the system clock to XTL.

Step8: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register in turn to enable register rewrite. Step9: Set SYSCTRL0.rch_en to0 turn off the RCH oscillator.

3.2.3 Switching from RCH to XTH Example

The operation process is as follows.

Step1: Set P0ADS.P0ADS1 and P0ADS.P0ADS1 as 1, and configure P01/P02 pins as analog ports.

Step2: Configure XTH_CR.Driver according to the crystal characteristics.

Step3: Set XTH_CR.Startup to 3, and select the longest crystal stabilization time.

Step4: Write 0x5A5A, 0xA5A5 to 1 SYSCTRL2 register in order to enable register rewriting.

Step7: After the query waits for XTH_CR.stable flag to change1, the software delays for more than 10ms and the crystal outputs a stable clock.

Step8: Write 0x5A5A, 0xA5A5 to 1 SYSCTRL2 register in order to enable register rewrite. Step11: Set SYSCTRL0.RCH_EN to0 turn off the RCH oscillator.

The following diagram shows the clock switching timing.

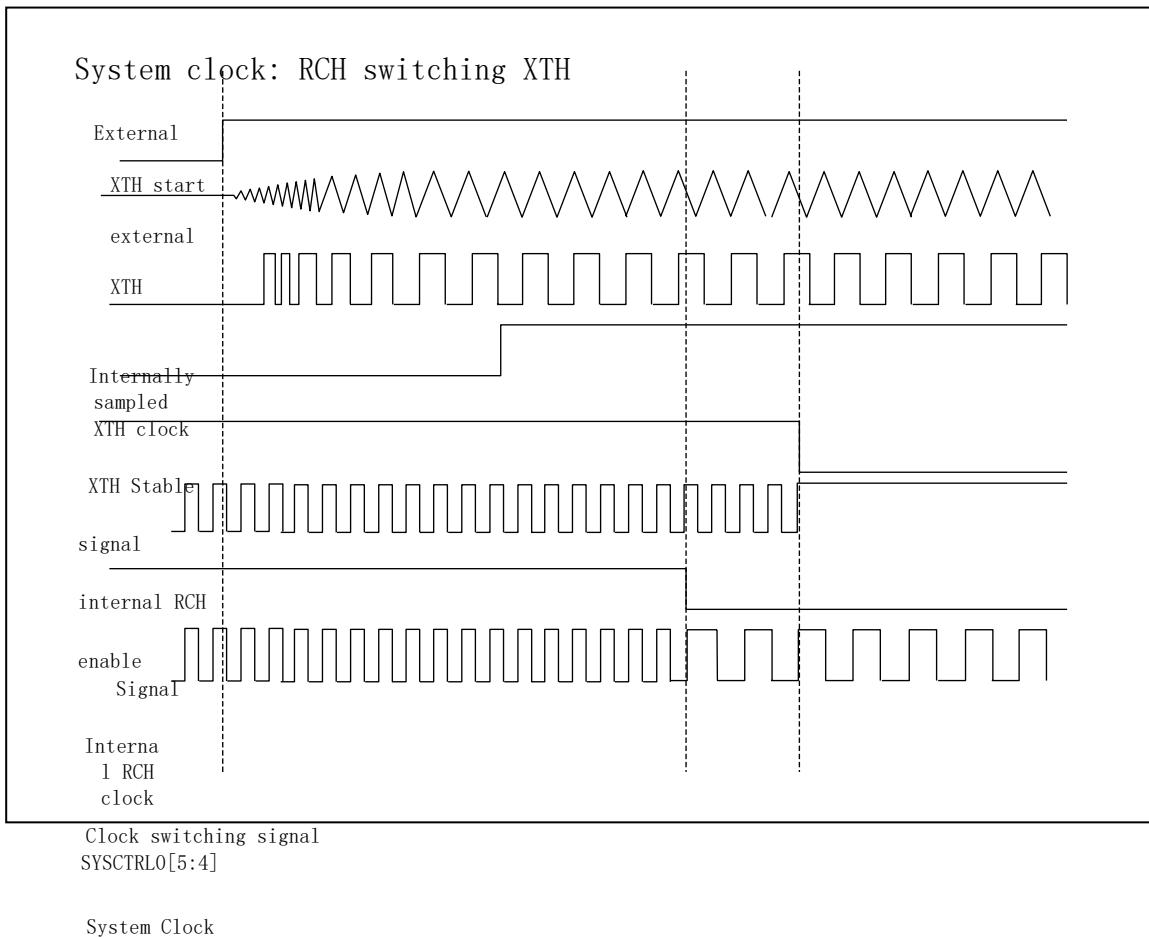


Figure 3-3 Clock switching diagram

3.2.4 Switching from RCL to XTH Example

The operation process is as follows.

Step1: Set P0ADS.P0ADS1 and P0ADS.P0ADS1 as 1, and configure P01/P02 pins as analog ports.

Step2: Configure XTH_CR.Driver according to the crystal characteristics.

Step3: Set XTH_CR.Startup to 3, and select the longest crystal stabilization time.

Step4: Write 0x5A5A, 0xA5A5 to 1 SYSCTRL2 register in order to enable register rewriting.

Step7: After the query waits for XTH_CR.stable flag to change1, the software delays for more than 10ms and the crystal outputs a stable clock.

Step8: Write 0x5A5A, 0xA5A5 to 1 SYSCTRL2 register in order to

Step10: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register to enable register rewrite. Step11: Set SYSCTRL0.

3.2.5 Switching from RCH to RCL Example

The operation process is as follows.

Step1: Configure RCL_CR.TRIM and RCL_CR.Startup.

Step2: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register in order1 to enable register rewrite. step3: Set SYSCTRL0. Step4: Query and wait for RCL_CR.Stable flag to become1 , RCL output stable clock.

Step5: Write 0x5A5A, 0xA5A5 to2 SYSCTRL2 register in order to enable register rewriting.

Step7: If you want to turn off the RCH oscillator, perform the following operations: write 0xAA55 and 0xA5A5 to the SYSCTRL2 register in order to enable the register rewrite; set SYSCTRL0.RCH_EN to0 turn off the RCH oscillator.

3.2.6 Switching from RCL to RCH Example

The operation process is as follows.

Step1: Configure RCH_CR.TRIM.

Step2: Write 0x5A5A, 0xA5A5 to1 SYSCTRL2 register in order to enable register rewrite. step3: Set SYSCTRL0.

Step5: Write 0x5A5A, 0xA5A5 to0 SYSCTRL2 register in order to enable register rewriting.

Step7: If you want to turn off the RCL oscillator, perform the following operations: write 055xAA and 0xA5A5 to the SYSCTRL2 register in order to enable the register rewrite; set SYSCTRL0. RCL_EN to0 turn off the RCL oscillator.

3.2.7 RCH Switching between different oscillation frequencies

There are two options for switching
between different oscillation
frequencies of the RCH. **Scheme1**

Step1: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register in order to
enable register rewriting.

Step3: Adjust the output frequency of RCH up or down step by step, 4M -> 8M ->
16M -> 24M/22.12M or

24M/22.12M -> 16M -> 8M -> 4M.

Step4: Write 0x5A5A, 0xA5A5 to SYSCTRL2 register in order to
enable register rewrite.

The sample code for switching from 4M to 24M is shown below.

```

M0P_SystemCtrl->SYSCTRL2 = 0X5A5A;
M0P_SystemCtrl->SYSCTRL2 = 0XA5A5;
M0P_SystemCtrl->SYSCTRL0_f.HCLK_PRS = 7;
M0P_SystemCtrl->RCH_CR    =*
                (*(0X00100C08))      ;
                //4M
M0P_SystemCtrl->RCH_CR    =*
                (*(0X00100C06))      );
                //8M
M0P_SystemCtrl->RCH_CR
=*((uint16*)(0X00100C04)); //16M M0P_SystemCtrl->RCH_CR
=*((uint16*)(0X00100C04)); //16M RCH_CR    =*
                (*(0X00100C04))      )
                //16MM0P_SystemCtrl->RCH_CR
                =*
                (uint16*)(0X00100C00
) ); //24MM0P_SystemCtrl-

```

```
>SYSCTRL2 = 0X5A5A;
```

```
M0P_SystemCtrl->SYSCTRL2 = 0XA5A5;
```

```
M0P_SystemCtrl->SYSCTRL0_f.HCLK_PRS = 0;
```

Program2 :

Step1: Switch the system clock to RCL, see [Switching from 3.2.5RCH to RCL example](#).

Step2: Change the value of `RCH_CR.TRIM` to change the frequency of RCH oscillation.

Step3: Switch the system clock to RCH, see the example of switching from [3.2.6RCL internal low speed to RCH](#).

3.3 Clock Calibration Module

When the reference clock and the calibrated clock are selected, set the register REFcnt value and set cali.start to start the clock calibration circuit, then two bit32 counters (incremental and decremental) work at the same time, when the decremental counter is equal to 0, cali.finish is set to indicate the end of calibration. When the decremental counter is equal to, cali.finish is set,

indicating the end of calibration, then the software can read the CALCNT value, so that it is easy to get the frequency relationship between the reference clock and the calibrated clock.

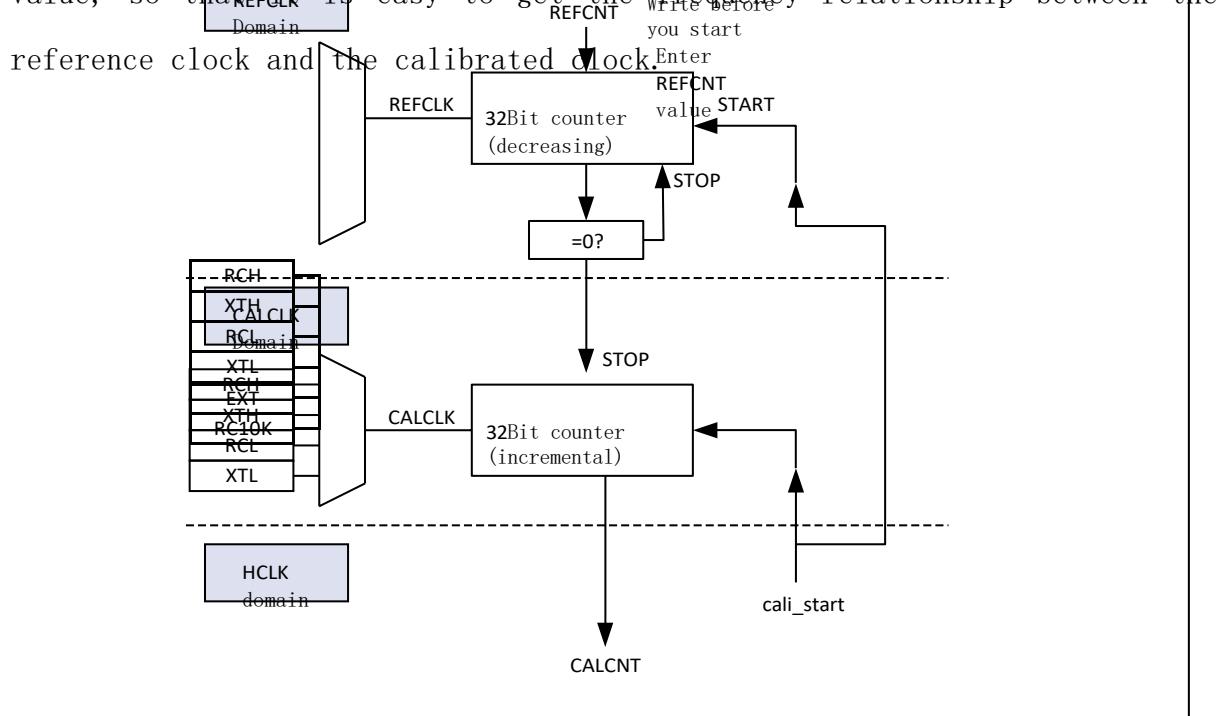


Figure 3-4 Clock calibration schematic

3.4 Interrupt wake-up control

When the processor executes a WFI instruction and enters the hibernate state, it stops executing the instruction. The processor wakes up when an interrupt request (higher priority) occurs in the hibernate state and needs to be processed.

The behavior of a processor in the dormant state when it receives an interrupt request is shown in the following table.

PRIMASK Status	WFI Behavior	Awakeni ng	ISR Executi on
0	IRQ Priority > Current Level	Y	Y
0	IRQ priority \leq Current level	N	N
1	IRQ Priority > Current Level	Y	N
1	IRQ priority \leq Current level	N	N

3.4.1 Method of executing interrupt service program after waking up from deep sleep mode

1. Enables the NVIC corresponding to the module that needs to wake up the processor
2. Enable the interrupt corresponding to the module that needs to wake up the processor
3. Set SCB->SCR.SLEEPDEEP to 1
4. Execute the WFI command to enter deep sleep mode
5. The system enters deep sleep mode and waits for an interrupt to wake up, after waking up, the interrupt service routine is executed.

```
SCB_SCR |= 0x00000004u;
```

```
while(1)
```

```
{
```

```
  asm("WFI");
```

3.4.2 Method of waking up from deep sleep mode without executing interrupt service program

1. Enables the NVIC corresponding to the module that needs to wake up the processor
2. Enable the interrupt corresponding to the module that needs to wake up the processor
3. Set PRIMASK to 1
4. Set SCB->SCR.SLEEPDEEP to 1

5. Execute the WFI command to enter deep sleep mode
6. The system enters deep hibernation mode and waits for an interrupt to wake up, and executes the next instruction after waking up
7. Clear interrupt flag, clear interrupt pending status routine.

```

asm("CPSID      I");//Set
PRIMASKSCB_SCR |= 0x00000004u;
while(1)
{
    asm("WFI");
    BTIMERLP_REG-
>TFCR_f.TFC=0;/          /Clear Int Flag
NVIC_ClearPendingIRQ(BASE_TIMER3_IRQn); //Clear Pending
Flag
}

```

3.4.3 Using the exit hibernation feature

Sleep-on-exit is ideal for interrupt-driven applications. When this feature is enabled, the processor enters sleep mode as soon as it has completed exception handling and returned to thread mode. With the sleep-on-exit feature, the processor can stay in sleep mode for as long as possible.

Cortex-M0 uses the exit hibernation feature to enter hibernation, which is similar to the effect of executing WFI immediately after an exception exit. However, the processor does not perform the process of exiting the stack so that the next time it enters an exception, it does not have to perform a stack press operation.

1. Enables the NVIC corresponding to the module that needs to wake up the processor
2. Enable the interrupt corresponding to the module that needs to wake up the processor
3. Set SCB->SCR.SLEEPDEEP to 1

-
4. Set SCB->SCR.SLEEPONEXIT to 1
 5. Execute the WFI command to enter deep sleep mode
 6. The system enters deep sleep mode and waits for interrupts to wake up, then executes the interrupt service subroutine after waking up
 7. Automatic sleep mode routines
when exiting interrupt service.

```
SCB_SCR |= 0x00000004u;  
SCB_SCR |= 0x00000002u;  
while(1)  
{  
    asm("WFI");  
}  
}
```

3.5 Register

Base address 0x40002000

Register	Offset Address	Description
SYSCTRL0	0x000	System Control Register0
SYSCTRL1	0x004	System Control Register1
SYSCTRL2	0x008	System Control Register2
RCH_CR	0x00C	RCH control register
XTH_CR	0x010	XTH control register
RCL_CR	0x014	RCL control register
XTL_CR	0x018	XTL control register
PERI_CLKEN	0x020	Peripheral module clock control register
SYSTICK_CR	0x034	Systick clock control

Table 3-1 System Control Register Table

3.5.1 System Control Register0 (SYSCTRL0)

Offset address: 0x000

Reset value: 0x0000 0001

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

Wakeup_byRCH	Reserved	PCLK_PRS	HCLK_PRS	clk_sw4_sel	XTL_EN	RCL_EN	XTH_EN	RCH_EN
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function Description
31:16	Reserved	Reserved
15	wakeup_byRCH	1: After waking up from Deep Sleep, the system clock source is RCH and the original clock continues to be enabled. 0: After waking up from Deep Sleep, the system clock source is not changed.
14:11	Reserved	Reserved
10:9	PCLK_PRS	PCLK crossover selection00: HCLK 01: HCLK/2 10: HCLK/4 11: HCLK/8
8:6	HCLK_PRS	HCLK crossover selection000: SystemClk 001: SystemClk/2010 : SystemClk/4011 : SystemClk/8100 : SystemClk/16101 : SystemClk/32110 : SystemClk/64 111: SystemClk/128
5:4	Clk_sw4_sel	System clock source selection 00: Internal high-speed clock RCH 01: External high-speed crystal XTH 10: Internal low-speed clock RCL 11: External low-speed crystal XTL

3	XTL_EN	External low-speed crystal XTL enable control 0: Close 1: Enabled Note: P14 and P15 need to be set as analog ports.
---	--------	--

2	RCL_EN	Internal low-speed clock RCL enable control 0: Close 1: Enabled
1	XTH_EN	External High Speed Crystal XTH Enable Control 0: Close 1: Enabled Note: When the system enters DeepSleep, this high-speed clock will be automatically turned off.
0	RCH_EN	Internal high-speed clock RCH enable signal. 0: Close 1: Enabled Note: When the system enters DeepSleep, this high-speed clock will be automatically turned off.

Caution.

- This step can effectively prevent the misoperation of SYSCTRL0 and SYSCTRL1 registers.

3.5.2 System Control Register1 (SYSCTRL1)

Offset address:

0x004 Reset value:

0x00000008

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

Reserved	RTC_FREQ_ADJUST	SWD_UIO	RES_UIO	LOCK_EN	RTC_LPW	Res	XTL_alwayson	EXTL_EN	EXTH_EN	Re s
	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	

position	Marker	Function Description
31:12	Reserved	
11:9	RTC_FREQ_ADJUST	RTC high-speed clock compensation clock frequency selection 000 4M; 001 6M; 010 8M; 011 12M 100 16M; 20M101; 24M110; 32M111;
8	SWD_USE_IO	SWD port function configuration 0: SWD port 1: GPIO port
7	RESET_USE_IO	RESET port function configuration 0: RESET port 1: GPIO port
6	LOCKUP_EN	Cortex-M0+ LockUp Function Configuration 0: Close 1: Enabled Note: When enabled, the CPU will reset the MCU when it reads an invalid instruction, which can enhance system reliability.
5	RTC_LPW	RTC module low power control 1: Low power mode enable 0: Low power mode disable Note: When enabled, the RTC module enters a low-power state and its registers cannot be read or written.
4	Reserved	
3	XTL_ALWAYS_ON	XTL Advanced Enable Control 1SYSCTRL0.XTL_EN can be set only. 0SYSCTRL0.XTL_EN can be set or cleared.

2	EXTL_EN	External XTL clock input control 1: XTL output clock is input from P14. 0: The XTL output clock is generated by a crystal. Note: When using P14 input clock, SYSCTRL0.XTL_EN should be set to 1.
---	---------	---

1	EXTH_EN	External XTH input control 1: XTH output clock is input from P01. 0: The XTH output clock is generated by a crystal. Note: When using P01 input clock, you need to set SYSCTRL0.XTH_EN as1.
0	Reserved	

Caution.

- This step can effectively prevent the misoperation of SYSCTRL0 and SYSCTRL1 registers.

3.5.3 System Control Register2 (SYSCTRL2)

Offset address:

0x008 Reset value:

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCTRL2															
WO															

position	Marker	Function Description
31:16	Reserved	
15:0	SYSCTRL2	<p>Register SYSCTRL0 and SYSCTRL1 protect the series control registers, write 0x5A5A to SYSCTRL2 first, then write 0xA5A5 to start the write operation for register SYSCTRL0 and SYSCTRL1, as long as the write operation for register SYSCTRL0 and SYSCTRL1, this protection bit automatically returns to the protection status.</p> <p>Need to rewrite the series to open the protection.</p>

3.5.4 RCH Control Register (RCH_CR)

Offset address:

0x00C Reset value:

0x00000126

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				Stable	TRIM										R/W
				RO											

position	Marker	Function Description
31:12	Reserved	
11	stable	<p>Internal high-speed clock RCH stability flag bit. 1: means that the RCH is stable and can be used by the internal circuit. 0: means that the RCH is not stable and cannot be used by the internal circuit.</p>
10:0	TRIM	<p>Clock frequency adjustment, change the value of this register can adjust the output frequency of RCH. Each increase of the register value increases the output frequency of RCH by about 0.2%, and the total adjustment range is 4~24MHz. The calibration value of the group frequency is already stored in Flash, read out the calibration value in Flash and write it to RCH_CR.TRIM to get the accurate frequency.</p> <p>24M calibration value address: 0x00100C00 - 0x00100C01 22.12M calibration value address: 0x00100C02 - 0x00100C03 16M calibration value address: 0x00100C04 - 0x00100C05 8M calibration value address: 0x00100C06 - 0x00100C07 4M calibration value address: 0x00100C08 - 0x00100C09 RCH output frequency changes need to follow a specific timing sequence, as detailed in the System Clock Switching3.2.7 chapter.</p>

3.5.5 Oscillation XTH control register (XTH_CR)

Offset address:

0x010 Reset value:

0x00000022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Stable	Startup	Driver					
								RO	R/W						R/W

position	Marker	Function Description
31:6	Reserved	
6	stable	<p>External high-speed clock XTH stability flag bit. 1: means that the XTH is stable and can be used by the internal circuitry. 0: means XTH is not stable and cannot be used by internal circuits. Note: In order to increase the reliability of the system, after the flag is queried, a software delay of 10ms or more is required before the system clock can be switched to XTH.</p>
5:4	Startup	<p>External high-speed clock XTH stable time selection 00:256 Cycle; 01:1024 Cycle; 10:4096 cycles; 11: 16384 cycles; Note: It is highly recommended to set the XTH stabilization time 11. If the XTH stabilization time is not sufficient, the system will not work stably during clock switching or when waking up from deep hibernation.</p>

		3:2Freq selects the operating frequency of the crystal 11: 24M~32M 10: 16M~24M 01: 8M~16M 00: 4M~8M
3:0	Driver	<p>1:0Driver selects the drive capability of the crystal 11: The strongest drive capability 10: Default drive capability (recommended value) 01: Weak drive capability 00: Weakest drive capability</p> <p>Note: You need to select the appropriate drive capability according to the crystal characteristics, load capacitance and parasitic parameters of the board. The larger the drive capability, the greater the power consumption; the weaker the drive capability, the smaller the power consumption.</p>

3.5.6 RCL control register (RCL_CR)

Offset address: 0x014

Reset value: 0x0000033Fh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	Stable	Startup	TRIM												
	RO	R/W	R/W												

position	Marker	Function Description
31:13	Reserved	
12	stable	Internal low-speed clock RCL stability flag bit. 1: means that the RCL is stable and can be used by the internal circuitry. 0: means that RCL is not stable and cannot be used by internal circuits.
11:10	Startup	Internal low-speed clock RCL stabilization time selection 11:256 cycles; 10:64 cycles; 01:16 Cycle; 00:4 Cycle;
9:0	TRIM	Internal low-speed clock frequency adjustment, with 2group frequency calibration values stored in Flash. The calibration value in Flash is read out and written to RCL_CR.TRIM to obtain the accurate frequency. 38.4K calibration value address: 0x00100C20 - 0x00100C21 32.768K calibration value address: 0x00100C22 - 0x00100C23

3.5.7 XTL Control Register (XTL_CR)

Offset address:

0x018 Reset value:

0x00000021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Stable	Startup	Driver					
								RO	R/W						R/W

position	Marker	Function Description
31:6	Reserved	
6	stable	External low-speed crystal XTL stabilization flag bit. 1: means that the XTL is stable and can be used by the internal circuitry. 0: means that XTL is not stable and cannot be used by internal circuits.
5:4	Startup	External low-speed crystal XTL stabilization time selection 00: 256 cycles; 01: 1024Cycle; 10: 4096cycles; 11: 16384cycles;
3:0	Driver	External low-speed crystal XTL drive option 1111: Maximum drive 0000: Minimum drive 3:2 Amp_controlXTLfine adjustment of oscillation amplitude. 11: Maximum amplitude 10: Larger amplitude (recommended value) 01: Normal amplitude 00: Minimum amplitude 1:0 Driver external crystal drive capability selection 11: The strongest drive capability 10: Stronger drive capability 01: Default drive capability (recommended) 00: The weakest drive capability. Note: It is necessary to select the appropriate drive energy according to the crystal characteristics, load capacitance and parasitic parameters of the board.

		Power. The higher the drive capability, the higher the power consumption; the weaker the drive capability, the lower the power consumption. 00: Weakest drive capability
--	--	--

3.5.8 Peripheral module clock control register (PERI_CLKEN)

Reset value: 0xC080_0000

Offset address: 0x020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Flash	Res.	GPIO	Res.	CRC	Res.	TICK	Res.	Trim	RTC	Res.	VC	ADC			
R/W		R/W		R/W		R/W		R/W	R/W		R/W	R/W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDT	PCA	Res.	ADV	LP	BASE	Res.	SPI	Res.	I2C	Res.	LPUART	UART1	UART0		
R/W	R/W		TIM	TIM	TIM		R/W	R/W	R/W		R/W	R/W	R/W		

position	Marker	Function Description
31	flash	Flash controller module clock enable, which is required to enable the clock when operating the flash registers 1: Enabled;0 : Off Note: This bit is not affected by executing programs from flash.
30:29	Res.	Reserved Bits
28	GPIO	GPIO module clock enable. 1: Enabled;0 : Off
27	Res.	Reserved Bits
26	CRC	CRC module clock enable. 1: Enabled;0 : Off
25	Res.	Reserved Bits
24	TICK	SysTick timer reference clock enable. 1: Enabled;0 : Off
23:22	Res.	Reserved Bits
21	Trim	CLKTRIM module clock enable. 1: Enabled;0 : Closed
20	RTC	RTC module clock enable. 1: Enabled;0 : Closed
19:18	Res.	Reserved Bits
17	VC	VC, LVD, module clock enable. 1: Enabled;0 : Closed
16	ADC	ADC module clock enable. 1: Enabled;0 : Closed
15	WDT	WDT module clock enable. 1: Enabled;0 : Off

14	PCA	PCA module clock enable. 1: Enabled;0 : Closed
13:11	Res.	Reserved Bits
10	ADVTIM	Timer456 Module clock enable. 1: Enabled;0 : Closed
9	LPTIM	LPTimer module clock enable. 1: Enabled;0 : Closed
8	BASETIM	Timer012 Module clock enable. 1: Enabled;0 : Closed
7	Res.	Reserved Bits
6	SPI	SPI module clock enable. 1: Enabled;0 : Off
5	Res.	Reserved Bits
4	I2C	I2C module clock enable. 1: Enabled;0 : Closed
3	Res.	Reserved Bits
2	LPUART	LPUART module clock enable. 1: Enabled;0 : Closed
1	UART1	UART1 module clock enable. 1: Enabled;0 : Closed
0	UART0	UART0 module clock enable. 1: Enabled;0 : Closed

3.5.9 SysTick clock control (SYSTICK_CR)

Reset value:

0x0100_0147 Offset

address: 0x034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved						CLK_SEL	NO REF	SKE W	STCALIB [23:16]							
									R/W							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
STCALIB [15:0]																R/W

position	Marker	Function Description
31-28	Reserved	
27-26	CLK_SEL	SysTick external reference clock selection 00: External low-speed clock XTL 01: Internal low-speed clock RCL 10: System clock8 divider SystemClk/811 : External high-speed clock XTH
25	NOREF	SysTick clock source selection 1: Internal high frequency clock HCLK 0: External reference clock, selected by SYSTICK_CR[27:26] Note: When using an external reference clock, the reference clock frequency is not allowed to be higher than HCLK
24	SKEW	STCALIB Accuracy indication 1: STCALIB value represents roughly 10ms 0: STCALIB value represents the exact 10ms
23:0	STCALIB	10Millisecond calibration value when the reference clock is XTL

4 Reset controller (RESET)

4.1 Introduction of reset controller

Each reset signal allows the CPU to re-run, most registers are reset to their reset values, and the program counter PC is reset to point 00000000 to.

- POR/BOR reset (VCC domain and Vcore domain)
- External Reset PAD reset
- WDT reset
- PCA reset
- LVD reset
- Cortex-M0+ SYSRESETREQ Software Reset
- Cortex-M0+ LOCKUP Hardware Reset

Each reset source is indicated by the corresponding reset flag. The reset flags are set by hardware and need to be cleared by user software. If `Reset_flag.POR15V` or `Reset_flag.POR5V` is queried when the chip is reset, it is a power-on **reset**. The user program should clear the `Reset_flag` register during power-on **reset**, so that the reset source can be determined by the bits of `Reset_flag` during the next reset.

The following figure depicts the source of the reset in each region.

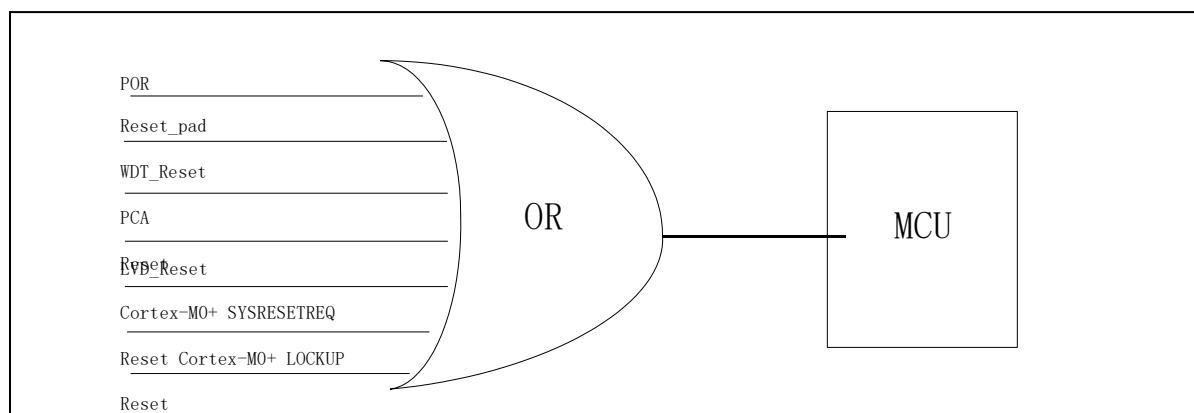


Figure 4-1 Schematic diagram of reset source

4.1.1 Power-up Power-down Reset POR/BOR

This product has two power supply areas: VCC area and Vcore area. All analog modules and IO operate in VCC

area; the other modules work in the Vcore area.

When the VCC area is powered up, the POR5V signal is generated when the VCC voltage is below the POR threshold voltage (typical value is 1.65V). When the VCC area is powered down, the POR5V signal is generated when the VCC voltage is below the BOR threshold voltage (typical value is 1.5V). The POR5V signal is generated when the Vcore voltage is below the POR threshold voltage when the Vcore area is powered up, and the POR15V signal is generated when the Vcore voltage is below the BOR threshold voltage when the Vcore area is powered down.

The POR15V signal is generated when the Vcore voltage is below the POR threshold voltage when the Vcore area is powered up, and the POR15V signal is generated when the Vcore voltage is below the BOR threshold voltage when the Vcore area is powered down.

Both POR5V signal and POR15V signal will reset the chip's registers to the initialized state.

4.1.2 External reset pin reset

A system reset is generated when the external reset pin detects a low level. This reset pin has a built-in pull-up resistor and an integrated burr filter circuit. The burr filter circuit filters out burr signals less than 20us (typical), so the low signal added to the reset pin must be greater than 20us to ensure a reliable reset of the chip.

4.1.3 WDT reset

See the WDT chapter for a description of the watchdog reset.

4.1.4 PCA reset

For PCA reset, please refer to the PCA chapter for instructions.

4.1.5 LVD Low Voltage Reset

For LVD reset, please refer to the LVD chapter for instructions.

4.1.6 Cortex-M0+ SYSRESETREQ reset

Cortex-M0+ Software Reset

4.1.7 Cortex-M0+ LOCKUP reset

When the Cortex-M0+ encounters a serious exception, it stops its PC pointer at the current address and locks up the self

The entire CORE area is reset after a few clock cycles of delay.

4.2 Register

4.2.1 Reset flag register (Reset_flag)

Reset value: 00000000_00000000_00000000_xxxxxx11b

Address: 0x4000201C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RSTB	sysreq	lockup	PCA	WDT	LVD	Por15	Por5v
								RW0	RW0	RW0	RW0	RW0	RW0	RW0	RW0

position	Marker	Function Description
31:8	Reserved	
7	RSTB	RESETB port reset flag, need to be initialized and cleared by software, power-on state is variable 1: Port reset occurs 0: No port reset occurs
6	Sysreq	Cotrex-M0+ CPU software reset flag, requires software initialization and clearing, power-on state is variable 1: Cotrex-M0+ CPU software reset occurs 0: No Cotrex-M0+ CPU software reset occurs
5	Lockup	Cotrex-M0+ CPU Lockup reset flag, requires software initialization and clearing, power-on state is variable 1: Cotrex-M0+ CPULockupreset occurs 0: No Cotrex-M0+ CPULockupreset occurs
4	PCA	PCA reset flag, requires software initialization and clearing, power-on state is variable 1: PCA reset occurs 0: No PCA resets occur
3	WDT	WDT reset flag, need to be initialized and cleared by software, power-on state is variable 1: WDT reset occurs 0: No WDT reset occurs
2	LVD	LVD reset flag, requires software initialization and clearing, power-on state is variable 1: LVD reset occurs 0: No LVD reset occurs
1	POR15V	Vcore domain reset flag 1: Reset0 occurs in the Vcore domain: No

		reset occurs in the Vcore domain
0	POR5V	VCC power domain reset flag1: VCC power domain reset occurred 0: No reset occurs in the VCC power domain

4.2.2 Peripheral module reset control register (PERI_RESET)

Reset value: 0xD7B3C757

Address: 0x40002028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	GPIO	Res.	CRC	Res.	TICK	Res.	TRIM	RTC	Res.	VC	ADC	Res.	R/W	R/W	R/W	
			R/W		R/W		R/W	R/W								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	PCA.	Res.	ADV	LP	BASE	Res.	SPI	Res.	I2C	Res	LPUA	UART1	UART0	R/W	R/W	R/W
			TIM	TIM	TIM		R/W		R/W		RT					

position	Marker	Function Description
31:29	Res.	Reserved Bits
28	GPIO	GPIO module reset enable. 1: Normal operation;0 : Module in reset state
27	Res.	Reserved Bits
26	CRC	CRC module reset enable. 1: Normal operation;0 : Module in reset state
25	Res.	Reserved Bits
24	TICK	SYSTICK module reset enable. 1: Normal operation;0 : Module in reset state
23:22	Res.	Reserved Bits
21	TRIM	CLKTRIM module reset enable. 1: Normal operation;0 : Module in reset state
20	RTC	RTC module reset enable. 1: Normal operation;0 : Module in reset state
19-18	Res.	Reserved Bits
17	VC	VC, LVD, module reset enable. 1: Normal operation;0 : Module in reset state
16	ADC	ADC module reset enable. 1: Normal operation;0 : Module in reset state
15	Res.	Reserved Bits
14	PCA	PCA module reset enable. 1: Normal operation;0 : Module in reset state
13:11	Res.	Reserved Bits
10	ADVTIM	Timer456 module reset enable. 1: Normal operation;0 : Module in reset state

9	LPTIM	LPTimer module reset enable. 1: Normal operation;0 : Module in reset state
8	BASETIM	Timer012 module reset enable. 1: Normal operation;0 : Module in reset state
7	Res.	Reserved Bits
6	SPI	SPI module reset enable. 1: Normal operation;0 : Module in reset state
5	Res.	Reserved Bits
4	I2C	I2C module reset enable. 1: Normal operation;0 : Module in reset state
3	Res.	Reserved Bits
2	LPUART	LPUART module reset enable. 1: Normal operation;0 : Module in reset state
1	UART1	UART1 module reset enable. 1: Normal operation;0 : Module in reset state
0	UART0	UART0 module reset enable. 1: Normal operation;0 : Module in reset state

5 Interrupt Controller (NVIC)

5.1 Overview

The Cortex-M0+ processor has a built-in nested vector interrupt controller (NVIC). It supports up to one 32 interrupt request (IRQ) input and one 1non-maskable interrupt (NMI) input (not used in this product system). In addition, the processor supports multiple internal exceptions.

Each exception source has a separate exception number, and each exception type has a corresponding priority, which is fixed for some exceptions and

Exception programmable number	Exception Type	Priority	Description
1	Reset	-3 (maximum)	Reset
2	NMI	-2	Non-maskable interrupts (not used in this system)
3	Hardware error	-1	Error Handling Exceptions
4-10	Reserved	NA	...
11	SVC	Programmable	Calling the hypervisor via SVC command
12-13	Reserved	NA	...
14	PendSV	Programmable	Suspendable requests for system services
15	SysTick	Programmable	SysTick Timer
16	Interrupt #0	Programmable	External Interrupt #0
17	Interrupt #1	Programmable	External Interrupt #1
...	Table 5-1 Cortex-M0+ Processor Exception List		

This section only describes in detail the processor's external interrupt requests (interrupt #0 to interrupt #31). Also, this section only discusses the interrupt handling mechanism of the NVIC in the processor core, and the interrupt generation mechanism of the peripheral module itself is not discussed here.

5.2 Interrupt priority

Each external interrupt corresponds to a priority register, each priority is 2wide and uses the top two bits of the interrupt priority register, each register occupying one byte (8bit). With this setting, the priority levels

available are 0x00 (highest) 0x40, 0x80 and 0xc0 (lowest)

Figure 5-1 uses only the high two priority registers

Preemption occurs if the processor is already running another interrupt handler and the new interrupt has a higher priority than the one being executed. The running interrupt processing is suspended and the new interrupt is executed instead, a process often referred to as interrupt nesting. After the new interrupt is executed, the previous interrupt processing continues to execute and returns to the program thread when it is finished.

If the processor is running another interrupt with the same or higher priority, the new interrupt will wait and enter a pending state. The pending interrupt will wait until the current interrupt level changes, for example, when the currently running interrupt returns from processing and the current priority is reduced to less than the pending interrupt.

If two interrupts occur at the same time and they have the same priority, the interrupt with the lower interrupt number will be executed first. For example, if interrupt #0 and interrupt #1 are enabled and have the same priority, interrupt #0 will be executed first when they are triggered at the same time.

5.3 Interrupt vector table

When the Cortex-M0+ processor wants to process an interrupt service request, it needs to first determine the starting address for exception processing, and the required information is called a vector table, as shown in Figure 5-2. The vector table is stored at the beginning of the memory space and contains the exception (interrupt) vector of exceptions (interrupts) available in the system, as well as the initial value of the main stack pointer (MSP).

Memory Address	Exception number
0x0000004C	19
0x00000048	18
0x00000044	17
0x00000040	16
0x0000003C	15
	SysTick vector

0x00000038	PendSV vector	14
0x00000034	Not used	13
0x00000030	Not used	12
0x0000002C	SVC vector	11
0x00000028	Not used	10
0x00000024	Not used	9
0x00000020	Not used	8
0x0000001C	Not used	7
0x00000018	Not used	6
0x00000014	Not used	5
0x00000010	Not used	4
0x0000000C	Hardware error exception	3
0x00000008	NMI vectors	2
0x00000004	Reset vector	1
0x00000000	MSP initial value	0

Figure 5-2 Interrupt vector table

The interrupt vector is stored in the same order as the interrupt number, and since each vector is a 1word (4byte) the address of the interrupt vector is the interrupt number multiplied4 by , and each interrupt vector is the starting address for interrupt processing.

5.4 Interrupt input and hang behavior

In the NVIC module of the Cortex-M0+ processor, each interrupt input corresponds to a pending status register, and each register has only bits1 to hold the interrupt request, regardless of whether the request is acknowledged or not. When the processor starts processing the interrupt, the hardware will automatically clear the pending status bits.

The system's peripherals use level-triggered interrupt outputs, and when an interrupt event occurs, the signal is acknowledged because the peripheral is connected to the NVIC. This signal remains high until the processor performs interrupt service and clears the peripheral's interrupt signal. Inside the NVIC, when an interrupt is detected, the pending state of the interrupt is set and cleared when the processor receives the interrupt and starts executing the interrupt service program. The process is illustrated in Figure 5-3.

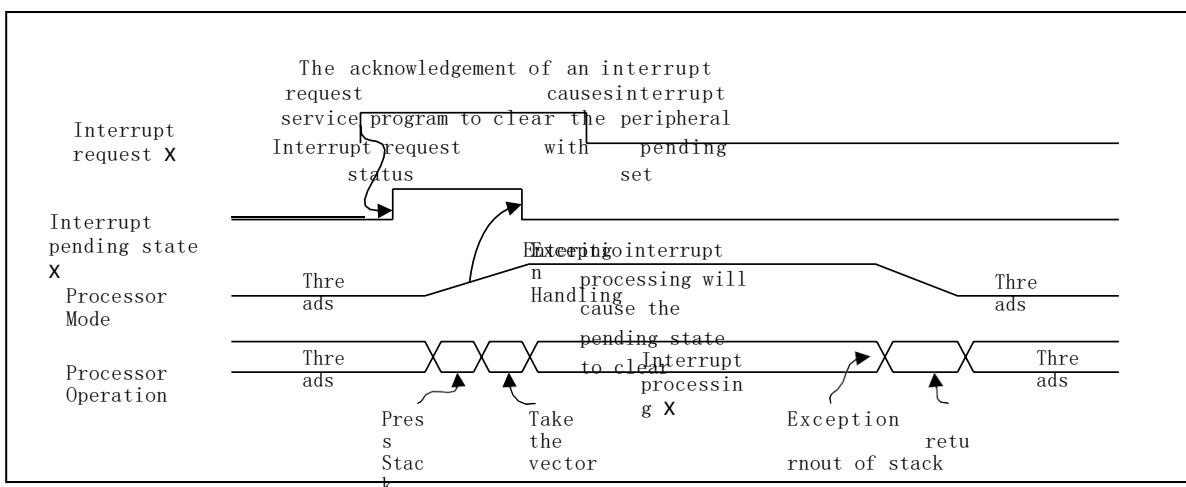


Figure 5-3 Interrupt Activation and Suspend Status

If the interrupt request is not executed immediately and is cleared by software before it is acknowledged, the processor will then ignore the current request and will not perform interrupt processing. The interrupt pending status can be cleared by writing to the NVIC_CLRPEND register. This processing is useful when setting up a peripheral that may have generated an interrupt request before it is set.

If the peripheral still holds the interrupt request when the software clears the pending state, the pending state is also generated immediately. The process is illustrated in Figure 5-4.

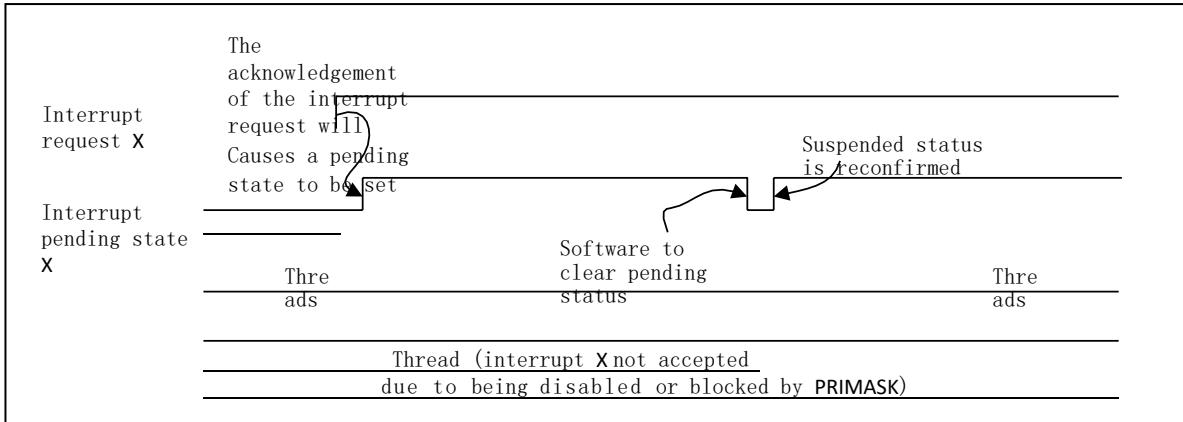
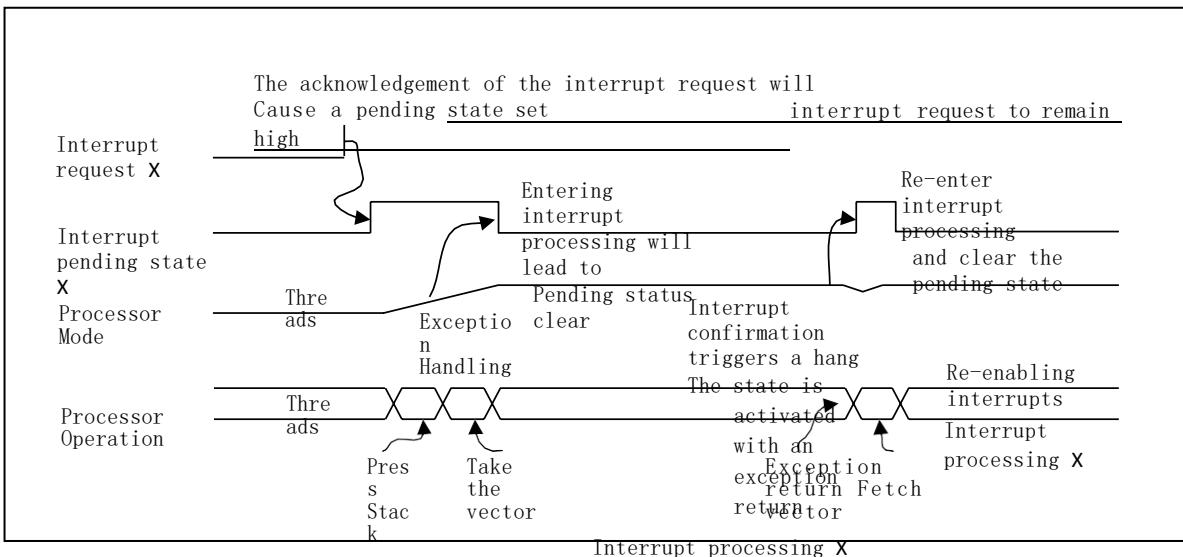


Figure 5-4 Interrupt pending state is cleared and then re-acknowledged

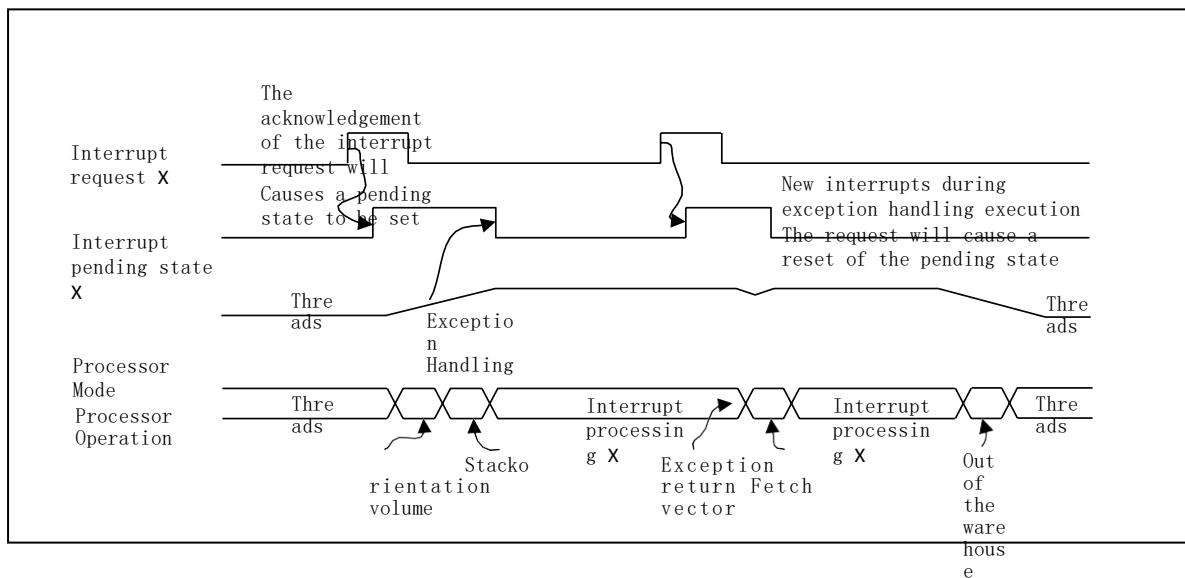
If the interrupt request generated by the peripheral is not cleared during exception handling, the pending state will be activated again after the



exception returns, so that the interrupt service program will be executed again. The process is shown in Figure 5-5.

Figure 5-5 Interrupt request held high at interrupt exit will cause interrupt processing to be executed again

If a peripheral interrupt request is generated during the execution of the terminal service program, the request is treated as a new interrupt request



and causes the interrupt service program to execute again after the current interrupt is exited. The process is shown in Figure 5-6.

Figure 5-6 An interrupt hang generated during interrupt processing can also be acknowledged

5.5 Interrupted waiting

Typically, the NVIC's interrupt wait time is one16 cycle. This wait time starts from the processor clock cycle of the interrupt acknowledgement and continues until the interrupt processing starts to execute. The following prerequisites are required to calculate interrupt wait.

- This interrupt is enabled and not blocked by SCS_PRIMASK or other exception handling being performed.
- The memory system does not have any wait state. Bus transfers are used for interrupt processing, stacking, orientation, or fetching at the beginning of interrupt processing, and if the memory system needs to wait, the wait state generated when the bus transfer occurs may make the interrupt delayed.

The following scenarios may result in different interruption waits.

- The end of the interrupt is chained and if another interrupt request is generated when the interrupt returns, the processor skips the out-stack and stack-press process, which reduces the interrupt wait time.
- Delayed arrival, if an interrupt occurs while another lower priority interrupt is being stacked, the higher priority interrupt will be executed first due to the delayed arrival mechanism, which will also result in less waiting time for the higher priority interrupt.

5.6 Interrupt source

Because the NVIC of the Cortex-M0+ processor supports a maximum of one32 external interrupt, and in this system, there are more than one32 external interrupt source, some external interrupts are multiplexed on the same NVIC interrupt input,

NVIC interrupt and NMI (non-maskable interrupt) input	External interrupt source	Active mode	Sleep mode	DeepSleep mode
external pin#0	PORT0 sources and	NVIC interrupt inputs in this system is shown		
Interrupt #1 in the following table.	PORT1	v	v	v
Interrupt #2	PORT2	v	v	v
Interrupt #3 10 Series User's	PORT3	v	v	v
Interrupt #4	Reserved	-	-	-
Interrupt #5	Reserved	-	-	-

Interrupt #12	I2C	v	v	-
Interrupt #13	Reserved	-	-	-
Interrupt #14	TIM0	v	v	-
Interrupt #15	TIM1	v	v	-
Interrupt #16	TIM2	v	v	-
Interruption #17	LPTIM	v	v	v
Interruption #18	TIM4	v	v	-
Interruption #19	TIM5	v	v	-
Interrupt #20	TIM6		v	-
Interrupt #21	PCA	v	v	-
Interrupt #22	WDT	v	v	v
Interruption #23	RTC	v	v	v
Interrupt #24	ADC	v	v	-
Interrupt #25	Reserved	-	-	-
Interruption #26	VC0	v	v	v
Interruption #27	VC1	v	v	v
Interruption #28	LVD	v	v	v
Interruption #29	Reserved	-	-	-
Interruption #30	EFCTRL/RAMCTRL	v	v	-
Interrupt #31	CLK_TRIM	v	v	v

Table 5-2 Correspondence between external interrupts and NVIC interrupt inputs

Caution.

- Since some module interrupts are multiplexed to the same IRQ interrupt source, when the CPU enters this interrupt operation, it must first determine which module generated the interrupt and then perform the corresponding interrupt operation.

5.7 Interrupt Structure Diagram

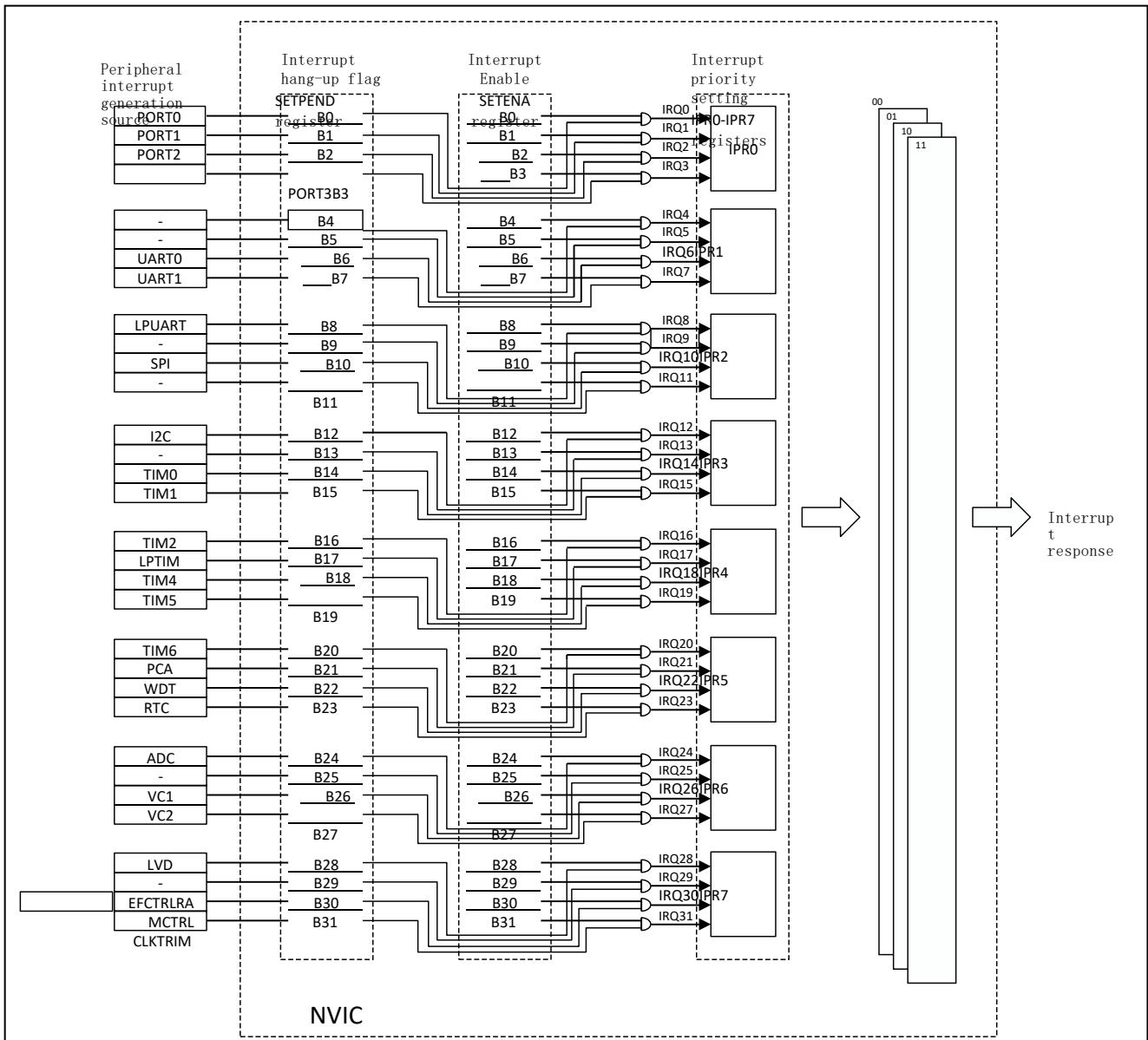


Figure 5-7 Interrupt Structure Diagram

The block diagram of the interrupt structure of this system is shown in Figure 5-7. Several points need to be noted.

- The respective interrupt enable of the peripheral interrupt source is not indicated in the diagram; only the interrupt signal logic block diagram after the peripheral interrupt generation is included here.
- IRQ30 has multiplexed 2peripheral interrupt inputs, and the interrupt flags of each of these 2peripherals must be read separately to determine which peripheral interrupt is being used.

- If a peripheral interrupt source is generated high, the interrupt pending register `SCS_SEPEND` will be set regardless of whether the NVIC interrupt enable register `SCS_SETENA` is set or not, indicating that an interrupt is generated by the corresponding peripheral interrupt source.

- Only when the interrupt enable register **SCS_SETENA** is set, the corresponding interrupt IRQ will be given to the processor to execute the corresponding interrupt program.
- In the interrupt program must clear the peripheral interrupt source high level interrupt signal, interrupt pending register **SCS_SETPEND**
Automatically cleared by hardware.
- The interrupt priority registers **SCS_IPR0**- **SCS_IPR7** set the priority of each 32 interrupt source, 00 the highest priority and 11 the lowest priority. When the priority is the same, the priority is determined by the interrupt number, the smaller the number the higher the priority.

5.8 Register

Base address: 0xE000 E000

Register	Offset Address	Description
SCS_SETENA	0x100	Interrupt request enable register
SCS_CLRENA	0x180	Interrupt request clear enable register
SCS_SETPEND	0x200	Interrupt set pending register
SCS_CLRPEND	0x280	Interrupt to clear pending registers
SCS_IPR0	0x400	Interrupt #0 - Interrupt #3 priority register
SCS_IPR1	0x404	Interrupt #4 - Interrupt #7 priority register
SCS_IPR2	0x408	Interrupt #8 - Interrupt #11 priority register
SCS_IPR3	0x40C	Interrupt #12 - Interrupt #15 priority register
SCS_IPR4	0x410	Interrupt #16 - Interrupt #19 priority register
SCS_IPR5	0x414	Interrupt #20 - Interrupt #23 Priority Register
SCS_IPR6	0x418	Interrupt #24 - Interrupt #27 Priority Register
SCS_IPR7	0x41C	Interrupt #28 - Interrupt #31 Priority Register
SCS_PRIMASK	-	Interrupt Mask Special Register

5.8.1 Interrupt enable setting register (SCS_SETENA)

Offset address: 0x100

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETENA[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETENA[15:0]															
RW															

position	Marker	Function Description
n		

31:0	SETENA [31:0]	Set enable interrupt #0 to interrupt #31; write 1"" is set, write0 "" is invalid [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31
------	------------------	---

5.8.2 Interrupt enable clear register (SCS_CLRENA)

Offset address: 0x180

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA															
RW															

position	Marker	Desc ription
31:0	CLRENA	Clear enable interrupt #0 to interrupt #31; write 1"" to clear, write0 "" to void

5.8.3 Interrupt pending status setting register (SCS_SETPEND)

Offset address: 0x200

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SETPEND[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SETPEND[15:0]															
RW															

position	Marker	Function Description
31:0	SETPEND	Set the pending status of interrupt #0 to interrupt #31; write "1" to set, write "0" to invalidate [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

5.8.4 Interrupt pending status clear register (**SCS_CLRPEND**)

Offset address: 0x280

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRPEND[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRPEND[15:0]															
RW															

position	Marker	Description
31:0	CLRPEND	Clear the pending status of interrupt #0 to interrupt #31; write "1" to clear, write "0" to invalidate [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

5.8.106 Interrupt priority register (SCS_IPR0)

Offset address: 0x400

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR0[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR0[15:0]															
RW															

position	Marker	Function Description
31:0	IPR0[31:0]	<p>The priority of interrupt #0 to interrupt #3.</p> <p>[31:30]: priority of interrupt #3 [23:22]: priority of interrupt #2</p> <p>[15:14]: priority of interrupt #1 [7:6]: priority of interrupt #0</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.107 Interrupt Priority Register (SCS_IPR1)

Offset address: 0x404

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR1 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR1[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR1[31:0]	<p>The priority of interrupt #4 to interrupt #7.</p> <p>[31:30]: priority of interrupt #7 [23:22]: priority of interrupt #6</p> <p>[15:14]: priority of interrupt #5 [7:6]: priority of interrupt #4</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.108 Interrupt Priority Register (SCS_IPR)108)

Offset address: 0x408

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR2 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR2[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR2[31:0]	<p>The priority of interrupt #8 to interrupt #11.</p> <p>[31:30]: priority of interrupt #11 [23:22]:</p> <p>priority of interrupt #10</p> <p>[15:14]: priority of interrupt #9 [7:6]:</p> <p>priority of interrupt #8</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.109 Interrupt Priority Register (SCS_IPR)109)

Offset address: 0x40C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR3 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR3 [31:0]	<p>The priority of interrupt #12 to interrupt #15.</p> <p>[31:30]: priority of interrupt #15 [23:22]: priority of interrupt #14</p> <p>[15:14]: priority of interrupt #13 [7:6]: priority of interrupt #12</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.110 Interrupt Priority Register (SCS_IPR)110)

Offset address: 0x410

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR4 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR4[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR4[31:0]	<p>The priority of interrupt #16 to interrupt #19.</p> <p>[31:30]: priority of interrupt #19 [23:22]: priority of interrupt #18</p> <p>[15:14]: priority of interrupt #17 [7:6]: priority of interrupt #16</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.111 Interrupt Priority Register (SCS_IPR)111)

Offset address: 0x414

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR5 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR5[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR5 [31:0]	<p>The priority of interrupt #20 to interrupt #23.</p> <p>[31:30]: priority of interrupt #23 [23:22]: priority of interrupt #22</p> <p>[15:14]: priority of interrupt #21 [7:6]: priority of interrupt #20</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.112 Interrupt Priority Register (SCS_IPR)112)

Offset address: 0x418

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR6 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR6[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR6 [31:0]	<p>The priority of interrupt #24 to interrupt #27.</p> <p>[31:30]: priority of interrupt #27 [23:22]: priority of interrupt #26</p> <p>[15:14]: priority of interrupt #25 [7:6]: priority of interrupt #24</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.113 Interrupt Priority Register (SCS_IPR)113)

Offset address: 0x41C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7 [31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]]															
RW															

position	Marker	Function Description
31:0	IPR7 [31:0]	<p>The priority of interrupt #28 to interrupt #31.</p> <p>[31:30]: priority of interrupt #31 [23:22]: priority of interrupt #30</p> <p>[15:14]: priority of interrupt #29 [7:6]: priority of interrupt #28</p> <p>Among them, 00 has the highest priority and 11 has the lowest priority</p>

5.8.13 Interrupt Mask Special Register (SCS_PRIMASK)

Offset address: ---

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
R															

SCS_PRIMASK															
Reset value: 0x0000_0000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
RO															
BIT	Symbols	Description													
31:1	Reserved														
0	PRIMASK	When set, all interrupts except NMI and hardware error exceptions will be masked out when cleared, all exceptions and interrupts will not be masked out This special register needs to be accessed through the MSR and MRS special register operation instructions, and can also be accessed with the change processor state instruction CPS. When dealing with time-sensitive applications, it is necessary to operate the PRIMASK register.													

5.9 Basic Software

5.9.1 Operation

External interrupt enable

Each peripheral module has its own internal interrupt enable register, and when an interrupt operation is required, the peripheral's own interrupt enable must first be turned on. The operation of this enable bit is not discussed in this section, please refer to the respective section description of the peripheral module.

5.9.2 NVIC interrupt enable and clear enable

The Cortex-M0+ processor supports a maximum of one32 interrupt source, each of which has an interrupt enable bit and a clear enable bit. If you want to enable an interrupt, set the `SCS_SETENA` register to the corresponding location1. If you want to clear an interrupt, the corresponding location1 of the `SCS_CLRENA` register is used.

Note that the interrupt enable mentioned here is only for the processor NVIC. The interrupt generation for each peripheral is determined by the peripheral's interrupt control register and is not related to `SCS_SETENA` and `SCS_CLRENA`.

5.9.3 NVIC Interrupt Pause and Clear Pause

If an interrupt occurs but cannot be processed immediately, the interrupt request will be hung. The pending status is stored in a register and will remain legal if the processor's current priority has not been reduced to handle the pending request and the pending status is not cleared manually.

When the processor starts to enter interrupt processing, the hardware automatically causes the pending state to clear.

The interrupt pending status can be accessed or modified by manipulating the interrupt set pending `SCS_SETPEND` and interrupt clear pending `SCS_CLRPEND` registers. The interrupt pending status register allows the

5.9 Basic Software Operation

5.9.4 NVIC Interrupt Priority

Setting the SCS_IPR0- SCS_IPR7 registers determines the priority of SCS IRQ0- SCS IRQ32. Programming of the interrupt priority registers should be done before interrupt enable, which is usually done at the beginning of the program. Changing the interrupt priority after interrupt enable should be avoided; the result is unpredictable and is not supported by the Cortex-M0+ processor.

5.9.5 NVIC Interrupt Mask

Some time-sensitive applications that need to disable all interrupts for a short period of time can be achieved using the interrupt mask register SCS_PRIMASK. The special register SCS_PRIMASK is only bit 1valid and defaults to 0 after reset. When 0 this register is set to , all interrupts and exceptions are allowed; when 1 it is set to , only NMI (not supported by this system) and hardware error exceptions are enabled. In fact, when SCS_PRIMASK is set to 1 , the current priority of the processor is reduced to 0(the highest priority that can be summed)

The SCS_PRIMASK registers can be programmed in a variety of ways. Using assembly language, the MSR instruction can be used to set and clear the SCS_PRIMASK registers. If using the C language and the CMSIS device driver library, the user can use the following functions to set and clear PRIMASK.

```
void enable_irq(  
    void)  
;/ / c l e a r  
  
PRIMASK void  
    disable_irq(void);  
/ / s e t  PRIMASK
```

6 Port Controller (GPIO)

6.1 Port Controller Introduction

This product has one 16 digital universal input/output port P01-P03, P14-P15, P23-P27, P31-P36 and one 1digital universal input port P00. The input and output signals of analog signals ADC/VC/LVD, input and output signals of various function modules (such as SPI, UART, I2C, Timer, etc.) and input and output signals of test and debug functions can be multiplexed with the digital ports. The analog signals of ADC/VC/LVD, the input and output signals of function modules (such as SPI, UART, I2C, Timer, etc.) and the input and output signals of test and debug functions can be multiplexed with the digital ports.

Each port can be configured as internal pull up/pull down input, high resistance input (floating input) push-pull output (CMOS output), open drain output, and two-position drive capability output. In order to prevent the chip from being abnormally reset, the external device generates abnormal action, the chip reset port is configured as high resistance input. However, to avoid leakage due to high resistance input, the user should configure the port accordingly (configured as internal pull-up/pull-down input or output)~~as~~ started.

When the digital port is configured as an analog port, the digital function is isolated and the numbers "1" and "0" cannot be output, and the CPU reads the port as "0".

All ports can provide external interrupts, and each interrupt can be configured as4 high trigger, low trigger, rising edge trigger, or falling edge trigger, and the corresponding interrupt trigger port can be found by querying the interrupt flag bit of Px_STAT[n]. In addition, the interrupt of each port can wake up the chip from sleep mode/deep sleep mode to working mode.

6.2 Port Controller Key Features

The port controller supports the following features.

- Port Multiplexing Function
 - Analog function pin multiplexing
 - Debugging pin multiplexing
 - Digital Universal Pin Multiplexing
 - Multiplexing of digital function pins
- Configuration Features
 - Support pull-up/down
 - Low drive/high drive
 - Push-pull output
 - Open Drain Output
- External interrupt source
 - High level/low level
 - Rising/falling edge
- Support interrupt in work mode/sleep mode/deep sleep mode

6.3 Port Controller Function Description

6.3.1 Port Configuration Features

Each port can be configured as an analog or digital port through configuration registers (PxADS) depending on system requirements. When configured as a digital port, the corresponding registers can also be configured to implement the following features.

1. Internal pull-up (PxPU)/pull-down (PxPD)

The pull-up register (PxPU) and pull-down register (PxPD) correspond to the port pull-up enable and port pull-down enable respectively, when the corresponding bit is '1', set the corresponding bit pin pull-up/pull-down enable, when '0', disable the corresponding bit pin pull-up/pull-down .

2. Two-speed drive output (PxDR)

The drive capability can be changed via the PxDR register, with low drive capability when PxDR is '1' and high drive capability when PxDR is '0'.

3. Open-drain output (PxOD)

The pin output status is set via the PxOD register. When PxOD is '1', the port open-drain output is enabled, and when it is '0', the port open-drain output is disabled. If the open-drain pin is not connected to an external pull-up resistor, it can only output low level, and if it needs to have the function of outputting high level at the same time, pull-up resistor is required.

4. Direction selection (PxDIR)

Used to set the direction of the port pins. port is output when PxDIR is '0', port is input when PxDIR is '1'.

5. Output high and low level selection (PxOUT), accessible via the AHB bus

When the port pin is configured as an output, the port pin output is high if PxOUT is '1', or high with an external pull-up resistor if configured as

an open-drain output. If PxOUT is 0'', the output will be low.

6. Input level status (PxIN) is accessible via the AHB bus

The pin level after synchronization can be obtained by reading the PxIN register, which is high when PxIN is 1'' and low when PxIN is 0''

Note: The above features are not valid when configured as an analog port.

The relationship between the state of the port and the register configuration is shown in the following table.

IO Status	IO Direction	PxADS	PxDI	PxOUT	PxIN	PxPU	PxPD	PxOD	PxDR	Px_SEL
Simulation	Input/output	1	W	W	0	W	W	W	W	W
Float	Input	0	1	W	X	0	0	W	W	0
Pull down	Input	0	1	W	0	0	1	W	W	0
Pull up	Input	0	1	W	1	1	0	W	W	0
Pull up	Input	0	1	W	1	1	1	W	W	0
1	Input	0	1	W	1	W	W	W	W	0
0	Input	0	1	W	0	W	W	W	W	0
1	Output	0	0	1	1	W	W	0	W	0
0	Output	0	0	0	0	W	W	0	W	0
1	Output	0	0	W	1	W	W	0	W	0
0	Output	0	0	W	0	W	W	0	W	0
(SET)1/(CLR)0	Output	0	0	W	(SET)1/(CLR)0	W	W	0	W	0
0	Output	0	0	0	0	W	W	1	W	0
Z	Output	0	0	1	X	0	0	1	W	0
0	Output	0	0	1	0	0	1	1	W	0
1	Output	0	0	1	1	1	0	1	W	0
1	Output	0	0	1	1	1	1	1	W	0

Note: 0- Logic low1 -Logic highW- Whatever 0 or 1X- unknown stateZ- high impedance

Table 6-1 Port Status Truth Table

The port circuit structure is shown in the following figure.

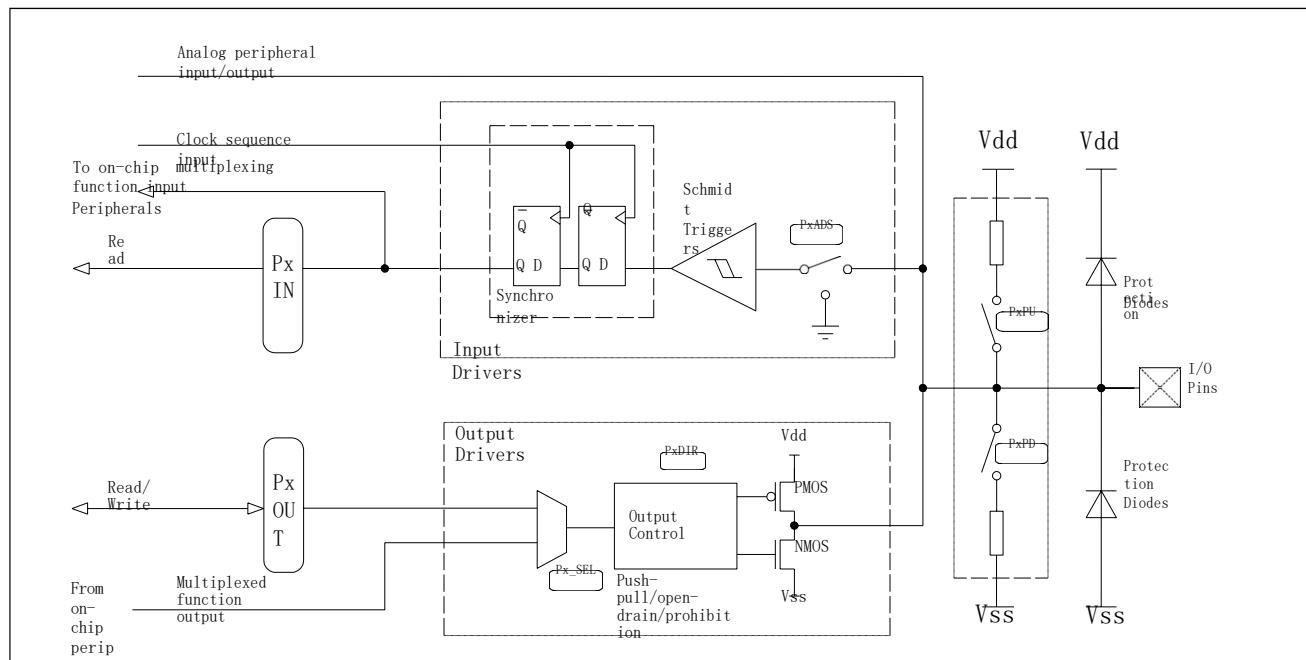


Figure 6-1 Port Circuit Schematic

6.3.2 Port writing

The Port Input Value/Output Value registers (PxIN/PxOUT) support AHB bus read/write. For the AHB bus, the system clock (HCLK) does not process the same period as the other buses, and the IO flips once every two HCLK cycles.

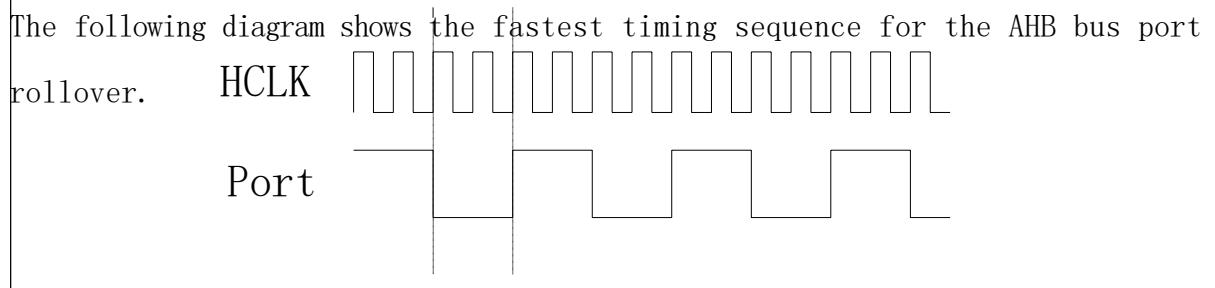


Figure 6-2 Variation of AHBbus port with system clock

6.3.3 Port reading

Each port can obtain the port pin level by reading the PxIN register, the bits of which form a synchronizer with the latch in front of it, thus avoiding signal instability caused by pin level changes in a short period of time when the system clock state changes, but also introducing delays.

The synchronization diagram for reading the port pin data is as follows.

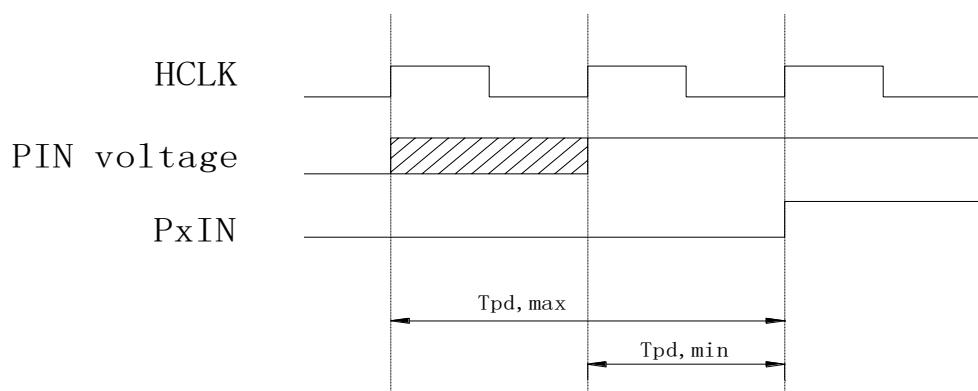


Figure 6-3 Read Port Pin Data Synchronization Diagram

During the clock period after the rising edge of the system clock, the pin level signal is latched into the internal register, as shown in the shaded area, and the stable pin level signal can be read after the next rising edge of the system clock. After that, the data is latched into the PxIN register on the rising edge of the system clock again. The signal loading delay Tpd is 1-2 system clocks.

Caution.

- Handling of unconnected pins.

If there are unconnected pins during use, it is recommended to give the pins a defined level in order to avoid dangling current consumption due to the lack of defined level of the pins during other digital input enable modes.

6.3.4 Port Multiplexing Function

Port multiplexing is one of the main functions of the port controller. By configuring the registers, the port can be flexibly configured as an analog port/test debug port/digital general purpose port/digital functional port. PxADS register is used for digital port/analog port switching, when PxADS is 1'', the port is configured as analog port, then the digital function is isolated, and the numbers "1" and "0" cannot be output. "When PxADS is ''', the port is configured as an analog port. When PxADS is 0'', the port is configured as a digital port, at this time, digital general

^{Simu} ^{Port/digital} ^{on}	Digital function port switching is realized by configuring Px_sel register.							
and each PxADS=1 system.	port can be independently configured as the function port required by the Please refer to the relevant section for the configuration P34 PCA_CH0 LPUART_TXD TIM5_CHA TIM0_EXT TIM4_CHA RTC_1Hz TIM1_TOG							
AIN4 VCIN4	P34	Px_sel=0 PCA_CH0	Px_sel=1 LPUART_TXD	Px_sel=2 TIM5_CHA	Px_sel=3 TIM0_EXT	Px_sel=4 TIM4_CHA	Px_sel=5 RTC_1Hz	Px_sel=7 TIM1_TOG
AIN5 information of the test debug port.	P36	UART1_RXD	TIM6_CHA	UART0_RXD	TIM0_GATE	TIM4_CHB	SPI_MISO	I2C_SDA
AIN6 ADC_VREF	VCIN6	P36	UART1_RXD	TIM6_CHA	UART0_RXD	PCA_CH4	TIM5_CHA	SPI_MOSI
AIN7 XTHI	VCIN7	P01	UART0_RXD	I2C_SDA	UART1_RXD	TIM0_TOG	TIM5_CHB	SPI_SCK
AIN8 XTHO		P02	UART0_TXD	I2C_SCL	UART1_RXD	TIM0_TOGN	TIM6_CHA	SPI_CS
LVDIN1		P03	PCA_CH3	SPI_CS	TIM6_CHB	LPTIM_EXT	RTC_1Hz	PCA_ECI
XTLO		P15	I2C_SDA	TIM2_TOG	UART1_RXD	LPTIM_GATE	SPI_SCK	UART0_RXD
XTLI		P14	I2C_SCL	TIM2_TOGN	PCA_ECI	ADC_RDY	SPI_CS	UART0_RXD
LVDIN2	VCIN0	P23	TIM6_CHA	TIM4_CHB	TIM4_CHA	PCA_CH0	SPI_MISO	UART1_RXD
AIN0		P24	TIM4_CHB	TIM5_CHB	HCLK_OUT	PCA_CH1	SPI_MOSI	UART1_RXD
LVDIN3	VCIN1	P25	SPI_SCK	PCA_CH0	TIM5_CHA	LVD_OUT	LPUART_RXD	I2C_SDA
AIN1		P26	SPI_MOSI	TIM4_CHA	TIM5_CHB	PCA_CH2	LPUART_RXD	I2C_SCL
		P27/SWDIO	SPI_MISO	TIM5_CHA	TIM6_CHA	PCA_CH3	UART0_RXD	RCH_OUT
		P31/SWCLK	LPTIM_TOG	PCA_ECI	PCLK_OUT	VC0_OUT	UART0_RXD	RCL_OUT
AIN2	VCIN2	P32	LPTIM_TOGN	PCA_CH2	TIM6_CHB	VC1_OUT	UART1_RXD	PCA_CH4
AIN3	VCIN3	P33	LPUART_RXD	PCA_CH1	TIM5_CHB	PCA_ECI	UART1_RXD	XTL_OUT
		P00 Reset						

Table 6-2 Port Multiplexing Table

6.3.5 Port interrupt function

Each digital general-purpose port can generate interrupts from external sources, which can be4 high level/low level/rising edge/falling edge signals, and the corresponding interrupt enable registers are high level interrupt enable register/low level interrupt enable register/rising edge interrupt enable register/falling edge interrupt enable register, respectively.

When an interrupt is triggered, you can determine which port triggered the interrupt by querying the interrupt status register, and clear the corresponding interrupt status flag bit by clearing the interrupt clear register.

6.4 Port Configuration Operations

6.4.1 Port multiplexing operation flow

Port multiplexing is configured as an analog port

Step1: Set register Px_ADS to 1

Port multiplexing configured as a digital universal port

a) Set register Px_ADS to 0

b) Set register Px_sel to 0

c) Set register PxDIR to 1: The direction of the port is input and the CPU can read the status of the port PxIN

d) Set register PxDIR to 0: Port direction is output

e) Set register PxOUT to 1: Port output high

f) Set register PxOUT to 0: Port output

low Port multiplexing configured as

digital function port

a) Set register Px_ADS to 0

b) Set register Px_sel to 1~7 (according to system requirements, refer to the port multiplexing table)

c) Set register PxDIR (according to system requirements)

Port multiplexing is

configured as a debug test

port refer to the Test

Debug related section.

Port multiplexing configuration for IR output signals

Port P23 can be configured as an IR output signal with a frequency of 38K.

a) Set register P23_ADS to 0

b) Set register P23_sel to 7

c) Set register P2DIR[3] to 0: Port direction is output

d) Set bit 14 of register GPIO_CTRL1 to select the output polarity of IR signal

e) Set register P2OUT[3] Output of gated IR signal

6.4.2 Port Interrupt Operation Flow

High level interrupt

- a) Set register Px_ADS to 0
- b) Set register Px_sel to 0
- c) Set register PxDIR to 1
- d) Set register PxHIE to 1
- e) Read interrupt status register after interrupt trigger Px_STAT
- f) Set register Px_ICLR to 0 Clear interrupt status register Px_STAT

Low level interrupt

- a) Set register Px_ADS to 0
- b) Set register Px_sel to 0
- c) Set register PxDIR to 1
- d) Set register PxLIE to 1
- e) Read interrupt status register after interrupt trigger Px_STAT
- f) Set register Px_ICLR to 0 Clear interrupt status register Px_STAT

Rising edge interruption

- a) Set register Px_ADS to 0
- b) Set register Px_sel to 0
- c) Set register PxDIR to 1
- d) Set register PxRIE to 1
- e) Read interrupt status register after interrupt trigger Px_STAT
- f) Set register Px_ICLR to 0 Clear interrupt status register Px_STAT

Interruptions along the descent

- a) Set register Px_ADS to 0
- b) Set register Px_sel to 0
- c) Set register PxDIR to 1
- d) Set register PxFIE to 1
- e) Read interrupt status register after interrupt trigger Px_STAT

f) Set register Px_ICLR to 0 Clear interrupt status register Px_STAT

6.4.3 Port configuration operation procedure

Pull-up enable

a) Set register PxPU to 1

Pull Down Enable

a) Set register PxPU to 0

b) Set register PxPD to 1

High drive capability

a) Set register PxDR to 0

Open Drain Output

a) Set register PxOD to 1

6.5 Port Controller Register Description

Register list

base address: 0x40020C00

Offset	Register Name	Access	Register Description
0x04	P01_SEL	RW	Port P01 Function Configuration Register
0x08	P02_SEL	RW	Port P02 Function Configuration Register
0x0c	P03_SEL	RW	Port P03 Function Configuration Register
0x50	P14_SEL	RW	Port P14 Function Configuration Register
0x54	P15_SEL	RW	Port P15 Function Configuration Register
0x8c	P23_SEL	RW	Port P23 Function Configuration Register
0x90	P24_SEL	RW	Port P24 Function Configuration Register
0x94	P25_SEL	RW	Port P25 Function Configuration Register
0x98	P26_SEL	RW	Port P26 Function Configuration Register
0x9c	P27_SEL	RW	Port P27 Function Configuration Register
0xc4	P31_SEL	RW	Port P31 Function Configuration Register
0xc8	P32_SEL	RW	Port P32 Function Configuration Register
0xcc	P33_SEL	RW	Port P33 Function Configuration Register
0xd0	P34_SEL	RW	Port P34 Function Configuration Register
0xd4	P35_SEL	RW	Port P35 Function Configuration Register
0xd8	P36_SEL	RW	Port P36 Function Configuration Register
0x100	P0DIR	RW	Port P0 Input/Output Configuration Register
0x104	P0IN	RO	Port P0 Input Value Register
0x108	P0OUT	RW	Port P0 Output Value Configuration Register
0x10c	P0ADS	RW	Port P0 Digital-to-analog configuration register
0x11c	P0DR	RW	Port P0 Drive Capability Configuration Register
0x120	P0PU	RW	Port P0 Pull-up Enable Configuration Register

0x124	P0PD	RW	Port P0 pull-down enable configuration register
0x12c	P0OD	RW	Port P0 Open Drain Output Configuration Register

0x130	P0HIE	RW	Port P0 High Interrupt Enable Configuration Register
0x134	P0LIE	RW	Port P0 Low Interrupt Enable Configuration Register
0x138	P0RIE	RW	Port P0 Rising Edge Interrupt Enable Configuration Register
0x13c	P0FIE	RW	Port P0 Falling Edge Interrupt Enable Configuration Register
0x200	P0_STAT	RO	Port P0 Interrupt Status Register
0x210	P0_ICLR	RW	Port P0 Interrupt Clear Register
0x140	P1DIR	RW	Port P1 Input/Output Configuration Register
0x144	P1IN	RO	Port P1 Input Value Register
0x148	P1OUT	RW	Port P1 Output Value Configuration Register
0x14c	P1ADS	RW	Port P1 Digital to analog configuration register
0x15c	P1DR	RW	Port P1 Drive Capability Configuration Register
0x160	P1PU	RW	Port P1 pull-up enable configuration register
0x164	P1PD	RW	Port P1 pull-down enable configuration register
0x16c	P1OD	RW	Port P1 Open Drain Output Configuration Register
0x170	P1HIE	RW	Port P1 High Interrupt Enable Configuration Register
0x174	P1LIE	RW	Port P1 Low Interrupt Enable Configuration Register
0x178	P1RIE	RW	Port P1 Rising Edge Interrupt Enable Configuration Register
0x17c	P1FIE	RW	Port P1 Falling Edge Interrupt Enable Configuration Register
0x240	P1_STAT	RO	Port P1 Interrupt Status Register
0x250	P1_ICLR	RW	Port P1 Interrupt Clear Register
0x180	P2DIR	RW	Port P2 Input/Output Configuration Register
0x184	P2IN	RO	Port P2 Input Value Register

0x188	P2OUT	RW	Port P2 Output Value Configuration Register
0x18c	P2ADS	RW	Port P2 Digital-to-Analog Configuration Register
0x19c	P2DR	RW	Port P2 Drive Capability Configuration Register
0x1a0	P2PU	RW	Port P2 pull-up enable configuration register
0x1a4	P2PD	RW	Port P2 pull-down enable configuration register

0x1ac	P2OD	RW	Port P2 Open Drain Output Configuration Register
0x1b0	P2HIE	RW	Port P2 High Interrupt Enable Configuration Register
0x1b4	P2LIE	RW	Port P2 Low Interrupt Enable Configuration Register
0x1b8	P2RIE	RW	Port P2 Rising Edge Interrupt Enable Configuration Register
0x1bc	P2FIE	RW	Port P2 Falling Edge Interrupt Enable Configuration Register
0x280	P2_STAT	RO	Port P2 Interrupt Status Register
0x290	P2_ICLR	RW	Port P2 Interrupt Clear Register
0x1c0	P3DIR	RW	Port P3 Input/Output Configuration Register
0x1c4	P3IN	RO	Port P3 Input Value Register
0x1c8	P3OUT	RW	Port P3 Output Value Configuration Register
0x1cc	P3ADS	RW	Port P3 Digital-to-Analog Configuration Register
0x1dc	P3DR	RW	Port P3 Drive Capability Configuration Register
0x1e0	P3PU	RW	Port P3 pull-up enable configuration register
0x1e4	P3PD	RW	Port P3 pull-down enable configuration register
0x1ec	P3OD	RW	Port P3 Open Drain Output Configuration Register
0x1f0	P3HIE	RW	Port P3 High Interrupt Enable Configuration Register
0x1f4	P3LIE	RW	Port P3 Low Interrupt Enable Configuration Register
0x1f8	P3RIE	RW	Port P3 Rising Edge Interrupt Enable Configuration Register
0x1fc	P3FIE	RW	Port P3 Falling Edge Interrupt Enable Configuration Register
0x2c0	P3_STAT	RO	Port P3 Interrupt Status Register
0x2d0	P3_ICLR	RW	Port P3 Interrupt Clear Register
0x304	GPIO_CTRL1	RW	Port Auxiliary Function Configuration

			Register 1	
0x308	GPIO_CTRL2	RW	Port Auxiliary Function Configuration Register 2	
0x30c	GPIO_CTRL3	RW	Port Auxiliary Function Configuration Register 3	
0x310	GPIO_CTRL4	RW	Port Auxiliary Function Configuration Register 4	

6.5.1 Port P0

6.5.1.1 Port P01 Function Configuration

Register (**P01_SEL**) Offset

Address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													P01_sel		
													RW		

position	Marker	Function Description
31:3	Reserved	
2:0	P01_sel	Port P01 Function Selection. 000: GPIO P01 001: UART0_RXUART0module RXD signal 010: I2C_SDAI2Cmodule data signal 011: UART1_RXUART1module TXD signal 100: TIM0_TOGTimer0flip signal 110: TIM5_CHBAdvancedTimer module channel B1 signal 111: SPI_SCKSPImodule clock signal 110: TIM2_EXTTimer2module external clock input signal

6.5.1.138 Port P02 Function

Configuration Register (P02_SEL)

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P02_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P02_sel	Port P02 Function Selection. 000: GPIO P02 001: UART0_TXDUART0module TXD signal 010: I2C_SCLI2Cmodule clock signal 011: UART1_RXDUART1module RXD signal 100: TIM0_TOGNTimer0: Inverse signal101 of module flip-flop signal: TIM6_CHAAdvancedTimer module channel A2 signal110: SPI_CSSPI module host mode chip select signal 111: TIM2_GATETimer2module gating signal

6.5.1.139 Port P03 Function

Configuration Register (P03_SEL)

Offset address: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P03_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P03_sel	Port P03 Function Selection. 000: GPIO P03 001: PCA_CH3PCA module channel 3 capture/compare signal 010: SPI_CSSPI host mode chip select signal 011: TIM6_CHB Advanced Timer module channel B2 signal100: LPTIM_EXTTimer3 module external clock input signal 101: RTC_1HzRTC module 1Hz output signal 110: PCA_ECIPCA module external clock input signal 111: VC0_OUTVC0 module output

6.5.1.4 Port P0 Input/Output Configuration Register (P0DIR)

Offset address:

0x100

Reset value: 0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P0D IR3	P0D IR2	P0D IR1	Res
												RW	RW	RW	

position	Marker	Function Description
31:4	Reserved	
3	P0DIR3	Port P03 Input/Output Configuration Register 1: configured as input 0: configured as output
2	P0DIR2	Port P02 Input/Output Configuration Register 1: configured as input 0: configured as output
1	P0DIR1	Port P01 Input/Output Configuration Register 1: configured as input 0: configured as output
0	Reserved	

6.5.1.5 Port P0 Input Value

Register (P0IN)

Offset address:

0x104

31	Reset value: 30 NA	29 28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0	
Reserved												POI N3	POI N2	POI N1	POI N0
												RO	RO	RO	RO

position	Marker	Function Description
31:4	Reserved	
3	P0IN3	Port P03 Input Value Register 1: Input is high 0: Input is low
2	P0IN2	Port P02 Input Value Register 1: Input is high 0: Input is low
1	P0IN1	Port P01 Input Value Register 1: Input is high 0: Input is low
0	P0IN0	Port P00 Input Value Register 1: Input is high 0: Input is low

6.5.1.6 Port P0 Output Value Configuration Register (P0OUT)

Offset address:

0x108

31	Reset value: NA	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P0O UT3	P0O UT2	P0O UT1	Res
												RW	RW	RW	

position	Marker	Function Description
31:4	Reserved	
3	P0OUT3	Port P03 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
2	P0OUT2	Port P02 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
1	P0OUT1	Port P01 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
0	Reserved	

6.5.1.7 Port P0 Digital-to-Analog Configuration Register (P0ADS)

Offset address:

0x10C

31	Reset value: 0x00000000	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P0A DS3	P0A DS2	P0A DS1	Res
												RW	RW	RW	

position	Marker	Function Description
31:4	Reserved	
3	P0ADS3	Port P03 Digital-to-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
2	P0ADS2	Port P02 Digital-to-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
1	P0ADS1	Port P01 Digital-to-analog configuration register 1: Configured as an analog port 0: Configured as a digital port
0	Reserved	

6.5.1.8 Port P0 Drive Capability Configuration Register (P0DR)

Offset address:

0x11C

31	Reset value: 0x00000000	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P0D R3	P0D R2	P0D R1	Res RW		

position	Marker	Function Description
31:4	Reserved	
3	P0DR3	Port P03 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
2	P0DR2	Port P02 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
1	P0DR1	Port P01 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
0	Reserved	

6.5.1.9 Port P0 pull-up enable configuration register (P0PU)

Offset address:

0x120

31	Reset value: 0x00000000	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P0P U3	P0P U2	P0P U1	Res
												RW	RW	RW	

position	Marker	Function Description
31:4	Reserved	
3	P0PU3	Port P03 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
2	P0PU2	Port P02 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
1	P0PU1	Port P01 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
0	Reserved	

6.5.1.10 Port P0 pull-down enable configuration register (P0PD)

Offset address:

0x124

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P0P D3	P0P D2	P0P D1	Res		
										RW	RW	RW			

position	Marker	Function Description
31:4	Reserved	
3	P0PD3	Port P03 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
2	P0PD2	Port P02 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
1	P0PD1	Port P01 Pull-down Enable Configuration Register 1: Enabled 0: Prohibition
0	Reserved	

6.5.1.11 Port P0 Open Drain Output Configuration Register (P0OD)

Offset address:

0x12C

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P0O	P0O	P0O	Res		
										D3	D2	D1			
										RW	RW	RW			

position	Marker	Function Description
31:4	Reserved	
3	P0OD3	Port P03 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
2	P0OD2	Port P02 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
1	P0OD1	Port P01 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
0	Reserved	

6.5.1.12 Port P0 High Interrupt Enable Configuration Register (P0HIE)

Offset address:

0x130

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P0H	P0H	P0H	P0H		
										IE3	IE2	IE1	IE0		
										RW	RW	RW	RW		

position	Marker	Function Description
31:4	Reserved	
3	P0HIE3	Port P03 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2	P0HIE2	Port P02 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
1	P0HIE1	Port P01 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
0	P0HIE0	Port P00 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition

6.5.1.13 Port P0 Low Level Interrupt Enable Configuration Register (P0LIE)

Offset address:

0x134

31	Reset value: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 0x0000 0006	27 26 25 24 23 22 21 20 19 18 17 16	Reserved													
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Reserved												POL IE3 RW	POL IE2 RW	POL IE1 RW	POL IE0 RW

position	Marker	Function Description
31:4	Reserved	
3	POLIE3	Port P03 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
2	POLIE2	Port P02 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
1	POLIE1	Port P01 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
0	POLIE0	Port P00 Low Interrupt Enable Configuration Register 1: Enabled 0: Prohibition

6.5.1.14 Port P0 Rising Edge Interrupt Enable Configuration Register (P0RIE)

Offset address:

0x138

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P0R	P0R	P0R	P0R		
										IE3	IE2	IE1	IE0		
										RW	RW	RW	RW		

position	Marker	Function Description
31:4	Reserved	
3	P0RIE3	Port P03 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2	P0RIE2	Port P02 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
1	P0RIE1	Port P01 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
0	P0RIE0	Port P00 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition

6.5.1.15 Port P0 Falling Edge Interrupt Enable Configuration Register (P0FIE)

Offset address:

0x13C

31	Reset value: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 0x0000 0006	27 26 25 24 23 22 21 20 19 18 17 16	Reserved													
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Reserved												P0F IE3	P0F IE2	P0F IE1	P0F IE0
													RW	RW	RW	RW

position	Marker	Function Description
31:4	Reserved	
3	P0FIE3	Port P03 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2	P0FIE2	Port P02 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
1	P0FIE1	Port P01 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
0	P0FIE0	Port P00 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition

6.5.1.16 Port P0 Interrupt Status

Register (P0_STAT)

Offset address:

0x200

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
Reserved										POS TA3	POS TA2	POS TA1	POS TA0	
										RO	RO	RO	RO	

position	Marker	Function Description
31:4	Reserved	
3	P0STA3	Port P03 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
2	P0STA2	Port P02 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
1	P0STA1	Port P01 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
0	P0STA0	Port P00 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger

6.5.1.17 Port P0 Interrupt Clear

Register (**P0_ICLR**)

Offset address:

0x210

31	Reset value: 30 0xfffffff 29 28	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0	
												P0C	P0C	P0C	P0C
												LR3	LR2	LR1	LR0
												RW	RW	RW	RW

position	Marker	Function Description
31:4	Reserved	
3	P0CLR3	Port P03 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
2	P0CLR2	Port P02 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
1	P0CLR1	Port P01 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
0	P0CLR0	Port P00 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit

6.5.2 Port P1

6.5.2.1 Port P14 Function Configuration

Register (**P14_SEL**) Offset

Address: 0x50

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P14_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P14_sel	Port P14 Function Selection. 000: GPIO P14 001: I2C_SCL/I2C module clock signal 010: Reverse signal of the TIM2_TOGNTimer2 module flip signal 011: PCA_ECI/PCA module external clock input signal 100: ADC_RDY/ADC module RDY signal 101: SPI_CSSPI module host mode chip select signal 110: UART0_TXD/UART0 module TXD signal 111: NC

6.5.2.2 Port P15 Function Configuration Register (P15_SEL)

Offset address: 0x54

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P15_sel			
												RW			

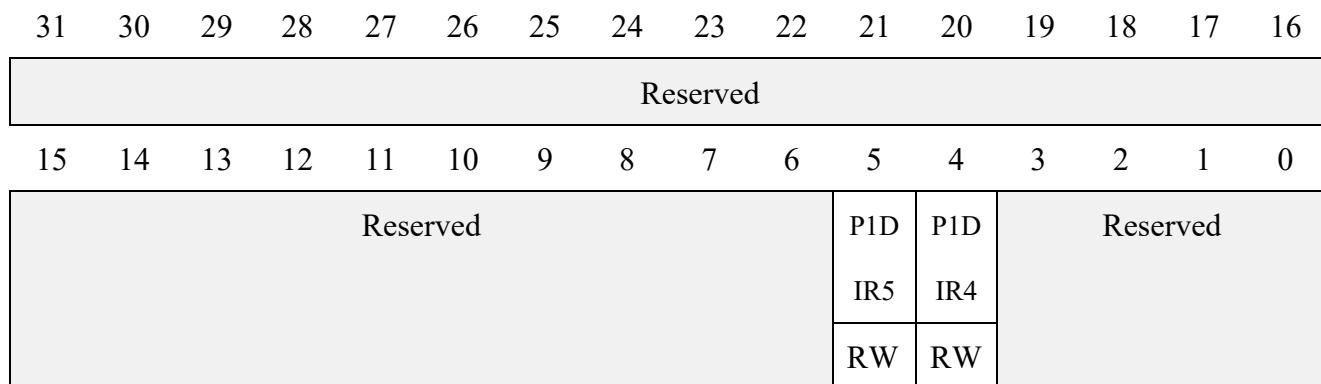
position	Marker	Function Description
31:3	Reserved	
2:0	P15_sel	Port P15 Function Selection. 000: GPIO P15 001: I2C_SDAI2Cmodule data signal 010 TIM2_TOGTimer2flip-flop signal011: TIM4_CHAdvancedTimer module channel B0 signal100: LPTIM_GATETimer3module gating signal 101: SPI_SCKSPImodule clock signal 110: UART0_RXDUART0module RXD signal 111: LVD_OUTLVDmodule output signal

6.5.2.3 Port P1 Input/Output Configuration Register (P1DIR)

Offset address:

0x140 Reset

value: 0xffff ffff



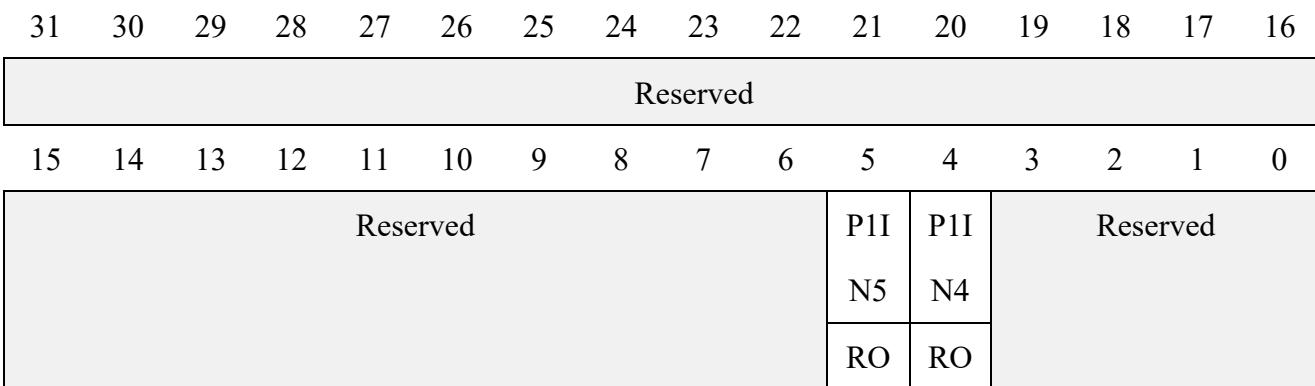
position	Marker	Function Description
31:6	Reserved	
5	P1DIR5	Port P15 Input/Output Configuration Register 1: configured as input 0: configured as output
4	P1DIR4	Port P14 Input/Output Configuration Register 1: configured as input 0: configured as output
3:0	Reserved	

6.5.2.4 Port P1 Input Value Register (P1IN)

Offset address:

0x144 Reset

value: NA



position	Marker	Function Description
31:6	Reserved	
5	P1IN5	Port P15 Input Value Register 1: Input is high 0: Input is low
4	P1IN4	Port P14 Input Value Register 1: Input is high 0: Input is low
3:0	Reserved	

6.5.2.5 Port P1 Output Value Configuration Register (P1OUT)

Offset address:

0x148 Reset

value: NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1O UT5	P1O UT4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1OUT5	Port P15 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
4	P1OUT4	Port P14 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
3:0	Reserved	

6.5.2.6 Port P1 Digital-to-Analog Configuration Register (P1ADS)

Offset address: 0x14C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1A DS5	P1A DS4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1ADS5	Port P15 Analog and Digital Configuration Register 1: Configured as an analog port 0: Configured as a digital port
4	P1ADS4	Port P14 Analog and Digital Configuration Register 1: Configured as an analog port 0: Configured as a digital port
3:0	Reserved	

6.5.2.7 Port P1 Drive Capability Configuration Register (P1DR)

Offset address: 0x15C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1D R5	P1D R4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1DR5	Port P15 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
4	P1DR4	Port P14 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
3:0	Reserved	

6.5.2.8 Port P1 pull-up enable configuration register (**P1PU**)

Offset address: 0x160

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1P U5	P1P U4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1PU5	Port P15 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
4	P1PU4	Port P14 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
3:0	Reserved	

6.5.2.9 Port P1 pull-down enable configuration register (P1PD)

Offset address: 0x164

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1P D5	P1P D4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1PD5	Port P15 pull-down enable configuration register 1: Enabled 0: Prohibition
4	P1PD4	Port P14 pull-down enable configuration register 1: Enabled 0: Prohibition
3:0	Reserved	

6.5.2.10 Port P1 Open Drain Output Configuration Register (P1OD)

Offset address: 0x16C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1O D5	P1O D4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1OD5	Port P15 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
4	P1OD4	Port P14 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
3:0	Reserved	

6.5.2.11 Port P1 High level interrupt enable configuration register (P1HIE)

Offset address: 0x170

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1H	P1H	Reserved			
										IE5	IE4				
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1HIE5	Port P15 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P1HIE4	Port P14 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3:0	Reserved	

6.5.2.12 Port P1 Low Level Interrupt Enable Configuration Register (**P1LIE**)

Offset address: 0x174

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1L	P1L	Reserved			
										IE5	IE4				
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1LIE5	Port P15 Low Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P1LIE4	Port P14 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
3:0	Reserved	

6.5.2.13 Port P1 Rising Edge Interrupt Enable Configuration Register (P1RIE)

Offset address: 0x178

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1R	P1R	Reserved			
										IE5	IE4				
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1RIE5	Port P15 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P1RIE4	Port P14 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3:0	Reserved	

6.5.2.14 Port P1 falling edge interrupt enable configuration register (P1FIE)

Offset address: 0x17C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1F IE5	P1F IE4	Reserved			
										RW	RW				

position	Marker	Function Description
31:6	Reserved	
5	P1FIE5	Port P15 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P1FIE4	Port P14 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3:0	Reserved	

6.5.2.15 Port P1 Interrupt Status Register (P1_STAT)

Offset address: 0x240

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										P1S TA5	P1S TA4	Reserved			
										RO	RO				

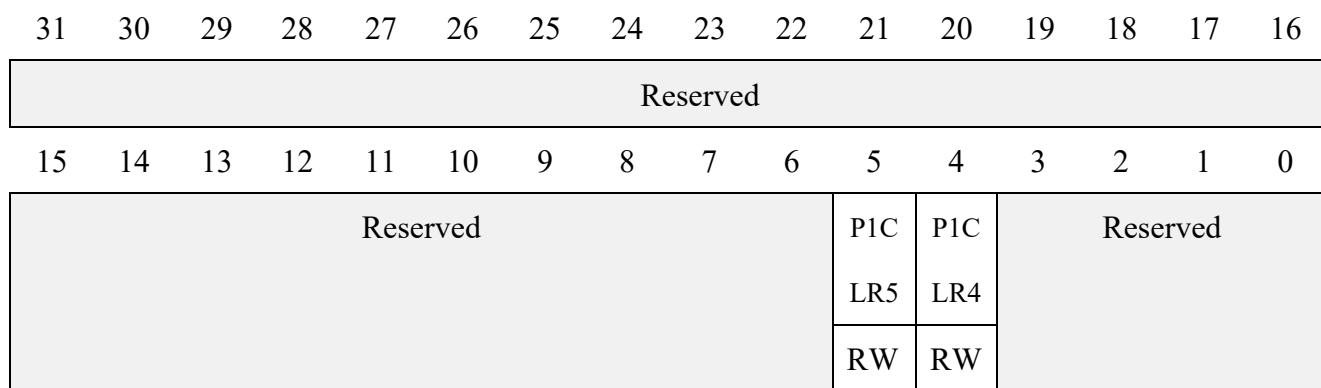
position	Marker	Function Description
31:6	Reserved	
5	P1STA5	Port P15 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
4	P1STA4	Port P14 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
3:0	Reserved	

6.5.2.16 Port P1 Interrupt Clear Register (P1_ICLR)

Offset address:

0x250 Reset

value: 0xffff ffff



position	Marker	Function Description
31:6	Reserved	
5	P1CLR5	Port P15 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
4	P1CLR4	Port P14 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
3:0	Reserved	

6.5.3 Port P2

6.5.3.1 Port P23 Function Configuration

Register (**P23_SEL**) Offset

Address: 0x8C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P23_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P23_sel	Port P23 Function Selection. 000: GPIO P23 001: TIM6_CHAAdvancedTimer Module Channel A2 Signal010: TIM4_CHBAdvancedTimer Module Channel B0 Signal011: TIM4_CHAAdvancedTimer Module Channel A 0Signal100: PCA_CHOPCAModule Channel 0Capture/Compare Signal 101: SPI_MOSISPImodule host input slave output data signal 110: UART1_RXDUART1module TXD signal 111: IR_OUT infrared output signal

6.5.3.2 Port P24 Function Configuration

Register (P24_SEL)

Offset address: 0x90

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P24_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P24_sel	Port P24 Function Selection. 000: GPIO P24 001: TIM4_CHBAdvancedTimer Module Channel B0 Signal 1010: TIM5_CHBAdvancedTimer Module Channel B1 Signal 1011: HCLK_OUTAHBBus Clock Output Signal 100: PCA_CH1PCA module channel 1Capture/compare signal 101: SPI_MOSISPI module host output slave input data signal 110: UART1_RXDUART1 module RXD signal 111: VC1_OUTVC1 module output

6.5.3.172 Port P25 Function

Configuration Register (P25_SEL)

Offset address: 0x94

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P25_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P25_sel	Port P25 Function Selection. 000: GPIO P25 001: SPI_SCKSPI module clock signal 010: PCA_CH0PCA channel 0Capture/compare signal 011: TIM5_CHA Advanced Timer module channel A1 signal 100: LVD_OUTLVD module output signal 101: LPUART_RXD LPUART module RXD signal 110: I2C_SDA I2C module data signal 111: TIM1_GATE Timer1 module gating signal

6.5.3.173 Port P26 Function

Configuration Register (P26_SEL)

Offset address: 0x98

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P26_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P26_sel	Port P26 Function Selection. 000: GPIO P26 001 SPI_MOSISPI Host Output Slave Input Data010 Signal: TIM4_CHAAdvancedTimer Module Channel A0 Signal1011: TIM5_CHBAdvancedTimer Module Channel 1 B Signal100: PCA_CH2PCAModule Channel 2Capture/Compare Signal 101: LPUART_TXD LPUART module TXD signal 110: I2C_SCL I2C module clock signal 111: TIM1_EXT Timer1 module external clock input signal

6.5.3.174 Port P174 Function configuration register (P174_SEL)

Offset address: 0x9C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P27_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P27_sel	Port P27 Function Selection. 000: GPIO P27 001SPI_MOSISP I Module Host Input Slave Output Data010 Signal: TIM5_CHAAdvancedTimer Module Channel A1 Signal1011: TIM6_CHAAdvancedTimer Module Channel A 2Signal1100: PCA_CH3PCAModule Channel 3Capture/Compare Signal 101: UART0_RXDUART0module RXD signal 110: RCH_OUT internal 24M RC clock output signal 111: XTH_OUT external 32M crystal output signal

6.5.3.6 Port P2 Input/Output Configuration Register (P2DIR)

Offset address:

0x180
Reset value: 0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P2D	P2D	P2D	P2D	P2D	Reserved		
								IR7	IR6	IR5	IR4	IR3			
								RW	RW	RW	RW	RW			

position	Marker	Function Description
31:8	Reserved	
7	P2DIR7	Port P27 Input/Output Configuration Register 1: configured as input 0: configured as output
6	P2DIR6	Port P26 Input/Output Configuration Register 1: configured as input 0: configured as output
5	P2DIR5	Port P25 Input/Output Configuration Register 1: configured as input 0: configured as output
4	P2DIR4	Port P24 Input/Output Configuration Register 1: configured as input 0: configured as output
3	P2DIR3	Port P23 Input/Output Configuration Register 1: configured as input 0: configured as output
2:0	Reserved	

6.5.3.7 Port P2 Input Value

Register (P2IN)

Offset address:

0x184

31	Reset value: NA	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P2I N7	P2I N6	P2I N5	P2I N4	P2I N3	Reserved		
								RO	RO	RO	RO	RO			

position	Marker	Function Description
31:8	Reserved	
7	P2IN7	Port P27 Input Value Register 1: Input is high 0: Input is low
6	P2IN6	Port P26 Input Value Register 1: Input is high 0: Input is low
5	P2IN5	Port P25 Input Value Register 1: Input is high 0: Input is low
4	P2IN4	Port P24 Input Value Register 1: Input is high 0: Input is low
3	P2IN3	Port P23 Input Value Register 1: Input is high 0: Input is low
2:0	Reserved	

6.5.3.8 Port P2 Output Value Configuration Register (P2OUT)

Offset address:

0x188

31	Reset value: NA	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P2O UT7	P2O UT6	P2O UT5	P2O UT4	P2O UT3	Reserved		
								RW	RW	RW	RW	RW			

position	Marker	Function Description
31:8	Reserved	
7	P2OUT7	Port P27 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
6	P2OUT6	Port P26 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
5	P2OUT5	Port P25 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
4	P2OUT4	Port P24 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.

6.5.3.8 Port P2 Output Value Configuration Register (P2OUT)

Offset address:

0x188

3	Reset Value: NA	Port P23 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
2:0	Reserved	

6.5.3.9 Port P2 Digital-to-Analog Configuration Register (P2ADS)

Offset address:

0x18C

31	Reset value: 0x00000000	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P2A DS7	P2A DS6	P2A DS5	P2A DS4	P2A DS3	Reserved		
								RW	RW	RW	RW	RW			

position	Marker	Function Description
31:8	Reserved	
7	P2ADS7	Port P27 Digital-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
6	P2ADS6	Port P26 Digital-to-analog configuration register 1: Configured as an analog port 0: Configured as a digital port
5	P2ADS5	Port P25 Digital-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
4	P2ADS4	Port P24 Digital-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
3	P2ADS3	Port P23 Analog and Digital Configuration Register 1: Configured as an analog port 0: Configured as a digital port
2:0	Reserved	

6.5.3.10 Port P2 Drive Capability Configuration Register (P2DR)

Offset address:

0x19C

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
Reserved								P2D R7	P2D R6	P2D R5	P2D R4	P2D R3	Reserved
								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2DR7	Port P27 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
6	P2DR6	Port P26 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
5	P2DR5	Port P25 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
4	P2DR4	Port P24 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
3	P2DR3	Port P23 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
2:0	Reserved	

6.5.3.11 Port P2 pull-up enable configuration register (P2PU)

Offset address:

0x1A0

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								P2P	P2P	P2P	P2P	P2P		Reserved	
								U7	U6	U5	U4	U3			
								RW	RW	RW	RW	RW			

position	Marker	Function Description
31:8	Reserved	
7	P2PU7	Port P27 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
6	P2PU6	Port P26 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
5	P2PU5	Port P25 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
4	P2PU4	Port P24 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
3	P2PU3	Port P23 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
2:0	Reserved	

6.5.3.12 Port P2 pull-down enable configuration register (P2PD)

Offset address:

0x1A4

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
Reserved								P2P	P2P	P2P	P2P	P2P	Reserved
								D7	D6	D5	D4	D3	
								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2PD7	Port P27 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
6	P2PD6	Port P26 Pull-down Enable Configuration Register 1: Enabled 0: Prohibition
5	P2PD5	Port P25 pull-down enable configuration register 1: Enabled 0: Prohibition
4	P2PD4	Port P24 pull-down enable configuration register 1: Enabled 0: Prohibition
3	P2PD3	Port P23 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
2:0	Reserved	

6.5.3.13 Port P2 Open Drain Output Configuration Register (P2OD)

Offset address:

0x1AC

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
								P2O D7	P2O D6	P2O D5	P2O D4	P2O D3	Reserved
								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2OD7	Port P27 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
6	P2OD6	Port P26 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
5	P2OD5	Port P25 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
4	P2OD4	Port P24 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
3	P2OD3	Port P23 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
2:0	Reserved	

6.5.3.14 Port P2 High Level Interrupt Enable Configuration Register (P2HIE)

Offset address:

0x1B0

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
Reserved								P2H	P2H	P2H	P2H	P2H	Reserved
								IE7	IE6	IE5	IE4	IE3	
								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2HIE7	Port P27 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
6	P2HIE6	Port P26 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
5	P2HIE5	Port P25 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P2HIE4	Port P24 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3	P2HIE3	Port P23 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2:0	Reserved	

6.5.3.15 Port P2 Low Level Interrupt Enable Configuration Register (P2LIE)

Offset address:

0x1B4

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
Reserved								P2L	P2L	P2L	P2L	P2L	Reserved
								IE7	IE6	IE5	IE4	IE3	
								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2LIE7	Port P27 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
6	P2LIE6	Port P26 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
5	P2LIE5	Port P25 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
4	P2LIE4	Port P24 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
3	P2LIE3	Port P23 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
2:0	Reserved	

6.5.3.16 Port P2 Rising Edge Interrupt Enable Configuration Register (P2RIE)

Offset address:

0x1B8

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
Reserved								P2R	P2R	P2R	P2R	P2R	Reserved
								IE7	IE6	IE5	IE4	IE3	
								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2RIE7	Port P27 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
6	P2RIE6	Port P26 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
5	P2RIE5	Port P25 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P2RIE4	Port P24 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3	P2RIE3	Port P23 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2:0	Reserved	

6.5.3.17 Port P2 Falling Edge Interrupt Enable Configuration Register (P2FIE)

Offset address:

0x1BC

31	Reset value: 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
Reserved								P2F	P2F	P2F	P2F	P2F	P2F	Reserved
								IE7	IE6	IE5	IE4	IE3		
								RW	RW	RW	RW	RW		

position	Marker	Function Description
31:8	Reserved	
7	P2FIE7	Port P27 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
6	P2FIE6	Port P26 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
5	P2FIE5	Port P25 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P2FIE4	Port P24 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3	P2FIE3	Port P23 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2:0	Reserved	

6.5.3.18 Port P2 Interrupt Status

Register (**P2_STAT**)

Offset address:

0x280

31	Reset value: 30 29 28 0x0000 0006	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								P2S TA7	P2S TA6	P2S TA5	P2S TA4	P2S TA3			
								RO	RO	RO	RO	RO			

position	Marker	Function Description
31:8	Reserved	
7	P2STA7	Port P27 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
6	P2STA6	Port P26 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
5	P2STA5	Port P25 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
4	P2STA4	Port P24 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
3	P2STA3	Port P23 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
2:0	Reserved	

6.5.3.19 Port P2 Interrupt Clear

Register (**P2_ICLR**)

Offset address:

0x290

31	Reset value: 30 0xfffffff 29 28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2
Reserved								P2C LR7	P2C LR6	P2C LR5	P2C LR4	P2C LR3	Reserved
RW								RW	RW	RW	RW	RW	

position	Marker	Function Description
31:8	Reserved	
7	P2CLR7	Port P27 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
6	P2CLR6	Port P26 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
5	P2CLR5	Port P25 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
4	P2CLR4	Port P24 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
3	P2CLR3	Port P23 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
2:0	Reserved	

6.5.4 Port P3

6.5.4.1 Port P31 Function Configuration

Register (**P31_SEL**) Offset

Address: 0xC4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P31_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P31_sel	Port P31 Function Selection. 000: GPIO P31 001: LPTIM_TOGTimer3Module flip signal 010: PCA_ECIPIPCAmodule external clock input signal 011: PCLK_OUTAPBbus clock output signal 100: VC0_OUTVC0module output 101: UART0_TXDUART0module TXD signal 110: RCL_OUT internal 38K RC clock output signal 111: HCLK_OUTAHBbus clock output signal

6.5.4.2 Port P32 Function Configuration

Register (P32_SEL)

Offset address: 0xC8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P32_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P32_sel	Port P32 Function Selection. 000: GPIO P32 001: LPTIM_TOGNNTimer3Inverse signal of the module flip signal 0102Capture/Compare Signal 1011: TIM6_CHBAdvancedTimer Module Channel B2 Signal 1100: VC1_OUTVC1Module Output 101: UART1_TXDUART1module TXD signal 110: PCA_CH4PCAmodule channel 4Capture/compare signal 111: RTC_1HzRTCmodule 1Hz output signal

6.5.4.192 Port P33 Function

Configuration Register (P33_SEL)

Offset address: 0xCC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P33_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P33_sel	Port P33 Function Selection. 000: GPIO P33 001: LPUART_RXD LPUART module RXD signal 010: PCA_CH1 PCA channel 1Capture/compare signal 011: TIM5_CHB Advanced Timer module channel B1 signal 100: PCA_ECI PCA module external clock input signal 101: UART1_RXD UART1 module RXD signal 110: XTL_OUT external 32K crystal output signal 111: TIM1_TOGN Timer1 Inverted signal of the module flip-flop signal

6.5.4.193 Port P34 Function

Configuration Register (P34_SEL)

Offset address: 0xD0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P34_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P34_sel	Port P34 Function Selection. 000: GPIO P34 001: PCA_CH0PCAmodule channel 0Capture/compare signal 010: LPUART_TXDLPUARTmodule TXD signal 011: TIM5_CHAAdvancedTimer Module Channel A1 Signal1100: TIM0_EXTTimer0External Clock Input Signal1101: TIM4_CHAAdvancedTimer Module Channel A0 Signal1110: RTC_1HzRTCModule 1Hz Output Signal 111: TIM1_TOGTimer1module flip signal

6.5.4.194 Port P194 Function configuration register (P194_SEL)

Offset address: 0xD4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P35_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P35_sel	Port P35 Function Selection. 000: GPIO P35 001: UART1_TXDUART1module TXD signal 010: TIM6_CHBAdvancedTimer Module Channel B 2signal 011: UART0_TXDUART0module TXD signal 100: TIM0_GATETimer0gating signal101: TIM4_CHBAdvancedTimer module channel 0 B signal110: SPI_MOSISPImodule host input slave output data signal 111: I2C_SDAI2Cmodule data signal

6.5.4.195 Port P195 Function configuration register (P195_SEL)

Offset address: 0xD8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												P36_sel			
												RW			

position	Marker	Function Description
31:3	Reserved	
2:0	P36_sel	Port P36 Function Selection. 000: GPIO P36 001: UART1_RXDUART1module RXD signal 010: TIM6_CHAAdvancedTimer module channel A 2signal 011: UART0_RXDUART0module RXD signal 100: PCA_CH4PCAchannel 4Capture/compare signal 101: TIM5_CHAAdvancedTimer module channel A1 signal 110: SPI_MOSISPImodule host output slave input data signal 111: I2C_SCLI2Cmodule clock signal

6.5.4.7 Port P3 Input/Output Configuration Register (P3DIR)

Offset address:

0x1C0 Reset

value: 0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3D	Res						
								IR6	IR5	IR4	IR3	IR2	IR1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3DIR6	Port P36 Input/Output Configuration Register 1: configured as input 0: configured as output
5	P3DIR5	Port P35 Input/Output Configuration Register 1: configured as input 0: configured as output
4	P3DIR4	Port P34 Input/Output Configuration Register 1: configured as input 0: configured as output
3	P3DIR3	Port P33 Input/Output Configuration Register 1: configured as input 0: configured as output
2	P3DIR2	Port P32 Input/Output Configuration Register 1: configured as input 0: configured as output

1	P3DIR1	Port P31 Input/Output Configuration Register
---	--------	--

		1: configured as input 0: configured as output
0	Reserved	

6.5.4.8 Port P3 Input Value Register (P3IN)

Offset address:

0x1C4 Reset

value: NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3I	P3I	P3I	P3I	P3I	P3I	Res	
								N6	N5	N4	N3	N2	N1		
								RO	RO	RO	RO	RO	RO		

position	Marker	Function Description
31:7	Reserved	
6	P3IN6	Port P36 Input Value Register 1: Input is high 0: Input is low
5	P3IN5	Port P35 Input Value Register 1: Input is high 0: Input is low
4	P3IN4	Port P34 Input Value Register 1: Input is high 0: Input is low
3	P3IN3	Port P33 Input Value Register 1: Input is high 0: Input is low
2	P3IN2	Port P32 Input Value Register 1: Input is high 0: Input is low

1	P3IN1	Port P31 Input Value Register
---	-------	-------------------------------

		1: Input is high 0: Input is low
0	Reserved	

6.5.4.9 Port P3 Output Value Configuration Register (P3OUT)

Offset address:

0x1C8 Reset

value: NA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3O	Res						
								UT6	UT5	UT4	UT3	UT2	UT1		
								RW							

position	Marker	Function Description
31:7	Reserved	
6	P3OUT6	Port P36 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
5	P3OUT5	Port P35 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
4	P3OUT4	Port P34 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
3	P3OUT3	Port P33 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.

2	P3OUT2	Port P32 Output Value Configuration Register 1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
1	P3OUT1	Port P31 Output Value Configuration Register

		1: Output high. If configured as an open-drain output, an external pull-up resistor is required to pull high. 0: Output low.
0	Reserved	

6.5.4.10 Port P3 Digital-to-Analog Configuration Register (P3ADS)

Offset address: 0x1CC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3A	Res						
								DS6	DS5	DS4	DS3	DS2	DS1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3ADS6	Port P36 Digital-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
5	P3ADS5	Port P35 Digital-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
4	P3ADS4	Port P34 Digital-Analog Configuration Register 1: Configured as an analog port 0: Configured as a digital port
3	P3ADS3	Port P33 Analog and Digital Configuration Register 1: Configured as an analog port 0: Configured as a digital port
2	P3ADS2	Port P32 Analog and Digital Configuration Register 1: Configured as an analog port 0: Configured as a digital port
1	P3ADS1	Port P31 Digital to analog configuration register

		1: Configured as an analog port 0: Configured as a digital port
0	Reserved	

6.5.4.11 Port P3 Drive Capability Configuration Register (P3DR)

Offset address: 0x1DC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3D	Res						
								R6	R5	R4	R3	R2	R1		
								RW							

position	Marker	Function Description
31:7	Reserved	
6	P3DR6	Port P36 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
5	P3DR5	Port P35 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
4	P3DR4	Port P34 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
3	P3DR3	Port P33 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
2	P3DR2	Port P32 Drive Capability Configuration Register 1: Low drive capability 0: High drive capability
1	P3DR1	Port P31 Drive Capability Configuration Register

		1: Low drive capability 0: High drive capability
0	Reserved	

6.5.4.12 Port P3 pull-up enable configuration register (P3PU)

Offset address: 0x1E0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3P	Res						
								U6	U5	U4	U3	U2	U1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3PU6	Port P36 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
5	P3PU5	Port P35 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
4	P3PU4	Port P34 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
3	P3PU3	Port P33 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
2	P3PU2	Port P32 Pull-up Enable Configuration Register 1: Enabled 0: Prohibition
1	P3PU1	Port P31 Pull-up Enable Configuration Register

		1: Enabled 0: Prohibition
0	Reserved	

6.5.4.13 Port P3 pull-down enable configuration register (P3PD)

Offset address: 0x1E4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3P	Res						
								D6	D5	D4	D3	D2	D1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3PD6	Port P36 Pull-down Enable Configuration Register 1: Enabled 0: Prohibition
5	P3PD5	Port P35 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
4	P3PD4	Port P34 Pull-down Enable Configuration Register 1: Enabled 0: Prohibition
3	P3PD3	Port P33 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
2	P3PD2	Port P32 Pull Down Enable Configuration Register 1: Enabled 0: Prohibition
1	P3PD1	Port P31 pull-down enable configuration register

		1: Enabled 0: Prohibition
0	Reserved	

6.5.4.14 Port P3 Open Drain Output Configuration Register (P3OD)

Offset address: 0x1EC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3O	Res						
								D6	D5	D4	D3	D2	D1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3OD6	Port P36 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
5	P3OD5	Port P35 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
4	P3OD4	Port P34 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
3	P3OD3	Port P33 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
2	P3OD2	Port P32 Open Drain Output Configuration Register 1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
1	P3OD1	Port P31 Open Drain Output Configuration Register

		1: Set the port output method to open-drain output 0: Set the port output method to push-pull output
0	Reserved	

6.5.4.15 Port P3 High level interrupt enable configuration register (P3HIE)

Offset address: 0x1F0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3H	P3H	P3H	P3H	P3H	P3H	Res	
								IE6	IE5	IE4	IE3	IE2	IE1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3HIE6	Port P36 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
5	P3HIE5	Port P35 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P3HIE4	Port P34 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3	P3HIE3	Port P33 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2	P3HIE2	Port P32 High Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
1	P3HIE1	Port P31 High Interrupt Enable Configuration Register

		1: Enabled 0: Prohibition
0	Reserved	

6.5.4.16 Port P3 Low Level Interrupt Enable Configuration Register (P3LIE)

Offset address: 0x1F4

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3L	P3L	P3L	P3L	P3L	P3L	Res	
								IE6	IE5	IE4	IE3	IE2	IE1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3LIE6	Port P36 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
5	P3LIE5	Port P35 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
4	P3LIE4	Port P34 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
3	P3LIE3	Port P33 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
2	P3LIE2	Port P32 Low-level interrupt enable configuration register 1: Enabled 0: Prohibition
1	P3LIE1	Port P31 Low-level interrupt enable configuration register

		1: Enabled 0: Prohibition
0	Reserved	

6.5.4.17 Port P3 Rising Edge Interrupt Enable Configuration Register (P3RIE)

Offset address: 0x1F8

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3R	Res						
								IE6	IE5	IE4	IE3	IE2	IE1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3RIE6	Port P36 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
5	P3RIE5	Port P35 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P3RIE4	Port P34 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3	P3RIE3	Port P33 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2	P3RIE2	Port P32 Rising Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
1	P3RIE1	Port P31 Rising Edge Interrupt Enable Configuration Register

		1: Enabled 0: Prohibition
0	Reserved	

6.5.4.18 Port P3 Falling Edge Interrupt Enable Configuration Register (P3FIE)

Offset address: 0x1FC

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3F	P3F	P3F	P3F	P3F	P3F	Res	
								IE6	IE5	IE4	IE3	IE2	IE1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3FIE6	Port P36 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
5	P3FIE5	Port P35 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
4	P3FIE4	Port P34 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
3	P3FIE3	Port P33 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
2	P3FIE2	Port P32 Falling Edge Interrupt Enable Configuration Register 1: Enabled 0: Prohibition
1	P3FIE1	Port P31 Falling Edge Interrupt Enable Configuration Register

		1: Enabled 0: Prohibition
0	Reserved	

6.5.4.19 Port P3 Interrupt Status Register (P3_STAT)

Offset address: 0x2C0

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3S	P3S	P3S	P3S	P3S	P3S	Res	
								TA6	TA5	TA4	TA3	TA2	TA1		
								RO	RO	RO	RO	RO	RO		

position	Marker	Function Description
31:7	Reserved	
6	P3STA6	Port P36 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
5	P3STA5	Port P35 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
4	P3STA4	Port P34 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
3	P3STA3	Port P33 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
2	P3STA2	Port P32 Interrupt Status Register 1: Interrupt trigger 0: No interrupt trigger
1	P3STA1	Port P31 Interrupt Status Register

		1: Interrupt trigger 0: No interrupt trigger
0	Reserved	

6.5.4.20 Port P3 Interrupt Clear Register (P3_ICLR)

Offset address:

0x2D0 Reset

value: 0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								P3C	Res						
								LR6	LR5	LR4	LR3	LR2	LR1		
								RW	RW	RW	RW	RW	RW		

position	Marker	Function Description
31:7	Reserved	
6	P3CLR6	Port P36 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
5	P3CLR5	Port P35 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
4	P3CLR4	Port P34 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
3	P3CLR3	Port P33 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
2	P3CLR2	Port P32 Interrupt Clear Register 1: Reserved interrupt flag bit 0: Clear the interrupt flag bit

1	P3CLR1	Port P31 Interrupt Clear Register
---	--------	-----------------------------------

		1: Reserved interrupt flag bit 0: Clear the interrupt flag bit
0	Reserved	

6.5.5 Port Assist Function

6.5.5.1 Port Auxiliary Function Configuration Register 1(GPIO_CTRL1)

Offset address: 0x304

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ir_p ol	hclk _en	pclk _en	hclk_sel	pclk_sel			ssn_sel			ext_clk_sel				
	RW	RW	RW	RW	RW			RW			RW				

position	Marker	Function Description
31:15	Reserved	
14	ir_pol	IR output polarity selection. 0 – Positive output 1 – Reverse Output
13	hclk_en	hclk output control. 0 – Outputs 0 1 – Output hclk
12	pclk_en	pclk output gating. 0 – Output 0 1 – Output pclk

11:10	hclk_sel	hclk output crossover selection00: hclk 01: hclk/210 : hclk/4 11: hclk/8
-------	----------	---

9:8	pclk_sel	pclk output crossover selection00: hclk 01: hclk/210 : hclk/4 11: hclk/8
7:4	ssn_sel	SPI SSN Signal Source Selection 0000: High level 1000: P23 0001: P351001: P24 0010: P361010: P25 0011: P011011: P26 0100: P021100: P27 0101: P031101: P31 0110: P151110: P32 0111: P141111: P33
3:0	ext_clk_sel	External clock signal source selection 0000: High level 1000: P23 0001: P351001: P24 0010: P361010: P25 0011: P011011: P26 0100: P021100: P27 0101: P031101: P31 0110: P151110: P32 0111: P141111: P33

6.5.5.2 Port Auxiliary Function Configuration Register 2(GPIO_CTRL2)

Offset address: 0x308

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					pca_cap4 _sel	pca_cap3 _sel	pca_cap2 _sel	pca_cap1 _sel	pca_cap0 _sel						
					RW	RW	RW	RW	RW						

position	Marker	Function Description
31:10	Reserved	
9:8	pca_cap4_sel	PCA Capture Channel Signal4 Source Selection 00: PCA_CH4 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
7:6	pca_cap3_sel	PCA Capture Channel Signal3 Source Selection 00: PCA_CH3 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
5:4	pca_cap2_sel	PCA Capture Channel Signal2 Source Selection 00: PCA_CH2 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
3:2	pca_cap1_sel	PCA Capture Channel Signal1 Source Selection

		00: PCA_CH1 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
1:0	pca_cap0_sel	PCA Capture Channel Signal0 Source Selection 00: PCA_CH0 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD

6.5.5.3 Port Auxiliary Function Configuration Register 3(GPIO_CTRL3)

Offset address: 0x30C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		tm6_a_sel	tm5_a_sel	tm4_a_sel	tm6_b_sel	tm5_b_sel	tm4_b_sel								
		RW	RW	RW	RW	RW	RW								

position	Marker	Function Description
31:12	Reserved	
11:10	tm6_a_sel	Timer6 A channel signal source selection 00 : TIM6_CHA 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
9:8	tm5_a_sel	Timer5 A channel signal source selection 00 : TIM5_CHA 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
7:6	tm4_a_sel	Timer4 A channel signal source selection 00: TIM4_CHA 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
5:4	tm6_b_sel	Timer6 B channel signal source selection 00: TIM6_CHB

		01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
3:2	tm5_b_sel	Timer5 B channel signal source selection 00: TIM5_CHB 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
1:0	tm4_b_sel	Timer4 B channel signal source selection 00: TIM4_CHB 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD

6.5.5.4 Port Auxiliary Function Configuration Register 4(GPIO_CTRL4)

Offset address: 0x310

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								tm3_gate_ sel	tm2_gate_ sel	tm1_gate_ sel	tm0_gate_ sel				
								RW	RW	RW	RW				

position	Marker	Function Description
31:8	Reserved	
7:6	tm3_gate_sel	Timer3 Gating input signal source selection 00: LPTIM_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
5:4	tm2_gate_sel	Timer2 Gating input signal source selection 00: TIM2_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
3:2	tm1_gate_sel	Timer1 Gating input signal source selection 00: TIM1_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
1:0	tm0_gate_sel	Timer0 Gating input signal source selection

		00: TIM0_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
--	--	---

7 FLASH Controller (FLASH)

7.1 Overview

This device contains a 32kByte FLASH memory divided into 64sectors, each sector is 512Byte, and this module supports erase, program and read operations on this memory. In addition, this module supports the protection of FLASH memory erase and write protection of control registers.

7.2 Structure Block Diagram

Address Range	Sector Serial Number	Address Range	Sector Serial Number
0x1E00-0x1FFF	Sector15	0x7E00-0x7FFF	Sector63
0x1C00-0x1DFF	Sector14	0x7C00-0x7DFF	Sector62
0x1A00-0x1BFF	Sector13	0x7A00-0x7BFF	Sector61
0x1800-0x19FF	Sector12	0x7800-0x79FF	Sector60
0x1600-0x17FF	Sector11	0x7600-0x77FF	Sector59
0x1400-0x15FF	Sector10	0x7400-0x75FF	Sector58
0x1200-0x13FF	Sector9	0x7200-0x73FF	Sector57
0x1000-0x11FF	Sector8	0x7000-0x71FF	Sector56
0x0E00-0x0FFF	Sector7	0x6E00-0x6FFF	Sector55
0x0C00-0x0DFF	Sector6	0x6C00-0x6DFF	Sector54
0x0A00-0x0BFF	Sector5	0x6A00-0x6BFF	Sector53
0x0800-0x09FF	Sector4	0x6800-0x69FF	Sector52
0x0600-0x07FF	Sector3	0x6600-0x67FF	Sector51
0x0400-0x05FF	Sector2	0x6400-0x65FF	Sector50
0x0200-0x03FF	Sector1	0x6200-0x63FF	Sector49
0x0000-0x01FF	Sector0	0x6000-0x61FF	Sector48

Figure 7-1 Memory Sector Segmentation

7.3 Function Description

This controller supports three bit width modes FLASH read and write: Byte (8bits) Half-word (16bits) Word

(32bits) Note that the address of a Byte operation must be aligned by Byte, the target address of a Half-word operation must be aligned by Half-word (the lowest bit of the address is 0) and the address of a Word operation must be aligned by Word (the lowest two bits of the address are 0). If the address of the read/write operation is not aligned as described above, the system will enter a Hard Fault error interrupt.

7.3.1 Page Erase (Sector Erase)

The page erase can erase one page (Sector) at a time specified by the user. If the erase operation is performed from FLASH, the CPU will stop fetching and the hardware will wait for the operation to complete automatically (FLASH_CR.BUSY becomes 0) if the erase operation is performed from RAM, the CPU will not stop fetching and the user software should wait for the operation to complete (FLASH_CR.BUSY becomes 0). BUSY becomes 0)

The page (Sector) erasure procedure is as follows.

Step1: Configure the FLASH erase parameters, see the Erase Timing chapter for details.

Step2: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step3: Configure FLASH_CR.OP to 2 set the Flash operation mode to Sector erase.

Step4: Check whether FLASH_CR.OP is 2, if not then skip to Step2.

Step5: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register to enable the rewrite of the register. **Step8:** Write any data to any address in the Sector to be erased to trigger the Sector erasure.

*Example: * (unsigned char *)0x00000200) = 0x00.*

Step9: Wait for FLASH_CR.BUSY to become 0, Sector

erase operation is finished. **step10:** If you want to
erase other Sectors, repeat Step5 - Step9.

7.3.2 Chip Erase

Full sheet erase can erase all pages (Sectors). After the erase operation is completed, all pages (Sectors) in the

If the erase operation is performed from FLASH, the CPU will stop fetching and the hardware will automatically wait for the operation to complete (FLASH_CR.BUSY becomes 0) if the erase operation is performed from RAM, the CPU will not stop fetching and the user software should wait for the operation to complete (FLASH_CR.BUSY becomes 0)

The steps for the full-slice erase operation are as follows.

Step1: Configure the FLASH erase parameters, see the Erase Timing chapter for details.

Step2: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step3: Configure FLASH_CR.OP to 3 set the Flash operation mode to Chip erase.

Step4: Check whether FLASH_CR.OP is 3, if not then skip to Step2.

Step5: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step6: Set FLASH_SLOCK to 0xFFFF to remove the erase protection of all Sectors. **Step7:** Check if FLASH_SLOCK is 0xFFFF, if not, skip to Step5. **Step8:** Write to any address in the Chip to be erased to trigger Chip erase.

*Example: * (unsigned char *)0x00000000) = 0x00.*

Step9: Wait for FLASH_CR.BUSY to be changed to 0, Chip erase operation is completed.

7.3.3 Write operation (Program)

The write operation can only write 10bits of 3data in FLASH, so you should make sure the data in the address to be written is 0xFF before writing the data. The written data is stored in FLASH in small-end mode, i.e., the low address stores the low byte of data. If the write operation is executed from FLASH, the CPU will stop fetching and the hardware will automatically wait for the operation to complete (FLASH_CR.BUSY becomes 0) if the write operation

is executed from RAM, the CPU will not stop fetching and the user software should wait for the operation to complete (FLASH_CR.BUSY becomes 0)

Byte write operation steps are as follows.

Step1: Configure the FLASH erase parameters, see the Erase Timing chapter for details.

Step2: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step3: Configure FLASH_CR.OP to set the Flash operation mode to write.

Step4: Check whether FLASH_CR.OP is 1, if not then skip to Step2.

Step5: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step6: Set the corresponding bit of FLASH_SLOCK to 1, to remove the erase protection.

Step7: Check if the corresponding bit of FLASH_SLOCK is 1, if not then1 skip to Step5.

Step8: Byte write operation to the target address to be written triggers the write operation.

*Example: * (unsigned char *)0x00001231) = 05xA.*

Step9: Wait for FLASH_CR.BUSY to change to 0, the write operation is completed.

Step10: If you need to write Byte to another address with erasure protection removed, repeat Step8 - Step9.

The half-word writing procedure is as follows.

Step1: Configure the FLASH erase time, see the Erase Timing chapter for details.

Step2: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step3: Configure FLASH_CR.OP to set the Flash operation mode to write.

Step4: Check whether FLASH_CR.OP is 1, if not then1 skip to Step2.

Step5: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step6: Set the corresponding bit of FLASH_SLOCK to 1, to remove the erase protection.

Step7: Check whether the corresponding bit of FLASH_SLOCK is 1, if not then1 skip to Step5

Step8: Perform a Half-word write operation to the target address to be written to, triggering the write operation.

*Example: * (unsigned short *)0x00001232) = 0xABCD.*

Step9: Wait for FLASH_CR.BUSY to change to 0, the write operation is completed.

Step10: If you need to write Half-word to another address with erasure protection removed, repeat Step8 - Step9.

The steps of the Word writing operation are as follows.

Step1: Configure the FLASH erase parameters, see the Erase Timing chapter for details.

Step2: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step3: Configure FLASH_CR.OP to set the Flash operation mode to write.

Step4: Check whether FLASH_CR.OP is 1, if not then skip to Step2.

Step5: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step6: Set the corresponding bit of FLASH_SLOCK to 1, to remove the erase protection.

Step7: Check whether the corresponding bit of FLASH_SLOCK is 1, if not then skip to Step5

Step8: Perform a Word write operation to the target address to be written to, triggering the write operation.

*Example: * (unsigned long *) 0x00001234) = 0x55667788.*

Step9: Wait for FLASH_CR.BUSY to change to 0, the write operation is completed.

Step10: If you need to write Word to another address with erasure protection removed, repeat Step8 - Step9.

7.3.4 Read operation

The read 3data lengths: Byte (8bits) Half-word (16bits) Word (32bits), the read data is in small-end mode, i.e., the low address stores the low byte of the data. The read operation does not require an operation step, and the data in FLASH can be read at any time.

Byte read operation temp = * (unsigned char *) 0x00001231)
example.

Half-word read temp = * (unsigned int*) 0x00001232)
operation example

Word read operation temp = * (unsigned long *) 0 x00001234)
example

7.4 Erase Timing

FLASH memory has strict timing requirements for the control signals of erase and program operations, and failure of the timing of the control signals will cause the erase and program operations to fail. If the HCLK frequency is not 4MHz, the user program should load the erase parameters corresponding to the HCLK frequency. 对 FLASH 进行操作时，需要 HCLK 的频率范围为 1MHz ~ 32MHz。擦写时序参数相关的寄存器为：FLASH_TNVS、FLASH_TPGS、FLASH_TPROG、FLASH_TSERASE、FLASH_TMERASE、FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV. If HCLK is raised from the default 4MHz to 8MHz, the value of the FLASH_Tx register above should be set to a multiple of 2 the default value, i.e., just keep the current Tsysclk*FLASH_Tx result equal to the default value.

The following table shows the corresponding FLASH erase timing parameters at different frequencies.

	4M	8M	16M	24M	32M
FLASH_TNVS	0x20	0x40	0x80	0xC0	0x100
FLASH_TPGS	0x17	0x2E	0x5C	0x8A	0xB8
FLASH_TPROG	0x1B	0x36	0x6C	0xA2	0xD8
FLASH_TSERAS	0x4650	0x8CA0	0x11940	0x1A5E0	0x23280
FLASH_TMERASE	0x222E0	0x445C0	0x88B80	0xCD140	0x111700
FLASH_TPRCV	0x18	0x30	0x60	0x90	0xC0
FLASH_TSRCV	0xF0	0x1E0	0x3C0	0x5A0	0x780
FLASH_TMRCV	0x3E8	0x7D0	0xFA0	0x1770	0x1F40

Table 7-1 FLASH erase time parameters at different frequencies

To configure the erase parameters for a system frequency of 8 MHz, proceed as follows.

Step1: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step2: Write 0x40 to FLASH_TNVS register, if the read value of this sender is HC32L110 Series User's Manual Rev2.31

not 0x40, then skip to the previous step.

Step3: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewrite.

Step4: Write 0x2E to FLASH_TPGS register, if the read value of this sender is not 0x2E, then skip to the previous step.

Step5: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step6: Write 0x36 to FLASH_TPROG register, if the read value of this sender is not 0x36, then skip to the previous step.

Step7: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step8: Write 0x8CA0 to FLASH_TSERASE register, if the value of the sender is not 0x8CA0, then skip to the previous step.

Step9: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step10: Write 0xC4450 to FLASH_TMERASE register, if the value of this sender read is not 0445xC0, then skip to the previous step.

Step11: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step12: Write 0x30 to FLASH_TPRCV register, if the value of this sender read is not 0x30, then skip to the previous step.

Step13: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step14: Write 0x1E0 to FLASH_TSRCV register, if the value of this sender read is not 0x1E0, then skip to the previous step.

Step15: Write 0x5A5A, 0xA5A5 to FLASH_BYPASS register in order to enable register rewriting.

Step16: Write 0x7D0 to FLASH_TMRCV register, if the read value of this sender is not 0x7D0, then skip to the previous step.

7.5 Read Waiting Cycle

When the HCLK frequency exceeds 24MHz, the CPU must set `FLASH_CR.WAIT` to insert a wait cycle for CPU fetch. When the wait cycle is inserted, the CPU only reads the instruction code in the FLASH memory once every two cycles.

7.6 Erase protection

7.6.1 Erase protection bit

The entire 32kByte FLASH memory is divided into 64Sectors, and each 4Sector shares a single erase protection bit. When a Sector is protected, the erase operation of the Sector is invalid and an alarm flag and interrupt signal are generated. When any Sector in the FLASH memory is protected, the Chip erase of the FLASH memory is invalid, and interrupt signal are generated.

7.6.2 PC address erase protection

When the CPU runs a program in FLASH, if the current PC pointer falls within the Sector address to be erased, the erase operation is invalidated and an alarm flag bit and interrupt signal are generated.

7.7 Register write protection

The important controllers of this module block ordinary write operations and must be modified only by the write sequence method. The registers that need to be changed by the write sequence method are shown below.

FLASH_TNVSFLASH_TPGSFLASH_TPROGFLASH_TSERASEFLASH_TMERASE,
FLASH_TPRCV, FLASH_TSRCV, FLASH_TMRCV, FLASH_CR, FLASH_SLOCK.

The registers that can be changed without writing sequences are shown below.

FLASH_ICLR, FLASH_BYPASS.

The steps to modify the register value by writing sequence are shown below: **Step1:** Write 0x5A5A to FLASH_BYPASS register. **Step2:** Write 0xA5A5 to FLASH_BYPASS register. **Step3:** Write the target value to the register to be modified.

Step4: Verify whether the current value of the register to be modified is the same as the target value, if not, jump to Step1.

Step5: Perform other operations.

Caution.

- Write 0x5a5a and write 0xa5a5 cannot insert any write operation between these two steps, and cannot be interrupted, otherwise the Bypass sequence will be invalidated and the 0x5a5a-0xa5a5 sequence will need to be rewritten.

7.8 Register

Base address: 0x4002 0000

Register	Offset Address	Description
FLASH_TNVS	0x00	Tnvs time parameter
FLASH_TPGS	0x04	Tpgs time parameter
FLASH_TPROG	0x08	Tprog time parameter
FLASH_TSERASE	0x0C	Tserase time parameters
FLASH_TMERASE	0x10	Tmerase time parameters
FLASH_TPRCV	0x14	Tprcv time parameter
FLASH_TSRCV	0x18	Tsrcv time parameter
FLASH_TMRCV	0x1C	Tmrcv time parameter
FLASH_CR	0x20	Control register
FLASH_IFR	0x24	Interrupt Flag Register
FLASH_ICLR	0x28	Interrupt Flag Clear Register
FLASH_BYPASS	0x2C	0x5a5a-0xa5a5 Bypass Sequence Register
FLASH_SLOCK	0x30	Sector Erasure protection register

7.8.1 TNVS parameter register (FLASH_TNVS)

Offset address: 0x00

Reset value: 0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TNVS							
R								RW							

position	Marker	Function Description
31:9	Reserved	
8:0	TNVS	Calculation formula: TNVS = 8*HCLK, the unit of HCLK is MHz, and the method of modifying this register value is detailed in 7.7 . 4MHz example: TNVS = 8*4 = 32.

7.8.2 TPGS parameter register (FLASH_TPGS)

Offset address: 0x04

Reset value: 0x0000 0017

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TPGS							
R								RW							

position	Marker	Function Description
31:8	RESERVED	
7:0	TPGS	Calculation formula: TPGS = 5.75*HCLK, the unit of HCLK is MHz. to modify this register value, please refer to 7.7 . 4MHz example: TPGS = 5.75*4 = 23.

7.8.3 TPROG parameter register (FLASH_TPROG)

Offset address: 0x08

Reset value: 0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TPROG							
R								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	TPROG	Calculation formula: TPROG = 6.75*HCLK, the unit of HCLK is MHz. to modify this register value, please refer to 7.7 . 4MHz example: TPROG = 6.75*4 = 27.

7.8.4 TSERASE register (FLASH_TSERASE)

Offset address: 0x0C

Reset value: 0x0000 4650

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														TSERASE	
														RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSERASE															
														RW	

position	Marker	Function Description
31:18	Reserved	
17:0	TSERASE	Calculation formula: TSERASE = 4500*HCLK, the unit of HCLK is MHz. to modify this register value, please refer to 7.7 . 4MHz example: TSERASE = 4500*4 = 18000.

7.8.5 TMERASE parameter register (FLASH_TMERASE)

Offset address: 0x10

Reset value: 0x000222E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														TMERASE	
														RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMERASE															
														RW	

position	Marker	Function Description
31:21	Reserved	
20:0	TMERASE	Calculation formula: TMERASE = 35000*HCLK, the unit of HCLK is MHz, the method to modify this register value is detailed in 7.7 . 4MHz example: TMERASE = 35000*4 = 140000.

7.8.6 TPRCV parameter register (FLASH_TPRCV)

Offset address: 0x14

Reset value: 0x0000 0018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TPRCV											
R				RW											

position	Marker	Function Description
31:12	Reserved	
11:0	TPRCV	Calculation formula: TPRCV = 6*HCLK, the unit of HCLK is MHz. 7.7 . 4MHz example: TPRCV = 6*4 = 24.

7.8.7 TSRCV parameter register (FLASH_TSRCV)

Offset address: 0x18

Reset value: 0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TSRCV											
R				RW											

position	Marker	Function Description
31:12	Reserved	
11:0	TSRCV	Calculation formula: TSRCV = 60*HCLK, the unit of HCLK is MHz. to modify this register value, please refer to 7.7 . 4MHz example: TSRCV = 60*4 = 240.

7.8.8 TMRCV parameter register (FLASH_TMRCV)

Offset address: 0x1C

Reset value: 0x0000 03E8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TMRCV													
R		RW													

position	Marker	Function Description
31:13	RESERVED	
12:0	TMRCV	Calculation formula: TMRCV = 250*HCLK, the unit of HCLK is MHz. to modify this register value, please refer to 7.7 . 4MHz example: TMRCV = 250*4 = 1000.

7.8.9 CR register (FLASH_CR)

Offset address: 0x20

Reset value: 0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
R																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								IE		BUSY		Res		WAIT		OP
R								RW		RO		RW		RW		

position	Marker	Function Description
31:7	Reserved	
6:5	IE	IE[6]: FLASH erase protected address interrupt enable;0: not enable;1: enable IE[5]: FLASH erase PC value interrupt enable;0: not enable;1: enable
4	BUSY	Idle/busy flag bit;0 : idle state; 1: busy state.
3	Reserved	
2	WAIT	Read FLASH wait cycle;0 :0 wait cycle; 1:1 wait cycle
1:0	OP	FLASH operation;00 : read;01 : program;10 : sector erase; 11: chip erase
For details on how to modify this register value, see 7.7 .		

7.8.10 IFR register (FLASH_IFR)

Offset address: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IF1 IF0							
R								RO RO							

position	Marker	Description
31:2	Reserved	
1	IF1	Erasure protection alarm interrupt flag bit
0	IF0	Erase PC address alarm interrupt flag bit

7.8.11 ICLR register (FLASH_ICLR)

Offset address: 0x28

Reset value: 0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ICLR1 ICLR0							
R								RW0 RW0							

position	Marker	Function Description
31:4	Reserved	
3:2	Reserved	Write is invalid, read is 0x3
1	ICLR1	Clear the protection alarm interrupt flag bit; write 0clear; write 1invalid.
0	ICLR0	Clear the PC address alarm interrupt flag bit; write 0clear; write 1invalid.

7.8.12 BYPASS register (FLASH_BYPASS)

Offset address: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYSEQ															
WO															

position	Marker	Description
31:16	Reserved	
15:0	BYSEQ	<p>Before modifying the registers of this module, you must write the 0x5a5a-0xa5a5 sequence to the BYSEQ[15:0] register. Each time you write to this</p> <p>After the Bypass sequence, the registers can be modified only once. If you need to modify the register again, you must enter the 0x5a5a-0xa5a5 sequence again. For details, see 7.7 .</p>

7.8.13 SLOCK register (FLASH_SLOCK)

Offset address: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK															
RW															

position	Marker	Description
31:16	Reserved	
15:0	SLOCK	<p>Sector erasure protection bit;0: no erasure allowed;1: erasure allowed</p> <p>SLOCK[0] corresponds to : Sector0-</p> <p>1-2-3 SLOCK[1] corresponds to :</p> <p>Sector4-5-6-7 SLOCK[2]</p> <p>corresponds to : Sector8-9-10-11</p> <p>SLOCK[3] corresponds to :</p> <p>Sector12-13-14-15</p> <p>.....</p> <p>SLOCK[15] Corresponding to: Sector60-61-62-63</p>

8 RAM Controller (RAM)

8.1 Overview

This product contains a static SRAM with a capacity of 24K/K byte, supporting by 8 bits~~16bits~~

word (32bits) Three types of read/write operations. The read/write operation can be performed at HCLK frequency without waiting cycle. In addition, the controller supports parity check, which can parity check each byte of SRAM data and generate a parity error interrupt.

8.2 Function Description

Read and write bit width

This controller supports ~~Bt(8bits) Hwd(16bits) and Wrd(32bits)~~ read and write operations in three bit widths.

The address of a Byte operation must be aligned by Byte, and the target address of a Half-word operation must be aligned by Half-word

(The address of the Word operation must be aligned by Word (the lowest two bits of the address are 00) If the target address of the read/write operation is not aligned according to the bit-width specification, the operation is invalid and the system generates a Hard Fault error interrupt.

Parity Check

This controller supports parity check of SRAM data. After reset, this function is turned off by default. When the parity function is turned on, when writing data to SRAM, parity check is done for each Byte of data, and the 1bit check value is stored in SRAM along with 8bits of data. When reading data to SRAM, the controller will read 8bits data and 1bit check value and do parity check, if the check is wrong, the parity check error flag bit will be set, and in the case of interrupt enable, an error interrupt will be generated.

Caution.

-
- When parity is enabled, the SRAM must be initialized before reading SRAM data, otherwise the parity alarm flag bit or interrupt may be triggered by mistake.

8.3 Register

Base address: 0x4002 0400

Register	Offset Address	Description
RAM_CR	0x00	Control register
RAM_ERRADDR	0x04	Error Address Register
RAM_IFR	0x08	Error interrupt flag register
RAM_ICLR	0x0C	Error interrupt flag clear register

8.3.1 Control register (RAM_CR)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													IE	CHK EN	
R													RW	RW	

position	Marker	Function Description
31:2	RESERVED	
1	IE	error alarm interrupt enable signal;1 : enable alarm interrupt,0 : disable alarm interrupt.
0	CHKEN	Parity check enable signal;1: enables parity check,0: disables parity check.

8.3.2 Parity error address register (**RAM_ERRADDR**)

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ERRADDR											
R				RO											

position	Marker	Function Description
31:12	Reserved	
11:0	ERRADDR	12bits Parity error byte address

8.3.3 Error Interrupt Flag Register (**RAM_IFR**)

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ERR	
R														RO	

position	Marker	Function Description
31:1	RESERVED	
0	ERR	Parity error flag bit

8.3.4 Error Interrupt Flag Clear Register (**RAM_ICLR**)

Offset address: 0x0C

Reset value: 0x0000 0001

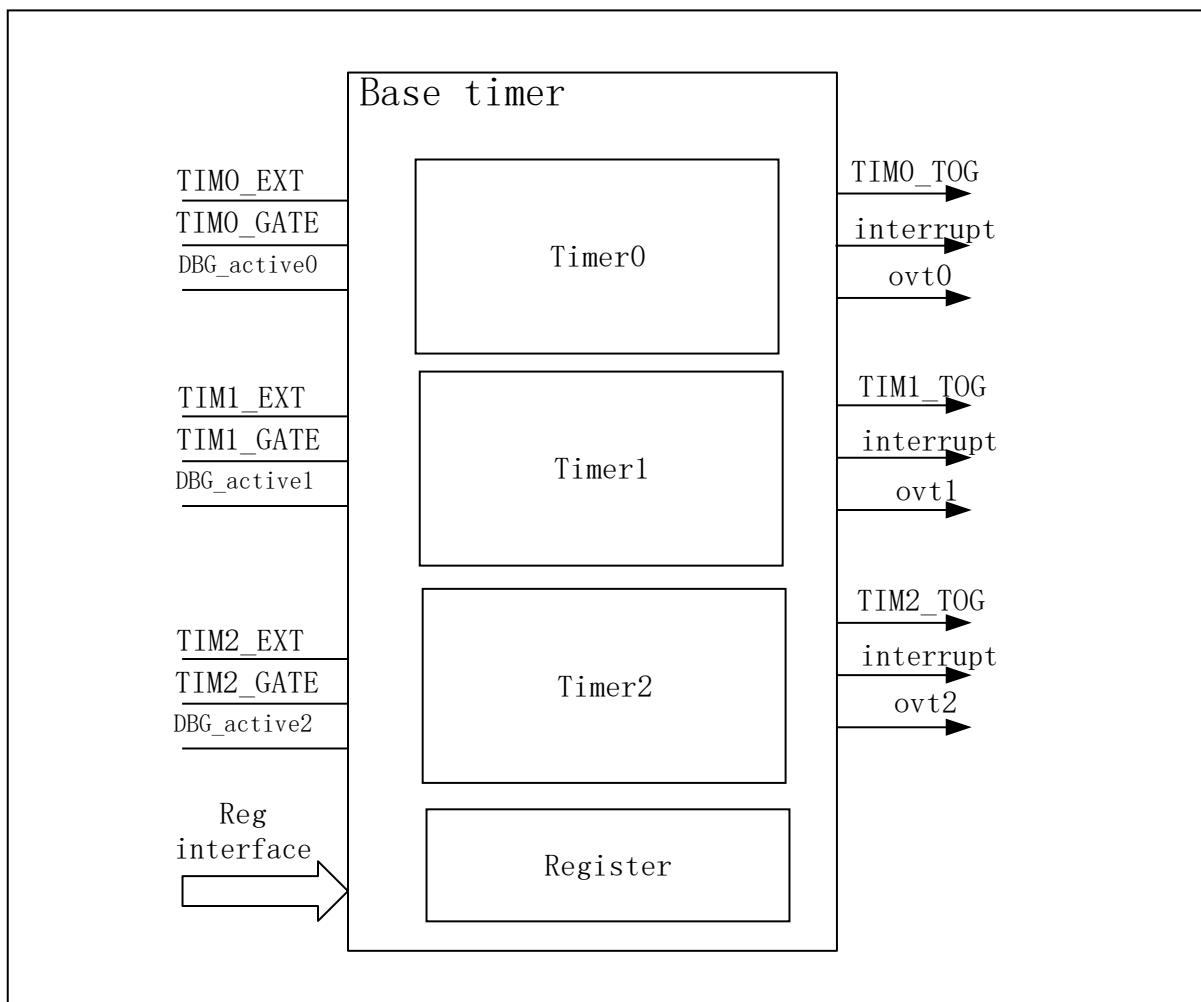
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
R															

position	Marker	Function Description	
31:1	Reserved		
0	ERRCLR	Error interrupt flag clear bit; writel: invalid, write0: clear	RW0

9 Basic timer (TIM0/1/2)

9.1 Basic Timer Introduction

The basic timer contains three timers Timer0/1/2. Timer0/1/2 are identical in function. timer0/1/2 are synchronous timers/counters that can be used as timers/counters with bit 16auto reload function or as timers/counters with bit



32no reload function. timer0/1/2 can count external pulses or implement system timing.

Figure 9-1Base Timer Block Diagram

9.2 Base Timer Function Description

Timer0/1/2 Each timer/counter has independent control start signal, and external input clock, gating signal. timer0/1/2 use EXT, GATE for counting function, EXT for external input clock signal of counter, GATE for valid level counting enable signal. When the gating function is enabled, the counter counts when and only when the external input GATE level is valid, otherwise the counter is in the hold state. The gating enable is controlled using CR.gate. The default gating function is off. GATE_P control is used for gating level selection. The default high level is the active gating level; when set to 1, the low gating level is active.

TIM0/1/2 uses PCLK, GATE for the timing function, PCLK for the internal input clock signal of the timer and GATE can be used for the valid level timing enable signal. When the gating function is enabled, the timer will count when and only when the external input GATE level is valid, otherwise the timer is in the timer counter stop state. Gate enable is controlled using CR. The default gating function is off. Gate level selection is controlled using CR. The default high

PRS[2:0]	the 000 active 001 gating 010 level; 0 when s100 to th101 level 1, 0 the adtive gating
Levelsdseelow.1 The timing function can be configured 32 pre-64 day the 256 frequency. Ratio	

The TIM0/1/2 timer/counter supports two modes of operation, by setting the MD in the Timer Control Register (CR)

Selects the operating mode. Mode is 1Bit 32free count mode. Mode Yes Bit 216reload mode.

32 Bit free count mode, count to maximum 0xFFFFFFFF overflow and generate interrupt, the timer/counter becomes 0X00000000; then continue counting; 16Bit reload mode, count to maximum 0xFFFF overflow and generate interrupt, the timer/counter value is loaded to the value of ARR, then continue counting up.

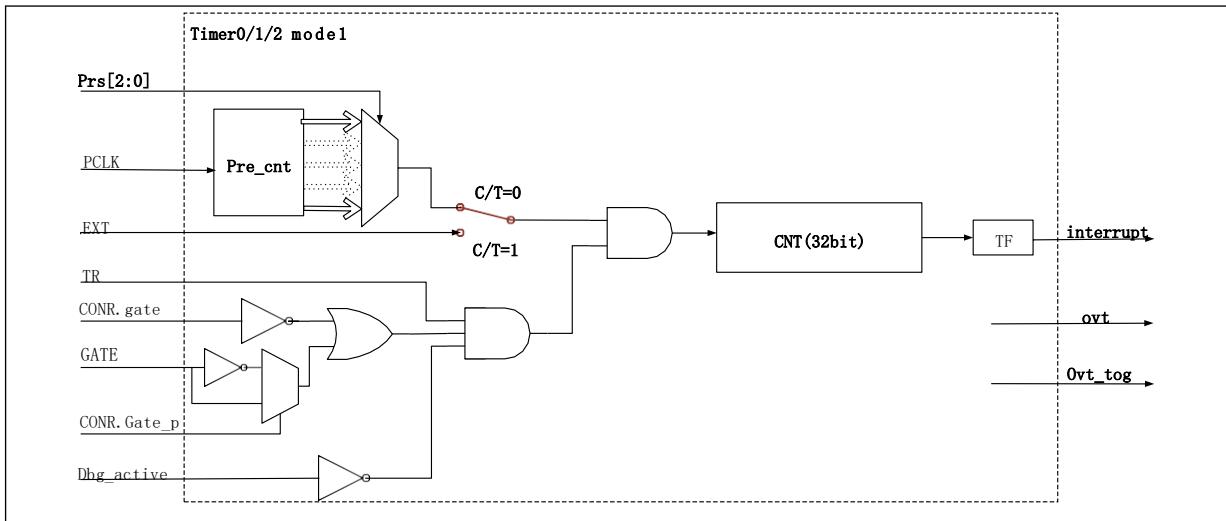


Figure 9-2 Timer Mode 1 Block Diagram

In mode 2 reload mode, if the timing time is set small, the software processing speed should be considered, otherwise the interrupt will not be processed in time and cause interrupt loss.

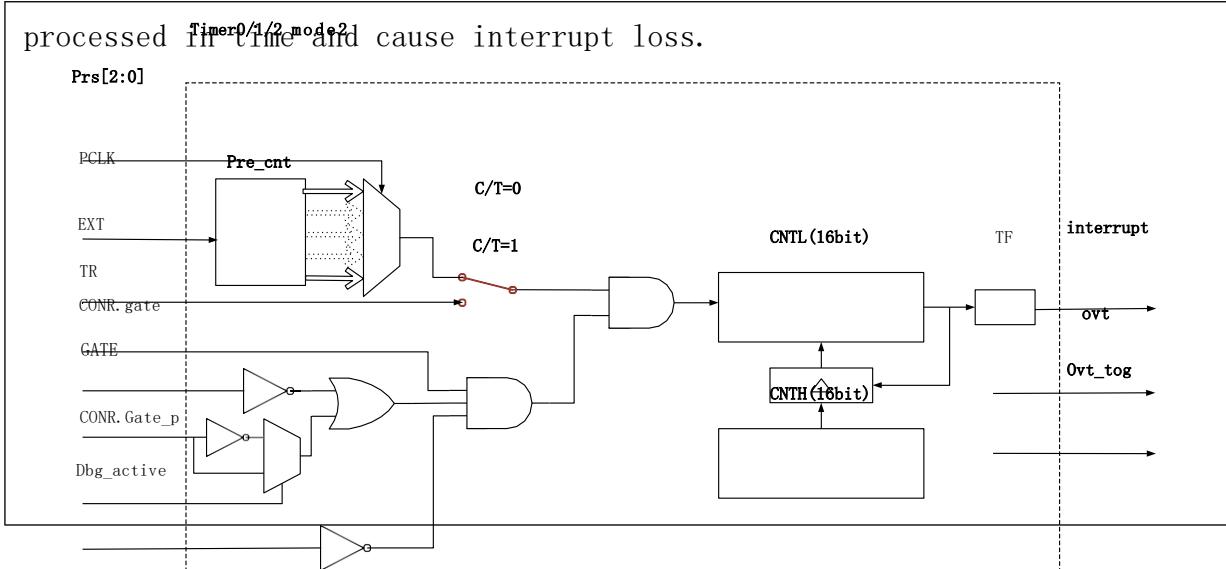


Figure 9-3 Timer Mode 2 Block Diagram

When the corresponding timer TR is set, the timer starts to run. The mode is 132-bit timer/counter, which starts counting from the initial value T set in the register after starting, and generates an overflow interrupt after counting to the maximum. Then the timer continues to count from 0. The mode is 2Bit 16Reload Timer/Counter, which starts counting from the initial value of CNT in the register after starting, counts up to the maximum 0xFFFF, generates an interrupt and then reloads the value of ARR in the register to the counter CNT, and continues counting up.

9.2.1 Counting function

The count function is used to determine the number of times an event has occurred. In the counting function, the counter is accumulated once on the falling edge of each corresponding input clock. The input signal is sampled

by Timer0/1/2 mode 1 ($\text{max}=0xFFFF FFFF$) so the external input clock frequency cannot exceed the system's Pclk clock. Counting to the maximum value will overflow and generate an interrupt. The interrupt flag needs to be cleared by software.

Count (32) TT+1T+2 T+3 T+4 ... max max 0 1 2 max 0 1 2 ...

012

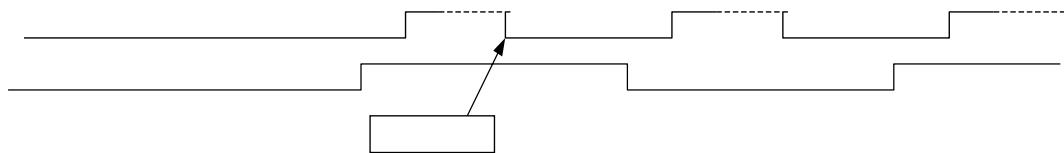
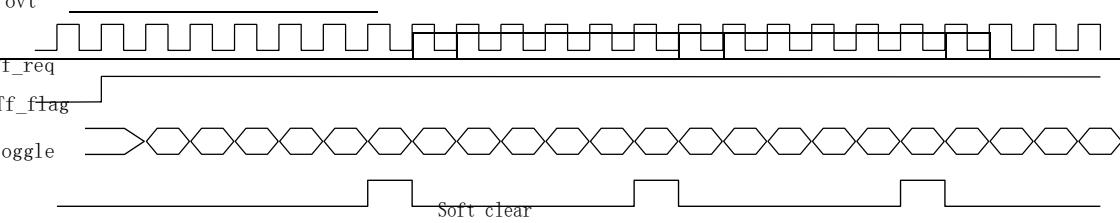


Figure 9-4 Mode Timing1 diagram

9.2.2 Timing function

The timer function is used to generate interval timing. In the timing function, the timer is pre-divided and the timer accumulates once at each

Timer0/1/2 mode 2 divided clock, counting to the maximum value and generating an ($\text{max}=0xFFFF$)

PCLK interrupt. The interrupt flag needs to be cleared by software.

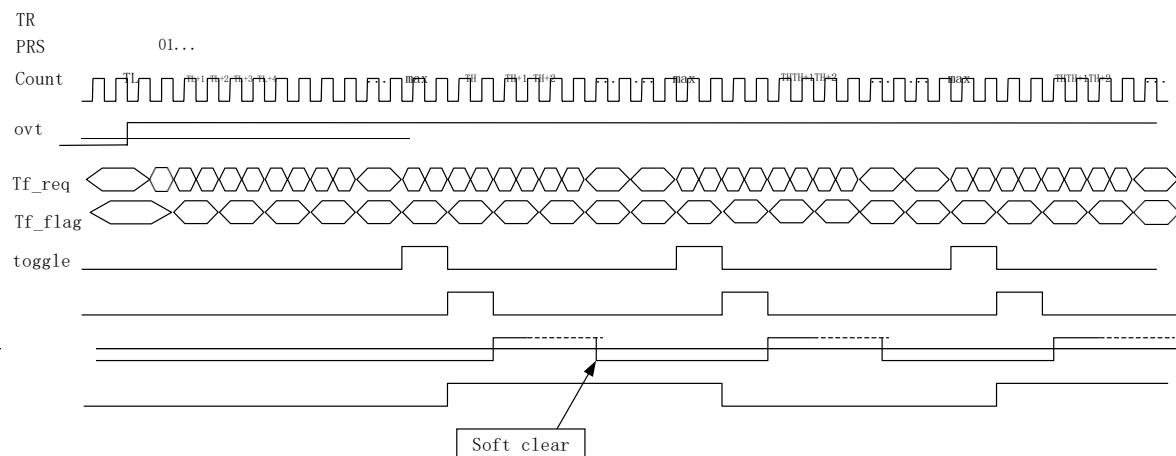


Figure 9-5 Mode 2 Timing Diagram (Prescaler set to 2)

9.2.3 Buzzer Features

When CR.TOG_EN is set, the TOG and TOGN outputs are reversed. TOG_EN is set to set the TOG and TOGN outputs to 0 at the same time. The timer reload mode of Buzzer output at different frequencies with 4M count clock is configured as

Buzzers. Frequency	Counter cycle	Counter count value	Counter reload value	CNT Initial Value	ARR Reload Value
1000Hz	0.5ms	2000	63536	0xF830	0xF830
2000Hz	0.25ms	1000	64536	0xFC18	0xFC18
4000Hz	0.125ms	500	65036	0xFE0C	0xFE0C

9.3 Base Timer Interconnection

9.3.1 GATE Interconnection

The GATE input can be fed directly from the port or into the RX signal of the UART; it can also be configured as VC

The GATE of Timer0/1/2 can be configured.

The internal interconnection configuration enables automatic recognition of the UART baud rate, measurement of the pulse width of the VC comparison output, and external control of the count.

Configuration selection RX input is controlled in the GPIO_CTRL register and VC control is controlled in the VC control register. For port selection, only one of the UART input selection and VC input selection can be valid as a gated input.

9.3.2 Toggle Output Interconnect

The flip output of TIM0, tog0, goes to the internal module UART0 to control the baud rate of UART0; the flip output of TIM1, tog1, goes to the internal module UART1 to control the baud rate of UART1; the flip output of TIM0/1/2 also goes to the port, which can drive the Buzzer to implement the buzzer control.

9.4 Base Timer Register Description

x=0,1,2;

Base Timer Base Address 0X40000C00

Timer	Offset Address	Description
TIM0	0x00	TIM0 Offset Address
TIM1	0x20	TIM1 Offset Address
TIM2	0x40	TIM2 Offset Address

Register	Offset Address	Description
TIMx_ARR	0X000	TIM0/1/2 Reload Register
TIMx_CNT	0X004	TIM0/1/2 16-bit mode counter register
TIMx_CNT32	0X008	TIM0/1/2 32-bit mode counter register
TIMx_CR	0X00C	TIM0/1/2 control register
TIMx_IFR	0X010	TIM0/1/2 interrupt flag
TIMx_ICLR	0X014	TIM0/1/2 Interrupt Clear Register

Table 9-1BaseTimer Register List

9.4.1 16 Bit Mode Reload Register (TIMx ARR)

Offset address: 0x000

Reset value: 0x0000 0000

position	Symbols	Description
31:16	Reserved	Reserved bits, read as0
15:0	ARR	Timer mode2 Reload register

9.4.2 16 Bit Mode Count Register (TIMx_CNT)

Offset address: 0x004

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
R/W															

position	Symbols	Description
31:16	Reserved	Reserved bits, read as 0
15:0	CNT	Timer mode 2Count value register

9.4.3 32 Bit Mode Count Register (TIMx_CNT32)

Offset address: 0x008

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT32 [31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT32 [15:0]															
R/W															

position	Symbols	Description
31:0	CNT32	Timer mode 1Count value register

9.4.4 Control register (TIMx_CR)

Offset address:

0x00C Reset value:

0x0000 0008

	31:11	10	9	8	7	6:4	3	2	1	0
Reserved	IE	GATE_P	GATE	Reserved	PRS	TOG_EN	CT	MD	TR	
	RW	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/w

position	Symbols	Description
31:11	Reserved	Reserved bits, read as0
10	IE	Interrupt enable control, writel and enable interrupt
9	GATE_P	Port GATE polarity control, default high gate active, set to1 active low
8	GATE	Timer Gate Control 0: no gating, timer operates when TR=1. 1: only works when port GATE is active and TR=1.
7	Reserved	Reserved bits, read as0
6:4	PRS	TIM pre-definition selection. 000:1; 001:2; 010:4; 011:8; 100:16; 101:32; 110:64. 111:256.
3	TOG_EN	TOG Output Enable 0: TOG and TOGN output simultaneously 1: TOG, TOGN output signals in opposite phase. Available for buzzer use.
2	CT	Counter/Timer Function Selection 0: Timer function, the timer is counted by PCLK. 1: Counter function, the counter is counted by the falling edge of the external input. The external input is sampled by PCLK and the external input clock frequency should be lower than 1/2 the sample clock.
1	MD	Timer operating mode 0Mode: 132Bit Counter/Timer 1: Mode 2auto reload 16bit counter/timer
0	TR	Timer operation control 0: Timer stop 1: Timer operation

9.4.5 Interrupt Flag Register (TIMx_IFR)

Offset address: 0x010

Reset value: 0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved								TF

position	Symbols	Description
31:1	REV	Reserved bits, read as0
0	TF	Interrupt flag, hardware set. Write clear register to clear zero

9.4.6 Interrupt Flag Clear Register (TIMx_ICLR)

Offset address: 0x014

Reset value: 0x0000 0001

31:8	7	6	5	4	3	2	1	0
Reserved								TFC

position	Symbols	Description
31:1	Reserved	Reserved bits, read as0
0	TFC	Interrupt flag clear, write 0clear, write 1invalid

10 Low Power Timer (LPTIM)

10.1 LPTimer Introduction

LPTimer is an asynchronous bit16 timer/counter that can be timed/countered by internal low-speed RC or external low-speed crystal oscillation even after the system clock is turned off. The system can be woken up in low-power mode via interrupt.

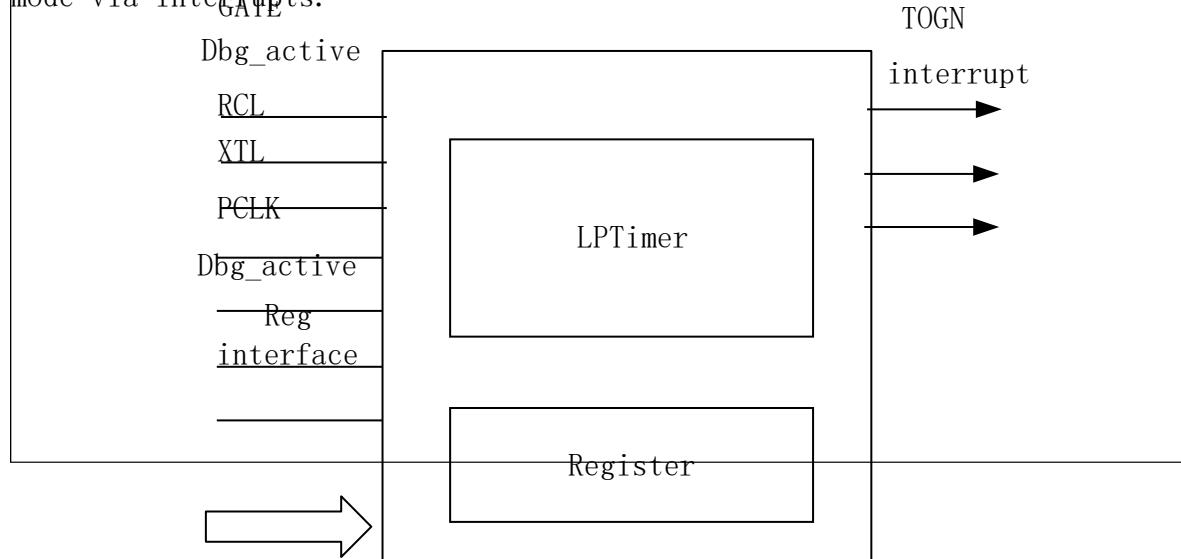


Figure 10-1 LPTimer block diagram

10.2 LPTimer Function Description

The LPTimer supports several operating modes, each timer/counter has an independent control start signal, as well as an external input clock, and a gating signal.

LPTimer uses EXTGATE for the counting function, and EXT for the external input clock signal of the counter.

Gate is used for valid level count enable signal.

The LPTimer timer supports two operating modes, which are selected by setting the MD in the Timer Control Register (CR). Mode is 1Bit 16free count mode. Mode is Bit 216reload mode.

The LPTimer automatically loads the value of the reload register ARR into the counter when it starts.

LPTimer can select three clocks as timer clocks, which are selected by control register CR.TCK_SEL. The default selection is PCLK. the clock

TCK_SEL	00	01	10	11
Timer Clock	PCLK	PCLK	XTL	RCL
Read timer count value	Read through synchronization	No synchronization	Read through synchronization	Read through synchronization

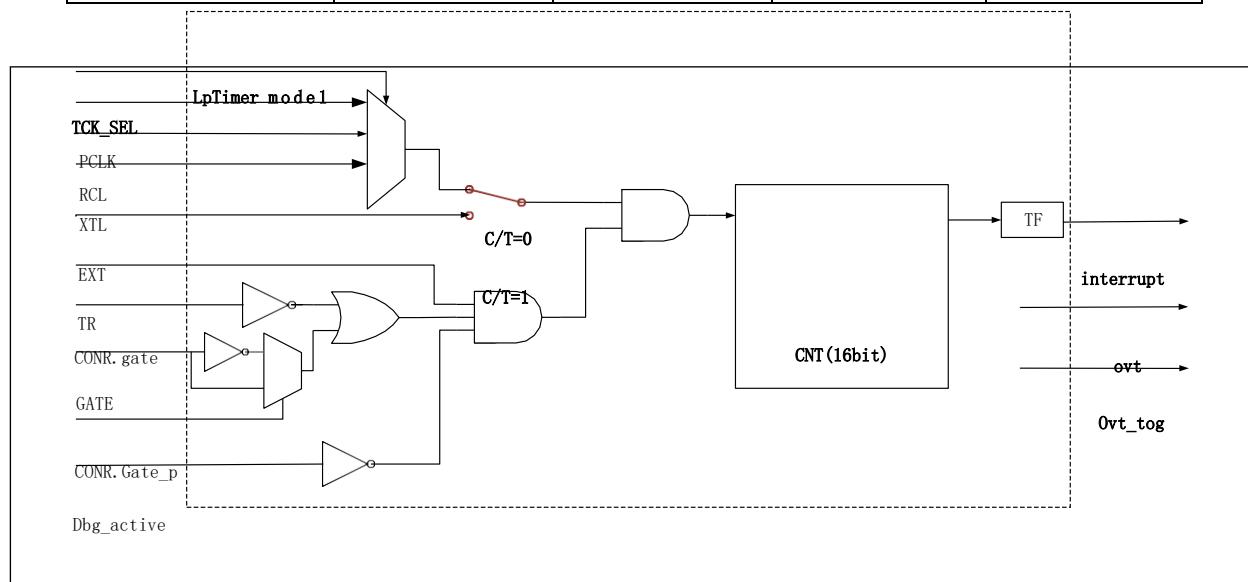


Figure 10-2 LPTimer mode 1

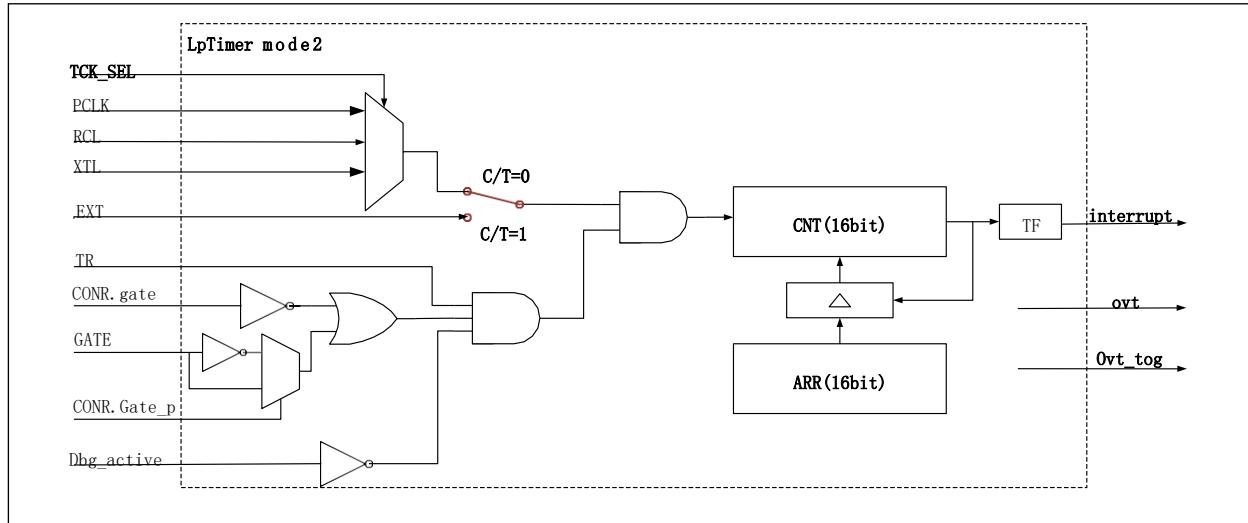


Figure 10-3 LPTimer mode 2

When the corresponding timer TR is set, the timer starts to run. The counter starts counting from the set value, and generates an overflow interrupt after counting to the maximum 0XFFFF. Mode After generating1 an interrupt, the counter 2is cleared to zero and continues to count upwards. The initial value of counter CNT is loaded automatically by ARR when the timer is started.

Since LPTimer is an asynchronous timer, timer interrupt is synchronized from Timer clock domain to pclk pre, the interrupt will be lost when the reload mode timer value is set higher than 0Xfffb, it is recommended to use the interrupt function if the reload register value is less than 0XFFFC. there is no such limitation without using interrupt.

10.2.1 Counting function

The count function is used to determine the number of times an event has occurred. In the counting function, the counter is accumulated once on the falling edge of each corresponding input clock. The input signal is sampled by the internal count clock, so the external input clock frequency cannot exceed the system's count clock. Counting to the maximum value will overflow and generate an interrupt.

10.2.2 Timing function

The timer function is used to generate interval timing. In the timing function, the timer accumulates once a clock and the count overflows to the maximum value and generates an interrupt.

10.3 LPTimer Interconnect

10.3.1 GATE Interconnection

The GATE input can be fed directly from the port or into the RX signal of the UART; it can also be configured as VC

The GATE of LPTIM can be configured as follows.

The internal interconnection configuration enables automatic recognition of the UART baud rate, measurement of the pulse width of the VC comparison output, and external control of the count.

The UART selection control register is in the port control register GPIO_CTRL4, the VC output control register is in the VC control module.

10.3.2 EXT Interconnection

The EXT input can be fed directly from the port or can be configured as an input to the VC as an EXT signal. the LPTIM

EXT can be configured as follows.

The VC output control register is in the VC control module.

10.3.3 Toggle Output Interconnect

The flip output of the LPTimer to the port can drive the Buzzer to implement the buzzer control.

10.4 LPTimer Register Description

Base address 0X40000C00

Register	Offset Address	Description
CNT	0X060	LPTimer count value read-only register
ARR	0X064	LPTimer reload register
CR	0X06C	LPTimer control register
IFR	0X070	LPTimer interrupt flag
ICLR	0X074	LPTimer interrupt clear register

Table 10-1 List of LPTimer Registers

10.4.1 Counter Count Value Register (LPTIM_CNT)

Offset address: 0x060

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
RO															

position	Symbols	Description
31:16	Reserved	Reserved bits, read as0
15:0	CNT	Low-power timer count value read-only register. When the timer is started, the initial value of CNT is automatically loaded by ARR. Load.

10.4.2 Reload Register (LPTIM_ARR)

Offset address: 0x064

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
R/W															
position	Symbols	Description													
31:16	Reserved	Reserved bits, read as0													
15:0	ARR	Low Power Timer Reload Register WT_FLAG before writing ARR, and the data can be written when and if WT_FLAG is1. WT2_FLAG will become low after writing ARR register.													

10.4.3 Control register (LPTIM_CR)

Offset address:

0x06C Reset value:

31:11	0x0000 0008	9	8	7	6	4:5	3	2	1	0
Reserved	IE	GATE_P	GATE	WT_FLAG	Reserved	TCK_SEL	TOG_EN	CT	MD	TR
	RW	R/W	R/W	RO		R/W	R/W	R/W	R/W	R/W

position	Symbols	Description
31:11	Reserved	Reserved bits, read as0
10	IE	Interrupt enable control, writel and enable interrupt
9	GATE_P	GATE polarity control, default high level gate active, set to1 low level active
8	GATE	Timer Gate Control 0: no gating, timer operates when TR=1. 1: operates only when gated 1and TR=1.
7	WT_FLAG	WT,Write sync flag 0: In sync, write ARR is invalid at this time 1: Synchronization is complete, at this point you can change the ARR
6	Reserved	Reserved bits, read as0
5:4	TCK_SEL	LPTimer clock selection 00PCLK; 10:XTL; ,11RCL
3	TOG_EN	TOG Output Enable 0: TOG and TOGN output simultaneously 1: TOG, TOGN output signals in opposite phase. Available for buzzer use.
2	CT	Counter/Timer Function Selection 0: Timer function, the timer uses the clock selected by TCK_SEL for counting. 1: Counter function, the counter uses the falling edge of the external input for counting. The sampling clock uses the clock selected by TCK_SEL, and the external input clock is lower than 1/2 the sampling clock.
1	MD	Timer operating mode 0: Mode 1without overload mode 16bit counter/timer 1: Mode 2auto reload 16bit counter/timer
0	TR	Timer run control bit 0: Timer stop 1: Timer operation

10.4.4 Interrupt Flag Register (LPTIM_IFR)

Offset address: 0x070

Reset value: 0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved								TF

position	Symbols	Description
31:1	Reserved	Reserved bits, read as0
0	TF	Interrupt flag, hardware set. Write clear register to clear zero

10.4.5 Interrupt Flag Clear Register (LPTIM_ICLR)

Offset address: 0x074

Reset value: 0x0000 0001

31:8	7	6	5	4	3	2	1	0
Reserved								TFC

position	Symbols	Description
31:1	Reserved	Reserved bits, read as0
0	TFC	Interrupt flag clear, write 0clear, write 1invalid

11 Programmable Counting Array (PCA)

11.1 PCA Introduction

The PCA (Programmable Counter Array) supports capture/compare modules of up to one 5bit16. Each module of the PCA can be programmed independently to provide input capture, output comparison, or pulse width modulation. Additional modules are available with an additional watchdog timer mode.

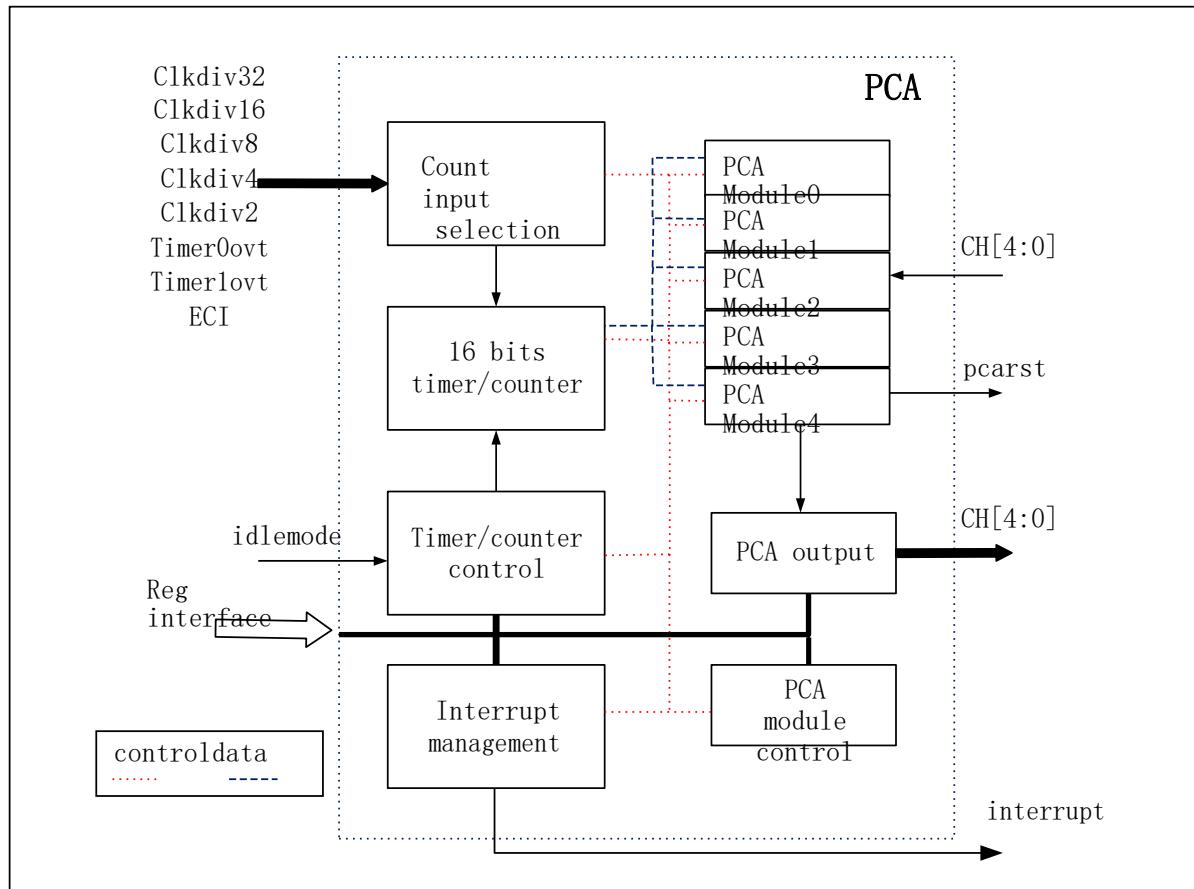


Figure 11-1 PCA overall block diagram

11.2 PCA Functional Description

Each module can be configured to operate independently in three ways: edge triggered capture, output comparison, and 8bit pulse width modulation. Each module has its own function registers in the system controller, which are used to configure how the module works and to exchange data with the module.

Each compare/capture module consists of a compare/capture register set (CCAPx), as well as a 1bit 16comparator, and various logic gate controls.

The register set is used to store the time or number of times for externally triggered capture conditions, or internally triggered comparison conditions. In PWM mode, the register (CCAPxL) is used to control the duty cycle of the output waveform. Each module can be independently programmed to operate in any of the following modes.

- 16 Bit capture mode is triggered on rising, falling or arbitrary edges.
- Comparison modes:16 bit software timer, 16 bit high speed output, 16 bit watchdog timer (module4) or bit 8pulse width modulation.
- Not started.

The Compare/Capture Module Mode Register (CCAPMx) determines the corresponding operating mode. For programming the compare/capture modules, they are based on a common time counter. The timer/counter on and off is controlled by the CCON.CR bit for PCA timer/counter operation. In a compare/capture module capture, the software timer, high speed output, sets the module's compare/capture flag (CCON.CCFx)and generates a PCA interrupt request if the corresponding enable bit is set in the CCAPMx register. the CPU can read and write to the CCAPx register at any time.

11.2.1 PCA Timer/Counter

This set of special function registers of CNT can be used as a bit 16timer/counter. This is a 16bit-up counter. If the CMOD.CFIE bit is set to 1, the counter will generate an interrupt when it reaches the value of the CNT.L register. The counter can be cleared by writing a 1 to the CCON.CCR bit.

"1", the hardware automatically sets the PCA overflow flag (CCON.CF) when the CNT overflows and generates a PCA interrupt request. CMOD.CPS[2:0] three bits selects eight signals to be input to the timer/counter.

- System PCLK The Clock crossover32 frequency.
- System PCLK The Clock crossover16 frequency.
- System PCLK The Clock crossover8 frequency.
- System PCLK The Clock crossover4 frequency.
- System PCLK The Clock crossover2 frequency.

- Timer 0overflow. The CNT is incremented each time the timer 0count overflows, thus providing a variable programming frequency input to the PCA.
- Timer 1overflow. The CNT is incremented each time the timer 1count overflows, thus providing a variable programming frequency input to the PCA.
- ECI. The CPU samples the PCA ECI every 4 PCLK clock cycles, and the CL is automatically added1 as each sample result changes from high to low, so the maximum ECI input frequency cannot be higher than 1/8 of the system clock PCLK to meet the sampling requirements.

Set the run controller (CCON.CR) to start the PCA timer/counter. When CMOD.CIDL is set to 1", the PCA timer/counter can continue to run in idle mode. the CPU can read the CNT value at any time, but when the count is started (CCON.CR=1), the CNT is disabled from writing in order to prevent counting errors.

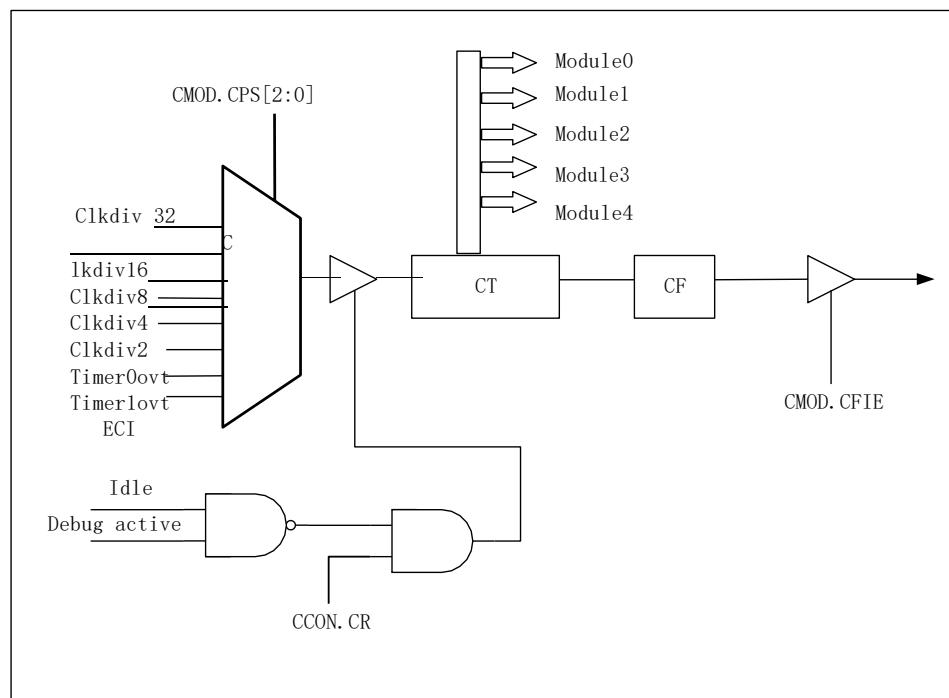


Figure 11-2PCAcounter block diagram

11.2.2 PCA capture function

The PCA capture mode provides the 5PCA with the ability to measure pulse period, pulse width, duty cycle and phase difference. A level jump on the pin causes the PCA to capture the PCA counter/timer value and load it into the corresponding module's bit16 capture/comparison register (CCAPx). negative edge) When a capture occurs, the capture/compare flag (CCFn) in CCON is set to a logic '1' and an interrupt request is generated (if the CCF interrupt is allowed) The CCFn bit is not automatically cleared by hardware when the CPU goes to the interrupt service routine, and the user software writes the INTCL register to clear this flag bit. When simultaneous capture of upper and lower edges is required, the recommended procedure is to enable one edge capture first, then after a capture occurs, enable the other capture more in the interrupt service program, and turn off the previous capture interrupt. Switch in sequence.

The resolution is equal to the timer/counter clock. The input signal must be held high or low for at least one 2clock cycle to ensure that the input signal is recognized by the hardware.

The CPU can read or write to the CCAPx registers at any time. Capture settings.

- CCPMx.CAPP = "1" and CCAPMx.CAPN = "0" when capture on external rising edge is required
- CCPMx.CAPP = "0" and CCAPMx.CAPN = "1" when an external falling edge capture is required
- CCPMx.CAPP =1 "" and CCAPMx.CAPN = 1 "" when external rising and falling edge capture is required
- When the external rising and falling edges need to be captured at the same time and the captured edge needs to be known, first set CCPMx.CAPP =1 "", after the rising edge capture occurs, switch to the falling edge capture in the interrupt service subroutine, set CCAPMx.CAPN = "1 CAPN = "1" and clear CCPMx.CAPP = "0". After the next capture, set it

to rising edge capture again, and switch the captured edges in turn.

Caution.

- Subsequent captures by the same module will overwrite the existing captures. In order to keep the captured value, it is saved in RAM in the interrupt service program, which must be done before the next event, otherwise the previous captured sample value will be lost.

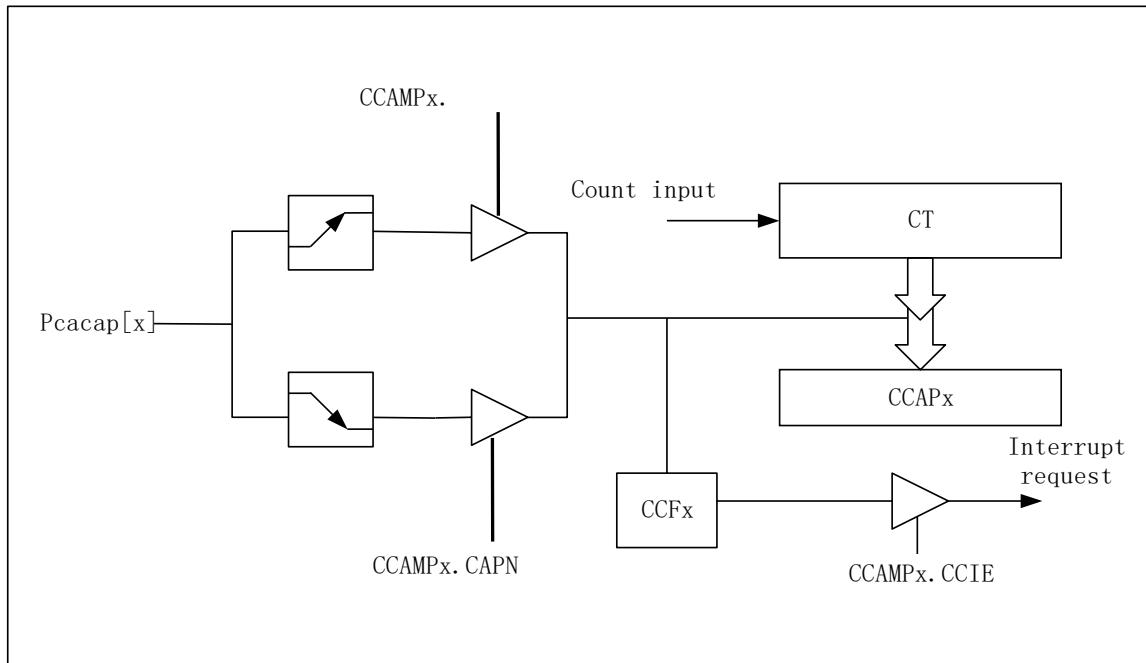


Figure 11-3PCACapture function block diagram

11.2.3 PCA comparison function

The comparison function provides five modules as follows: Timer, Event Counter, Pulse Width Modulation. Four modes employ the compare function: 16 bit software timer mode, high speed output mode, WDT mode, and PWM mode. In the first three functions, the compare/capture module compares the value of the bit 16PCA timer/counter with the bit 16value preloaded into the module's CCAPx register. In the PWM mode, the PCA module continuously compares the PCA timer

/The counter low byte register (CNT) is compared with a value in the CCAPxL module register bit8. Each

4 The module compares once every three clock cycles, i.e., matching the clock rate of the fastest PCA timer/counter. Set the CCAPMx.ECOM bit to select the comparison function for this module.

To properly use the module in comparison mode, follow these general procedures.

- Select the operating mode of the PCA module
- Selects the input signal for the PCA timer/counter.
- The comparison value is loaded into the module's compare/capture register pair.
- Set the PCA timer/counter run control bit.
- Generate an interrupt after matching and clear the module's compare/capture flag.

11.2.3.1 16 Bit software counting mode

To set a compare/capture module in bit 16software timer mode, the CCAPMx.ECOM and CCAPMx.MAT bits need to be set. Once a match has occurred between the PCA timer/counter and the compare/capture register (CCAPx), this will set the module's compare/capture flag (CCON.CCFx). This will generate an interrupt request if the corresponding interrupt enable bit (CCAPMx.CCIE) is set. Since the hardware does not clear the compare/capture

flag when the interrupt is processed, the user must clear the software flag.

In the interrupt service program, a new bit16 comparison value can be written to the comparison/capture register (CCAPx)

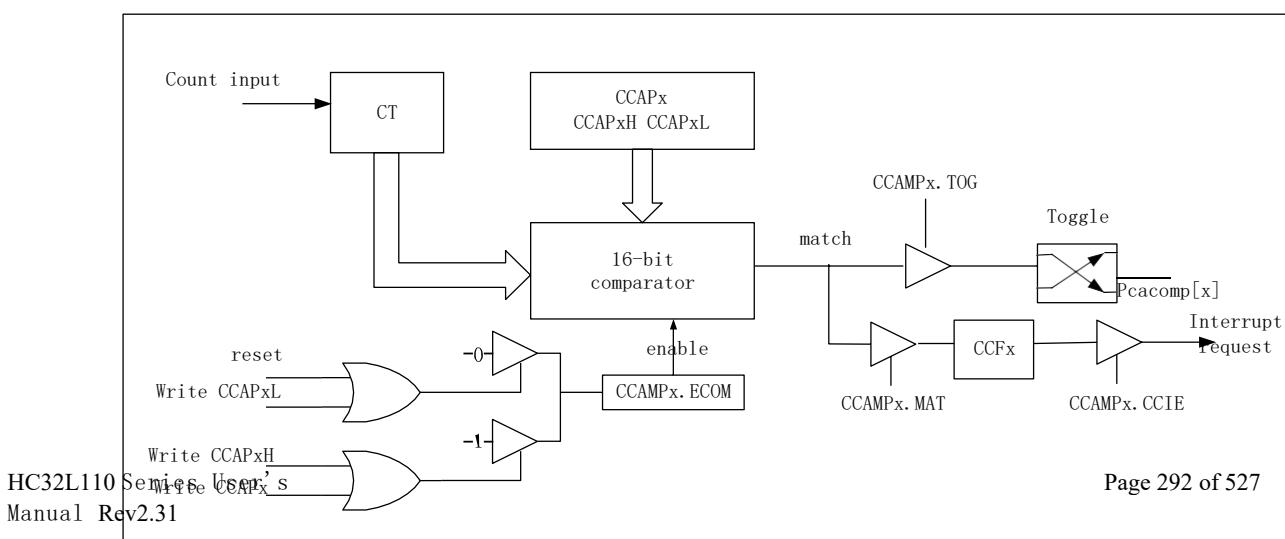
Note: When updating these registers, to prevent invalid matches from occurring, the user software should write CCAPxL first and CCAPxH second. once CCAPxL is written it will clear the disable compare function ECOMx bit, while writing CCAPxH will simultaneously set the ECOMx bit and re-enable the compare function. That is, when writing a bit16 value to the Capture/Compare register of PCA0, the low byte should be written first.

11.2.3.2 High-speed output mode

In the high-speed output mode, the logic level on the CAP/CMP[x] pin of the module PCA will change whenever the value in the PCA counter matches the bit16 capture/compare register (CCAPx) of the module. This provides higher accuracy than switching IO outputs, because this high-speed output is not affected by interrupt response, and relies on the CPU to switch IO outputs, which lacks power consumption and accuracy.

To set the high-speed output mode of a compare/capture module, set the CCAPMx.ECOM, CCAPMx.MAT, and CCAPMx.TOG bits. The match between the PCA timer/counter and the compare/capture register (CCAPx) toggles the PCA's CAP/CMP[x] signal and sets the module's compare/capture flag (CCON.CCFx). By setting or clearing the CAP/CMP[x] signal of the PCA by software, the user can select the match switching signal from low to high or high to low. The user can also choose to generate an interrupt request by setting the corresponding interrupt enable bit (CCAPMx.CCIE) when a match occurs. Since the compare/capture flag interrupt cannot be cleared by hardware, the user must clear this flag bit in software. If the user does not change the compare/capture register in the interrupt program, the PCA and recounts the compare value, and the next flip occurs if the match occurs. In the interrupt service program, a new bit16 comparison value can be written to the compare/capture register (CCAPx).

Note: To prevent invalid matches while updating these registers, the user



software should write to CCAPxL first then CCAPxH. write to CCAPxL to clear the disable compare function ECOM bit, and write to CCAPxH to set the ECOM bit to re-enable the compare function.

Figure 11-4PCAComparison Function Block Diagram

11.2.3.3 4WDT function for PCA modules

The PCA's 16WDT reset signal is used as a stand-alone reset signal. It is combined with external reset (RST) hardware watchdog reset (WDTRST) and LVD low voltage reset, POR power-up and power-down reset. The user is free to use them in combination or separately. Module is a module 4with WDT mode unique PCA. When not set to WDT, it can be used independently in other modes.

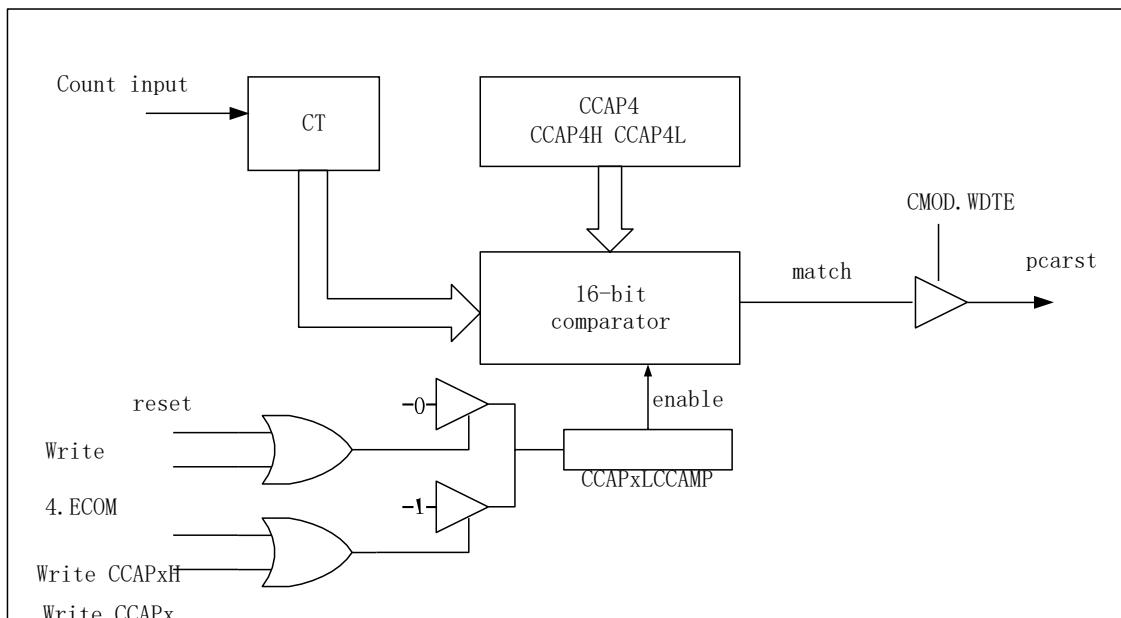


Figure 11-5 PCA WDT Function Block Diagram

ECOM4, CCAPM4.MAT4 and CMOD.WDTE must be set when using the PCA module as a 4WDT. In addition the PCA timer/counter can be set to CMOD.

Enter a one-bit16 comparison value in the Compare/Capture register (CCAP4)At the PCA Timer/Counter (CNT), an initial value 16of one bit or use the reset value (0x0000) These values are multiplied by the difference between the PCA input pulse rates to determine the WDT match run time. Set the Timer/Counter Run Control bit (CCON.CR) to start the PCA WDT. each time a match is made, the PCA WDT generates a reset signal. To prevent a PCA WDT from ~~reset~~ user has three options.

- Periodically the comparison value CCAP4 changes, so the match never happens.

-
- Change the PCA timer/counter value (CNT) periodically so matching never happens.

- Disable the module reset output signal by using the CMOD.WDTE bit that is cleared before matching and re-enabling it later. The first two options are more reliable because WDT is not disabled in the third option.

The second option is not recommended if the other PCA modules are in use, since the five modules share a common timebase. Therefore, the first option is the best in most applications.

PCA WDT configuration process.

1. Configure WDT compare/capture register PCA_CCAP4
2. Configure PCA Count Register PCA_CNT
3. Configure PCA_CCAMP4 to select the Compare Match function
4. Configure PCA_CMOD to select the input clock and enable the WDT function
5. Start PCA
6. Select Clear PCA WDT Clear to clear PCA WDT before PCA WDT reset

11.2.3.4 PCA bit pulse 8width modulation function

Pulse Width Modulation is a technique that uses a program to control the duty cycle, period, and phase of a waveform.⁵ Each PCA module can be used independently to generate a pulse width modulation (PWM) output at the CAP/CMP[x] pin of the corresponding PCA with a bit8 resolution pulse width. Use the module's capture/compare register CCAPxL to change the duty cycle of the PWM output signal. When the low byte (CNTL) of the PCA counter/timer is equal to the value in CCAPxL, the output on the CAP/CMP[x] pin of the PCA is set to "1"; when the count value in CNTL overflows, the CAP/CMP[x]

output of the PCA is reset to "0". When the low byte CNTL of the counter/timer overflows (from 0xFF to 0x00) the value saved in CCAPxH is automatically loaded into CCAPxL without software intervention.

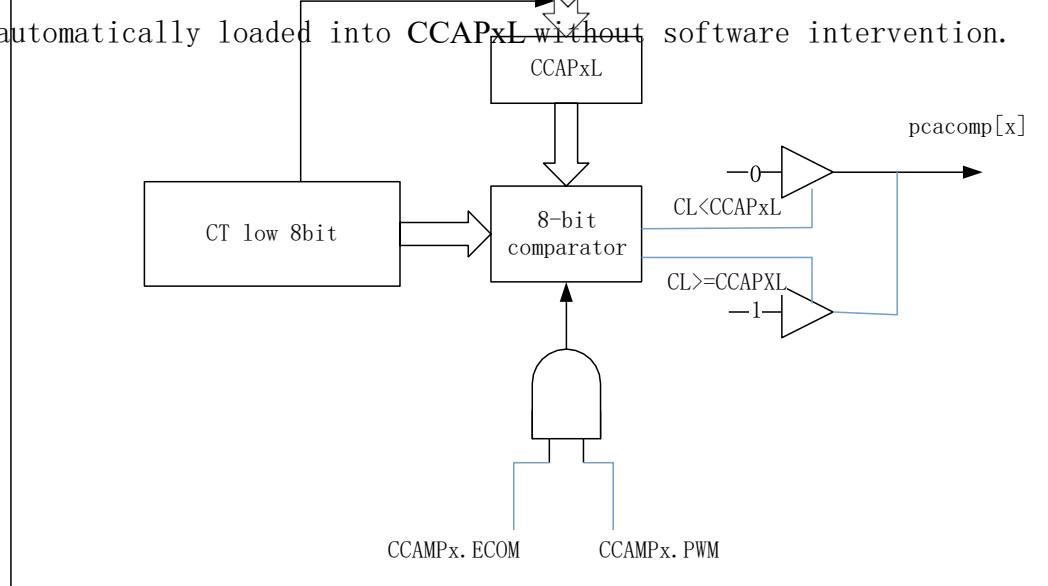


Figure 11-6 PCA PWM Function Block Diagram

In this mode, the value of the low byte (CNTL) of the PCA timer/counter is constantly compared with the value of the low byte (CCAPxL) of the compare/capture register. The output waveform is low when CNTL < CCAPxL and high when CNTL >= CCAPxL. When CNTL overflows, the system automatically loads the value of CCAPxH into CCAPxL to start a new

counting cycle.

The value of CCAPxL determines the duty cycle of the current cycle and the value of CCAPxH determines the duty cycle of the next cycle. The duty cycle of PWM can be changed by changing the value in CCAPxL. As shown in the figure below, adjusting the value of CCAPxL from 0 to 255 can achieve a duty cycle of ~0100.4%. To prevent burr, it is recommended not to change the value of CCAPxL directly.

The value of **CCAPxH** is automatically loaded by hardware to **CCAPxL** in the next cycle by changing the value of **CCAPxH**.

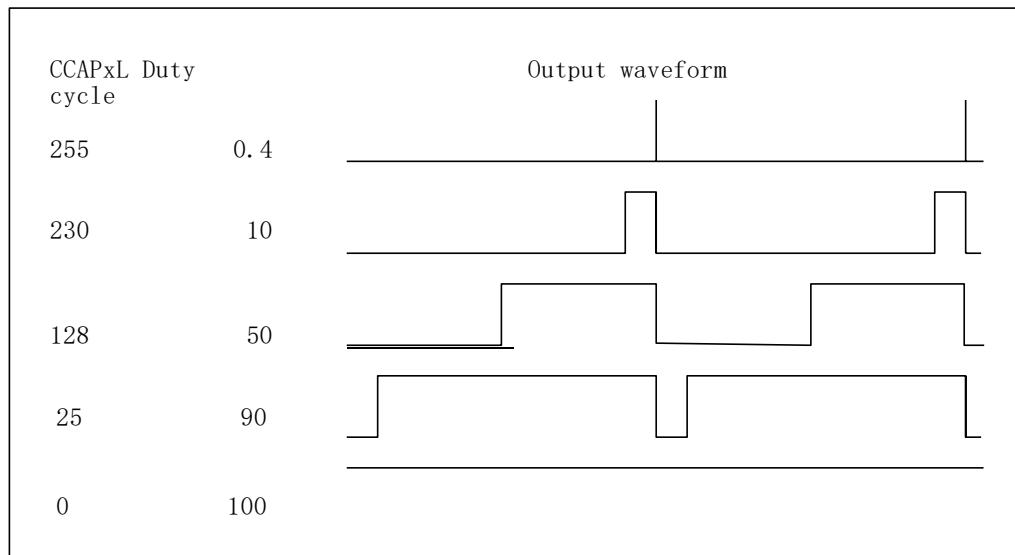


Figure 11-7 PCA PWM Output Waveform

To set a compare/capture module in PWM mode, set the CCAPMx.ECOM and CCAPMx.PWM bits. CSP[2:0] can be programmed to select the input count signal frequency. Enter a bit8 value in CCAPxL to specify the duty cycle of the first PWM waveform. Entering a value 8 of one bit in CCAPxH specifies the duty cycle of the second PWM waveform. Set the timer/counter run control bit (CCON.CR) to start the PCA timer/counter.

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Working method
X	1	0	0	0	0	X	Capture with positive edge trigger
X	0	1	0	0	0	X	Capture with negative edge trigger
X	1	1	0	0	0	X	Trigger capture with jump edge
1	0	0	1	0	0	X	Software Timer
1	0	0	1	1	0	X	High speed output
1	0	0	0	0	1	X	8Bit Pulse Width Modulator

Table 11-1PCAComparison Capture Function Module Settings

11.3 Interconnection and control of PCA modules with other modules

11.3.1 ECI Interconnection

The ECI input can be a different input port selected externally via IO MUX, or it can be a filtered output for internal VC comparison. the VC output control register is in the VC control module.

11.3.2 PCACAP0

The 0capture input of a channel can be.

- External IO MUX input port
- MUX input for RX of external UART
- Internal VC1's comparison filtered output

UART selection control register is in the port control register GPIO_CTRL2, VC output control register is in the VC control module.

11.3.3 PCACAP1

The 1capture input of a channel can be.

- External IO MUX input port
- MUX input for RX of external UART
- Internal VC2's comparison filtered output

UART selection control register is in the port control register GPIO_CTRL2, VC output control register is in the VC control module.

11.3.4 PCACAP [4:2]

The 2,3,4capture input of a channel can be.

- External IO MUX input port
- MUX input for RX of external UART

The UART selection control register is in the Port Control Register GPIO_CTRL2.

11.4 PCA Register Description

Base address 0X40001000

Register	Offset Address	Description
CCON	0X000	PCA control register
CMOD	0X004	PCA mode register
CNT	0X008	PCA Counting Register
ICLR	0X00C	PCA interrupt clear register
CCAPM0	0x010	PCA compare/capture module 0mode register
CCAPM1	0x014	PCA compare/capture module 1mode register
CCAPM2	0x018	PCA compare/capture module 2mode register
CCAPM3	0x01C	PCA compare/capture module 3mode register
CCAPM4	0x020	PCA compare/capture module 4mode register
CCAP0H	0X024	PCA compare/capture module 0high8 register
CCAP0L	0X028	PCA compare/capture module 0low8 register
CCAP1H	0X02C	PCA compare/capture module 1high8 register
CCAP1L	0X030	PCA compare/capture module 1low8 register
CCAP2H	0X034	PCA compare/capture module 2high8 register
CCAP2L	0X038	PCA compare/capture module 2low8 register
CCAP3H	0X03C	PCA compare/capture module 3high8 register
CCAP3L	0X040	PCA compare/capture module 3low8 register
CCAP4H	0X044	PCA compare/capture module 4high8 register
CCAP4L	0X048	PCA compare/capture module 4low8 register
CCAPO	0X04C	PCA PWM with high-speed output flag register
CCAP0	0X050	PCA compare/capture module 016bit register
CCAP1	0X054	PCA compare/capture module 116bit register
CCAP2	0X058	PCA compare/capture module 216bit register
CCAP3	0X05C	PCA compare/capture module 316bit register
CCAP4	0X060	PCA compare/capture module 416bit register

Table 11-2PCARegister List

11.4.1 Control register (PCA_CCON)

Offset address: 0x000

Reset value: 0x0000 0000

	31:8	7	6	5	4	3	2	1	0
Reserved		CF	CR	Reserved	CCF4	CCF3	CCF2	CCF1	CCF0
		RO	R/W		RO	RO	RO	RO	RO

position	Symbols	Description
31:8	Reserved	Reserved Bits
7	CF	<p>PCA counter overflow flag (write invalid)</p> <p>When the PCA count overflows, CF is set by hardware, and if the CFIE bit1 of the CMOD register is 1, the CF flag can generate an interrupt1 : counter overflow occurred; 0: no overflow.</p>
6	CR	<p>PCA counter operation control bit</p> <p>1: Start PCA counter counting</p> <p>0: Turn off the PCA counter count</p>
5	Reserved	Reserved Bits
4	CCF4	<p>PCA Counter Module 4Compare/Capture Flag Bits</p> <p>This bit is set by hardware when a match or capture occurs. (write invalid)</p> <p>When CCAPM4.CCIE is set, this flag bit generates a PCA interrupt</p>
3	CCF3	<p>PCA Counter Module 3Compare/Capture Flag Bits</p> <p>This bit is set by hardware when a match or capture occurs. (write invalid)</p> <p>When CCAPM3.CCIE is set, this flag bit generates a PCA interrupt</p>
2	CCF2	<p>PCA Counter Module 2Compare/Capture Flag Bits</p> <p>This bit is set by hardware when a match or capture occurs. (write invalid)</p> <p>When CCAPM2.CCIE is set, this flag bit generates a PCA interrupt</p>
1	CCF1	<p>PCA Counter Module 1Compare/Capture Flag Bits</p> <p>This bit is set by hardware when a match or capture occurs. (write invalid)</p> <p>When CCAPM1.CCIE is set, this flag bit generates a PCA interrupt</p>
0	CCF0	<p>PCA Counter Module 0Compare/Capture Flag Bits</p> <p>This bit is set by hardware when a match or capture occurs. (write invalid)</p> <p>When CCAPM0.CCIE is set, this flag bit generates a PCA interrupt</p>

11.4.2 Mode register (PCA_CMOD)

Offset address: 0x004

Reset value: 0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved	CIDL	WDTE	Reserved		CPS		CFIE	
	R/W	R/W			R/W		R/W	

position	Symbols	Description
31:8	Reserved	Reserved Bits
7	CIDL	Whether the PCA stops working in idle mode IDLE 1: In sleep mode, the PCA stops working 0: The PCA continues to work in sleep mode
6	WDTE	PCA WDT function enable control bit 1: Activates the PCA module 4WDT function 0: Disables the PCA module 4WDT function
5:4	Reserved	Reserved Bits
3:1	CPS	Clock division selection and clock source selection 000: PCLK/32 001: PCLK/16 010: PCLK/8 011: PCLK/4 100: PCLK/2 101: Timer0 overflow 110: Timer1 overflow 111: ECI external clock, clock PCLK quadrature sampling
0	CFIE	PCA Counter interrupt enable control signal 1: Enable interrupt 0: Turn off interruptions

11.4.3 Counting register (PCA_CNT)

Offset address: 0x008

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															

position	Symbols	Description
31:16	Reserved	Reserved Bits
15:0	CNT	Timer counter value Only in the PCA stop state CNT can write, otherwise the write is invalid

11.4.4 Interrupt Clear Register (PCA_ICLR)

Offset address: 0x00C

Reset value: 0x0000 009Fh

31:8	7	6	5	4	3	2	1	0
Reserved	CF	Reserved	CCF4	CCF3	CCF2	CCF1	CCF0	
	W0		W0	W0	W0	W0	W0	

position	Symbols	Description
31:8	Reserved	Reserved Bits
7	CF	PCA counter overflow flag cleared (software write 0clear, write 1invalid), read out value1
6:5	Res.	Reserved Bits
4	CCF4	PCA counter module 4Compare/capture flag bit cleared (software write 0clear, write 1invalid), read value1
3	CCF3	PCA counter module 3Compare/capture flag bit cleared (software write 0clear, write 1invalid), read value1
2	CCF2	PCA counter module 2Compare/capture flag bit cleared (software write 0clear, write 1invalid), read value1
1	CCF1	PCA counter module 1Compare/capture flag bit cleared (software write 0clear, write 1invalid), read value1
0	CCF0	PCA counter module 0Compare/capture flag bit cleared (software write 0clear, write 1invalid), read value1

11.4.5 Compare capture mode registers (PCA_CCAPM0~4)

Offset Address

CCAPM0: 0x010; CCAPM1: 0x014; CCAPM2:
0x018; CCAPM3: 0x01C; CCAPM4: 0x020.

Reset value: 0x0000 0000

	31:8	7	6	5	4	3	2	1	0
Reserved	ECOM	CAPP	CAPN	MAT	TOG	PWM	CCIE		
	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

position	Symbols	Description
31:7	Reserved	Reserved Bits
6	ECOM	Allows comparator function control bits 1: allows comparator function; 0 : disables comparison of strong functions. When PCA is used for software counter, high speed output, PWM mode, WDT mode, to set ECOM Writing CCAMPHx or CCAMPx register will automatically set ECOM; writing CCAMPLx register will automatically clear ECOM bit
5	CAPP	Positive edge capture control bit 1: allow rising edge capture; 0 : disable rising edge capture
4	CAPN	Negative edge capture control bit 1: Allow falling edge capture; 0 : Disable falling edge capture
3	MAT	Allow matching control bits 1: Once the PCA count value matches the value in the module's compare/capture register, the CCON-hosted interrupt flag CCFx (x=0-4) will be set 0: Prohibit matching function
2	TOG	Flip control position 1: Operating in PCA high-speed output mode, the CCPx pin is flipped once the value of the PCA counter matches the value of the module's compare/capture register 0: Disable flip function
1	PWM	Pulse Width Modulation Control Bits 1: Allows the CCPx pin to be used as a PWM output 0: Disable PWM pulse width modulation function PWM function is valid only when CCAPMx[6:0]=100_0010
0	CCIE	PCA enable interrupt 1: Enable compare/capture interrupt 0: PCA compare/capture function interrupt disable

11.4.6 Compare capture data register high bit8 (PCA_CCAP0~4H)

Offset Address

CCAP0H: 0x024; CCAP1H: 0x02C; CCAP2H:
0x034; CCAP3H: 0x03C; CCAP4H: 0x044.

Reset value: 0x0000 0000

	31:8	7	6	5	4	3	2	1	0
Reserved	CCAPx [15:8]								
	R/W								

position	Symbols	Description
31:8	Reserved	Reserved Bits
7:0	CCAPx[15:8]	<p>Compare/capture mode high8 register Used to save the high8 bit of the 16bit capture count value when PCA mode is used in compare/capture mode; write The CCAPxH register automatically sets the ECOM bit of register CCAPMx. When PCA mode is used in PWM mode, it is used to control the output duty cycle loading register, and the loading register is automatically updated to the PWM comparison register when the counter low8 overflows</p>

11.4.7 Compare capture data register low bit8 (PCA_CCAP0~4L)

Offset Address

CCAP0L: 0x028; CCAP1L: 0x030; CCAP2L:
0x038; CCAP3L: 0x040; CCAP4L: 0x048.

Reset value: 0x0000 0000

	31:8	7	6	5	4	3	2	1	0
Reserved	CCAPx [7:0]								
	R/W								

position	Symbols	Description
31:8	Reserved	Reserved Bits

7:0	CCAPx[7:0]	<p>Compare/capture mode low8 register</p> <p>Used to save the low8 bit of the 16bit capture count value when PCA mode is used in compare/capture mode; write</p> <p>The CCAPxH register automatically clears the ECOM bit of register CCAPMx.</p> <p>When PCA mode is used in PWM mode, it is used to control the output duty cycle comparison register. In PWM mode, the value of the low8 bit of the counter is less than the value of CCAPx[7:0] PWM outputs low, otherwise PWM outputs high.</p>
-----	------------	--

11.4.8 Compare capture bit16 registers (PCA_CCAP0~4)

Offset Address

CCAP0: 0x050; CCAP1: 0x054; CCAP2:

0x058; CCAP3: 0x05C; CCAP4: 0x060.

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCAPx [15:0]															
R/W															

position	Symbols	Description
31:16	Reserved	Reserved Bits
15:0	CCAPx	Compare/capture mode 16bit register Used to save the 16bit capture count value when PCA mode is used in compare/capture mode; writing the CCAPx register sets the ECOM bit of register CCAPMx. Writing the CCAPX register is equivalent to writing the CCAPxL and CCAPxH 8bit registers. In compare/capture mode, you can read and write this register directly, in PWM mode, use CCAPxL and CCAPxH register

11.4.9 Compare high-speed output flag register (PCA_CCAPO)

Offset address:

0x04C Reset value:

0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved			CCAPO4	CCAPO3	CCAPO2	CCAPO1	CCAPO0	R/W

position	Symbols	Description
31:5	Reserved	Reserved Bits
4	CCAPO4	Compare the output values of the module4
3	CCAPO3	Compare the output values of the module3

2	CCAPO2	Compare the output values of the module2
1	CCAPO1	Compare the output values of the module1
0	CCAPO0	Compare the output values of the module0

12 Advanced timer (TIM4/5/6)

12.1 Advanced Timer Introduction

The Advanced Timer is a high performance counter containing three timers, Timer4/5/6, ~~the~~ identical in function and can be used to generate different forms of clock waveforms by counting, and each timer can generate a complementary pair of PWM or independent 2PWM outputs that can capture external inputs for pulse width or period measurements.

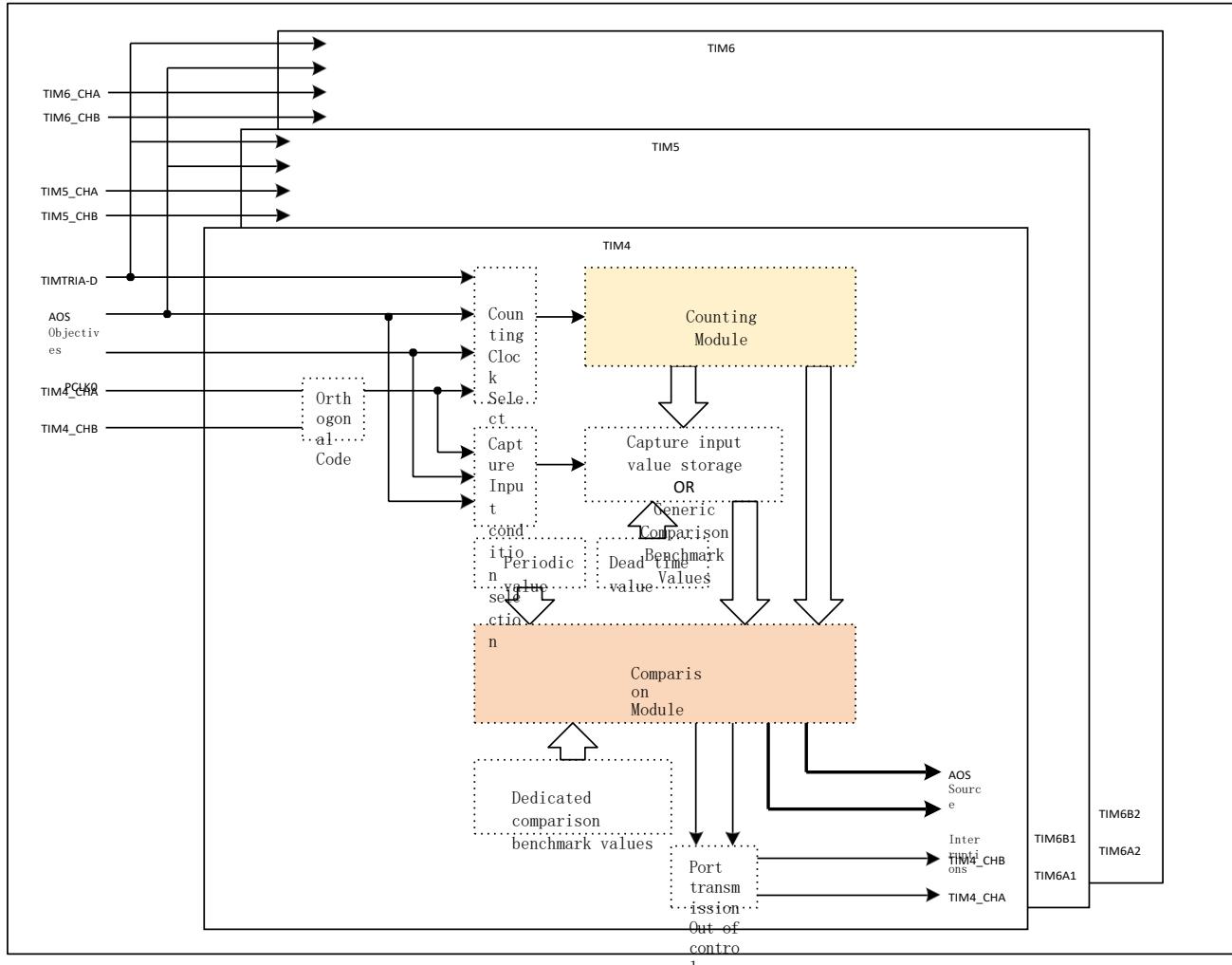
The basic functions and features of the advanced timer are shown in the table.

Waveform mode	Sawtooth wave, triangle wave
Basic Functions	● Incremental and decremental counting direction
	● Software Synchronization
	● Hardware Synchronization
	● Cache function
	● Orthogonal coding count
	● Universal PWM Output
	● Brake Protection
	● AOS Associated Action
Interrupt Type	Counting comparison match interrupts
	Counting cycle matching interrupts
	Dead time error interrupt

Table 12-1AdvancedTimer Basic Features

Port Name	Direction	Function
TIMx_CHA	Input/output	Quadrature coded count clock input port or capture input port or compare input port
TIMx_CHB		Out ports (x=4~6)
TIMTRIA	Input	
TIMTRIB		Hardware count clock input port or capture input port
TIMTRIC		Hardware start, stop, and clear condition input port
TIMTRID		

Table 12-2AdvancedTimer Port List



1

Figure 12-1 Advanced Timer Block Diagram

12.2 Advanced Timer Function Description

12.2.1 Basic movements

12.2.1.1 Basic waveform mode

Timer4/5/6 has two basic counting waveform modes, sawtooth mode and triangle waveform mode. The waveform modes are subdivided into different internal counting actions, and the triangle waveform mode is divided into triangle waveform A mode and triangle waveform B mode. The basic waveforms of sawtooth wave and triangle wave are shown in Figure 12-2 and Figure 12-3. The

difference between triangle A mode and triangle B mode is that there is a difference in cache transmission. Triangle A mode has only one cache transmission (valley) in one cycle, while triangle B mode has two cache transmissions (peak and valley) in one cycle.

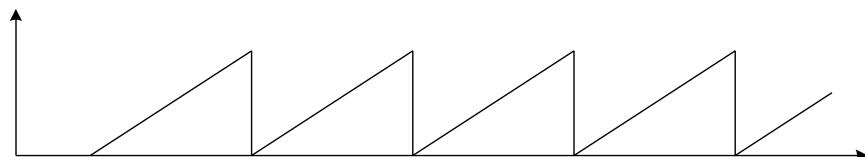


Figure 12-2 Sawtooth waveform (recursive counting)

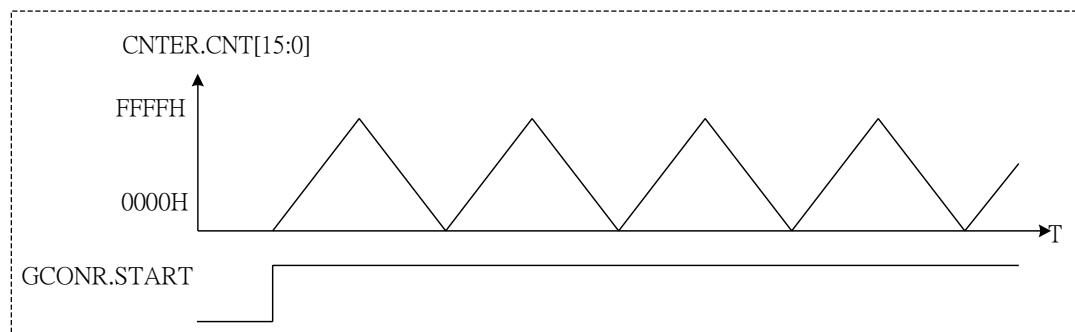


Figure 12-3 Triangular waveform

12.2.1.2 Compare Outputs

Timer4/5/6 A timer has a 2comparison output port (CHxA, CHxB) which can output the specified level when the count value matches with the count reference value; GCMAR and GCMBR registers correspond to the count reference value of CHxA and CHxB respectively. The CHxA port outputs the specified level when the counter count value is equal to GCMAR, and the CHxB port outputs the specified level when the counter count value is equal to GCMBR.

The count start level, count stop level, and count compare match level of CHxA and CHxB ports can be set by PCONR.STACA, PCONR.STPCA, PCONR.STASTPSA, PCONR.CMPCA[1:0], PCONR.PERCA[1:0], and PCONR.PERCA[1:0] of the Port Control Register (PCONR). STACB, PCONR.STPCB, PCONR.STASTPSB, PCONR.CMPCB[1:0], PCONR.PERCB[1:0] bits are set. Figure 12-4 shows

Compare the action examples of the output.

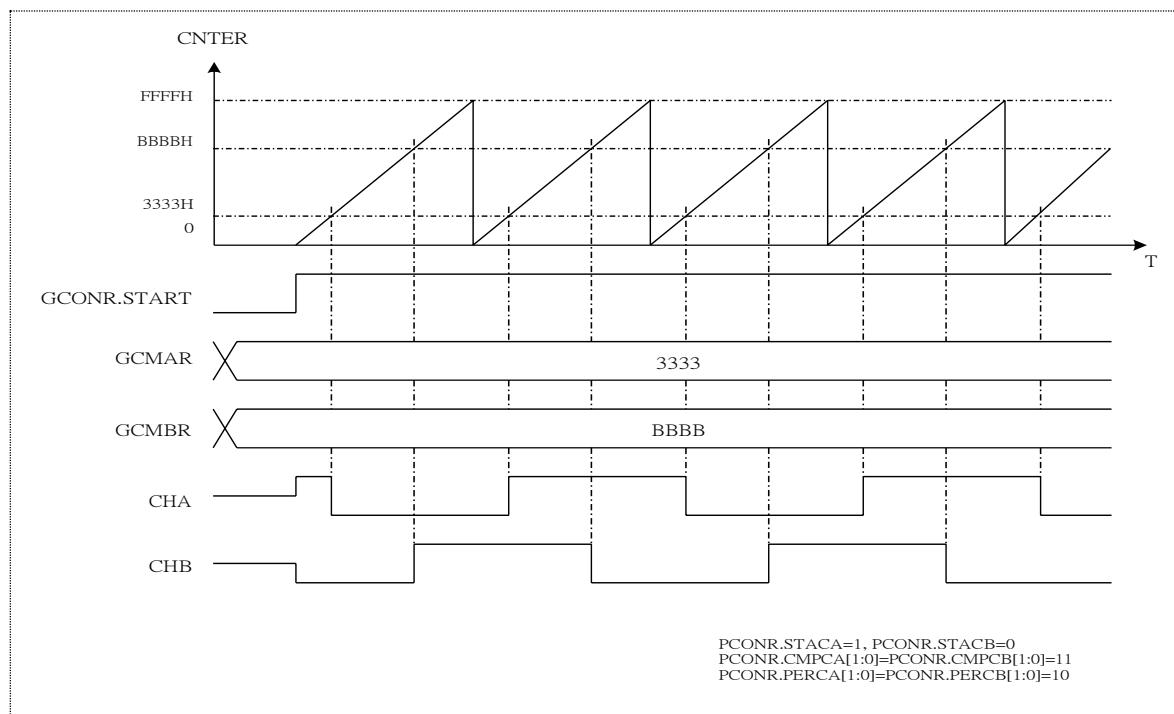


Figure 12-4 Comparison Output Action

12.2.1.3 Capture input

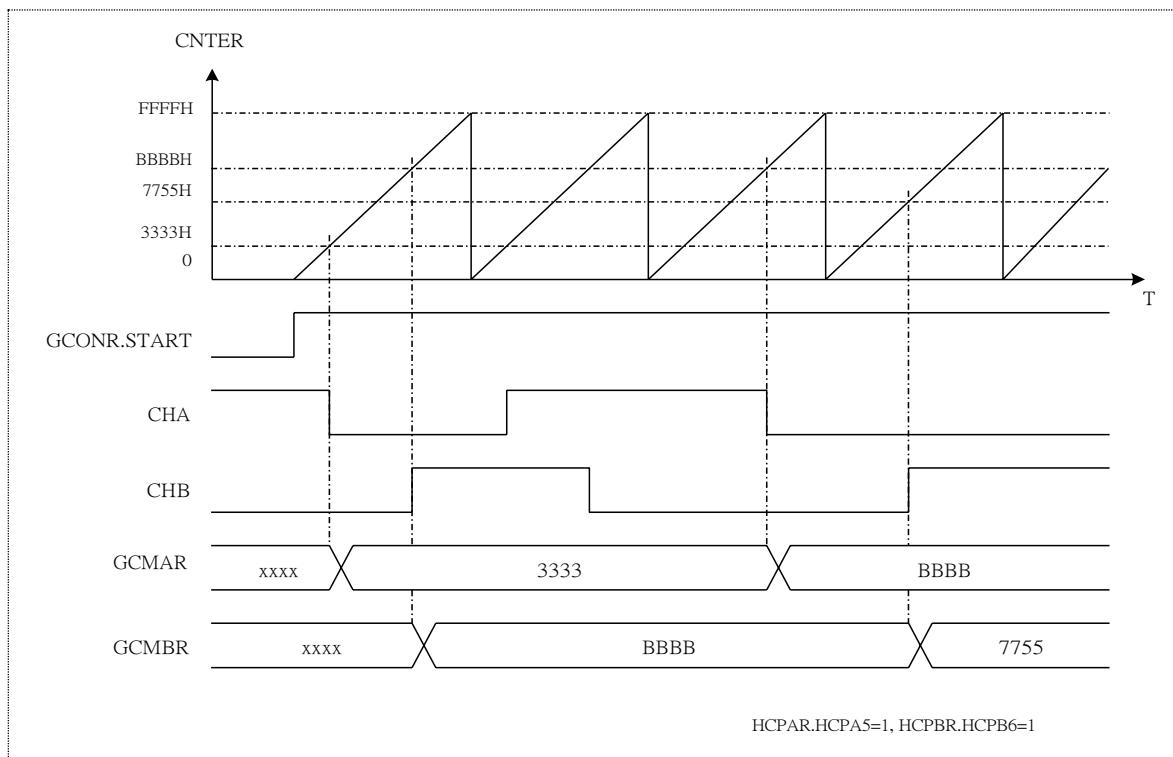


Figure 12-5 Capture Input Action

Timer4/5/6 have a capture input function with a set2 of capture input registers (GCMAR, GCMBR) to save the captured count value. CAPCA and PCONR.CAPCB bits1 of the Port Control Register (PCONR) are set to , and the capture input function is valid. When the corresponding capture input condition is set and the condition is valid, the current count value is saved in the corresponding registers (GCMAR, GCMBR).

The conditions for each set of capture inputs can be triggered by AOS events, TIMTRIA-TIMTRID inputs, CHxA or CHxB inputs, etc. The specific condition selection can be set ~~in~~ hardware capture event selection registers (HCPAR, HCPBR). Figure 12-5 shows an example of the capture input action.

12.2.2 Clock source selection

The counting clocks for Timer4/5/6 are available in the following options.

- a. PCLK and , 1024, , , , division of 2²⁴81664256 PCLK
(GCONR.CKDIV[2:0] setting)
- b. AOS event trigger input (HCUPR.HCUP [19:16] or HCDOR.HCDO [19:16] setting)
- c. Orthogonal coded inputs for CHxA and CHxB (HCUPR.HCUP[7:0] or HCDOR.HCDO [7:0])
(Setting)
- d. Port input for TIMTRIA-TIMTRID (HCUPR.HCUP [15:8] or HCDOR.HCDO [15:8])
(Setting)

As can be seen from the above description, the b, c and d clocks are independent of each other and can be set to be active or inactive respectively, and the a clock is automatically inactive when the b, c and d clocks are selected.

12.2.3 Counting direction

The counter counting direction of Timer4/5/6 can be changed by software.

The method of changing the count direction is slightly different for different waveform modes.

12.2.3.1 Sawtooth wave counting direction

In sawtooth wave mode, the counting direction can be set while the counter is counting or when it is stopped.

DIR=0 (decremental counting) ~~while in down~~ counting, the counter will change to decremental counting mode after counting up to overflow; while in decremental counting, GCONR.DIR=1 (decremental counting), ~~the~~ counter will change to decremental counting mode after counting down to overflow.

Set the GCONR.DIR bit when the count is stopped. ~~The~~ GCONR.DIR setting will not be reflected in the count until the overflow or underflow after the count starts.

12.2.3.2 Triangular wave counting direction

In triangle wave mode, the counting direction can only be set when the counter is stopped. Setting the count direction during counting is not valid. The GCONR.DIR bit is set when the counter is stopped. The setting of GCONR.DIR will not be reflected in the count until the overflow or underflow after the count starts.

12.2.4 Digital filtering

The CHxA, CHxB, and TIMTRIA~D port inputs of Timer4/5/6 have digital filtering functions. The filtering function of the corresponding port can be enabled by setting the relevant enable bit in the Filter Control Register (FCONR). The reference clock for filtering is also set through the Filter Control Register (FCONR).

After the filtered sample reference clock samples a 3consistent level on the port, the level is transmitted as a valid level to

The internal part of the module; levels that are less than sub-consistent3 are filtered out as external interference and are not transmitted to the internal part of the module. Its action

An example is shown in Figure 12-6.

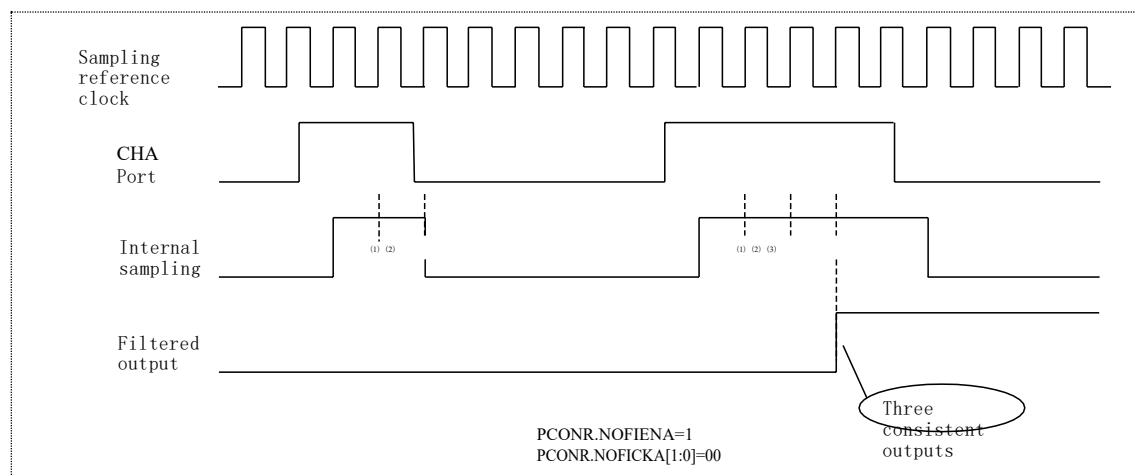


Figure 12-6 captures the filtering function of the input port

The TIMTRIA~D port is a set of ports shared between Timer4/5/6, and the digital filtering function of this set of ports is available only in the

Timer4 implementation, other timers Timer5/6 are not valid for the digital filtering function setting of this group of ports.

12.2.5 Software Synchronization

12.2.5.1 Software Synchronization Launch

Timer4/5/6 The target Timer4/5/6 can be achieved by setting the relevant bits of the Software Synchronization Start Register (SSTAR)

The simultaneous start of the

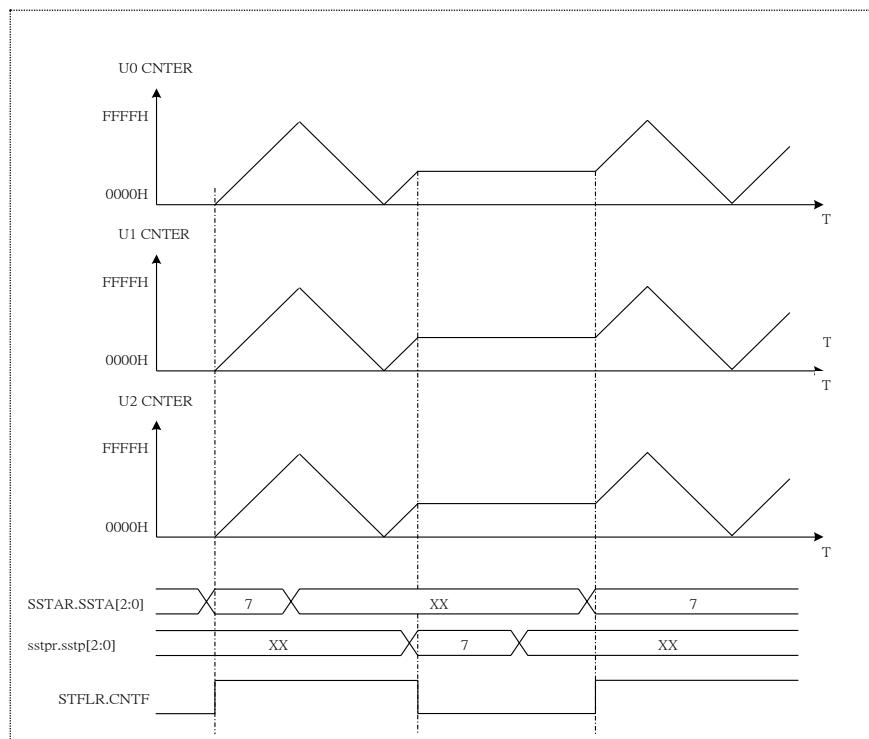


Figure 12-7 Software synchronization action

12.2.5.2 Software synchronization stop

Timer4/5/6 The target Timer4/5/6 can be achieved by setting the relevant bits of the Software Synchronization Stop Register (SSTPR)

The synchronization stop.

12.2.5.3 Software synchronization zeroing

Timer4/5/6 The target Timer4/5/6 can be achieved by setting the relevant bit in the Software Synchronized Clear Register (SCLRR)

The synchronized zeroing of the

SSTAR.SSTA0=SSTAR.SSTA1=SSTAR.SSTA2 of Timer4, as shown in Figure 12-7.

Timer4/5/6 software can be started simultaneously.

The Software Synchronization Action Related Registers (SSTAR, SSTPR,

SCLRR) are a set of registers that are independent of Timer4/5/6 and shared
HC32L110 Series User's
Manual Rev2.31

among TIMs. When reading

The counter status of each timer is read when the **SSTAR** register is read, and when the **SSTPR** or **SCLRR** is read0.

12.2.6 Hardware Synchronization

Each timer has a common 4external common input port (TIMTRIA, TIMTRIB, TIMTRIC, TIMTRID) and an 4AOS target, in addition to independent 2common input ports (CHxA, CHxB), which can realize hardware synchronization between timers.

12.2.6.1 Hardware Synchronized Start

Each Timer4/5/6 can be selected to start the counter in hardware mode. Timers with the same hardware start condition can be synchronized to start when the start condition is valid. The specific hardware start condition is defined by the hardware start event selection register (HSTAR) setting to determine.

12.2.6.2 Hardware synchronization stop

Each Timer4/5/6 can be selected to stop the counter in hardware mode. Timers with the same hardware stop condition can be selected to achieve synchronous stop when the stop condition is valid. The specific hardware stop condition is defined by the hardware stop event selection register (HSTPR) setting to determine.

12.2.6.3 Hardware Synchronous Zeroing

Each Timer4/5/6 can be selected to clear the counter in hardware. Timers with the same hardware clear condition can be synchronously cleared when the clear condition is valid. The specific hardware clearing condition is defined by the hardware clearing event selection register (HCLRR) setting to determine.

12.2.6.4 Hardware synchronized capture input

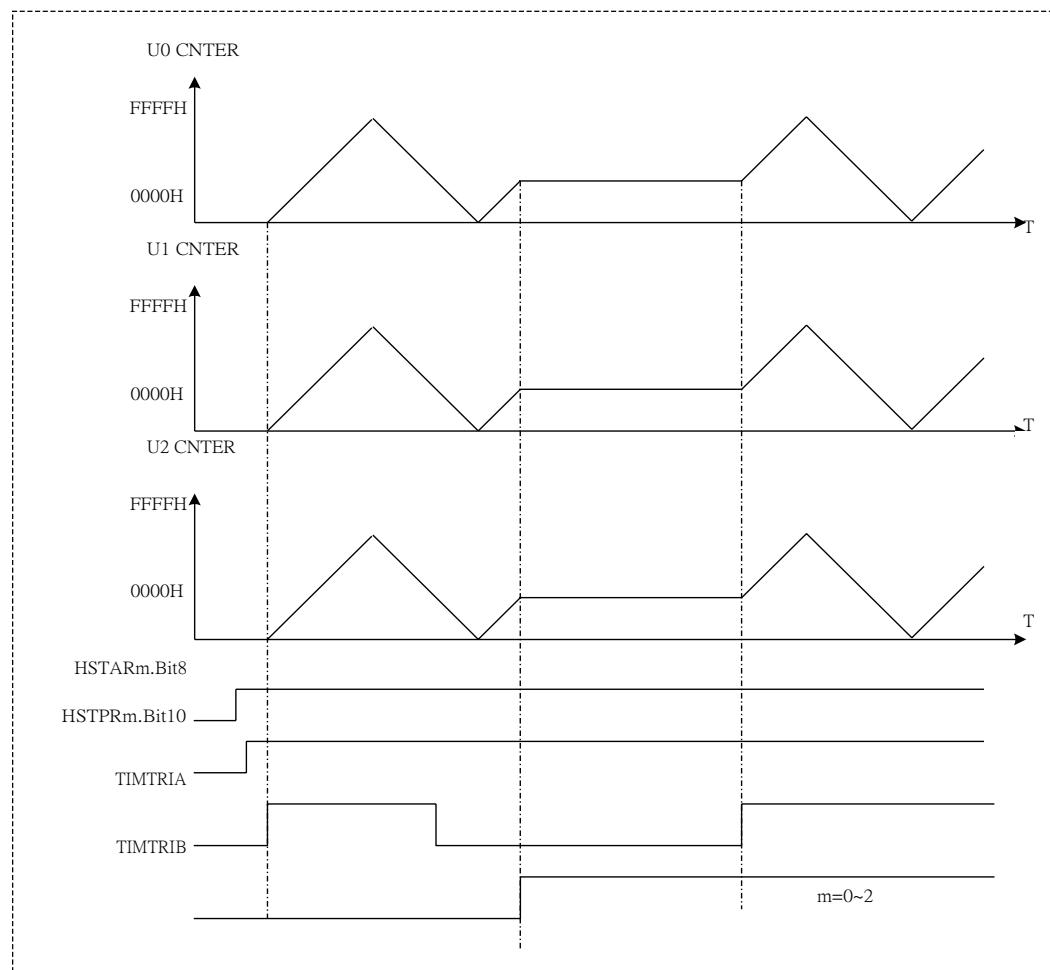
Each Timer4/5/6 can be selected to implement the capture input function in hardware. Timers with the same capture input function condition can be selected to achieve synchronous capture input when the capture input

function condition is valid. The specific hardware capture input function condition is determined by the setting of hardware capture event selection registers (HCPAR, HCPBR).

12.2.6.5 Hardware synchronized counting

Timer4/5/6 can choose to use hardware input as CLOCK for counting. Timer with the same hardware counting condition can realize synchronous counting when hardware counting CLOCK is valid. The specific hardware count condition is determined by the setting of the hardware decrement event select register (HCUPR) and hardware decrement event select register (HCDOR). When the hardware synchronous counting function is selected, only the external input clock source is selected, which does not affect the counter start, stop, or

Zeroing action. The start, stop, and clear of the counter need to be set separately. Figure 12-8 shows



an example of the hardware synchronization action of Timer4/5/6.

Figure 12-8 Hardware synchronization action

12.2.7 Cache function

A cache action is the selection of the following events to occur at the cache transfer time point by setting the cache control register (**BCONR**)

- a. The value of the General Periodic Reference Buffer Register (**PERBR**) is automatically transferred to the General Periodic Reference Register (**PERAR**) in
- b. The value of the General Comparison Reference Value Cache Register (**GCMCR, GCMDR**) is automatically transferred to the General Comparison Reference Value Register (**GCMAR, GCMBR**) (when comparing outputs)
- c. The value of the General Comparison Reference Value Register (**GCMAR, GCMBR**) is automatically transferred to the General Comparison Reference Value Buffer Register (**GCMCR, GCMDR**) (when capturing input)

Figure 12-9 shows the timing diagram of the single-cache mode of the General Comparison Reference Value Register during the compare output action. As can be seen from the figure, changing the value of the General Comparison Reference Register (**GCMAR**) during counting adjusts the output duty cycle, and changing the value of the General Periodic Reference Register (**PERAR**) adjusts the output period.

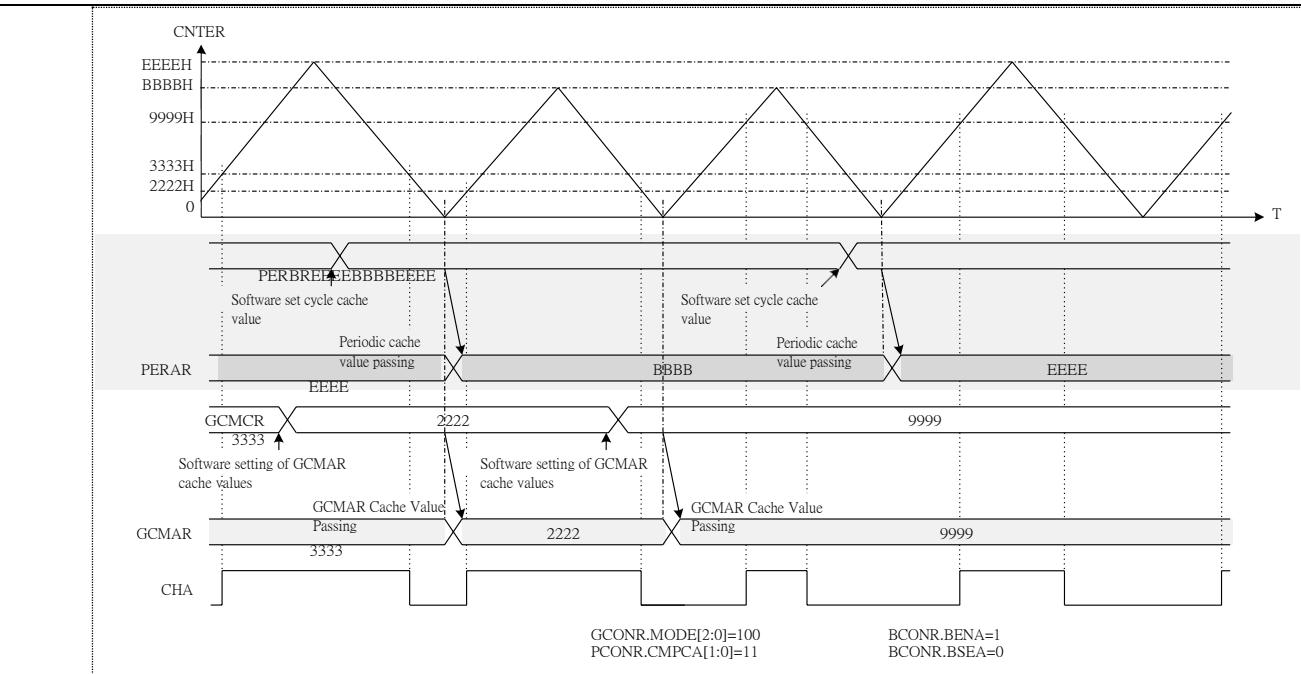


Figure 12-9 Comparison of output timings in single cache mode

12.2.7.1 Cache transfer time point

12.2.7.2 Generic cycle reference value cache transfer time point

The cycle reference value cache transfer time point is the incremental count up overflow point or decremental count down overflow point for sawtooth waves and the count valley point for delta waves.

12.2.7.3 Generic comparison of reference value cache transfer time points

BCONR.BENA=1 or BCONR.BNEB=1 is set during Ramp A mode, the cache action is valid. The cache transfer occurs at the upper overflow point or lower overflow point.

BCONR.BENA=1 or BCONR.BNEB=1 when triangle wave A mode is set, the cache action is valid. The cache transmission occurs at the count valley.

BCONR.BENA=1 or BCONR.BNEB=1 when triangle wave B mode is set, the cache action is valid. The cache transmission occurs at the count valley and count peak points.

12.2.7.4 Capture input value cache transfer time point

The capture input action cache transfer time point is when the input action is captured.

12.2.7.5 Cache transfer during clear action

If a clearing action is generated during the normal comparison output action in the ramp count mode or hardware count mode, a cache transfer will occur for the universal cycle reference value, universal comparison reference value, etc. according to the corresponding cache action setting status.

12.2.8 Universal PWM Output

12.2.8.1 PWM Spread Spectrum Output

To reduce external interference from the PWM output, there is a spread spectrum configuration in the PWM output stage. The phase of the PWM

output is fine-tuned for each PWM output cycle.

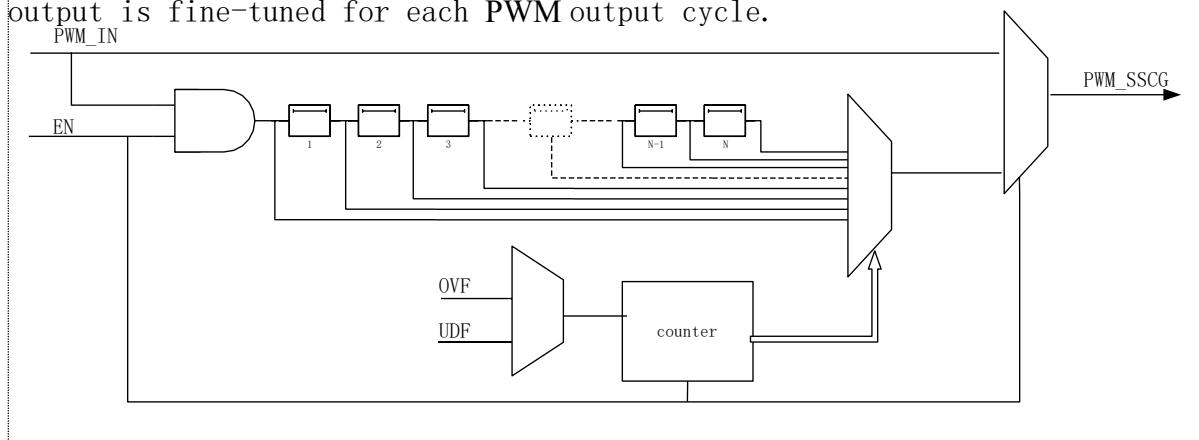


Figure 12-10 PWM spread spectrum output diagram

12.2.8.2 Independent PWM output

The CHxA and CHxB 2ports of each timer can output PWM waves independently.

As shown in Figure 12-11, the CHA port of Timer6 outputs a PWM waveform.

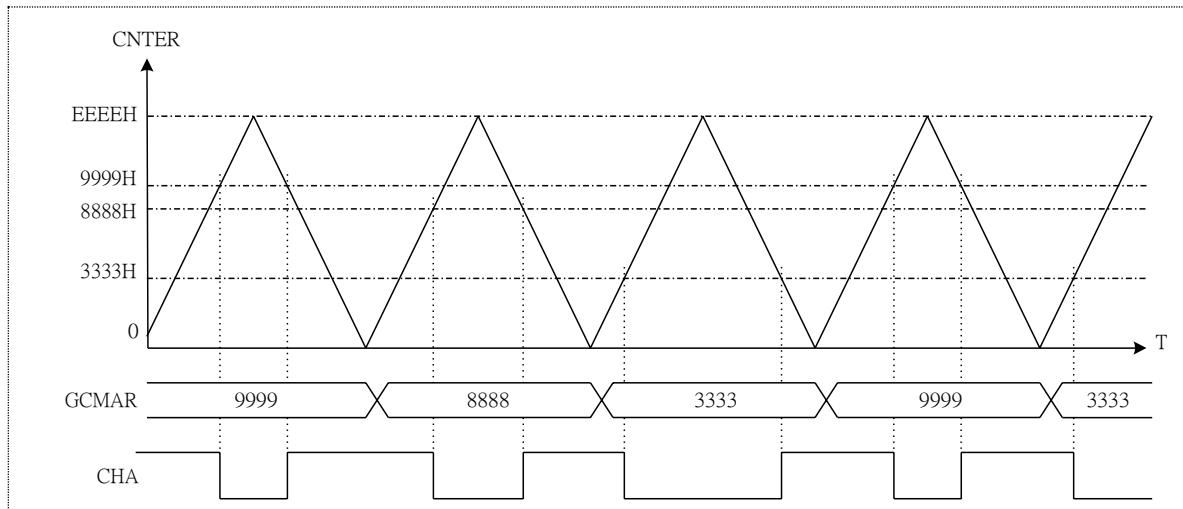


Figure 12-11 CHA output PWM Wave

12.2.8.3 Complementary PWM Outputs

The CHxA port and CHxB port can be combined to output complementary PWM waveforms in different modes.

12.2.8.3.1 Software Settings GCMBR Complementary PWM Output

The software setting GCMBR complementary PWM output means that the value of the General Comparison Base Value Register (GCMBR) used for waveform output of CHxB port in sawtooth mode and delta waveform A mode and delta waveform B mode is set directly by the register and has no direct relationship with the value of the General Comparison Base Value Register (GCMAR).

Figure 12-12 shows the output example of software setting GCMBR complementary PWM waveform.

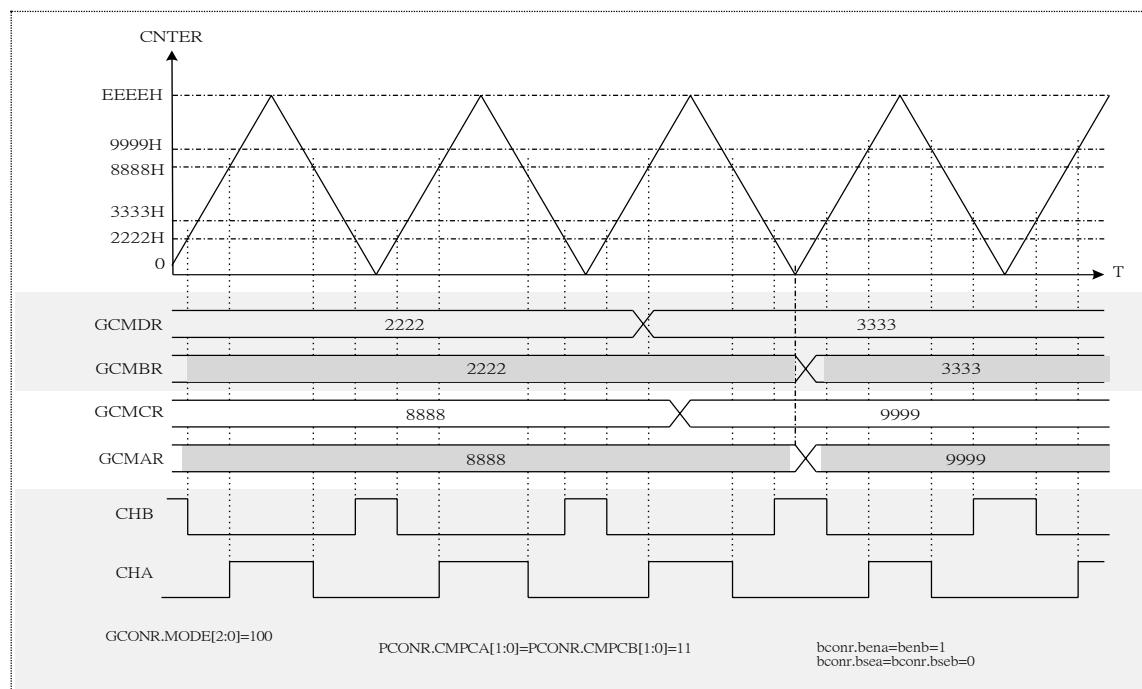


Figure 12-12 Software Setting GCMBR Complementary PWM Output in Delta Wave A Mode

12.2.8.3.2 Hardware Settings GCMBR Complementary PWM Output

The hardware-set GCMBR complementary PWM output means that the value of the General Comparison Reference Register (GCMBR) used for the waveform output of the CHxB port is determined by the operation of the values of the General Comparison Reference Register (GCMAR) and the Dead Time Reference Registers (DTUAR, DTDAR) in the Delta A mode and Delta B mode.

Figure 12-13 shows an example of hardware setting GCMBR complementary PWM waveform output.

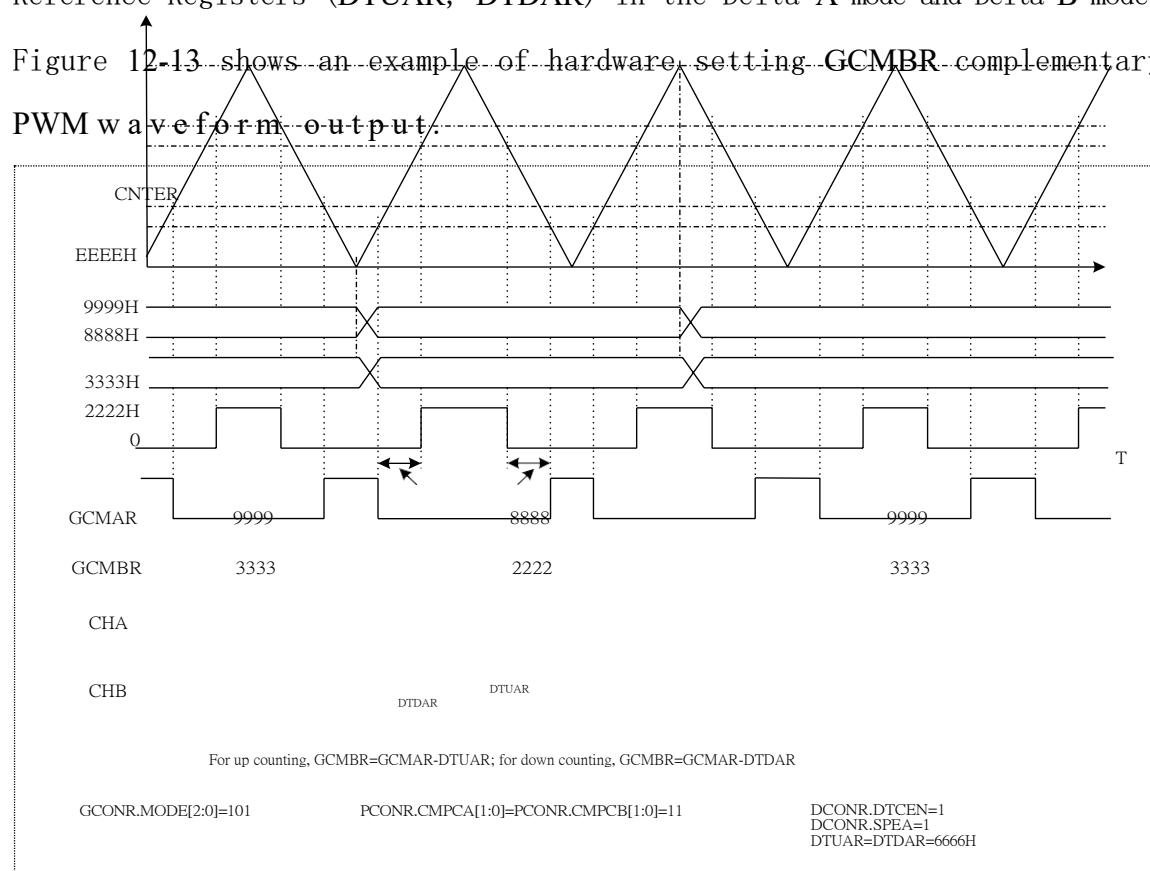
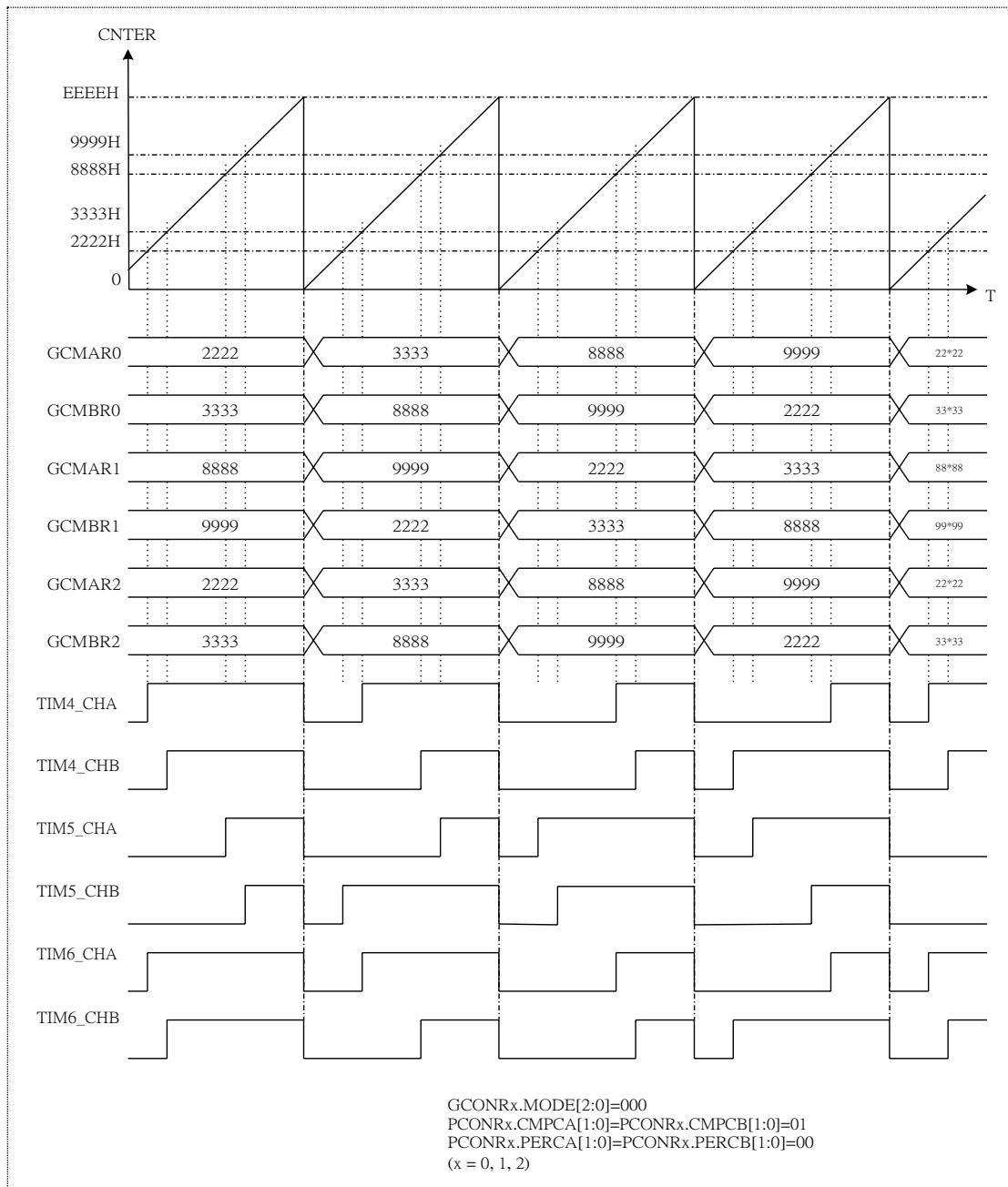


Figure 12-13 Hardware Setting GCMBR Complementary PWM Output in Delta B Mode (Symmetric Deadband)

12.2.8.4 Multi-phase PWM output

The CHxA and CHxB ports of each timer can output phase 2 independent PWM waveforms or a set of complementary PWM waveforms, and multiple timers can be combined with software and hardware synchronization to achieve multi-phase PWM output. As shown in Figure 12-14, Timer4, Timer5, Timer6



combine to output phase6 PWM waveforms; as shown in Figure 12-15, Timer4, Timer5, Timer6 combine to output phase3 complementary PWM waveforms.

Figure 12-146 Phase PWM Wave

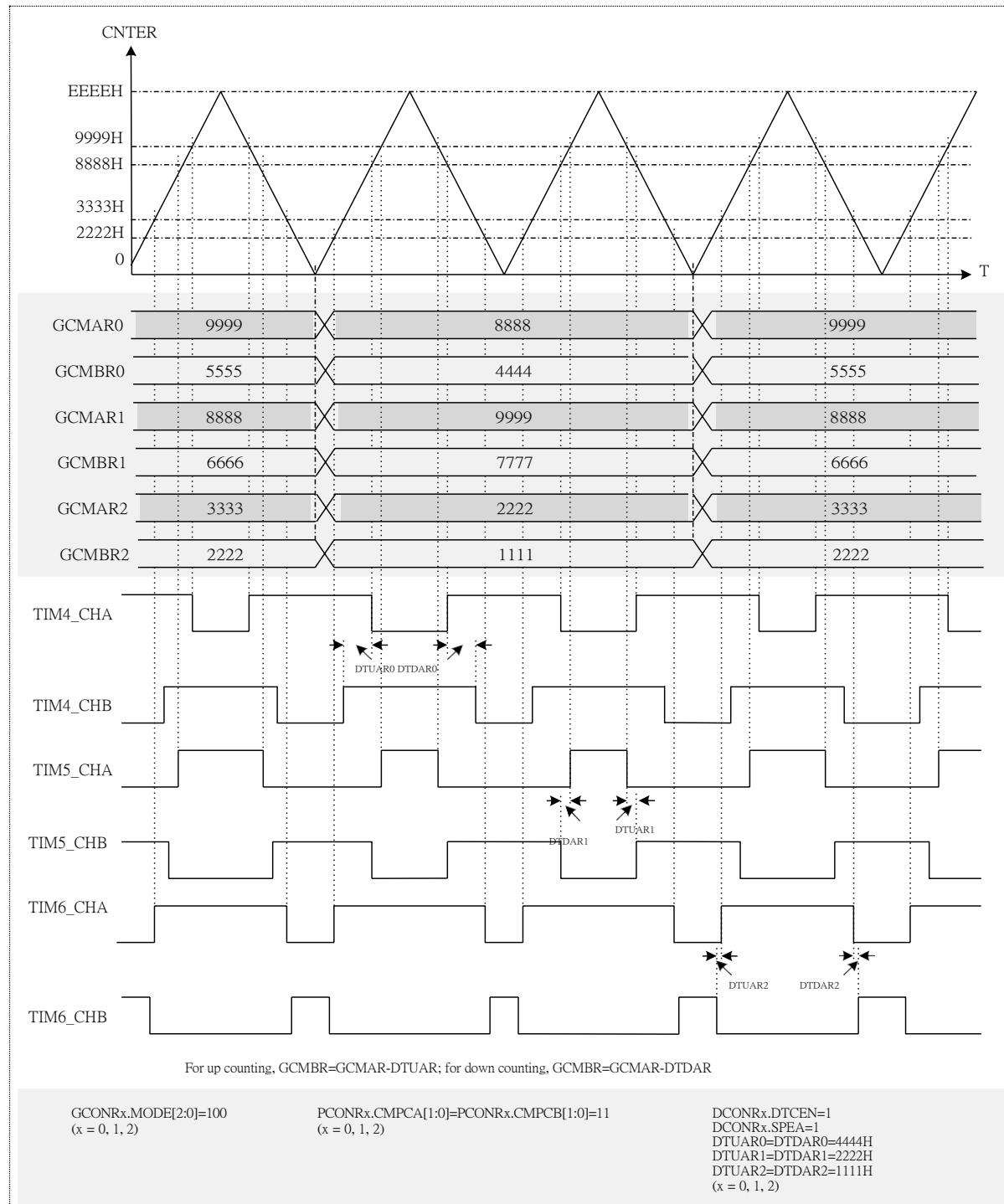


Figure 12-15 Triangular wave A mode with dead time three-phase complementary PWM wave output

12.2.9 Orthogonal coding count

By treating the CHxA input as an AIN input, the CHxB input as a BIN input, and any of the TIMTRIA-D inputs as a ZIN input, Advanced Timer can achieve quadrature coding of the three inputs.

One timer's AIN,BIN alone can realize position counting mode; two timers' AIN,BIN, ZIN combined action can realize revolution counting mode, one timer is used for position counting and one timer is used for revolution counting.

In revolution counting mode, every two timer combinations (timer 4 and timer 5 combination, timer as 4position counting unit and timer as 5revolution counting unit) realize position counting and revolution counting respectively.

The counting conditions of AIN and BIN are realized by setting the orthogonal relationship between CHxA and CHxB in the hardware incremental event selection register (HCUPR) and hardware decremental event selection register (HCDOR); the input action of ZIN is realized by setting the hardware zero event selection register (HCLRR) of the position counter unit to clear the position counter of the position counter unit, and by setting the hardware incremental event selection register (HCUPR) of the revolution counter unit to clear the revolution counter of the revolution counter unit. The input action of ZIN is realized by setting the hardware clear event selection register (HCLRR) of the position unit to clear the position counter of the position counter unit, and by setting the hardware incremental event selection register (HCUPR) of the revolution counter unit to count the revolution counter of the revolution counter unit.

12.2.9.1 Position counting mode

The orthogonal encoding position mode means that the basic counting function, phase difference counting function and direction counting function are

implemented according to the input of AIN and BIN.

12.2.9.1.1 Basic Count

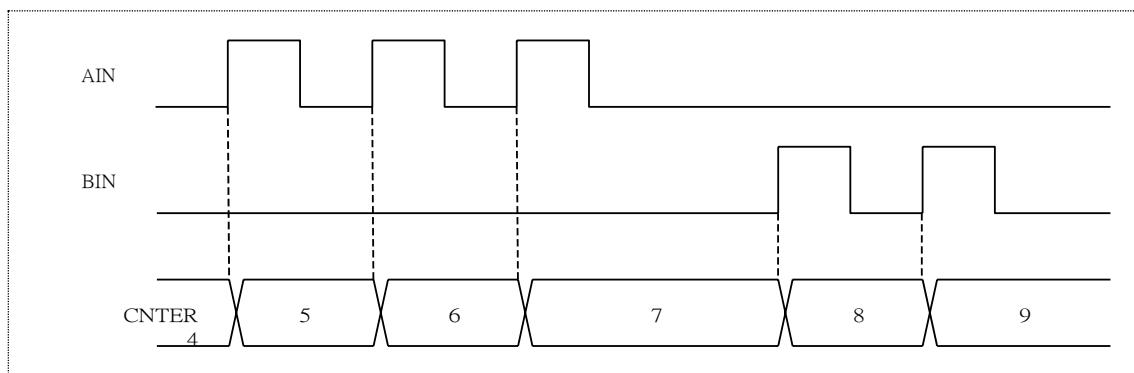


Figure 12-16 Basic counting action in position mode

By setting the HCUPR and HCDOR registers, various ways of phase difference counting can be flexibly implemented.

12.2.9.1.2 Phase difference counting

Phase difference counting means counting according to the phase relationship between AIN and BIN. Depending on the setting, multiplier1 counting, 2 multiplicative counting, 4 multiplicative counting, etc. can be realized, as shown in Figure 12-17 to Figure 12-19 below.

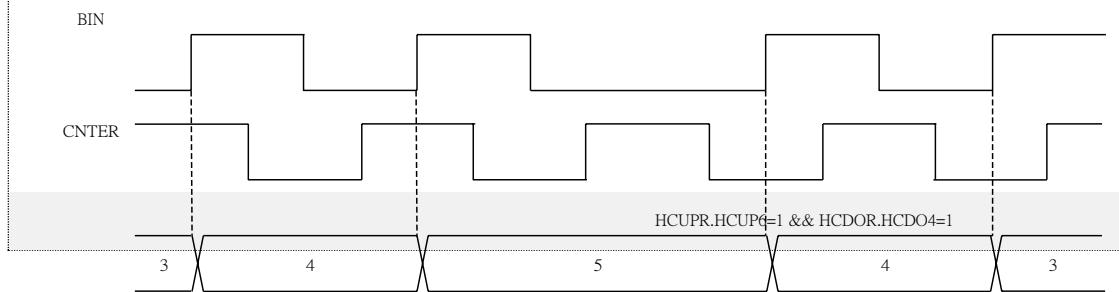


Fig. 12-17 Phase difference counting action setting in position mode (1X)

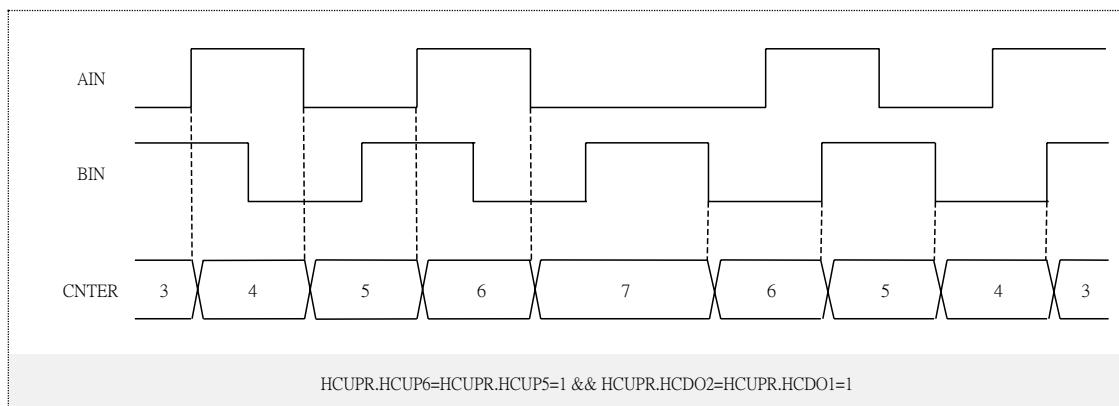


Fig. 12-18 Phase difference counting action setting in position mode (2 times)

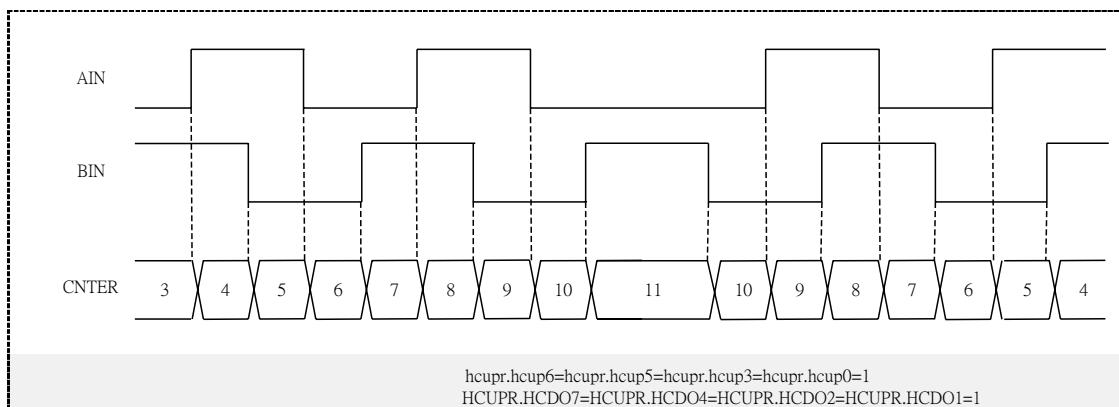


Figure 12-19 Phase difference counting action setting in position mode (4 times)

12.2.9.1.3 Direction Counting

Direction counting means setting the input state of AIN as direction control and the input of BIN as clock counting, as shown in Figure 12-20.

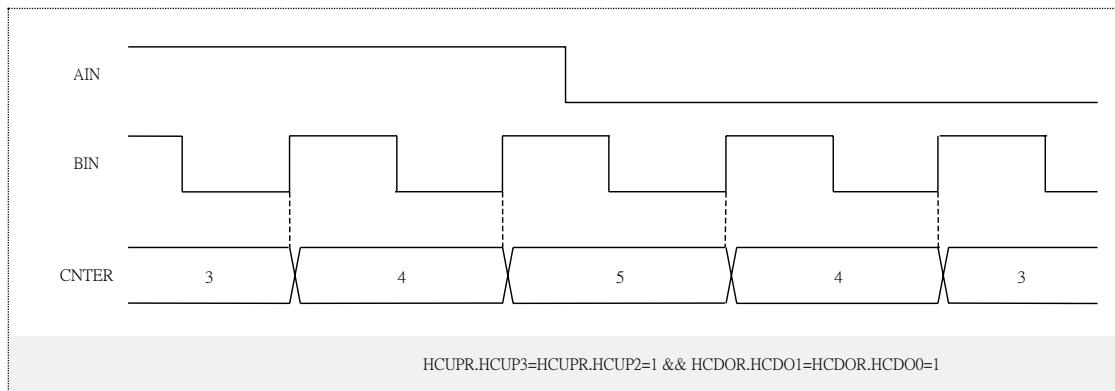


Figure 12-20 Direction counting action in position mode

12.2.9.2 Public Rotation Mode

The orthogonal coding revolution mode is to add ZIN input events to the AIN and BIN counts to judge the number of revolutions, etc. The Z-phase counting function, the position counter output counting function, and the mixed Z-phase counting and position counter output counting function can be implemented in the revolution mode according to the counting method of the revolution counter. This function is realized by using two Advanced Timers.

12.2.9.2.1 Z-phase count

Z-phase counting is a counting action in which the revolution counter unit counts according to the input of ZIN and the position counter unit is cleared to zero at the same time.

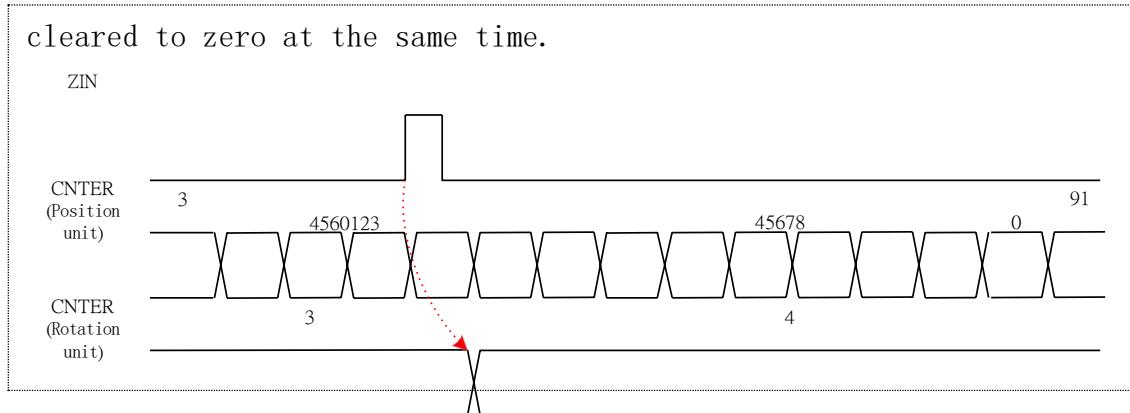


Fig. 12-21 Z-phase counting action during the rotation mode

12.2.9.2.2 Position Overflow Count

Position overflow counting means that an overflow event is generated when the count of the position counter unit overflows or underflows, which triggers the counter of the revolution counter unit to perform a count (the input of ZIN does not perform the count action of the revolution counter unit and the zero action of the position counter unit during this counting method)

The overflow event of the position counter unit is counted by the AOS module's linkage selector to achieve the count of the rotation counter unit, which enables the position overflow count. The hardware incremental (decremental) event selection register (HCUPR or HCDOR) of the rotary counter unit selects the bits inl Bit16:it19, and the AOS module sets the event

source of the corresponding incremental (decremental) event as the count ^{ZIN} overflow event of the position counter unit, please refer to the AOS chapter for details. As shown in Figure 12-22.

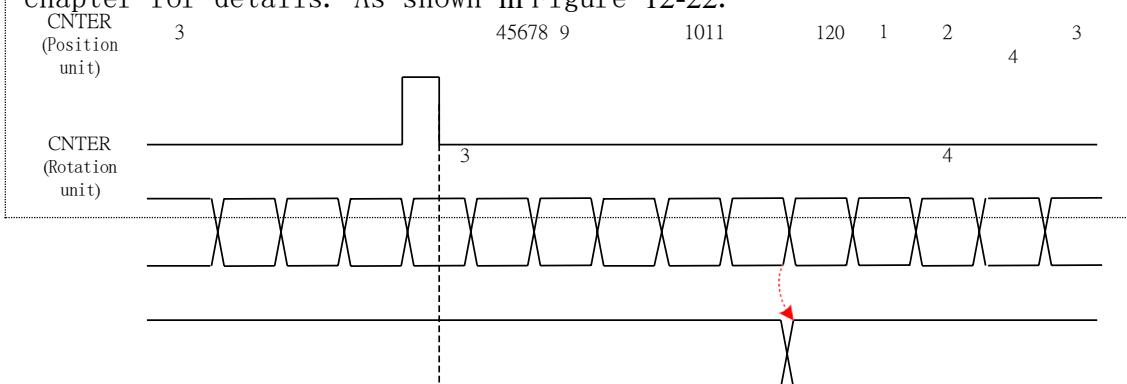


Fig. 12-22 Position counter output counting action in revolution mode

12.2.9.2.3 Mixed counting

Mixed counting is a counting action that combines the above two counting methods, Z-phase counting and position overflow counting, and its implementation is also a combination of the above two counting methods.

As ^{ZIN} shown in Figure 12-23.



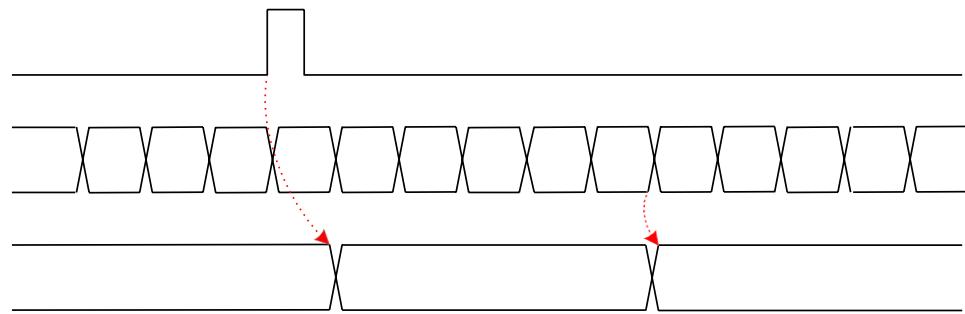


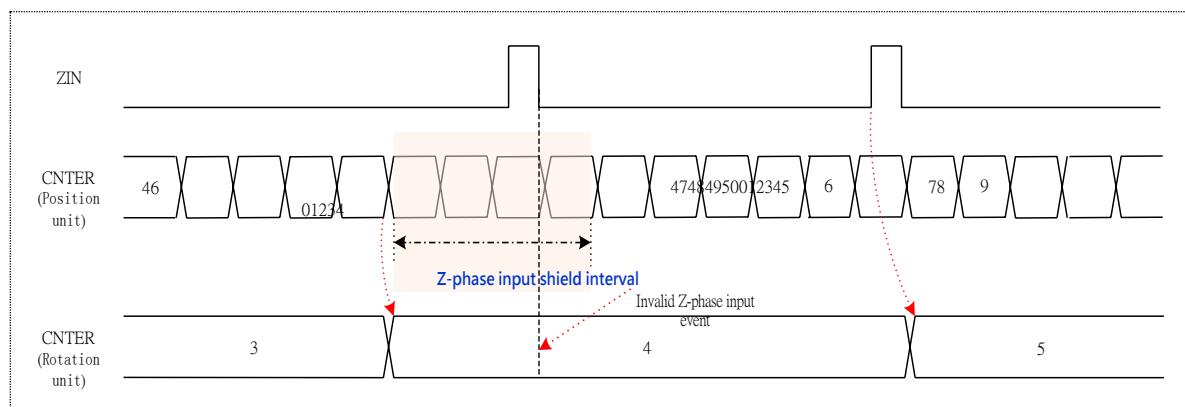
Fig. 12-23 Mixed counting action of Z-phase counting and position counter output during revolution mode

12.2.9.2.4 Z-phase action shield

During the Z-phase counting function or mixed counting function in the revolution counting mode, you can set the valid input of ZIN to be masked from counting of the revolution counter unit and clearing of the position counter unit for several cycles after the upper or lower overflow point of the position counter (GCONR.ZMSK[0:1] setting)

ZMSKPOS of the general control register (GCONR) of the position counter unit is 1, the Z-phase mask function of the position counter unit is enabled, and the number of cycles of Z-phase mask is set by GCONR.ZMSK. is enabled.

Figure 12-24 shows that when the ZIN phase is input within one 4counting cycle after the position counter unit is overflowed, the action of ZIN phase input is invalid, i.e., the revolution counter unit does not



count and the position counter unit is not cleared; after that, the ZIN phase input acts normally.

Figure 12-24 Rotation counting mode - mixed counting Z-phase shield action example 1

Figure 12-25 shows the mixed counting in revolution counting mode, in the first 3cycle after the position counter unit counts overflow, the counting direction changes and the set masking period of one 4cycle becomes invalid (the actual ZIN phase masking function is maintained for one3 ~~4~~ downward counting starts. After the count overflow of the position counter unit, the ZIN phase masking function is turned back on and becomes invalid after one4

cycle. During the ZIN phase blocking period, the ZIN phase input function is invalid, i.e., the revolution counter unit does not count and the position counter unit does not clear; the ZIN phase input afterwards acts normally.

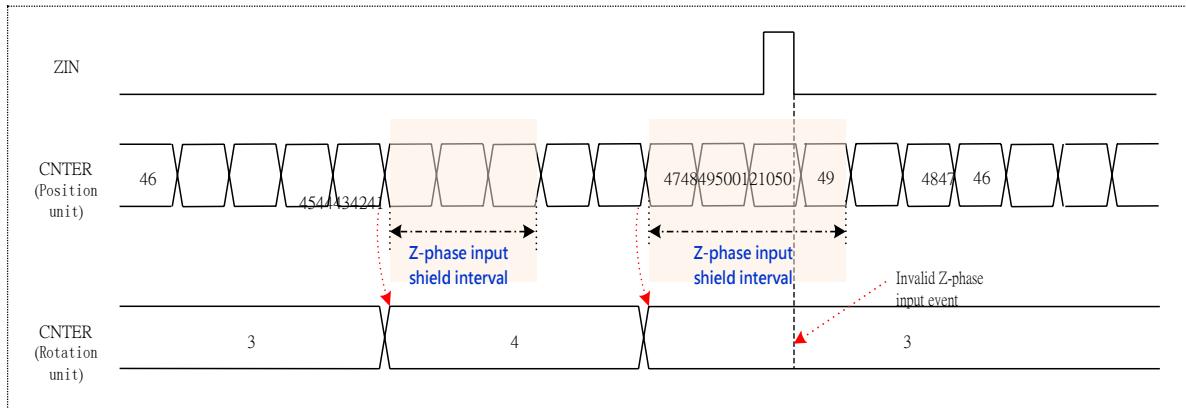


Figure 12-25 Rotation counting mode - mixed counting Z-phase shield action example 2

12.2.10 Cycle interval response

Timer4/5/6's General Comparison Reference Value Registers (GCMAR~GCMDR) can generate a dedicated valid request signal when the counts are compared and matched, respectively, to the AOS module for associating actions with other modules.

The request signal can be generated after every few cycles. By setting the VPERR.PCNTS bit of the Valid Period Register (VPERR) to specify how many cycles the request signal is valid, a valid request signal will not be output in other cycles even if the count value is equal to the

value of the Comparison Reference Value Register GCMAR or GCMBR.

Figure 12-26 shows an example of a period interval valid request signal.

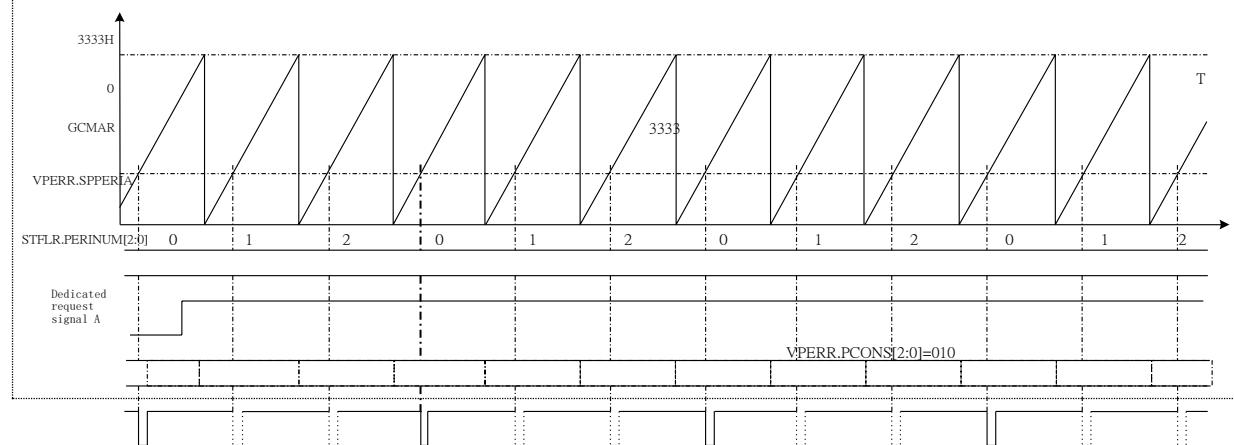


Figure 12-26 Cycle interval valid request signal action

12.2.11 Protection mechanism

Advanced Timer provides protection control over the output state of the port.

The Advanced Timer has a 4common port output invalid event 4interface that connects to a group4 of brake events output by the brake control module. The abnormal condition events selected on each interface can be set from the brake control to enable control of the generic PWM output when an abnormal condition is monitored on these interfaces.

The port output state can change to a pre-defined state if a brake event from the brake control is monitored during normal output. The general-purpose PWM output port can change its port state to output high resistance, output low, or output high (as determined by the PCONR.DISVALA, PCONR.DISVALB settings) exception event occurs.

For example, if PCONR.DISSEL[1:0]=01&PCONR.DISVALA=01 is set, then at the CHxA end

If a brake event is generated on the output invalid condition during the normal output of the CHxA port, the output on the CHxA port becomes a high resistance state.

12.2.12 Interrupt description

Timer4/5/6 each contain a total of one class of 39 interrupt. They are a general count compare match interrupt (including a capture input interrupt), a count cycle match interrupt, and a dead time error interrupt.

12.2.12.1 Counting comparison match interrupts

The General Comparison Reference Registers (GCMAR-GCMDR) are used to generate a valid comparison match signal by comparing with the count value. CMAF to STFLR.CMDF bits in the Status Flag Register (STFLR) are set to match. If the interrupt control register is set to

(If the corresponding bit in ICONR.INTENA~ICONR.INTEND of (ICONR) is enable interrupt, the corresponding interrupt request will also be triggered.

Input action occurs when the capture input valid condition selected by the hardware capture event selection registers (HCPAR, HCPBR) is generated. If the ICONR.INTENA or ICONR.INTENB bit of the Interrupt Control Register (ICONR) is set to enable interrupt, the corresponding interrupt request is triggered.

12.2.12.2 Counting cycle matching interrupts

The STFLR.OVFF or STFLR.UDFF bit of the Status Flag Register (STFLR) is set to when the ramp count reaches the upper overflow point, the ramp decrement count reaches the lower overflow point, the triangle count reaches the valley point, or the triangle count reaches the peak point. If the ICONR.INTENOVF and ICONR.INTENUDF bits of the Interrupt Control Register (ICONR) are set to enable the interrupt, the count cycle matching interrupt can be triggered at the corresponding time point.

12.2.12.3 Dead time error interrupt

Load the value of the dead time reference value registers (DTUAR, DTDAR) into the general comparison reference value register

(If the cycle limit is exceeded in the GCMBR, a dead time error is generated and the STFLR.DTEF bit in the Status Flag Register (STFLR) is set to . 1 If the ICONR.INTENDE bit of the Interrupt Control Register (ICONR) is set to enable the interrupt, the dead time error interrupt will be triggered at that time.

12.2.13 Brake Protection

DISVALA, PCONR.DISVALB. The hardware automatically changes the port state to the preset state (high, low, high resistance state, keep normal output) if an invalid condition is valid.

12.2.13.1 Port Brakes vs. Software Brakes

After the port is polarity-selected and enabled, it is digitally filtered and synchronized to generate the port brake flag; the port brake flag is used as

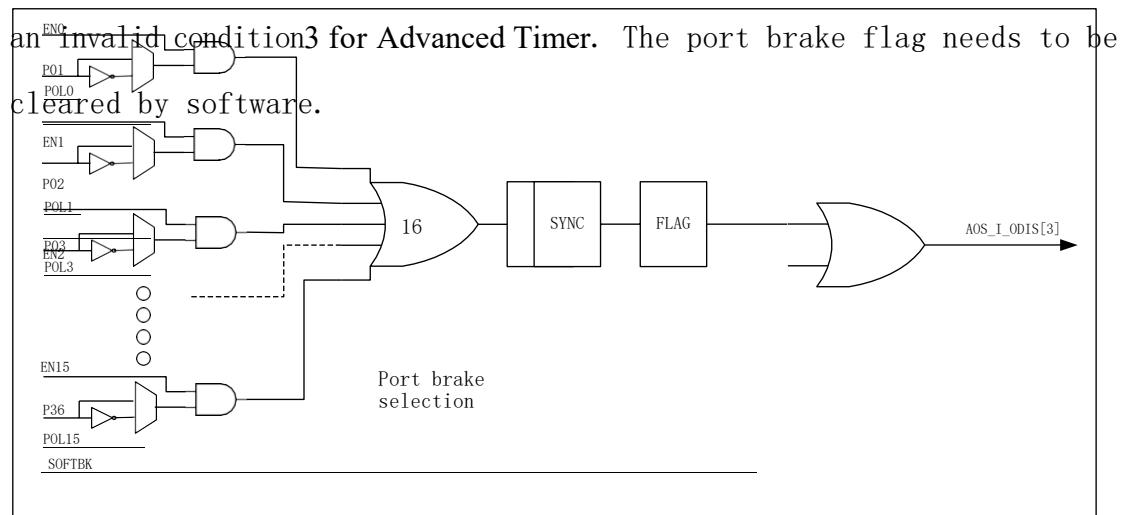


Figure 12-27 Schematic diagram of port brakes and software brakes

12.2.13.2 Low power mode automatic brake

The system enters low-power mode, and the PWM will not work properly after the clock is stopped. Low-power mode is used as Advanced Timer's invalid condition controls the 2PWM brake.

12.2.13.3 Output level same high and low brake

After the output level is monitored and enabled, it is synchronized to generate the same high and same low brake flags; the port brake flags are used

as invalid conditions1 for Advanced Timer. The same high and same low brake

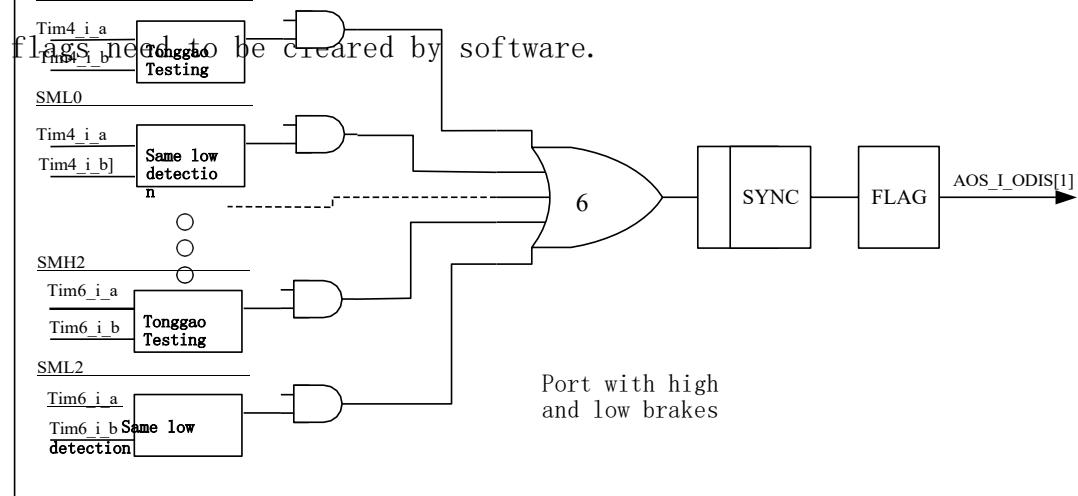


Fig. 12-28 Schematic diagram of the same high and low brake output

12.2.13.4 VC Brakes

VC1, VC2 interrupt flags are enabled as invalid conditions0 for Advanced Timer.

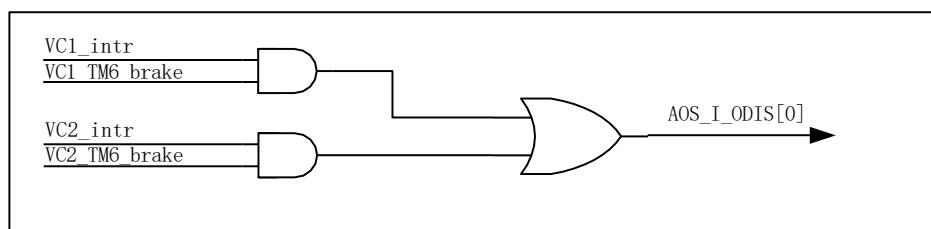


Figure 12-29VCBrake Control Schematic

12.2.14 Internal Interconnection

12.2.14.1 Interrupt trigger output

Since one interrupt of Timer4/5/6 contains multiple interrupt sources. The interrupt signal controlling the trigger ADC and the interrupt signal controlling the AOS have separate control to select different sources,

and can ^{INT_CMA}select any interrupt source for the upper overflow, lower overflow, or comparative matching of a total of one 6TIMx interrupt source as the ^{CMB}trigger condition.

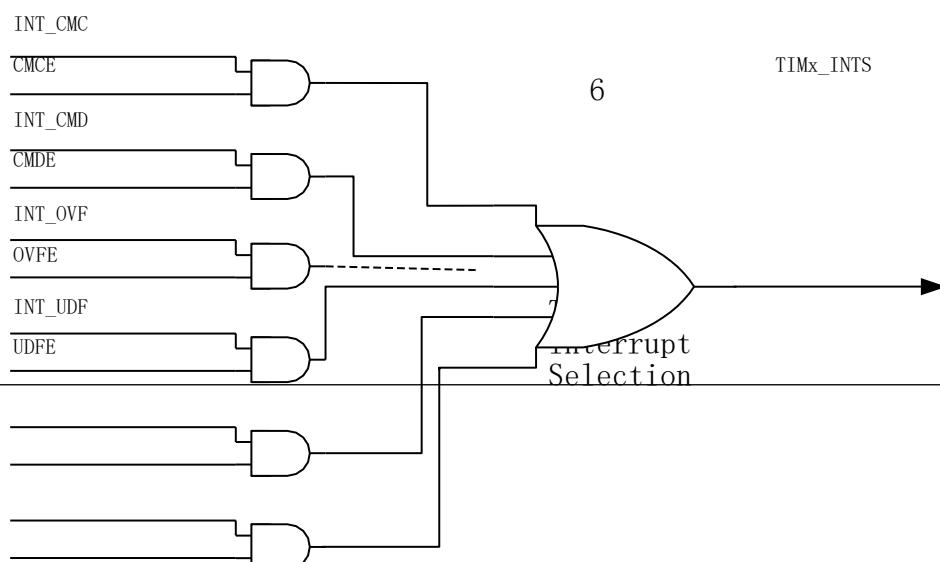


Figure 12-30 Timer4/5/6 Interrupt Selection

12.2.14.2 AOS Trigger

The Advanced Timer has 4 AOS triggers, each of which can be selected as an interrupt source for a different module. The selected signal generates a single pulse trigger input to Advanced Timer to control the start, stop, and reset of Advanced Timer's counter.

Timer4/5/6 internally use registers to select different AOS_I_TRIG as its own trigger signal. For example, the HSTAR register can be used to trigger

the hardware start of the corresponding timer using interrupt	AOS_i_trig0	AOS_i_trig1	AOS_i_trig2	AOS_i_trig3
Selecting control signals user's	ITRIG.IAOS0S	ITRIG.IAOS1S	ITRIG.IAOS2S	ITRIG.IAOS3S
0000	TIM0_INT	TIM0_INT	TIM0_INT	TIM0_INT
0001	TIM1_INT	TIM1_INT	TIM1_INT	TIM1_INT

0011	LPTIMER_INT	LPTIMER_INT	LPTIMER_INT	LPTIMER_INT
0100	TIM4_INTS	TIM4_INTS	TIM4_INTS	TIM4_INTS
0101	TIM5_INTS	TIM5_INTS	TIM5_INTS	TIM5_INTS
0110	TIM6_INTS	TIM6_INTS	TIM6_INTS	TIM6_INTS
0111	UART0_INT	UART0_INT	UART0_INT	UART0_INT
1000	UART1_INT	UART1_INT	UART1_INT	UART1_INT
1001	LPUART_INT	LPUART_INT	LPUART_INT	LPUART_INT
1010	VC1_INT	VC1_INT	VC1_INT	VC1_INT
1011	VC2_INT	VC2_INT	VC2_INT	VC2_INT
1100	RTC_INT	RTC_INT	RTC_INT	RTC_INT
1101	PCA_INT	PCA_INT	PCA_INT	PCA_INT
1110	SPI_INT	SPI_INT	SPI_INT	SPI_INT
1111	ADC_INT	ADC_INT	ADC_INT	ADC_INT

Table 12-3AOSSource Selection

12.2.14.3 Port Trigger TRIGA-TRIGD

Port triggering can control the hardware start, stop, clear, capture, counter plus or minus counting functions of Advanced Timer, with digital

Selection control, and the port signals are chip independent Control	TRIGA	TRIGB	TRIGC	TRIGD
0000	P01	P01	P01	P01
0001	P02	P02	P02	P02
0010	P03	P03	P03	P03
0011	P15	P15	P15	P15
0100	P14	P14	P14	P14
0101	P23	P23	P23	P23
0110	P24	P24	P24	P24
0111	P25	P25	P25	P25
1000	P26	P26	P26	P26
1001	P27	P27	P27	P27
1010	P31	P31	P31	P31
1011	P32	P32	P32	P32
1100	P33	P33	P33	P33
1101	P34	P34	P34	P34
1110	P35	P35	P35	P35
1111	P36	P36	P36	P36

Table 12-4 Port Trigger Selection

12.2.14.4 Compare output VC with Advanced Timer interconnection

The VC can be internally interconnected to the capture input of the Advanced Timer, allowing capture of the edges of the VC output.

12.2.14.5 UART interconnects with Advanced Timer

UARTx_RX / LPUART_RX can be interconnected internally to BaseTimer, LPTimer, PCA and

Advanced Timer is a software that allows automatic baud rate recognition.

UART selection control register is in the port control register GPIO_CTRL3, VC output control register is in the

VC control module.

12.3 Register Description

CH0 base address

0x40003000 CH1 base

address 0x40003400

Register base address	Offset Address	Description
CH2		
0x40006800ER	0x000	General purpose count reference value register
TIMx_PERAR	0x004	General Cycle Reference Value Register
TIMx_PERBR	0x008	General purpose cycle reference value cache register
TIMx_GCMAR	0x010	General comparison A reference value register
TIMx_GCMBR	0x014	General comparison B reference register
TIMx_GCMCR	0x018	General-purpose comparison C reference value register
TIMx_GCMDR	0x01C	General comparison D reference register
TIMx_DTUAR	0x040	Dead time reference value register
TIMx_DTDAR	0x044	Dead time reference value register
TIMx_GCONR	0x050	General Control Register
TIMx_ICONR	0x054	Interrupt control register
TIMx_PCONR	0x058	Port Control Register
TIMx_BCONR	0x05C	Cache Control Register
TIMx_DCONR	0x060	Deadband control register
TIMx_FCONR	0x068	Filter control register
TIMx_VPERR	0x06C	Effective period register
TIMx_STFLR	0x070	Status Flag Register
TIMx_HSTAR	0x074	Hardware boot event selection register
TIMx_HSTPR	0x078	Hardware stop event selection register
TIMx_HCELR	0x07C	Hardware Clear Event Select Register
TIMx_HCPAR	0x080	Hardware capture event selection register
TIMx_HCPBR	0x084	Hardware capture event selection register
TIMx_HCUPR	0x088	Hardware decrement event selection register
TIMx_HCDOR	0x08C	Hardware decrement event selection register
TIMx_IFR	0x100	Interrupt Flag Register
TIMx_ICLR	0x104	Interrupt clear register
TIMx_CR	0x108	Spread Spectrum and Interrupt Trigger Selection Register
TIMx_AOSSR	0x110	AOS selection register, common to all three channels
TIMx_AOSCL	0x114	AOS brake flag clear register, shared by three channels
TIMx_PTBK5	0x118	Port brake control register, shared by three channels
TIMx_TTRIG	0x11C	Port trigger control register, shared by three channels

TIMx_PTBKP	0x124	Port brake polarity control register, shared by three channels
TIMx_SSTAR	0x3F4	Software synchronization start register
TIMx_SSTPR	0x3F8	Software synchronization stop register
TIMx_SCLRR	0x3FC	Software synchronous zeroing register

Table 12-5AdvancedTimer Register List

12.3.1 General purpose count reference value register (TIMx_CNTER)

Address offset:

0x000 Reset value:

0x0000 0000

31302928272625242322212019181716	Reserved
----------------------------------	----------

CNT [15:0]

R/W

1514131211109876543210

position	Symbols	Function Description
31:16	Reserved	-
15:0	CNT [15:0]	Current counter count value

12.3.2 General Periodic Reference Value Register (TIMx_PERAR)

Address offset:

0x004 Reset value:

0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Reserved								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PERA [15:0]								

position	Symbols	Function Description
31:16	Reserved	-
15:0	PERA [15:0]	Counting cycle value, set the counting cycle value for each round of counting

12.3.3 General Cycle Buffer Register (TIMx_PERBR)

Address offset:

0x008 Reset value:

0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERB [15:0]															
R/W															

position	Symbols	Function Description
31:16	Reserved	-
15:0	PERB [15:0]	Cache count cycle value, cache value of count cycle

12.3.4 General Comparison Reference Value Register (TIMx_GCMAR-GCMDR)

Address offsets: 0x0010, 0x0014, 0x0018, 0x001C

Reset value: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GCMA-D [15:0]															
R/W															

position	Symbols	Function Description
31:16	Reserved	-
15:0	GCMA-D [15:0]	Counting comparison reference value, comparison reference value setting, and count value equal when the matching signal is valid

12.3.5 Dead Time Reference Value Register (**TIMx_DTUAR-DTDAR**)

Address offsets: 0x040, 0x044

Reset value: 0x0000 FFFF

31302928272625242322212019181716	Reserved
----------------------------------	----------

DTUA/DTDA [15:0]
R/W

1514131211109876543210

position	Symbols	Function Description
31:16	Reserved	-
15:0	DTUA/DA [15:0]	Dead time value, dead time setting value

12.3.6 General Control Register (TIMx_GCONR)

Address offset:

0x050 Reset value:

0x00000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ZMSK	ZMSK	ZMSK	
												R/W	R/W	R/W	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DIR	Res.	CKDIV	MODE		START		
								R/W		R/W	R/W		R/W		

position	Marker	Function Description
31:20	Reserved	-
19:18	ZMSK	<p>Number of Z-phase input shield cycles Count cycle value of orthogonally encoded Z-phase input masked 00: Z-phase input shield function is invalid 01: The Z-phase input is masked for 4one count period after the position count overflow or underflow. 10: The Z-phase input is masked for 8one count period after the position count overflow or underflow. 11: The Z-phase input is masked for 16one count period after the position count overflow or underflow.</p>
17	ZMSKPOS	<p>Z-phase input position counter selection 0: This timer acts as a position counter during the Z-phase input, and the position counter clear function operates normally during the masking cycle. 1: This timer acts as a position counter during the Z-phase input, and the position counter is cleared during the masking cycle. Blocked</p>
16	ZMSKREV	<p>Z-phase input revolution counter selection 0The timer acts as a revolution counter when the Z-phase is input, and the revolution counter function operates normally during the masking period. 1The timer acts as a revolution counter when the Z-phase is input, and the revolution counter counts during the masking period. Blocked</p>
15:9	Reserved	-
8	DIR	<p>Counting direction 0: decreasing count;1 : increasing count</p>

7	Reserved	-
6:4	CKDIV	Counting clock selection 000: PCLK0001: PCLK0/2010: PCLK0/4011: PCLK0/8 100:101 PCLK0/64110: PCLK0/256 111 : PCLK0/1024 PCLK0/16
3:1	MODE	Counting Mode 000: Sawtooth wave A mode 100: Triangle wave A mode 101: Triangle wave B mode

		Please do not set other values
0	START	0Counter start : Counter off ;1 : Counter start <i>Note: This bit will automatically change when a software stop condition or hardware stop condition is in effect0</i>

12.3.7 Interrupt Control Register (TIMx_ICONR)

Address offset:

0x054 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved					INTEN	INTEN	INTEN	Reserved	INTEN	INTEN	INTEN	INTEN	R/W	R/W	R/W	R/W

position	Marker	Function
31:9	Reserved	-
8	INTENDE	Dead time error interrupt enable 0: The interrupt is invalid if the dead time is wrong 1: This interrupt is enabled when the dead time is wrong
7	INTENUDF	Underflow interrupt enable 0: The interrupt is invalid if the underflow occurs during sawtooth wave or if the count reaches the valley point during delta wave. 1: The interrupt is enabled when the underflow occurs during a sawtooth wave or when the count reaches the valley point during a triangle wave.
6	INTENOVF	Overflow interrupt enable 0: The interrupt is invalid if the overflow occurs during a sawtooth wave or if the count reaches the peak during a triangle wave. 1: The interrupt is enabled when the overflow occurs during a sawtooth wave or when the count reaches the peak during a delta wave.
5:4	Reserved	-
3	INTEND	Count Match Interrupt Enable D 0: If the GCMDR register is equal to the count value, the interrupt is invalid. 1: The interrupt is enabled when the GCMDR register is equal to the count value.
2	INTENC	Count Match Interrupt Enable C 0: If the GCMCR register is equal to the count value, the interrupt is invalid. 1: The interrupt is enabled when the GCMCR register is equal to the count value.

1	INTENB	Count Match Interrupt Enable B 0: This interrupt is invalid when the GCMBR register is equal to the count value, or when a capture input event occurs. 1: This interrupt is enabled when the GCMBR register is equal to the count value, or when a capture input event occurs. Can
0	INTENA	Count Match Interrupt Enable A 0: This interrupt is invalid when the GCMAR register is equal to the count value, or when a capture input event occurs 1: When the GCMAR register is equal to the count value, or when a capture input event occurs, this interrupt enables the Can

12.3.8 Port Control Register (TIMx_PCONR)

Address offset:

0x058 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved			DISVALB		DISSELB		OUTENB		PERCB		CMPBC		STASTPS		STPC	
			R/W		R/W		R/W		R/W		R/W		R/W		R/W	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved			DISVALA		DISSELLA		OUTENA		PERCA		CMPCA		STASTPS		STPC	
			R/W		R/W		R/W		R/W		R/W		R/W		R/W	

position	Marker	Function
31:29	Reserved	-
28:27	DISVALB	CHxB output status control 00: The CHxB port outputs normally when the selected conditions 0 to 3 of the forced output invalidation conditions are met. 01: When the selected conditions 0 to 3 of the forced output invalidation conditions are met, the CHxB port outputs a high resistance state 10: The CHxB port outputs low when the selected conditions 0 to 3 of the forced output invalidation conditions hold. 11: The CHxB port outputs high when the selected conditions 0 to 3 of the forced output invalidation conditions are met.
26:25	DISSELB	Forced output invalid condition option B 00: select the forced output invalid condition0;01 : select the forced output invalid condition1 10: select the forced output invalid condition2;11 : select the forced output invalid condition3
24	OUTENB	Output enable B 0: Invalid CHxB port output during Advanced Timer function 1: CHxB port output valid for Advanced Timer function
23:22	PERCB	Port status setting when cycle value matches B 00: When the counter count value is equal to the cycle value, the CHxB port output is held low 01: When the counter count value is equal to the cycle value, the CHxB port output is set high 10: When the counter value is equal to the cycle value, the CHxB port output is set to the previous state 11: When the counter count value is equal to the cycle value, the CHxB port output is set to the inverted level

21:20	CMPCB	<p>Port status setting B when comparison values match</p> <p>00: When the counter count is equal to GCMBR, the CHxB port output is held low</p> <p>01: The CHxB port output is set high when the counter count value is equal to GCMBR</p> <p>10: When the counter value is equal to GCMBR, the CHxB port output is set to the previous state</p> <p>11: When the counter count value is equal to GCMBR, the CHxB port output is set to the inverse level</p>
19	STASTPSB	<p>Count Start Stop Port Status Selection B</p> <p>0: The CHxB port output is determined by STACB, STPCB when counting starts or stops</p> <p>1: The CHxB port output is set to the previous state when counting starts or stops</p> <p><i>Note: The count start here refers to the initial count start or stop and start again; the count stop refers to the initial stop or stop after the count start</i></p>

18	STPCB	Counting stop port status setting B 0: The CHxB port output is set low when the count is stopped 1: The CHxB port output is set high when the count is stopped
17	STACB	Counting start port status setting B 0: The CHxB port output is set low at the start of the count 1: The CHxB port output is set high when counting begins
16	CAPCB	Function mode selection B 0: Comparison output function;1 : Capture input function
15:13	Reserved	-
12:11	DISVALA	CHxA output status control 00: The CHxA port outputs normally when the selected conditions 0 to 3 of the forced output invalidation conditions are met. 01: The CHxA port outputs a high resistance state when the selected conditions 0 to 3 of the forced output invalidation conditions hold. 10: The CHxA port outputs low when the selected conditions 0 to 3 of the forced output invalidation conditions hold. 11: The CHxA port outputs high when the selected conditions 0 to 3 of the forced output invalidation conditions are met.
10:9	DISSELA	Forced output invalid condition selection A 00: select the forced output invalid condition0;01 : select the forced output invalid condition1 10: select the forced output invalid condition2;11 : select the forced output invalid condition3
8	OUTENA	Output enable A 0: Invalid CHxA port output during Advanced Timer function 1: CHxA port output valid for Advanced Timer function
7:6	PERCA	Port status setting when cycle value matches A 00: When the counter count value is equal to the cycle value, the CHxA port output is held low 01: The CHxA port output is set high when the counter count value is equal to the cycle value 10: The CHxA port output is set to the previous state when the counter count value is equal to the cycle value 11: When the counter count value is equal to the cycle value, the CHxA port output is set to the inverted level
5:4	CMPCA	Port status setting A when the comparison value matches 00: When the counter count is equal to GCMAR, the CHxA port output is held low 01: The CHxA port output is set high when the counter count is equal to the GCMAR 10: When the counter count is equal to the GCMAR, the CHxA port output is set to the previous state 11: When the counter count is equal to the GCMAR, the CHxA port output is set to the inverted level

3	STASTPSA	Count Start Stop Port Status Selection A 0: The CHxA port output is determined by STACA, STPCA when counting starts or stops 1: The CHxA port output is set to the previous state when counting starts or stops <i>Note: The count start here refers to the initial count start or stop and start again; the count stop refers to the initial stop or stop after the count start</i>
2	STPCA	Counting stop port status setting A 0: The CHxA port output is set low when the count is stopped. 1: The CHxA port output is set high when the count is stopped
1	STACA	Counting start port status setting A 0: The CHxA port output is set low at the start of the count 1: The CHxA port output is set high when counting begins
0	CAPCA	Function mode selection A

		0: Comparison output function;1 : Capture input function
--	--	--

12.3.9 Cache Control Register (**TIMx_BCONR**)

Address offset:

0x05C Reset value:

0x0000 0000

31:30	2928272625242322212019181716	Reserved
-------	------------------------------	----------

1514131211109876543210

Reserved	BENP R/W	Reserved	BENB R/W	Res.	BENA R/W
----------	-------------	----------	-------------	------	-------------

position	Marker	Function
31:9	Reserved	-
8	BENP	Periodic value cache transfer 0: Cache transfer invalid 1: Cache transfer enable (PERBR->PERAR)
7:3	Reserved	-
2	BENB	Generic comparison value cache transfer B0 : cache transfer invalid 1: Cache transfer enable When comparing output functions: (GCMDR->GCMBR) ; When capturing input functions. (GCMBR->GCMDR)
1	Reserved	-
0	BENA	Generic comparison value cache transfer A0 : cache transfer invalid 1: Cache transfer enable When comparing output functions: (GCMCR->GCMAR) ; When capturing input functions. (GCMAR->GCMCR)

12.3.10 Deadband Control Register (TIMx_DCONR)

Address offset:

0x060 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								SEPA	Reserved							
								R/W								

position	Marker	Function
31:9	Reserved	-
8	SEPA	Separation settings 0: DTUAR and DTDAR are set separately 1: The values of DTDAR and DTUAR are automatically equal
7:1	Reserved	-
0	DTCEN	Dead zone function 0: Deadband function is invalid 1: Dead zone function is effective

12.3.11 Filter Control Register (TIMx_FCONR)

Address offset:

0x068 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	NOFICKTD	NOFI ENTD	Res.	NOFICKTC	NOFI ENTC	Res.	NOFICKTB	NOFI ENTB	Res.	NOFICKTA	Res.	NOFI ENTA			
	R/W	R/W		R/W	R/W		R/W	R/W		R/W		R/W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								NOFICKGB	NOFI	Res.	NOFICKGA	NOFI			
								R/W	R/W		R/W	R/W			

position	Marker	Function
31	Reserved	-
30:29	NOFICKTD[1:0]	TRID port filter sampling reference clock selection 00: PCLK001: PCLK0/410: PCLK0/16 11: PCLK0/64
28	NOFIENTD	TRID port capture input filter enable, 0 disabled;1 enable
27	Reserved	-
26:25	NOFICKTC[1:0]	TRIC port filter sampling reference clock selection 00: PCLK001: PCLK0/410: PCLK0/16 11: PCLK0/64
24	NOFIENTC	TRIC port capture input filter enable, 0 disabled;1 enable
23	Reserved	-
22:21	NOFICKTB[1:0]	TRIB port filter sampling reference clock selection 00: PCLK001: PCLK0/410: PCLK0/16 11: PCLK0/64
20	NOFIENTB	TRIB port capture input filter enable, 0 disabled;1 enable
19	Reserved	-
18:17	NOFICKTA[1:0]	TRIA port filtering sample reference clock selection 00: PCLK001: PCLK0/410: PCLK0/16 11: PCLK0/64
16	NOFIENTA	TRIA port capture input filter enable, 0 disabled;1 enable
15:7	Reserved	-
6:5	NOFICKGB[1:0]	CHxIB port filter sampling reference clock selection 00: PCLK001: PCLK0/410: PCLK0/16 11: PCLK0/64
4	NOFIENGB	CHxIB port capture input filter enable, 0 disabled;1 enable

3	Reserved	-
2:1	NOFICKGA[1:0]	CHxIA port filter sampling reference clock selection 00: PCLK001: PCLK0/410: PCLK0/16 11: PCLK0/64
0	NOFIENGA	CHxIA port capture input filter enable, 0 disabled;1 enable

Caution.

- The TRIGA-D filter setting is valid only for TIM4 and not for Timer5/6.

12.3.12 Valid Period Register (TIMx_VPERR)

Address Offset:

0x06C Reset Value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										PCNTS		PCNTE			
										R/W		R/W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										GE PERI	GE PERIC	GE PERI	GE PERI		
										R/W	R/W	R/W	R/W		

position	Marker	Function
31:21	Reserved	-
20:18	PCNTS	Effective period selection 000: Valid period selection function is invalid 001: Valid every1 other cycle 010: Valid every2 other cycle 011: Valid every3 other cycle 100: Valid every4 other cycle 101: Valid every5 other cycle 110: Valid every6 other cycle 111: Valid every7 other cycle
17:16	PCNTE	Effective cycle counting condition selection 00: Valid period selection function is invalid 01: Sawtooth wave counting upper and lower overflow points or triangular wave valley as counting conditions 10: Sawtooth wave counting upper and lower overflow points or triangular wave crest as counting conditions 11: Sawtooth wave counting upper and lower overflow points or triangular wave valley and wave peak as counting conditions
15:4	Reserved	-
3	GEPERID	Generic signal effective period selection D 0: Valid period selection function is disabled;1 : Valid period selection function is enabled
2	GEPERIC	Generic signal effective period selection C 0: Valid period selection function is disabled;1 : Valid period selection function is enabled

1	GEPERIB	Generic signal effective period selection B 0: Valid period selection function is disabled;1 : Valid period selection function is enabled
0	GEPERIA	Generic signal effective period selection A 0: Valid period selection function is disabled;1 : Valid period selection function is enabled

12.3.13 Status Flag Register (**TIMx_STFLR**)

Address offset:

0x070 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
DIR R	Reserved							VPERNUM	Reserved					
								R						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
Reserved							DTEF R/W	UDFF R/W	OVFF R/W	Reserved	CMD R/W	CMC R/W	CMB R/W	CMA R/W

position	Marker	Function
31	DIRF	Counting direction 0: decreasing count 1: Recursive counting
30:24	Reserved	-
23:21	VPERNUM	Number of cycles The number of cycles after counting when the effective cycle selection function is enabled
20:9	Reserved	-
8	DTEF	Dead time error 0: Dead time error did not occur;1 : Dead time error occurred
7	UDFF	Underflow matching 0: No sawtooth wave underflow or delta wave count to valley 1: Sawtooth wave overflow or triangular wave count to the valley point occurs
6	OVFF	Overflow matching 0: No sawtooth wave overflow or delta wave counting to the peak 1: Sawtooth wave overflow or triangular wave counting to the peak point occurs
5:4	Reserved	-
3	CMDF	Counting Match D 0: The value of GCMDR register is not equal to the count value;1 : The value of GCMDR register is equal to the count value
2	CMCF	Counting Match C 0: The value of GCMCR register is not equal to the count value;1 : The value of GCMCR register is equal to the count value

1	CMBF	Counting Match B 0: The value of GCMBR register is not equal to the count value, and no CHxB capture completion action occurs. 1: The value of the GCMBR register is equal to the count value, or the CHxB capture completion action occurs.
0	CMAF	Counting Match A 0: The value of GCMAR register is not equal to the count value, and no CHxA capture completion action occurs.

		1: The value of the GCMAR register is equal to the count value, or the CHxA capture completion action occurs.
--	--	---

12.3.14 Hardware start event selection register (TIMx_HSTAR)

Address offset:

0x074 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STARTS	Reserved														
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSTA 15	HSTA 14	HSTA 13	HSTA 12	HSTA 11	HSTA 10	HSTA 9	HSTA 8	HSTA 7	HSTA 6	HSTA 5	HSTA 4	HSTA 3	HSTA 2	HSTA 1	HSTA 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function
31	STARTS	Hardware Boot Enable 0: Hardware boot is invalid 1: Hardware boot valid <i>Note: The SSTAR setting is not valid when hardware boot is in effect</i>
30:16	Reserved	-
15	HSTA15	Hardware start condition15: Timrid port up-sampled to falling edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
14	HSTA14	Hardware start condition14: Timrid port sampled to rising edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
13	HSTA13	Hardware start condition13: Timric port up-sampled to falling edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
12	HSTA12	Hardware start condition12: Timric port sampled to rising edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
11	HSTA11	Hardware start condition11: Timrib port up-sampled to falling edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
10	HSTA10	Hardware start condition10: Timrib port sampled to rising edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
9	HSTA9	Hardware start-up condition9: Timria port up-sampled to falling edge 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched

8	HSTA8	Hardware start-up condition8: sampling to rising edge on TIMTRIA port 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
7	HSTA7	Hardware start condition7: sampling to falling edge on CHxB port 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
6	HSTA6	Hardware start condition6: sampling to rising edge on CHxB port 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
5	HSTA5	Hardware start condition5: sampling to falling edge on CHxA port 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
4	HSTA4	Hardware start condition4: sampling to rising edge on CHxA port 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
3	HSTA3	Hardware start condition3: event trigger3 from AOS is valid 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
2	HSTA2	Hardware start condition2: event trigger2 from AOS is valid 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
1	HSTA1	Hardware start condition1: event trigger1 from AOS is valid 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched
0	HSTA0	Hardware start condition0: event trigger0 from AOS is valid 0: Hardware boot is invalid when conditions are matched 1: Hardware boot is valid when conditions are matched

12.3.15 Hardware Stop Event Select Register (TIMx_HSTPR)

Address offset:

0x078 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STOPS	Reserved														
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSTP 15	HSTA 14	HSTP 13	HSTP 12	HSTP 11	HSTP 10	HSTP 9	HSTP 8	HSTP 7	HSTP 6	HSTP 5	HSTP 4	HSTP 3	HSTP 2	HSTP 1	HSTP 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function
31	STOPS	Hardware Stop Enable 0: Hardware stop invalid 1: Hardware stop valid <i>Note: When hardware stop is valid, the setting of software stop is invalid</i>
30:16	Reserved	-
15	HSTP15	Hardware stop condition15: Timrid port sampled to falling edge 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
14	HSTP14	Hardware stop condition14: sampling to rising edge on TIMTRID port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
13	HSTP13	Hardware stop condition13: Timtric port up-sampled to falling edge 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
12	HSTP12	Hardware stop condition12: sampling to rising edge on TIMTRIC port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
11	HSTP11	Hardware stop condition11: Timtrib port sampled to falling edge 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
10	HSTP10	Hardware stop condition10: Timtrib port sampled to rising edge 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
9	HSTP9	Hardware stop condition9: sampling to falling edge on TIMTRIA port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched

8	HSTP8	Hardware stop condition8: sampling to rising edge on TIMTRIA port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
7	HSTP7	Hardware stop condition7: sampling to falling edge on CHxB port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
6	HSTP6	Hardware stop condition6: sampling to rising edge on CHxB port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
5	HSTP5	Hardware stop condition5: sampling to falling edge on CHxA port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
4	HSTP4	Hardware stop condition4: sampling to rising edge on CHxA port 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
3	HSTP3	Hardware stop condition3: event trigger3 from AOS is valid 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
2	HSTP2	Hardware stop condition2: event trigger2 from AOS is valid 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
1	HSTP1	Hardware stop condition1: event trigger1 from AOS is valid 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched
0	HSTP0	Hardware stop condition0: event trigger0 from AOS is valid 0: Hardware stop is invalid when conditions are matched 1: Hardware stop valid when conditions are matched

12.3.16 Hardware Clear Event Select Register (TIMx_HCELR)

Address offset:

0x07C Reset value:

0x0000 0000

31302928272625242322212019181716

CLEAR	Reserved														
R/W															

1514131211109876543210

HCEL 15	HCEL 14	HCEL 13	HCEL 12	HCEL 11	HCEL 10	HCEL 9	HCEL 8	HCEL 7	HCEL 6	HCEL 5	HCEL 4	HCEL 3	HCEL 2	HCEL 1	HCEL 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function
31	STARTS	Hardware Clear Enable 0: Hardware zeroing is invalid 1: Hardware clearing valid <i>Note: When hardware zeroing is valid, the software zeroing setting is invalid</i>
30:16	Reserved	-
15	HCEL15	Hardware clear condition15: Timrid port is sampled to the falling edge 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
14	HCEL14	Hardware clear condition14: Timrid port sampled to rising edge 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
13	HCEL13	Hardware clear condition13: Timtric port up-sampled to falling edge 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
12	HCEL12	Hardware clear condition12: Timtric port sampled to rising edge 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
11	HCEL11	Hardware clear condition11: Timtrib port is sampled to the falling edge 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
10	HCEL10	Hardware clear condition10: TIMTRIB port sampled to rising edge 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched

9	HCEL9	Hardware clear condition 9: sampling to falling edge on TIMTRIA port 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
---	-------	--

8	HCEL8	Hardware clear condition8: sampling to rising edge on TIMTRIA port 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
7	HCEL7	Hardware clear condition7: sampling to falling edge on CHxB port 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
6	HCEL6	Hardware clear condition6: sampling to rising edge on CHxA port 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
5	HCEL5	Hardware clear condition5: sampling to falling edge on CHxA port 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
4	HCEL4	Hardware clear condition4: sampling to rising edge on CHxA port 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
3	HCEL3	Hardware clear condition3: event trigger3 from AOS is valid 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
2	HCEL2	Hardware clear condition2: event trigger2 from AOS is valid 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
1	HCEL1	Hardware clear condition1: event trigger1 from AOS is valid 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched
0	HCEL0	Hardware clear condition0: event trigger0 from AOS is valid 0: Hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched

12.3.17 Hardware Capture A Event Select Register (TIMx_HCPAR)

Address offset:

0x080 Reset value:

0x0000 0000

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

| HCRA |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| R/W |

position	Marker	Function
31:16	Reserved	-
15	HCRA15	Hardware capture A condition15: Timrid port sampled to falling edge 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
14	HCRA14	Hardware capture A condition14: Timrid port sampled to rising edge 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
13	HCRA13	Hardware capture A condition13: sampling to falling edge on TIMTRIC port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
12	HCRA12	Hardware capture A condition12: sampling to rising edge on TIMTRIC port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
11	HCRA11	Hardware capture A condition11: TIMTRIB port sampled to falling edge 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
10	HCRA10	Hardware capture A condition10: TIMTRIB port sampled to rising edge 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
9	HCRA9	Hardware capture A condition9: sampling to falling edge on TIMTRIA port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
8	HCRA8	Hardware capture A condition8: sampling to rising edge on TIMTRIA port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched

7	HCPA7	Hardware capture A condition7: sampling to falling edge on CHxB port 0: Hardware capture A is invalid when the condition is matched
---	-------	--

		1: Hardware capture A is valid when the conditions are matched
6	HCPA6	Hardware capture A condition6: sampling to rising edge on CHxB port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
5	HCPA5	Hardware capture A condition5: sampling to falling edge on CHxA port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
4	HCPA4	Hardware capture A condition4: sampling to rising edge on CHxA port 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
3	HCPA3	Hardware capture A condition3: event trigger3 from AOS is valid 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
2	HCPA2	Hardware capture A condition2: event trigger2 from AOS is valid 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
1	HCPA1	Hardware capture A condition1: event trigger1 from AOS is valid 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched
0	HCPA0	Hardware capture A condition0: event trigger0 from AOS is valid 0: Hardware capture A is invalid when the condition is matched 1: Hardware capture A is valid when the conditions are matched

12.3.18 Hardware Capture B Event Select Register (TIMx_HCPBR)

Address offset:

0x084 Reset value:

0x0000 0000

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

HCPB 15	HCPB 14	HCPB 13	HCPB 12	HCPB 11	HCPB 10	HCPB 9	HCPB 8	HCPB 7	HCPB 6	HCPB 5	HCPB 4	HCPB 3	HCPB 2	HCPB 1	HCPB 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function
31:16	Reserved	-
15	HCPB15	Hardware capture B condition15: sampling to falling edge on TIMTRID port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
14	HCPB14	Hardware capture B condition14: sampling to rising edge on TIMTRID port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
13	HCPB13	Hardware capture B condition13: sampling to falling edge on TIMTRIC port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
12	HCPB12	Hardware capture B condition12: sampling to rising edge on TIMTRIC port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
11	HCPB11	Hardware capture B condition11: Timtrib port sampled to falling edge 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
10	HCPB10	Hardware capture B condition10: sampling to rising edge on TIMTRIB port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
9	HCPB9	Hardware capture B condition9: sampling to falling edge on TIMTRIA port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
8	HCPB8	Hardware capture B condition8: sampling to rising edge on TIMTRIA port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched

7	HCPB7	Hardware capture B condition7: sampling to falling edge on CHxB port 0: Hardware capture B is invalid when the condition is matched
---	-------	--

		1: Hardware capture B is valid when the conditions are matched
6	HCPB6	Hardware capture B condition6: sampling to rising edge on CHxB port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
5	HCPB5	Hardware capture B condition5: sampling to falling edge on CHxA port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
4	HCPB4	Hardware capture B condition4: sampling to rising edge on CHxA port 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
3	HCPB3	Hardware capture B condition3: event trigger3 from AOS is valid 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
2	HCPB2	Hardware capture B condition2: event trigger2 from AOS is valid 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
1	HCPB1	Hardware capture B condition1: event trigger1 from AOS is valid 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched
0	HCPB0	Hardware capture B condition0: event trigger0 from AOS is valid 0: Hardware capture B is invalid when the condition is matched 1: Hardware capture B is valid when the conditions are matched

12.3.19 Hardware recursive event selection register (TIMx_HCUPR)

Address offset:

0x088 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved														HCUP 19	HCUP 18	HCUP 17	HCUP 16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
HCUP 15	HCUP 14	HCUP 13	HCUP 12	HCUP 11	HCUP 10	HCUP 9	HCUP 8	HCUP 7	HCUP 6	HCUP 5	HCUP 4	HCUP 3	HCUP 2	HCUP 1	HCUP 0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

position	Marker	Function
31:20	Reserved	-
19	HCUP19	Hardware recurrence condition: event trigger3 from AOS is valid 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
18	HCUP18	Hardware recurrence condition: event trigger2 from AOS is valid 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
17	HCUP17	Hardware recurrence condition: event trigger1 from AOS is valid 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
16	HCUP16	Hardware recurrence condition: event trigger0 from AOS is valid 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
15	HCUP15	Hardware recurrence condition: TIMTRID port is sampled to the falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
14	HCUP14	Hardware recurrence condition: TIMTRID port sampled to rising edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
13	HCUP13	Hardware recurrence condition: TIMTRIC port up-sampling to falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched

12	HCUP12	Hardware recurrence condition: Timtric port sampled to rising edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
----	--------	--

11	HCUP11	Hardware recurrence condition: TIMTRIB port up-sampling to falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
10	HCUP10	Hardware recurrence condition: TIMTRIB port sampled to rising edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
9	HCUP9	Hardware recurrence condition: TIMTRIA port is sampled to the falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
8	HCUP8	Hardware recurrence condition: TIMTRIA port sampled to rising edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
7	HCUP7	Hardware recurrence condition: CHxA port is sampled to falling edge when CHxB port is high 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
6	HCUP6	Hardware recurrence condition: CHxA port is sampled to rising edge when CHxB port is high 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
5	HCUP5	Hardware recurrence condition: when CHxB port is low, CHxA port is upsampled to falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
4	HCUP4	Hardware recurrence condition: CHxA port is sampled to rising edge when CHxB port is low 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
3	HCUP3	Hardware recurrence condition: when CHxA port is high, CHxB port is upsampled to falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
2	HCUP2	Hardware recurrence condition: when CHxA port is high, CHxB port is upsampled to rising edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
1	HCUP1	Hardware recurrence condition: when CHxA port is low, CHxB port is upsampled to falling edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched
0	HCUP0	Hardware recurrence condition: when CHxA port is low, CHxB port is upsampled to rising edge 0: Hardware recursion is invalid when conditions are matched 1: Hardware recursion is valid when conditions are matched

12.3.20 Hardware decrement event selection register (TIMx_HCDOR)

Address offset:

0x08C Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												HCD O	HCD O	HCD O	HCD O
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCD O															
R/W															

position	Marker	Function
31:20	Reserved	-
19	HCDO19	Hardware decrement condition: event trigger3 from AOS is valid 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
18	HCDO18	Hardware decrement condition: event trigger2 from AOS is valid 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
17	HCDO17	Hardware decrement condition: event trigger1 from AOS is valid 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
16	HCDO16	Hardware decrement condition: event trigger0 from AOS is valid 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
15	HCDO15	Hardware decrement condition: TIMTRID port up-sampled to falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
14	HCDO14	Hardware decrement condition: TIMTRID port is sampled to the rising edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
13	HCDO13	Hardware decrement condition: TIMTRIC port up-sampled to falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
12	HCDO12	Hardware decrement condition: sampling to rising edge on TIMTRIC port 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched

11	HCDO11	Hardware decrement condition: TIMTRIB port up-sampled to falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
10	HCDO10	Hardware decrement condition: TIMTRIB port is sampled to the rising edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
9	HCDO9	Hardware decrement condition: TIMTRIA port up-sampled to falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
8	HCDO8	Hardware decrement condition: sampling to rising edge on TIMTRIA port 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
7	HCDO7	Hardware decrement condition: when CHxB port is high, CHxA port is upsampled to the falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
6	HCDO6	Hardware decrement condition: when CHxB port is high, CHxA port is upsampled to rising edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
5	HCDO5	Hardware decrement condition: when CHxB port is low, CHxA port is upsampled to the falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
4	HCDO4	Hardware decrement condition: when CHxB port is low, CHxA port is upsampled to rising edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
3	HCDO3	Hardware decrement condition: when CHxA port is high, CHxB port is upsampled to the falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
2	HCDO2	Hardware decrement condition: when CHxA port is high, CHxB port is upsampled to rising edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
1	HCDO1	Hardware decrement condition: when CHxA port is low, CHxB port is upsampled to the falling edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched
0	HCDO0	Hardware decrement condition: when CHxA port is low, CHxB port is upsampled to rising edge 0: Hardware decrement is invalid when conditions are matched 1: Hardware decrement is valid when conditions are matched

12.3.21 Software Synchronization Start Register (TIMx_SSTAR)

Address offset:

0x3F4 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
SSTA2 SSTA1 SSTA0 R/W R/W R/W															

position	Marker	Function
31:3	Reserved	
2	SSTA2	Timer6 Software Launch 0: Software startup is invalid 1: Software startup enable
1	SSTA1	Timer5 Software Launch 0: Software startup is invalid 1: Software startup enable
0	SSTA0	Timer4 Software Launch 0: Software startup is invalid 1: Software startup enable

12.3.22 Software Synchronization Stop Register (**TIMx_SSTPR**)

Address offset:

0x3F8 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SSTP2	SSTP1	SSTP0	
												R/W	R/W	R/W	

position	Marker	Function
31:3	Reserved	
2	SSTP2	Timer6 Software Stop 0: The software is not valid 1: Software stop enable
1	SSTP1	Timer5 Software Stop 0: The software is not valid 1: Software stop enable
0	SSTP0	Timer4 Software Stop 0: The software is not valid 1: Software stop enable

12.3.23 Software Synchronous Clear Register (TIMx_SCLRR)

Address offset:

0x3FC Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
Reserved																					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Reserved																					
<table border="1" style="margin-left: auto; margin-right: 0;"> <tr> <td>SCLR2</td><td>SCLR1</td><td>SCLR0</td></tr> <tr> <td>R/W</td><td>R/W</td><td>R/W</td></tr> </table>																SCLR2	SCLR1	SCLR0	R/W	R/W	R/W
SCLR2	SCLR1	SCLR0																			
R/W	R/W	R/W																			

position	Marker	Function
31:3	Reserved	
2	SCLR2	Timer6 Software Zeroing 0: Software zeroing is invalid 1: Software zero enable
1	SCLR1	Timer5 Software Zeroing 0: Software zeroing is invalid 1: Software zero enable
0	SCLR0	Timer4 Software Zeroing 0: Software zeroing is invalid 1: Software zero enable

12.3.24 Interrupt Flag Register (TIMx_IFR)

Address offset:

0x100 Reset value:

0x0000 0000

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

SAMHF	SAMLF	Reserved	DTEF	UDFF	OVFF	Reserved	CMDF	CMCF	CMBF	CMAF
RO	RO		RO	RO	RO		RO	RO	RO	RO

position	Marker	Place Name	Function
31:16	Reserved	-	
15	SAMHF	CHxA/B port high status interrupt flag 0: No simultaneous high levels on CHxA and CHxB ports 1: High level on both CHxA and CHxB ports	
14	SAMLF	CHxA/B port low status interrupt flag 0: No simultaneous low levels on CHxA and CHxB ports 1: Low level on both CHxA and CHxB ports	
13:9	Reserved	-	
8	DTEF	Dead time error interrupt flag 0: Dead time error did not occur;1 : Dead time error occurred	
7	UDFF	Underflow match interrupt flag 0: No sawtooth wave underflow or delta wave count to valley 1: Sawtooth wave overflow or triangular wave count to the valley point occurs	
6	OVFF	Overflow match interrupt flag 0: No sawtooth wave overflow or delta wave counting to the peak 1: Sawtooth wave overflow or triangular wave counting to the peak point occurs	
5:4	Reserved	-	
3	CMDF	Counting Match D interrupt flags 0: The value of GCMDR register is not equal to the count value;1 : The value of GCMDR register is equal to the count value	
2	CMCF	Counting Matching C Interrupt Flags 0: The value of GCMCR register is not equal to the count value;1 : The value of GCMCR register is equal to the count value	

1	CMBF	<p>Counting Match B interrupt flags</p> <p>0: The value of GCMBR register is not equal to the count value, and no CHxB capture completion action occurs.</p> <p>1: The value of the GCMBR register is equal to the count value, or the CHxB capture completion action occurs.</p>
0	CMAF	<p>Counting Match A Interrupt Flags</p> <p>0: The value of GCMAR register is not equal to the count value, and no CHxA capture completion action occurs.</p>

		1: The value of the GCMAR register is equal to the count value, or the CHxA capture completion action occurs.
--	--	---

12.3.25 Interrupt Flag Clear Register (TIMx_ICLR)

Address offset:

0x104 Reset value:

0x0000 0000

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

SAMHC	SAMLC	Reserved -	DTEC	UDFC	OVFC	Reserved	CMDC	CMCC	CMBC	CMAC
W0	W0		W0	W0	W0		W0	W0	W0	W0

position	Marker	Function
31:16	Reserved	-
15	SAMHC	CHxA/B port high status interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
14	SAMLC	CHxA/B port low status interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
13:9	Reserved	-
8	DTEC	Dead time error interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
7	UDFC	Underflow match interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
6	OVFC	Overflow match interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
5:4	Reserved	-
3	CMDC	Count match D interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
2	CMCC	Count match C interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
1	CMBC	Count match B interrupt flag clear, write 1invalid, write 0clear corresponding interrupt
0	CMAC	Count match A interrupt flag clear, write 1invalid, write 0clear corresponding interrupt

12.3.26 Spread Spectrum and Interrupt Trigger Selection (TIMx_CR)

Address offset:

0x108 Reset value:

0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DITENS	DITENB	DITNA	UDFE	OVFE	Reserved		CMDE	CMCE	CMBE	CMAE
					R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W

position	Marker	Function
31:11	Reserved	-
10	DITENS	PWM Spread Spectrum Count Selection 0: select lower overflow, 1 : select upper overflow
9	DITENB	PWM Channel B Spread Spectrum Enable 0: Enable is disabled, 1: Enable is enabled, and the output delay of the PWM is changed every cycle
9	DITENA	PWM Channel A Spread Spectrum Enable 0: Enable is disabled, 1: Enable is enabled, and the output delay of the PWM is changed every cycle
7	UDFE	Underflow match enable triggers ADC 0: Enable is disabled, 1: Enable is enabled, this interrupt can control ADC/AOS_i_tirg
6	OVFE	Overflow match enable triggers ADC 0: Enable is disabled, 1: Enable is enabled, this interrupt can control ADC/AOS_i_tirg
5:4	Reserved	-
3	CMDE	Count Match D Enable Trigger ADC 0: Enable is disabled, 1: Enable is enabled, this interrupt can control ADC/AOS_i_tirg
2	CMCE	Count Match C Enable Trigger ADC 0: Enable is disabled, 1: Enable is enabled, this interrupt can control ADC/AOS_i_tirg
1	CMBE	Count Match B Enable Trigger ADC 0: Enable is disabled, 1: Enable is enabled, this interrupt can control ADC/AOS_i_tirg
0	CMAE	Count Match A Enable Trigger ADC 0: Enable is disabled, 1: Enable is enabled, this interrupt can control ADC/AOS_i_tirg

12.3.27 AOS Selection Control Register (TIMx_AOSSR)

Address offset:

0x110 Reset value:

0x0000 0000

Timer4/5/6 use the same entity register, after any one timer is changed, the value in the other two timers will be changed at the same time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	SMH2	SMH1	SMH0	SML2	SML1	SML0	SOFTB	Reserved	BFILTEN	BFILTS	FSAME	F BRAKE	R/W	R/W	R	R
	R/W		R/W	R/W												

position	Marker	Function
31:14	Reserved	-
13	SMH2	Channel2 same height selection 0: selection is invalid, 1: selection is valid, when the same high appears AOS_i_odis[1]
12	SMH1	Channel1 same height selection 0: selection is invalid, 1: selection is valid, when the same high appears AOS_i_odis[1]
11	SMH0	Channel0 same height selection 0: selection is invalid, 1: selection is valid, when the same high appears AOS_i_odis[1]
10	SML2	Channel2 same low selection 0: selection is invalid, 1: selection is valid, appears with the same low AOS_i_odis[1]
9	SML1	Channel1 same low selection 0: selection is invalid, 1: selection is valid, appears with the same low AOS_i_odis[1]
8	SML0	Channel0 same low selection 0: selection is invalid, 1: selection is valid, appears with the same low AOS_i_odis[1]
7	SOFTBK	Software brakes: write to 1 implement software brakes
13	Reserved	-
4	BFILTEN	Port brake filter enable
3:2	BFILTS	Port brake filter clock selection
1	FSAME	Same high and low brake symbol, read only
0	FBRAKE	Port brake flag, read-only

12.3.28 AOS Select Control Register Flag Clear (TIMx_AOSCL)

Address offset:

0x114 Reset value:

0x0000 0000

Timer4/5/6 use the same entity register, after any one timer is changed, the value in the other two timers will be changed at the same time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FSAM	FBRAKE
														R	R

position	Marker	Function
31:2	Reserved	-
1	FSAME	Same high and same low brake flag clear, write 0clear, write 1invalid, read constant as1
0	FBRAKE	Port brake flag clear, write 0clear, write 1invalid, read constant for1

12.3.29 Port Brake Control Register (TIMx_PTBKs)

Address offset:

0x118 Reset value:

0x0000 0000

Timer4/5/6 use the same entity register, after any one timer is changed, the value in the other two timers will be changed at the same time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function
31:16	Reserved	-
15	EN15	P36 Brake Port Enable: 1Select, 0Invalid
14	EN14	P35 Brake Port Enable: 1Select, 0Invalid
13	EN13	P34 Brake Port Enable: 1Select, 0Invalid
12	EN12	P33 Brake Port Enable: 1Select, 0Invalid
11	EN11	P32 Brake Port Enable: 1Select, 0Invalid
10	EN10	P31 Brake port enable: 1Select, 0 disabled
9	EN9	P27 Brake Port Enable: 1Select, 0Invalid
8	EN8	P26 Brake Port Enable: 1Select, 0Invalid
7	EN7	P26 Brake Port Enable: 1Select, 0Invalid
6	EN6	P24 Brake port enable: 1Select, 0 disabled
5	EN5	P23 Brake Port Enable: 1Select, 0Invalid
4	EN4	P15 Brake Port Enable: 1Select, 0Invalid
3	EN3	P14 Brake Port Enable: 1Select, 0Invalid
2	EN2	P03 Brake Port Enable: 1Select, 0Invalid
1	EN1	P02 Brake Port Enable: 1Select, 0Invalid
0	EN0	P01 Brake Port Enable: 1Select, 0Invalid

12.3.30 Port Trigger Control Register (TIMx_TTRIG)

Address offset:

0x11C Reset value:

0x0000 0000

Timer4/5/6 use the same entity register, after any one timer is changed, the value in the other two timers will be changed at the same time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGDS				TRIGCS				TRIGBS				TRIGAS			
R/W				R/W				R/W				R/W			

position	Mark er	Function
31:16	Reserved	-
15:12	TRIGDS	TIMx triggers D-port selection
11:8	TRIGCS	TIMx triggers C-port selection
7:4	TRIGBS	TIMx triggers B-port selection
3:0	TRIGAS	TIMx triggers A-port selection

The control signals and ports are selected as follows

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
P01	P02	P03	P15	P14	P23	P24	P25	P26	P27	P31	P32	P33	P34	P35	P36

12.3.31 AOS Trigger Control Register (**TIMx_ITRIG**)

Address offset:

0x120 Reset value:

0x0000 0000

Timer4/5/6 use the same entity register, after any one timer is changed, the value in the other two timers will be changed at the same time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAOS3S				IAOS2S				IAOS1S				IAOS0S			
R/W				R/W				R/W				R/W			

position	Marker	Function
31:16	Reserved	-
15:12	IAOS3S	TIMx AOS3 Trigger Source Selection
11:8	IAOS2S	TIMx AOS2 Trigger Source Selection
7:4	IAOS1S	TIMx AOS1 Trigger Source Selection
3:0	IAOS0S	TIMx AOS0 Trigger Source Selection

The control signal (IAOSxS) and the interrupt source are selected as follows (x=0,1,2,3)

0000	0001	0010	0011	0100	0101	0110	0111
TIM0_INT	TIM1_INT	TIM2_INT	LPTIMER_INT	TIM4_INTS	TIM5_INTS	TIM6_INTS	UART0_INT
1000	1001	1010	1011	1100	1101	1110	1111
UART1_INT	LPUART_INT	VC1_INT	VC2_INT	RTC_INT	PCA_INT	SPI_INT	ADC_INT

12.3.32 Port Brake Polarity Control Register (TIMx_PTBKPx)

Address offset:

0x124 Reset value:

0x0000 0000

Timer4/5/6 use the same entity register, after any one timer is changed, the value in the other two timers will be changed at the same time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL15	POL14	POL13	POL12	POL11	POL10	POL9	POL8	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function
31:16	Reserved	-
15	POL15	P36 Brake port polarity selection:1 active low,0 active high
14	POL14	P35 Brake port polarity selection:1 active low,0 active high
13	POL13	P34 Brake port polarity selection:1 active low,0 active high
12	POL12	P33 Brake port polarity selection:1 active low,0 active high
11	POL11	P32 Brake port polarity selection:1 active low,0 active high
10	POL10	P31 Brake port polarity selection:1 active low,0 active high
9	POL9	P27 Brake port polarity selection:1 active low,0 active high
8	POL8	P26 Brake port polarity selection:1 active low,0 active high
7	POL7	P26 Brake port polarity selection:1 active low,0 active high
6	POL6	P24 Brake port polarity selection:1 active low,0 active high
5	POL5	P23 Brake port polarity selection:1 active low,0 active high
4	POL4	P15 Brake port polarity selection:1 active low,0 active high
3	POL3	P14 Brake port polarity selection:1 active low,0 active high
2	POL2	P03 Brake port polarity selection:1 active low,0 active high
1	POL1	P02 Brake port polarity selection:1 active low,0 active high
0	POL0	P01 Brake port polarity selection:1 active low,0 active high

13 Real Time Clock (RTC)

13.1 Real Time Clock Introduction

The real-time clock/calendar provides information in seconds, minutes, hours, days, weeks, months, and years, and the number of days per month and days in leap years can be automatically adjusted. Clock operation can be determined by the AM/PM register bits in either or 24 hour 12 format. Table 14-

1 The basic characteristics are shown.

Clock Source	Off-chip low-speed crystal XTL (32.768kHz) On-chip low-speed oscillator RCL (32kHz, 1% accuracy) Off-chip high-speed crystal XTH
Basic Functions	Calculate seconds, minutes, hours, days, weeks, months and years from 00 to 99 years
	Automatic leap year adjustment available
	Configurable to 24 or 12 hourly format
	Programmable start or stop
	With alarm clock function
	High precision 1Hz square wave signal output
Interruptions	With periodic interruptions
	With alarm clock interruption

Table 13-1 Basic characteristics of the RTC

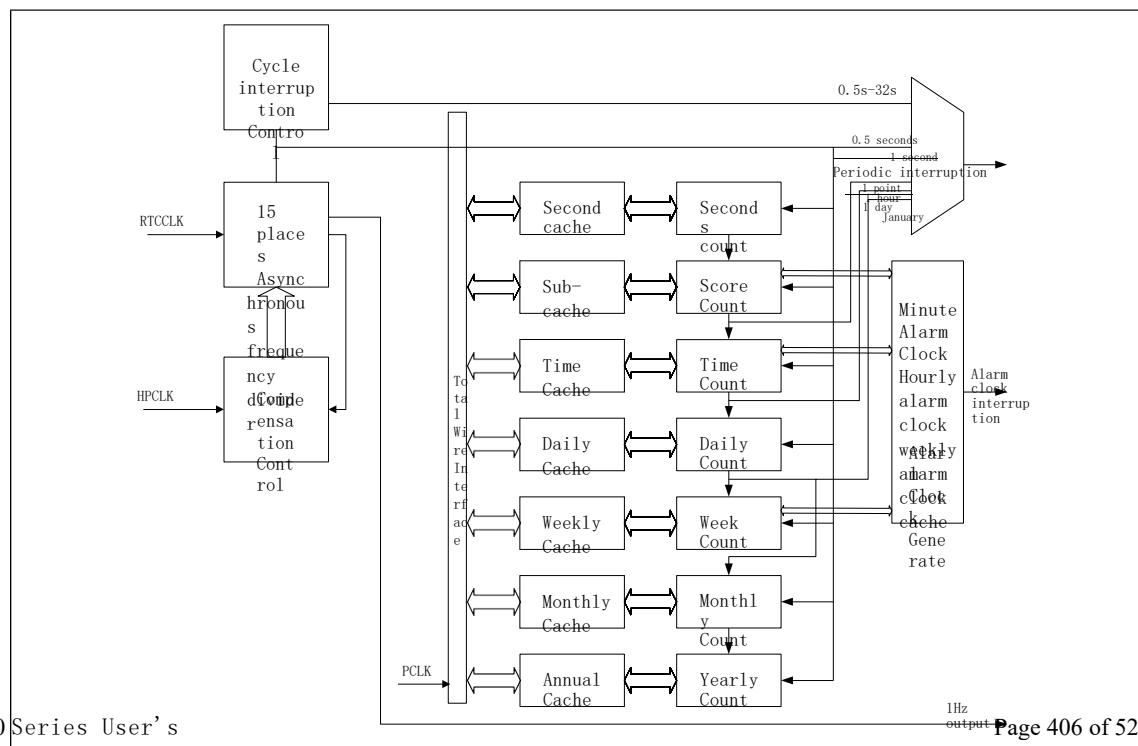


Figure 13-1 RTC Block Diagram

13.2 Real Time Clock Function Description

The clock source of the real-time clock can be configured as external low-speed crystal, external high-speed crystal, and internal low-speed RC; the default is to use the external low-speed crystal. Control registers CR0, CR1 and COMPEN are controlled by power-on reset only, other reset sources cannot reset these three control registers. Other data registers have variable power-on status, need to be initialized after power-on, and are not affected by any reset.

All date and time values written and read by the software are in BCD code and do not require hexadecimal conversion to decimal. Any invalid date and time will not be written, such as day32, 25hour, 70second, B month, etc.

13.2.1 Power-up settings

The RTC is reset once after power on. Without powering down the system, various external reset requests cannot reset the RTC and the RTC will remain in the counting state. After power-on, the RTC is started after setting the calendar initial value, alarm setting, error compensation, interrupt, etc.

13.2.2 RTC Counting Start Setting

1. Set CR0.START=0, counting stops.
2. Sets CR0.AMPM and CR0.PRDS, and CR0.PRDX sets the timing and interrupt period.
3. Set CR1.CKSEL to select the timing clock for the RTC.
4. Setting the calendar count register for seconds, minutes, hours, weeks, days, months and years.
5. (b) When clock error compensation is required, set the count clock error compensation register COMPEN.
6. Clear the interrupt flag bits CR1.ALMF, CR1.PRDF and enable the

interrupt.

7. Set CR0.START=1, counting starts.

13.2.3 System low-power mode switching

If the system switches to low power mode immediately after the RTC count starts, perform any of the following checks before switching modes.

The control register is in the system control register SYSCTRL1.RTC_LPW

1. After CR0.START=1 is set, the mode is switched after 2 more than one RTC count clock.
2. After setting CR0.START=1, set CR1.WAIT=1 and query CR1.WAITF=1.
Set again

CR1.WAIT=0, query CR1.WAITF=0 and then switch the mode.

In RTC low-power mode, the RTC registers cannot be read or written. In low-power mode, the RTC consumes less current.

There is no waiting while the RTC is running to switch to low-power mode.

13.2.4 Read out the count register

There are three ways to read the count register.

- Method1 : Any time reading method 1
 1. Set CR1.WAIT=1 to stop calendar register counting and enter read/write mode.
 2. Query until CR1.WAITF=1.
 3. Reading of seconds, minutes, hours, week, day, month and year count register values.
 4. Set CR1.WAIT=0, counter counts.
 5. Query until CR1.WAITF=0.
- Method2 : Any time reading method 2
 1. Reads the minute, hour, week, day, month and year count register values.
 2. Reading of the seconds counter register value.
 3. Reading out the seconds counter register value again.
 4. Determine whether the readout values of two seconds are the same, different restart from the first step, the same readout ends.
- Method3 : Interrupt reading method

Reads the seconds, minutes, hours, week, day, month, and year count register values in the RTC cycle interrupt service. This is because there is at least 0.5s time between the interrupt occurring and the next data change.

13.2.5 Write to count register

1. Set CR1.WAIT=1 to stop calendar register counting and enter read/write mode.
2. Query until CR1.WAITF=1.
3. Write the seconds, minutes, hours, week, day, month, year count register

values.

4. Set CR1.WAIT=0 and the counter will start counting again. Note that all write operations must be completed within seconds1.
5. Query until CR1.WAITF=0.

There is no need to wait for WAIT to write seconds, minutes, hours, weeks, days, months, and years count registers in RTC unboot mode.

Notes.

- Changing the seconds register in counting mode will reset the seconds count.
Writing the minutes, hours, weeks, days, months, and years count register values will not affect the RTC count.

13.2.6 Alarm settings

1. Set CR1.ALMEN=0, alarm clock disabled.
2. Set CR1.ALMIE=1, alarm interrupt permit.
3. minute alarm ALMMIN, hour alarm ALMHOUR, week alarm ALMWEEK settings.
4. Set CR1.ALMEN=1, alarm clock permit.
5. Waiting for an interruption to occur.
6. Since the alarm interrupt and the fixed-cycle interrupt share the interrupt request signal, the alarm interrupt is processed when CR1.ALMF=1; otherwise, the fixed-cycle interrupt is processed.

13.2.7 1Hz Output

The RTC can select to output 31Hz clocks of normal, higher and higher accuracy. The higher precision 1Hz clock is output when the clock error compensation function is active, and the higher precision 1Hz clock is output when the PCLK of different frequencies is used. The system control register needs to be configured according to the PCLK frequency, where

- The 1Hz output setting for general accuracy is as follows: (no clock compensation)
 1. Set CR0.START=0, counting stops.
 2. RTC output pin setting.
 3. CR0.1HZOE=1, clock output permission.
 4. Set CR0.START=1, counting starts.
 5. Waiting for more than one 2count cycle.
 6. 1Hz output start.

- The higher precision 1Hz output settings are as follows: (low speed compensation)
 1. Set CR0.START=0, counting stops.
 2. RTC output pin setting.
 3. CR0.1HZOE=1, clock output permission.
 4. Clock error compensation register COMPEN.C R Compensation number setting.

5. clock error compensation register COMPEN.EN=1, error compensation is valid.
 6. Set CR0.START=1, counting starts.
 7. Waiting for more than one 2count cycle.
 8. 1Hz output start.
- When the high precision 1Hz output is required, it is necessary to provide the RTC with a higher precision output based on the 4M,6M,8M,12M,16M,20M,24M,32MHz high speed PCLK clocks with the following output settings.
 1. Set CR0.START=0, counting stops.
 2. RTC output pin setting.
 3. CR0.1HZOE=1, clock output permission.
 4. CR0.1HZSEL=1, to select the output high precision 1Hz clock.
 5. Configure high-speed clock compensation clock SYSCTRL1.RTC_FREQ_ADJUST
 6. Clock error compensation register COMPEN.CR[8:0] Compensation number setting.
 7. clock error compensation register COMPEN.EN=1, accuracy compensation is valid.
 8. Set CR0.START=1, counting starts.
 9. Waiting for more than one 2count cycle.
 10. 1Hz output start.

13.2.8 Clock error compensation

Since there is an error in the external crystal, it is necessary to compensate for this error when high precision counting results are required. There are two types of compensation methods: the first one, error compensation based on its own clock; the second one, error compensation based on high-speed clock.

Principle and calculation of error compensation based on own clock.

Since the counter uses a 32.768KHz clock, if it is necessary to compensate the accuracy per second, it can only be compensated by the integer period of

32.768KHz, so the minimum unit of compensation per second is $(1/32768)^6 \times 10 = 30.5\text{ppm}$, which cannot meet the requirement of high accuracy.

To achieve higher accuracy clock compensation with a 32.768KHz counter clock, the algorithm needs to be adjusted to expand the maximum compensation period by a factor of 32 one. Then, if the minimum compensation unit is 30.5ppm, the average compensation unit per second becomes $30.5\text{ppm}/32=0.96\text{ppm}$, which satisfies the requirement of higher accuracy clock compensation. And

The compensation occurs in a relatively uniform range per second32.

Therefore, a bit5 decimal setting is introduced in this register.

Example1

When the 1Hz clock is output directly in the default state, the compensation target value is calculated by measuring the accuracy of this clock. Assuming that the actual measured value is 0.9999888Hz, the

$$\begin{aligned} \text{Actual oscillation frequency} &= 32768 \times \\ &0.999988832767.63 \end{aligned}$$

Compensation target value = (actual oscillation frequency - target frequency)/target frequency $\times 10^6$

$$\begin{aligned} &= (32767.96 - 32768) / 32768 \times 10^6 \\ &= -11.29 \text{ ppm} \end{aligned}$$

According to

$$\text{CR}[8 : 0] = \frac{\text{Compensation target value } [\text{ppm}]^{15}}{10^6}$$

0001.00000B
Take the
2's
complement

If the compensation target value is -11.29ppm, the corresponding register value is calculated as follows: CR[8:0] = (-11.29 $\times 2^{15}/10^6$) taken as the 2complement + 0001.00000B

$$\begin{aligned} &= (-0.37) \quad \text{taken as the 2complement of + 0001.00000B} \\ &= 1111.10101B + 0001.00000B \\ &= 0000.10101B \end{aligned}$$

Principle and calculation of error compensation based on a high-speed 24MHz clock.

This method is calculated in the same way as the error compensation based on its own clock. With the introduction of the 4M-32MHz high-speed clock, the 1/32768 second error that would otherwise be accumulated over a maximum of seconds32 can be spread out over each second1, and a minimum of 0.96ppm (2324MHz clock period) can be compensated for each second1, resulting in

an average high precision 1Hz per second clock output.

13.3 RTC interrupt

The RTC supports two interrupt types. Alarm interrupt, and fixed-cycle interrupt. The alarm interrupt and the fixed-cycle interrupt share a common interrupt signal.

13.3.1 RTC alarm interrupt

When CR1.ALMIE=1, if the current calendar time is the same as the minute alarm register (ALMMIN) the hour alarm register (ALMHOUR) and weekly alarm register (ALMEEK) are equal, the alarm interrupt is triggered.

13.3.2 RTC Cycle Break

When ALMIE=1 in the control register 1(CR1), the selected cycle occurs and triggers the fixed-cycle wake-up interrupt, which is distinguished by the flag register bits since the alarm and fixed-cycle share the same interrupt.

13.4 RTC Register Description

Base address 0X40001400

Register	Offset Address	Description
RTC_CR0	0X000	Control register0
RTC_CR1	0X004	Control register1
RTC_SEC	0X008	Second counter register
RTC_MIN	0X00C	Sub-count register
RTC_HOUR	0X010	Time counter register
RTC_WEEK	0X014	Week Count Register
RTC_DAY	0X018	Day Count Register
RTC_MON	0X01C	Monthly Count Register
RTC_YEAR	0X020	Year Count Register
RTC_ALMMIN	0X024	Minute Alarm Register
RTC_ALMHOUR	0X028	Time alarm clock register
RTC_ALMWEEK	0X02C	Weekly alarm clock register
RTC_COMPEN	0X030	Clock error compensation register

Table 13-2RTCRegister List

13.4.1 Control register0 (RTC_CR0)

*Only power-on reset of
 this register is valid for
 the address offset: 0x000
 Reset value 0x0000 0000

31302928272625242322212019181716	Reserved
----------------------------------	----------

151413:7654382:0

Res.	PRDSEL	PRDX	START	HZ1SEL	HZ1OE	Res.	AMPM	PRDS
	R/W	R/W	R/W	R/W	R/W		R/W	R/W

position	Symbols	Function Description
31:15	Reserved	-
14	PRDSEL	0: Periodic interrupt interval set using PRDS 1: Cycle interval set using PRDX
13:8	PRDX	Set the time interval for generating a periodic interrupt. The settable range is from 0.5seconds to 32 seconds, in steps of 0.5seconds. 000000: 0.5seconds 000001: 1Seconds 111110: 31.5Seconds 111111: 32seconds
7	START	0: Stop RTC counters 1: Enables RTC counters
6	1HZSEL	0: Normal accuracy 1Hz output 1: High precision 1Hz output
5	1HZOE	0: 1Hz output disabled 1: Enable 1Hz output
4	Reserved	-
3	AMPM	0: 12Hourly 1: 24Hourly

2:0	PRDS	<p>Set the time interval for generating interrupts.</p> <p>000: No cycle interruptions</p> <p>001: 0.5seconds</p> <p>010: 1seconds</p> <p>011: 1Minutes</p> <p>100: 1Hours</p> <p>101: 1Sky</p> <p>11x: 1Month</p> <p>Note: If you need to write to change the interval of the cycle interrupt at START=1, proceed as follows.</p>
-----	------	--

		<p>step1, turn off the RTC interrupt in the NVIC; step2, change the interval of the periodic interrupt; step3, clear the RTC interrupt flag. step4, enable RTC interrupt.</p>
--	--	---

13.4.2 Control register1 (RTC_CR1)

*Only power-on reset of

this register is valid

address offset: 0x004

Reset value 0X00000000

31302928272625242322212019181716	Reserved
----------------------------------	----------

Reserved		CKSEL	ALMEN	ALMIE	Res.	ALMF	PRDF	Res.	WAITF	WAIT
		R/W	R/W	R/W		RO	RO		R/W	R/W

position	Symbols	Function Description
31:11	Reserved	-
10:8	CKSEL	RTC clock selection 00x: XTL 32.768k 01x: RCL 32k 100: XTH/128(select this item when the crystal is 4M) 101: XTH/256(select this when the crystal is 8M) 110: XTH/512(select this when the crystal is 16M) 111: XTH/1024(select this when the crystal is 32M)
7	ALMEN	0: No alarm clock 1: Enabling alarm clock Note: Enabled if START=1 calendar count is in progress and ALMIE=1 interrupt is allowed When ALMEN is enabled, please turn off the system interrupt to prevent misoperation. Please clear the ALMF flag bit after enabling.
6	ALMIE	0: Disable alarm clock interruption 1: Enables alarm interruption
5	Reserved	-
4	ALMF	0: No alarm clock interruptions 1: Alarm clock interruption has occurred Note: This bit is only valid when ALMEN=1. 32.768KHz One clock post1 when alarm is matched. Clear flag 0on write, writel is invalid.
3	PRDF	0: No cycle interruptions 1: Cycle interruption has occurred Note: This location1 after a cycle interrupt occurs. The flag is cleared 0on write and the write is 1invalid.

2	Reserved	-
1	WAITF	<p>0: Non-write/read status 1: Write/read status</p> <p>Note: The WAIT bit sets the validity flag. Check if this bit is 1"" before writing/reading out. count</p>

		During the counting process, wait for the WAIT bit0 to clear "" after the write is completed before clearing 0"".
0	WAIT	<p>0: Normal counting mode 1: Write/read mode</p> <p>Note: Please set this position1 "" when writing/reading out. Since the counter is counting continuously, please complete the write/read operation and clear0 this bit "" within 1seconds.</p>

13.4.3 Second Count Register (RTC_SEC)

Address Offset:

0x008 Reset Value:

Variable

31:0	31302928272625242322212019181716	Reserved
------	----------------------------------	----------

1514131211109876543210

Reserved		SECH	SECL
		R/W	R/W

position	Symbols	Function Description
31:7	Reserved	-
6:4	SECH	Seconds count tens of values
3:0	SECL	Second count single digit value

Indicates 0-59 seconds, using decimal counting. Please write the BCD code of decimal 0-59, and the written value will be ignored when writing the wrong value.

13.4.4 Sub-count register (RTC_MIN)

Address Offset:

0x00C Reset Value:

Variable

31:0	31302928272625242322212019181716	Reserved
------	----------------------------------	----------

1514131211109876543210

Reserved		MINH	MINL
		R/W	R/W

position	Symbols	Function Description
31:7	Reserved	-
6:4	MINH	Subtotal tens of values
3:0	MINL	Single digit value of score count

Indicates 0-59 points, using decimal counting. Please write the BCD code of

decimal 0-59, and the written value will be ignored when you write the wrong value.

13.4.5 Time Count Register (RTC_HOUR)

Address Offset:

0x010 Reset Value:

Variable

Reserved
31302928272625242322212019181716

1514131211109876543210

Reserved	HOURH	HOURL
	R/W	R/W

position	Symbols	Function Description
31:6	Reserved	-
5:4	HOURH	Timer tens values
3:0	HOURL	Time count single digit value

24 In hourly time system, it means 0-23 hours. 12 In hourly time, b5=0 means AM, then 01:12 means AM; b5=1 means PM, then 21:32 means PM.

Please set the BCD code of 0:23 or 01:12,21:32 in the correct decimal according to the value controlled as AM/PM. Writing values out of range will be ignored.

Refer to the following table for specific time indications.

24Hourly hourly system	AMPM=1	12Hourly hourly system	AMPM=0
Time	Register representation	Time	Register representation
00Time	00H	AM 12time	12H
01Time	01H	AM 01time	01H
02Time	02H	AM 02time	02H
03Time	03H	AM 03time	03H
04Time	04H	AM 04time	04H
05Time	05H	AM 05time	05H
06Time	06H	AM 06time	06H
07Time	07H	AM 07time	07H
08Time	08H	AM 08time	08H
09Time	09H	AM 09time	09H
10Time	10H	AM 10time	10H
11Time	11H	AM 11time	11H
12Time	12H	PM 12:00 pm	32H
13Time	13H	PM 01 hour	21H
14Time	14H	PM 02 hours	22H
15Time	15H	PM 03 hours	23H
16Time	16H	PM 04 hours	24H
17Time	17H	PM 05 hours	25H
18Time	18H	PM 06 hours	26H
19Time	19H	PM 07 hours	27H
20Time	20H	PM 0800 hours	28H
21Time	21H	PM 0900 hours	29H
22Time	22H	PM 10 when	30H
23Time	23H	PM 11:00 pm	31H

13.4.6 Day Count Register (RTC_DAY)

Address Offset:

0x018 Reset Value:

Variable

Reserved
31302928272625242322212019181716

1514131211109876543210

Reserved	DAYH	DAYL
	R/W	R/W

position	Symbols	Function Description
31:6	Reserved	-
5:4	DAYH	Daily counting tens of values
3:0	DAYL	Daily count single digit value

The decimal representation of 1:31 days automatically calculates leap years and months. This is expressed as follows.

Month	Day count representation
2Months (ordinary year)	01:28
2Month (leap year)	01:29
4, 6, 9, 11month	01:30
1357810、、、、、、12month	01:31

13.4.7 Week Count Register (RTC_WEEK)

Address Offset:

0x014 Reset Value:

Variable

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															WEEK
															R/W

position	Symbols	Function Description
31:3	Reserved	-
2:0	WEEK	Weekly count value

Decimal 0:6 means Sunday:Saturday. Please write the correct BCD code of decimal 0:6, write other values and they will be ignored. The weekly count

Week values correspond as follows.	The week count indicates
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

13.4.8 Monthly Count Register (RTC_MON)

Address Offset:

0x01C Reset Value:

Variable

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

Reserved	MON
	R/W

position	Symbols	Function Description
31:5	Reserved	-
4:0	MON	Monthly Value

Decimal 1:12 means 1:12 month. Please write the correct decimal 1:12 BCD code, write other values and they will be ignored.

13.4.9 Year Count Register (RTC_YEAR)

Address Offset:

0x020 Reset Value:

Variable

31302928272625242322212019181716	Reserved
----------------------------------	----------

1514131211109876543210

Reserved	YEARH	YEARL
	R/W	R/W

position	Symbols	Function Description
31:8	Reserved	-
7:4	YEARH	Annual tens of values
3:0	YEARL	Annual single digit values

Decimal 0:99 means 0:99 years. Counting according to the month progression. Automatic calculation of leap years such as:00,04,08,...,92

ignored.

13.4.10 Sub-alarm register (RTC_ALMMIN)

Address Offset:

0x024 Reset Value:

Variable

31:0	302928272625242322212019181716	Reserved
------	--------------------------------	----------

1514131211109876543210

Reserved		ALMMINH	ALMMINL
		R/W	R/W

position	Symbols	Function Description
31:6	Reserved	-
5:4	ALMMINH	Minute alarm clock matching value decimal
3:0	ALMMINL	Minute alarm clock matching value single digit

Please set the BCD code of decimal 0:59. Writing other values will not cause the alarm to match.

13.4.11 Time Alarm Clock Register (RTC_ALMHOUR)

Address Offset:

0x028 Reset Value:

Variable

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										ALMHOURH	ALMHOURL				
										R/W	R/W				

position	Symbols	Function Description
31:6	Reserved	-
5:4	ALMHOURH	Hourly alarm clock 10-bit matching value
3:0	ALMHOURL	Time Alarm Clock Digit Matching Value

Please set the correct alarm matching value according to the time system, otherwise the time alarm matching will not occur.

13.4.12 Weekly alarm clock register (RTC_ALMWEEK)

Address Offset:

0x02C Reset Value:

Variable

Reserved
31302928272625242322212019181716

1514131211109876543210

Reserved	ALMWEEK
	R/W

position	Symbols	Function Description
31:7	Reserved	-
6:0	ALMWEEK	Weekly alarm clock matching values. b0:b6 corresponds to Sunday:Saturday respectively, when the corresponding setting is 1 set to "", it means the alarm is valid on that day of the week. For example, b0=1, b5=1 means the alarm is valid on Sunday and Friday.

Please set the correct alarm matching value according to the time system, otherwise the time alarm matching will not occur.

13.4.13 Clock error compensation register (RTC_COMPEN)

Address Offset: 0x030

*Only power-on reset is valid for this register, reset value: 0x00000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	Reserved										CR				
											R/W				

position	Symbols	Function Description																																																																		
31:16	Reserved	-																																																																		
15	EN	Compensation Enable 0: Disable clock error compensation 1: Enables clock error compensation																																																																		
14:b	Reserved	-																																																																		
8:0	CR	Compensation value The compensation value can be set to compensate for accuracy of +/- 0.96ppm per second. The compensation value is the complement of the decimal point, and the last digit is the decimal part. Compensable range is +/- 0.96 ppm. The minimum difference is 0.048 ppm. Detailed description: CR0.96 ppm. Please refer to the following table for the compensation value: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>range</td> <td>274.6 ppm</td> <td>212.6 ppm</td> <td>minimum difference</td> <td>Number of compensation</td> <td>minimum</td> </tr> <tr> <td>resolution</td> <td>CR0.96 ppm</td> <td>please refer to the following table for the</td> <td></td> <td></td> <td></td> </tr> <tr> <td>compensation</td> <td>0 accuracy</td> <td>0</td> <td>0</td> <td>0</td> <td>-274.6 ppm</td> </tr> <tr> <td></td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>-273.7 ppm</td> </tr> <tr> <td></td> <td>:</td> <td>:</td> <td>:</td> <td>:</td> <td>:</td> </tr> <tr> <td></td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-0.95 ppm</td> </tr> <tr> <td></td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0 ppm</td> </tr> <tr> <td></td> <td>:</td> <td>:</td> <td>:</td> <td>:</td> <td>:</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>+211.7 ppm</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>+212.6 ppm</td> </tr> <tr> <td></td> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>No compensation</td> </tr> </table>	range	274.6 ppm	212.6 ppm	minimum difference	Number of compensation	minimum	resolution	CR0.96 ppm	please refer to the following table for the				compensation	0 accuracy	0	0	0	-274.6 ppm		1	0	0	0	-273.7 ppm		:	:	:	:	:		0	0	0	1	-0.95 ppm		0	0	0	1	0 ppm		:	:	:	:	:		0	1	1	1	+211.7 ppm		0	1	1	1	+212.6 ppm		0	X	X	X	No compensation
range	274.6 ppm	212.6 ppm	minimum difference	Number of compensation	minimum																																																															
resolution	CR0.96 ppm	please refer to the following table for the																																																																		
compensation	0 accuracy	0	0	0	-274.6 ppm																																																															
	1	0	0	0	-273.7 ppm																																																															
	:	:	:	:	:																																																															
	0	0	0	1	-0.95 ppm																																																															
	0	0	0	1	0 ppm																																																															
	:	:	:	:	:																																																															
	0	1	1	1	+211.7 ppm																																																															
	0	1	1	1	+212.6 ppm																																																															
	0	X	X	X	No compensation																																																															

Description and calculation of the compensation principle.

Since the counter uses a 32.768KHz clock, if it is necessary to compensate for the accuracy per second, it can only be compensated for the integer period of 32.768KHz, so the minimum unit of compensation per second is $(1/32768)^6 * 10 = 30.5\text{ppm}$, which cannot meet the requirement of high accuracy.

Then, to achieve a high precision clock compensation with a 32.768KHz

counter clock, the algorithm needs to be tuned

The maximum compensation period is expanded by a factor of 32 one. The average compensation unit per second becomes $30.5\text{ppm}/32=0.96\text{ppm}$ when the minimum unit of compensation is 30.5ppm only. Moreover, the compensation occurs in a relatively uniform range per second³². Therefore, a bit5 decimal setting is introduced in this register.

The set values are calculated as follows.

$$\text{CR[8 : 0]} \quad \frac{\text{Compensation target value [ppm}}{10^6} \quad 2^{15} \quad 0001.00000B$$

Take the
2's
complement

If the compensation target value is +20.6ppm, calculate the corresponding register value as follows: CR[8:0] = $(20.3 \times 2^{15} / 10^6)$ taken as the 2complement + 0001.00000B

$$\begin{aligned}
 &= (0.6651904) \quad \text{taken as the 2complement of } + 0001.00000B \\
 &= 0000.10101B + 0001.00000B \\
 &= 0001.10101B
 \end{aligned}$$

If the compensation target value is -20.6 ppm, calculate the corresponding register value as follows.

$$\begin{aligned}
 \text{CR[8:0]} &= (-20.3 \times 2^{15} / 10^6) \text{ taken as the 2complement of } + 0001.00000B \\
 &= (-0.6651904) \text{ taken as the 2complement of } + 0001.00000B \\
 &= 1111.01011B + 0001.00000B \\
 &= 0000.01011B
 \end{aligned}$$

14 Watchdog Timer (WDT)

14.1 WDT Introduction

The WDT can be used to detect and resolve faults caused by software errors.

The WDT is driven by a dedicated 10KHz on-chip oscillator and triggers an

interrupt or generates a system reset when the WDT counter reaches a set
overflow time.

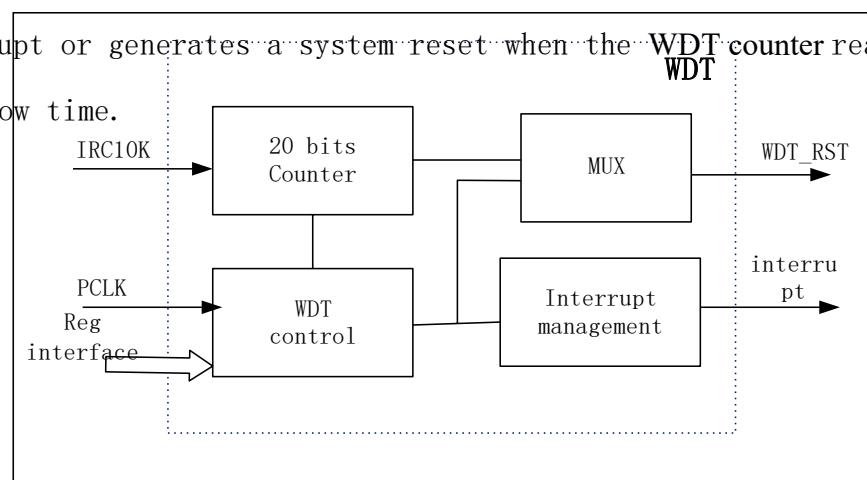


Figure 14-1 WDT overall block diagram

14.2 WDT Function Description

- 20Bit free-running incremental counter with configurable overflow time of 1.6ms - 50s
- The action after an overflow can be configured as an interrupt or a reset.
- The WDT clock is provided by a separate RC oscillator and can operate in Sleep and DeepSleep modes.
- The WDTCON register can only be modified when the WDT is not started, to prevent unintentional modifications after startup

The configuration of the WDT.

14.2.1 Interrupt generation after WDT overflow

In this mode, the WDT will generate interrupts periodically at the set time. The interrupt service program needs to be cleared in the

The WDT interrupt

flag. The

configuration

method is shown

below.

Step1: Configure WDT_CON.WOV and select WDT timing overflow time. Step2: Set WDT_CON.WINT_EN to 1, and select the interrupt generated after WDT overflow. step3: Enable WDT interrupt in NVIC interrupt vector table.

Step4: Write 0x1E, 0xE1 to WDT_RST register to start the WDT timer in turn.

Step5: Write 0xE10 xE1 to the WDT_RST register in order to clear the interrupt flag in the interrupt service program

14.2.2 Reset generated after WDT overflow

In this mode, a Reset signal is generated when the WDT counter overflows, which resets the MCU, and the user program needs to clear the WDT counter before the WDT overflows to avoid generating a WDT reset.

The configuration method is shown below.

Step1: Configure WDT_CON.WOV and select WDT counter overflow time. Step2: Set WDT_CON.WINT_EN to 0, select WDT overflow to generate reset. Step3: Write 0x1E, 0xE1 to WDT_RST register in sequence to start WDT timer. Step4: Write 0x1E, 0xE1 to the WDT_RST register to clear the WDT counter before the WDT overflow.

Note: Since the WDT oscillator is a low precision RC oscillator, it is highly recommended that the WDT be cleared before the WDT counter count reaches half of the overflow value.

14.3 WDT Register Description

Base address 0X40000C00

Register	Offset Address	Description
WDT_RST	0X080	WDT Clear Control Register
WDT_CON	0X084	WDT control register

Table 14-1 WDT Register List

14.3.1 WDT Clear Control Register (WDT_RST)

Offset address: 0x080

Reset value: 0x0000 0000

31:8	7	6	5	4	3	2	1	0
Reserved	WDTRST							
	WO							

position	Symbols	Description
31:8	Reserved	Reserved bits, read as 0
7:0	WDTRST	Watchdog start/zero control When the watchdog is not started, write 0xE1 to this register in turn to start the WDT timer. When the watchdog has been started, write 0x1E and 0xE1 to this register in turn to clear the WDT timer and interrupt flags.

14.3.2 WDT_CON register

Offset address: 0x084

Reset value: 0x0000 000F

Notes.

- This register can only be written when the WDT is not running.

	31:16	15:8	7	6	5	4	3	2	1	0
Reserved	WCNTL	WDINT	Res.	WINT_EN	WDTR	WOV				
	RO	RO		R/W	RO	R/W				

position	Symbols	Description
31:16	Reserved	Reserved bits, read as0
15:8	WCNTL	WDT counter low8
7	WDTINT	WDT interrupt flag 1: If a WDT interrupt has occurred, write 0x1E and 0xE1 to the WDT_RST register in order to clear the interrupt flag. 0: No WDT interruptions have occurred.
5	WINT_EN	Action configuration1 after WDT overflow: interrupt generation after WDT overflow. 0: Reset is generated after WDT overflow.
4	WDTR	WDT running flag1: WDT is running 0: WDT stop
3:0	WOV[3:0]	WDT timing overflow time configuration 0000: 1.6ms 1000: 500ms 0001: 3.2ms 1001: 820ms 0010: 6.4ms 1010: 1.64s 0011: 13ms 1011: 3.28s 0100: 26ms 1100: 6.55s 0101: 51ms 1101: 13.1s 0110: 102ms 1110: 26.2s 0111: 205ms 1111: 52.4s

15 Universal Synchronous Asynchronous Transceiver (UART)

15.1 Overview

This product comes with a2 Universal UART module (UART0/1) a Universal Synchronous Asynchronous Transceiver (UART) that provides the flexibility to exchange full-duplex data with external devices and supports synchronous one-way communication as well as multi-processor communication. The UART provides multiple baud rates through the programmable baud rate generator, which is generated by TIMER0 for UART0 and by TIMER1 for UART1.

15.2 Structure Block Diagram

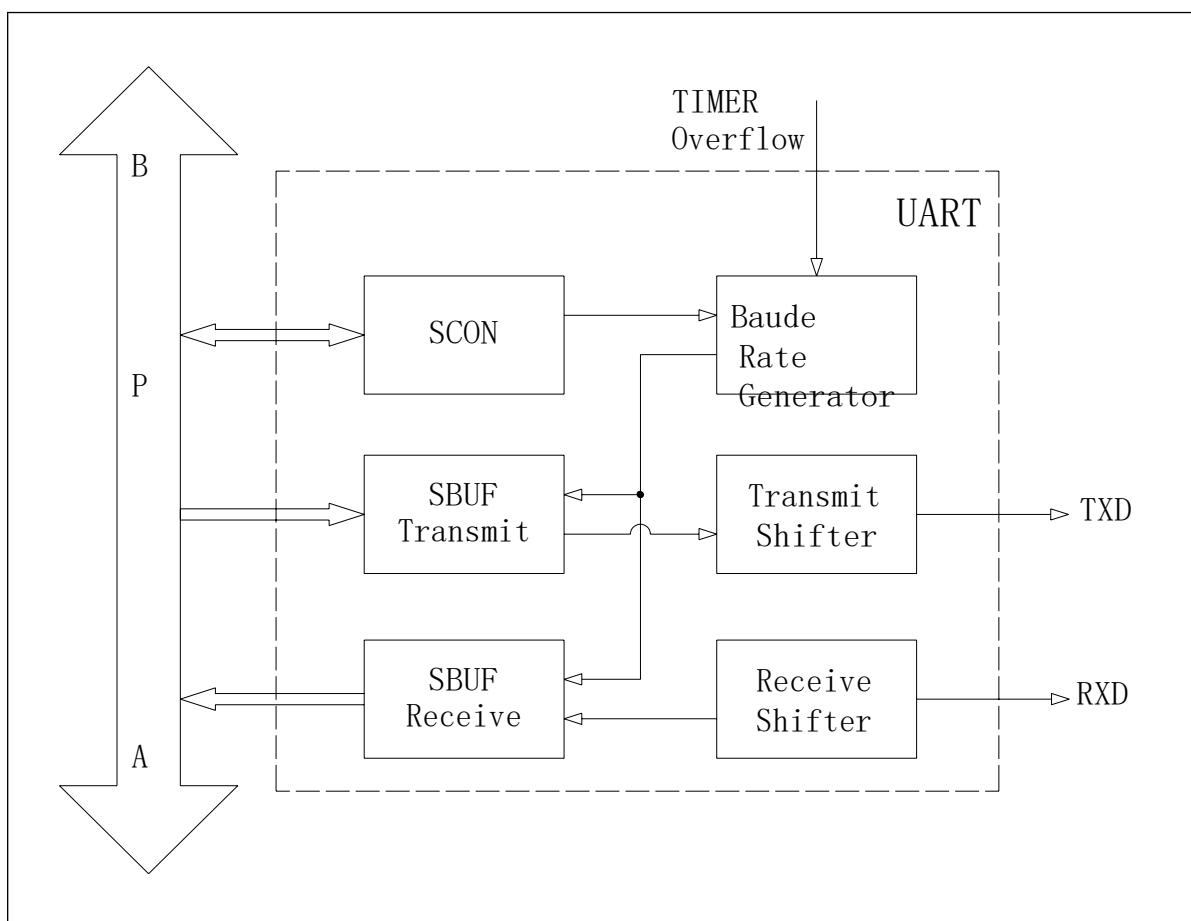


Figure 15-1 Block diagram of UART structure

15.3 Main Features

The Universal UART module supports the following basic functions.

- Full duplex transmission, half duplex transmission
- Programmable serial communication capability
 - Two character lengths: 8 bit, bit9
 - Mode0/1/2/3 Four transmission modes
- 16 Bit Baud Rate Generator
- Multi-machine communication
- Automatic address recognition

15.4 Function Description

15.4.1 Working mode

The UART supports multiple operating modes: synchronous half-duplex mode, asynchronous full-duplex mode. The `UARTx_SCON.SM0`

`SM1` can be configured with `UARTx_SCON`.

15.4.1.1 Mode0~Mode3 Function Comparison

SM can select different transmission modes: Mode0~Mode3. The main function pairs of these four operating modes are shown in the following

Working mode		Transfer bit width	Data Composition	Baud rate
Mode0	Synchronous mode Half Duplex	8bit	Data(8bit)	$BaudRate = \frac{f_{PCLK}}{12}$
Mode1	Asynchronous mode Full Duplex	10bit	Start (1bit) + Data(8bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{PCLK}}{32 * (65536 - TM)}$
Mode2	Asynchronous mode Full Duplex	11bit Table	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit) 15-1Mode0/1/2/3Data Structure	$BaudRate = \frac{(DBAUD + 1)f_{PCLK}}{64}$
Notes.	- Mode0 can only be used as a UART synchronous shift clock for the slave to receive external input. - Mode3 as a UART synchronous shift clock for the slave to receive external input. f_{PCLK} represents the current frequency	11bit	Start (1bit) + Data(8bit) shift + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{PCLK}}{32 * (65536 - TM)}$

- The definition of DBAUD is detailed in `UARTx_SCON`.
- TM is the TIMER count value. Note that TIMER must be configured in Bit 16Auto Reload mode, and both the Count Register and Reload Register must be written to the TM value.

15.4.1.2 Mode0 (synchronous mode, half duplex)

When operating in Mode0, theUART operates in synchronous mode with a baud rate of 1/12 of the fixed PCLK clock.

UART receive data is input by RXD, UART send data is output by TXD, RXD is the input and output port at this time, UART synchronous shift clock is output by TXD, TXD is the output port at this time. Note that this mode can only transmit the synchronous shift clock as the host, and cannot receive the clock from outside as the slave. In this mode, the transmitted data bit width can only be bit8, there is no start bit and end bit.

将 `UARTx_SCON.SM0` 和 `UARTx_SCON.SM1` 清零，可进入 Mode0 工作模式。发送数据时，清除 `UARTx_SCON.REN` 位，并将数据写入 `UARTx_SBUF` 寄存器。此时，发送数据将从 RXD 输出（低位在先，高位在后），同步移位时钟从 TXD 输出。

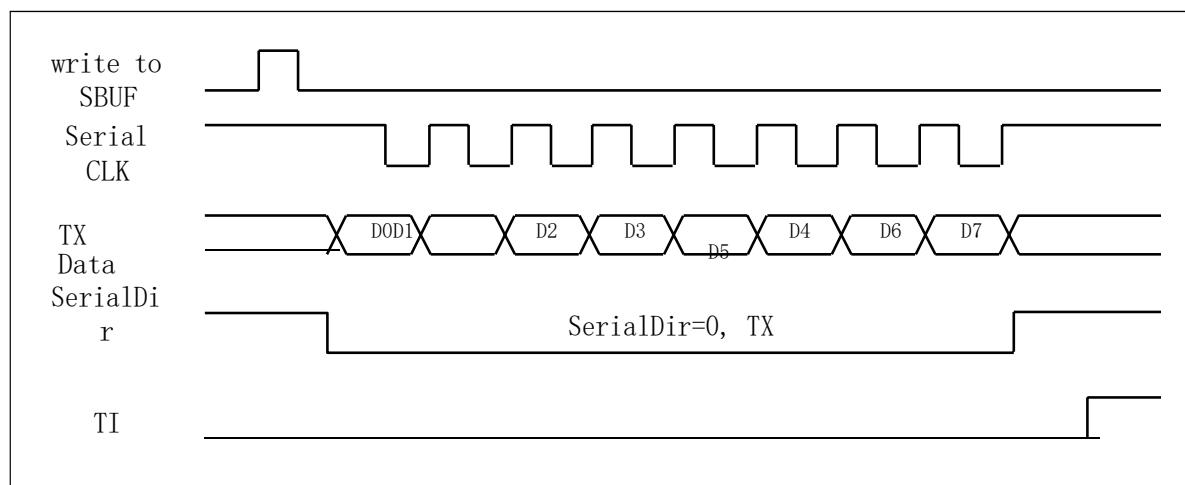


Figure 15-2 Mode0 Sending Data

接收数据时，将 `UARTx_SCON.REN` 位置 1，并将 `UARTx_ISR.RI` 位清零。当接收结束，数据可从 `UARTx_SBUF` 寄存器读出。此时，接收数据从 RXD 输入（低位在先，高位在后），同步移位时钟从 TXD 输出。

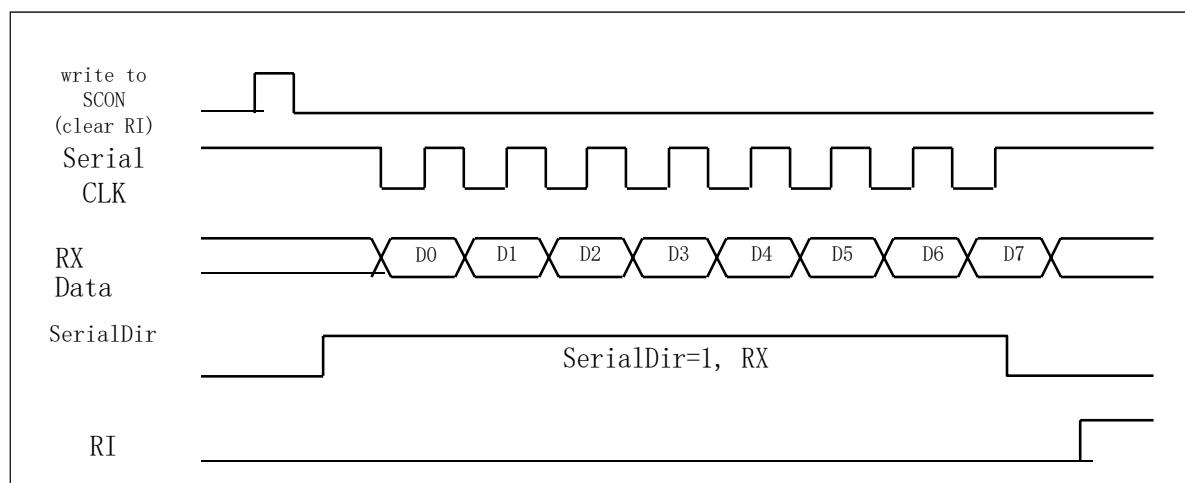


Figure 15-3 Mode0 Receiving Data

15.4.1.3 Mode1 (asynchronous mode, full duplex)

When operating in Mode1, transmit data is sent via TXD and receive data is received via RXD. The data consists of bits10: the start bit0 starts with "", followed by the 8data bits (low bit first, high bit second) and finally the end bit "1".

In this mode, the baud rate is generated by the timer module and is programmable. the baud rate of UART0 is generated by TIMER0

The baud rate of UART1 is generated by TIMER1.

将 `UARTx_SCON.SM0` 清 0, `UARTx_SCON.SM1` 置 1，可进入 Mode1 工作模式。

发送数据时，与 `UARTx_SCON.REN` 的值无关，将所发送数据写入 `UARTx_SBUF` 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

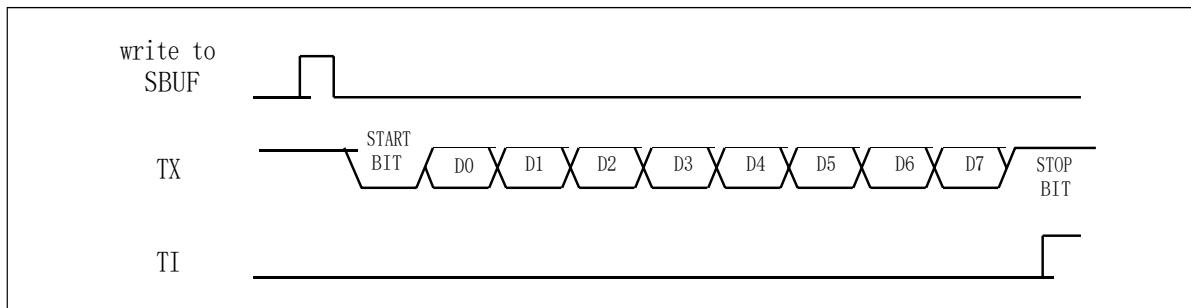


Figure 15-4 Mode1 Sending Data

接收数据时，需将 `UARTx_SCON.REN` 位置 1，并将 `UARTx_ISR.RI` 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 `UARTx_SBUF` 寄存器读出。

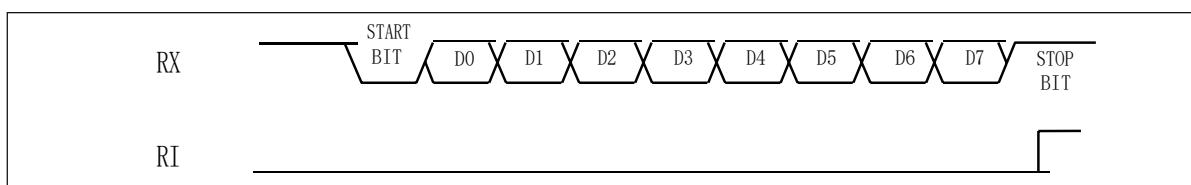


Figure 15-5 Mode1 Receiving Data

15.4.1.4 Mode2 (asynchronous mode, full duplex)

When operating in Mode2, transmit data is sent via TXD and receive data is received via RXD. The data consists of bits11: a start bit "0", followed by a 8data bit, 1 a TB8 bit and an end bit. The extra TB8 bit is used in a multi-computer communication environment, when TB=81, it indicates that the received

frame is an address frame; when 8TB=0, it indicates that the received frame is a data frame. This bit can also be used as a parity bit when multi-computer communication is not required.

In this mode, the baud rate can be generated independently without the need for external TIMER generation.

将 `UARTx_SCON.SM0` 置 1, `UARTx_SCON.SM1` 清 0, 可进入 Mode2 工作模式。

发送数据时, 与 `UARTx_SCON.REN` 的值无关, 并将所发送数据写入 `UARTx_SBUF` 寄存器中, 数据就会从 TXD 移出 (低位在先, 高位在后)。

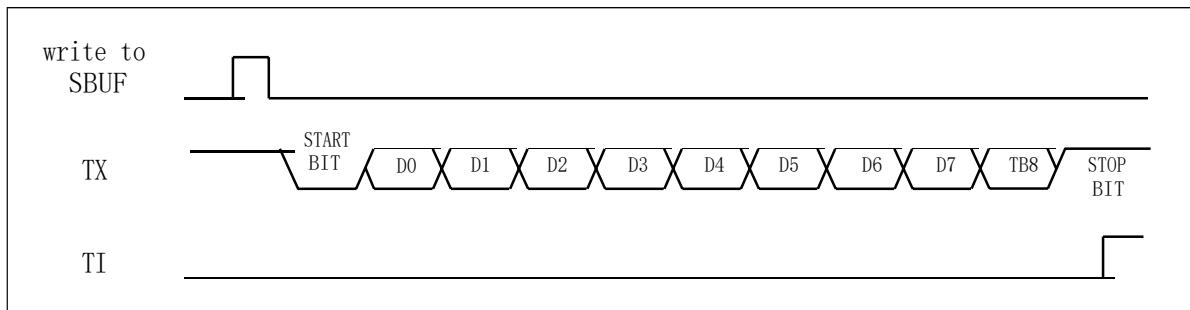


Figure 15-6Mode2 Sending Data

接收数据时, 需将 `UARTx_SCON.REN` 位置 1, 并将 `UARTx_ISR.RI` 位清 0。开始接收 RXD 上数据 (低位在先, 高位在后), 当接收完毕, 可以从 `UARTx_SBUF` 寄存器读出。

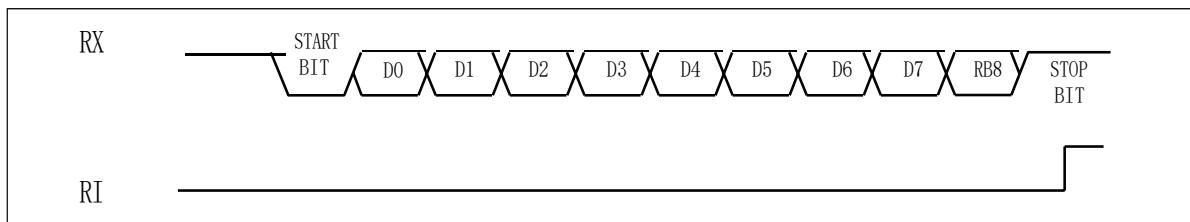


Figure 15-7Mode2 Receiving Data

15.4.1.5 Mode3 (asynchronous mode, full duplex)

The data format, transmission timing and operation mode of Mode3 are the same as Mode2, the only difference is that the baud rate of Mode3 is generated by TIMER, not by the device itself as Mode2. In this product, the baud rate of UART0 is generated by TIMER0, and the baud rate of UART1 is generated by TIMER1.

Set `UARTx_SCON.SM0` 1and `UARTx_SCON.SM1` to enter Mode3 operation mode.

When sending data, it is independent of the value of `UARTx_SCON.REN`, and the data sent is written into the `UARTx_SBUF` register, and the data is moved out from TXD (low bit first, high bit second)

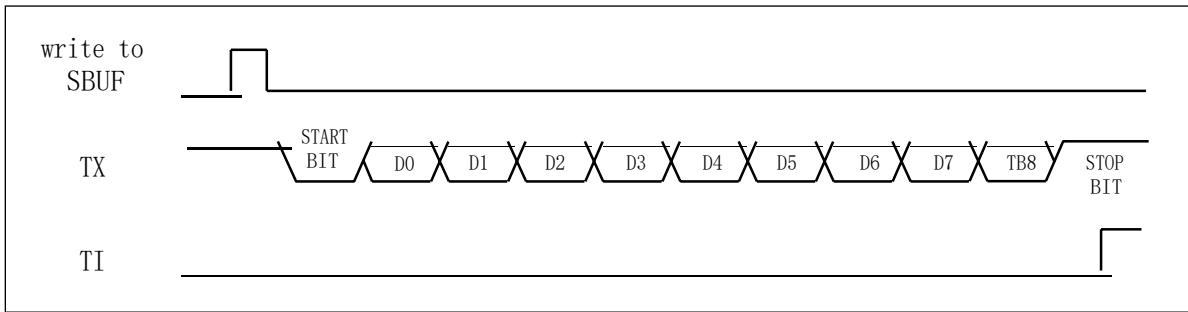


Figure 15-8Mode3Sending Data

接收数据时，需将 `UARTx_SCON.REN` 位置 1，并将 `UARTx_ISR.RI` 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 `UARTx_SBUF` 寄存器读出。

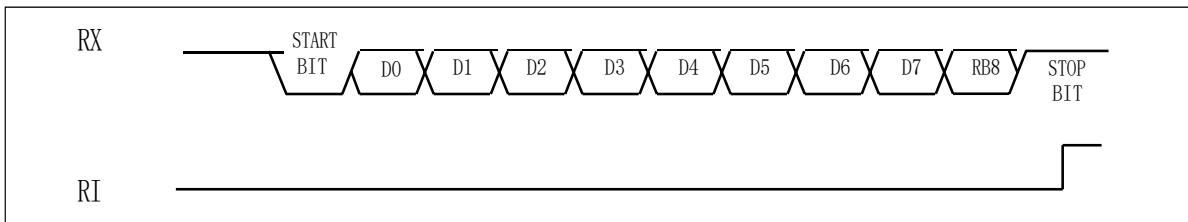


Figure 15-9Mode3Receiving Data

15.4.2 Baud rate generation

Mode0~Mode3 have different formulas for generating baud rates, as shown below.

$$\text{Mode0 baud rate generation formula: } \text{BaudRate} = \frac{f_{PCLK}}{12}$$

$$\text{Mode1 Baud rate generation formula: } \text{BaudRate} = \frac{(DBAUD+1)f_{PCLK}}{32*(65536-TM)}$$

$$\text{Mode2 baud rate generation formula: } \text{BaudRate} = \frac{(DBAUD+1)f_{PCLK}}{64}$$

$$\text{Mode3 Baud rate generation formula: } \text{BaudRate} = \frac{(DBAUD+1)f_{PCLK}}{32*(65536-TM)}$$

Notes.

f_{PCLK} represents the current frequency

- The definition of DBAUD is detailed in **UARTx_SCON**.
- TM is the TIMER count value. Note that TIMER must be configured in Bit 16Auto Reload mode, and both the Count Register and Reload Register must be written to the TM value.

15.4.2.1 Model/Mode3 baud rate setting example

Baud rate	PCLK = MHz1					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	26	2403.85	0.16%	13	2403.85	0.16%
4800	13	4807.69	0.16%	7	4464.29	-6.99%
9600	7	8928.57	-6.99%	3	10416.67	8.51%
19200	3	20833.33	8.51%	2	15625.00	-18.62%
38400	2	31250.00	-18.62%	1	31250.00	-18.62%
57600	1	62500.00	8.51%	1	31250.00	-45.75%
76800	1	62500.00	-18.62%	0	#DIV/0!	#DIV/0!
115200	1	62500.00	-45.75%	0	#DIV/0!	#DIV/0!

Baud rate	PCLK = MHz4					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	104	2403.85	0.16%	52	2403.85	0.16%
4800	52	4807.69	0.16%	26	4807.69	0.16%
9600	26	9615.38	0.16%	13	9615.38	0.16%
19200	13	19230.77	0.16%	7	17857.14	-6.99%
38400	7	35714.29	-6.99%	3	41666.67	8.51%
57600	4	62500.00	8.51%	2	62500.00	8.51%
76800	3	83333.33	8.51%	2	62500.00	-18.62%
115200	2	125000.00	8.51%	1	125000.00	8.51%

Baud rate	PCLK = MHz10					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	260	2403.85	0.16%	130	2403.85	0.16%

4800	130	4807.69	0.16%	65	4807.69	0.16%
9600	65	9615.38	0.16%	33	9469.70	-1.36%
19200	33	18939.39	-1.36%	16	19531.25	1.73%
38400	16	39062.50	1.73%	8	39062.50	1.73%
57600	11	56818.18	-1.36%	5	62500.00	8.51%

76800	8	78125.00	1.73%	4	78125.00	1.73%
115200	5	125000.00	8.51%	3	104166.67	-9.58%

Baud rate	PCLK = MHz14					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	365	2397.26	-0.11%	182	2403.85	0.16%
4800	182	4807.69	0.16%	91	4807.69	0.16%
9600	91	9615.38	0.16%	46	9510.87	-0.93%
19200	46	19021.74	-0.93%	23	19021.74	-0.93%
38400	23	38043.48	-0.93%	11	39772.73	3.57%
57600	15	58333.33	1.27%	8	54687.50	-5.06%
76800	11	79545.45	3.57%	6	72916.67	-5.06%
115200	8	109375.00	-5.06%	4	109375.00	-5.06%

Baud rate	PCLK = MHz20					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	521	2399.23	-0.03%	260	2403.85	0.16%
4800	260	4807.69	0.16%	130	4807.69	0.16%
9600	130	9615.38	0.16%	65	9615.38	0.16%
19200	65	19230.77	0.16%	33	18939.39	-1.36%
38400	33	37878.79	-1.36%	16	39062.50	1.73%
57600	22	56818.18	-1.36%	11	56818.18	-1.36%
76800	16	78125.00	1.73%	8	78125.00	1.73%
115200	11	113636.36	-1.36%	5	125000.00	8.51%

Baud rate	PCLK = MHz24					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
76800	8	78125.00	1.73%	4	78125.00	1.73%
115200	5	125000.00	8.51%	3	104166.67	-9.58%

2400	625	2400.00	0.00%	313	2396.17	-0.16%
4800	313	4792.33	-0.16%	156	4807.69	0.16%
9600	156	9615.38	0.16%	78	9615.38	0.16%
19200	78	19230.77	0.16%	39	19230.77	0.16%
38400	39	38461.54	0.16%	20	37500.00	-2.34%

57600	26	57692.31	0.16%	13	57692.31	0.16%
76800	20	75000.00	-2.34%	10	75000.00	-2.34%
115200	13	115384.62	0.16%	7	107142.86	-6.99%

Baud rate	PCLK = MHz2					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	52	2403.85	0.16%	26	2403.85	0.16%
4800	26	4807.69	0.16%	13	4807.69	0.16%
9600	13	9615.38	0.16%	7	8928.57	-6.99%
19200	7	17857.14	-6.99%	3	20833.33	8.51%
38400	3	41666.67	8.51%	2	31250.00	-18.62%
57600	2	62500.00	8.51%	1	62500.00	8.51%
76800	2	62500.00	-18.62%	1	62500.00	-18.62%
115200	1	125000.00	8.51%	1	62500.00	-45.75%

Baud rate	PCLK = MHz8					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	208	2403.85	0.16%	104	2403.85	0.16%
4800	104	4807.69	0.16%	52	4807.69	0.16%
9600	52	9615.38	0.16%	26	9615.38	0.16%
19200	26	19230.77	0.16%	13	19230.77	0.16%
38400	13	38461.54	0.16%	7	35714.29	-6.99%
57600	9	55555.56	-3.55%	4	62500.00	8.51%
76800	7	71428.57	-6.99%	3	83333.33	8.51%
115200	4	125000.00	8.51%	2	125000.00	8.51%

Baud	PCLK = 11.0592 MHz					
	Dual baud rate			Single baud rate		

rate	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	288	2400.00	0.00%	144	2400.00	0.00%
4800	144	4800.00	0.00%	72	4800.00	0.00%
9600	72	9600.00	0.00%	36	9600.00	0.00%

19200	36	19200.00	0.00%	18	19200.00	0.00%
38400	18	38400.00	0.00%	9	38400.00	0.00%
57600	12	57600.00	0.00%	6	57600.00	0.00%
76800	9	76800.00	0.00%	5	69120.00	-10.00%
115200	6	115200.00	0.00%	3	115200.00	0.00%

Baud rate	PCLK = MHz16					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	417	2398.08	-0.08%	208	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%
57600	17	58823.53	2.12%	9	55555.56	-3.55%
76800	13	76923.08	0.16%	7	71428.57	-6.99%
115200	9	111111.11	-3.55%	4	125000.00	8.51%

Baud rate	PCLK = MHz32					
	Dual baud rate			Single baud rate		
	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	833	2400.96	0.04%	417	2398.08	-0.08%
4800	417	4796.16	-0.08%	208	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%
76800	26	76923.08	0.16%	13	76923.08	0.16%
115200	17	117647.06	2.12%	9	111111.11	-3.55%

Baud	PCLK = MHz22.12					
	Dual baud rate			Single baud rate		

rate	CNT	Actual Baud Rate	Error %	CNT	Actual Baud Rate	Error %
2400	576	2400.17	0.01%	288	2400.17	0.01%
4800	288	4800.35	0.01%	144	4800.35	0.01%

9600	144	9600.69	0.01%	72	9600.69	0.01%
19200	72	19201.39	0.01%	36	19201.39	0.01%
38400	36	38402.78	0.01%	18	38402.78	0.01%
57600	24	57604.17	0.01%	12	57604.17	0.01%
76800	18	76805.56	0.01%	9	76805.56	0.01%
115200	12	115208.33	0.01%	6	115208.33	0.01%

15.5 Frame error detection

When operating in Mode1/2/3, the UART has a frame error detection feature where the hardware automatically detects if the received frame data has a valid Stop bit. If the received data is not received by the hardware with a valid Stop bit in expectation, and thus there is a synchronization failure or excessive noise, the `UARTx_ISR.FE` is set 1. The `UARTx_ISR.FE` bit is set by the hardware 1 and cleared 0 by the software, if not cleared 0 by the software in time, the subsequent received data will not clear 0 the `UARTx_ISR.FE` flag even if it carries a valid Stop bit.

15.6 Multi-machine communication

Mode2/3 has a multi-computer communication function, for which the TB8/RB8 bit1 has been added to the frame format. The `UARTx_SCON.SM2` is set to "1" to enable the multi-computer communication bit. When sending data with the multicomunication bit enabled, the host can use `UARTx_SCON.TB8` to distinguish whether the current frame is an address frame (`UARTx_SCON. TB=81`) or a data frame (`UARTx_SCON. 8TB=0`). When receiving data, the slave ignores the current received frame if the RB8 bit (first bit9) is "0".

- When it is a data frame, the frame data will not be stored into the `UARTx_SBUF` register of the slave, and the slave will not generate a receive interrupt.
- When the address frame is an address frame, the automatic address recognition function is enabled in the multi-machine communication, so that the slave can detect whether the received address matches its own address.
 - If the addresses match, the slave will set "1" for `UARTx_SCON.RB8=1` and `UARTx_ISR.RI=1`, the slave software will clear the `UARTx_SCON.SM2` bit to "0". Accept data frames.
 - If the address does not match, the host is not addressing the slave and the slave hardware holds `UARTx_SCON.RB8` and `RI` is "0", the software keeps the `UARTx_SCON.SM2` bit as "1" and continues to listen to the address. Note: If necessary, you can also enable the multi-machine communication bit under Mode1, where the TB8 bit is replaced by the stop bit. RI will be set to "1" when the slave receives a matching address frame and a valid stop bit.

15.7 Automatic address recognition

When the multi-machine communication bit is enabled (UARTx_SCON.SM2 is set to 1"), automatic address recognition function is also enabled. This function is implemented in hardware and allows the slave to detect the reception of each address frame, and if the address matches the slave address, the receiver will give the UARTx_ISR.RI receive flag. If the address does not match, the receiver does not give any receive flags.

15.7.1 Given address

The UARTx_SADDR register of the UART device is used to indicate the given address of its own device. The UARTx_SADEN register is an address mask, which can be used to define the irrelevant bits in the address. When one bit of UARTx_SADEN is "0", it means the address of that bit is irrelevant, which means that the address of that bit does not participate in the address matching process. These irrelevant bits increase the addressing flexibility, allowing the host to address one or more slave devices at the same time. Note that the UARTx_SADEN register must be set to 8'hFF if a unique match address is required.

GivenAddr SADDR & SADEN

15.7.1.1 Broadcast Address

The broadcast address is used to address all slave devices at the same time, typically the broadcast address is 8'hFF.

BroadCastAddr SADDR | SADEN

15.7.1.2 Examples

Suppose a slave has the following UARTx_SADDR and UARTx_SADEN configurations.

SADDR: 8'b01101001

SADEN: 8'b11111011

Then its given address and broadcast address are as follows.

Given: 8'b01101x01

Broadcast: 8'b11111x11

As can be seen, the host can address this slave with four addresses: 8'b01101001 and 8'b01101101 (given address) 8'b11111011 and 8'b11111111 (broadcast address).

15.8 Sender-sender cache

15.8.1 Receiving Cache

Universal UART (UART0/1) receiver has a frame length (8/9bits) receive buffer, which means that when one frame of data is received, the data in the receive buffer will be held until the next frame of data is received by

the Stop_{D4}^{START}_{D1}^{BIT}_{D0}^{DATA}_{D3}^{BIT}_{D2}^{DATA}_{D5}^{BIT}_{D6}^{DATA}_{D7}^{TOP}_{D8}^{BIT}_{D9}^{DATA} bit, and then the receive buffer will be updated to a new frame of data.

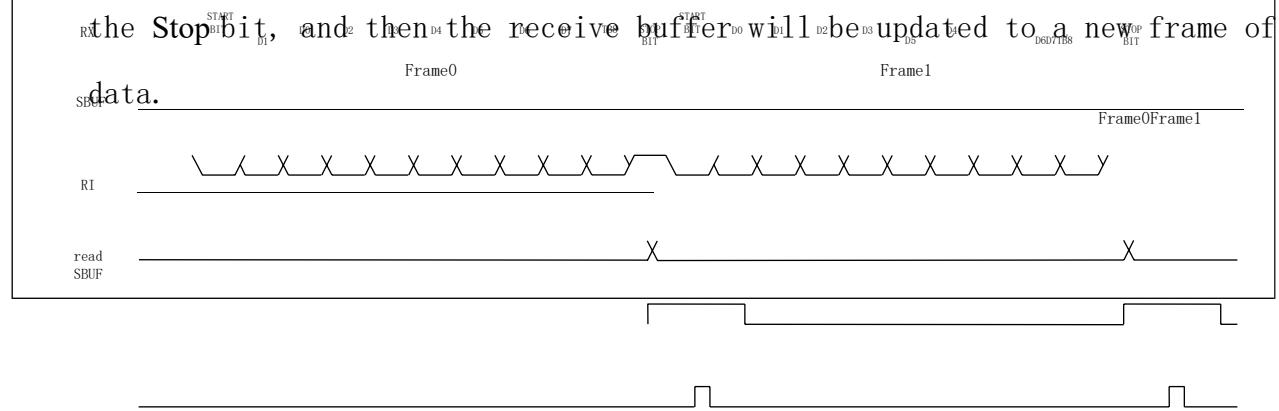


Figure 15-10 Receive Cache

15.8.2 Send cache

The transmitter side of Universal UART (UART0/1) does not support transmit cache. If the `UARTx_SBUF` register is filled during sending data, it will destroy the data currently being sent. The software should avoid this operation.

15.9 Register

UART0 Base address: 0x4000 0000

UART1 base address: 0x4000 0100

Register	Offset Address	Description
UARTx_SBUF	0x00	Data Register
UARTx_SCON	0x04	Control register
UARTx_SADDR	0x08	Address Register
UARTx_SADEN	0x0C	Address Mask Register
UARTx_ISR	0x10	Interrupt Flag Bit Register
UARTx_ICR	0x14	Interrupt flag bit clear register

15.9.1 Data register (UARTx_SBUF)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SBUF							
R								RW							

pos iti on	Marker	Function Descriptio n
31:8	Reserved	
7:0	SBUF	<p>(a) When sending data, the data to be sent is written to this register. Read the received data from this register when receiving data.</p> <p>Note that the value read to this register is actually the value in RxBuffer, and the value written to this register is actually written to TXShifter.</p>

15.9.2 Control register (UARTx_SCON)

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DBA UD	Reser ved	SM01	SM2	REN	TB8	RB8	TIEN	RIEN	
R						RW	R	RW	RW	RW	RW	RW	RW	RW	RW

position	Marker	Function Description
31:10	Reserved	
9	DBAUD	Baud rate multiplier setting 0: Single baud rate. 1: Double Baud Rate
8	Reserved	
7:6	SM01	Operating mode configuration n00: mode0;01 : mode1; 10: mode2. 11: mode3
5	SM2	Multi-machine communication enable control 0: Disable multi-camera communication function 1: Enables multi-computer communication function
4	REN	Receive Enable Control Mode0: 0 : send, 1 : receive Others: 0 ::send, 1 ::receive/send
3	TB8	The TB8 bits to be sent when sending data
2	RB8	RB8 bits received when receiving data
1	TIEN	Send completion interrupt enable 0: Disable send completion interrupt 1: Enables the send completion interrupt
0	RIEN	Receive completion interrupt enable 0: Disable receive completion interrupt 1: Enables the receive completion interrupt

15.9.3 Address register (UARTx_SADDR)

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADDR							
R								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	SADDR	Slave device address

15.9.4 Address Mask Register (UARTx_SADEN)

Offset address: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADEN							
R								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	SADEN	Slave device address mask

15.9.5 Flag Bit Register (UARTx_ISR)

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
R															

position	Symbols	Description
31:3	Reserved	
2	FE	Receive frame error flag bit; set in hardware, cleared in software 1: FE interrupt flag valid 0: FE interrupt flag is invalid
1	TI	Send completion interrupt flag bit; set in hardware, cleared in software 1: TI interrupt flag valid 0: TI interrupt flag is invalid
0	RI	Receive completion interrupt flag bit; hardware set, software clear 1: RI interrupt flag valid 0: RI interrupt flag is invalid

15.9.6 Flag Bit Clear Register (UARTx_ICR)

Offset address: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												FE CLR	TI CLR	RI CLR	
R												W	W	W	

position	Marker	Function Description
31:3	RESERVED	
2	FECLR	Receive frame error flag cleared Write 0clear, write 1invalid
1	TICLR	Send complete interrupt flag cleared Write 0clear, write 1invalid
0	RICLR	Receive complete interrupt flag cleared Write 0clear, write 1invalid

16 Low Power Synchronous Asynchronous Transceiver (LPUART)

16.1 Overview

The product comes with an LPUART module, which contains all the necessary hardware support to allow asynchronous serial communication with minimal power consumption; the baud rate can be generated either by external TIMER2 or by the module's internal logic.

The LPUART module internal register configuration logic works in the PCLK clock domain and the data sending and receiving logic works in the SCLK clock domain. When the system enters the low-power mode, turn off the high-frequency PCLK clock and turn on the low-frequency SCLK clock, the LPUART can still send and receive data normally.

SCLK clock source can be selected: PCLK, external low-speed clock (XTL), internal low-speed clock (RCL), In

When LPMODE=1, the SCLK clock also supports 1/2/4/8/16/32/64/128 times prescaling.

Note that with LPMODE=0, the LPUART receives the Toggle output signal from the TIMER2 clock instead of the

Overflow signal, so the Toggle output of TIMER2 must be enabled.

16.2 Structure Block Diagram

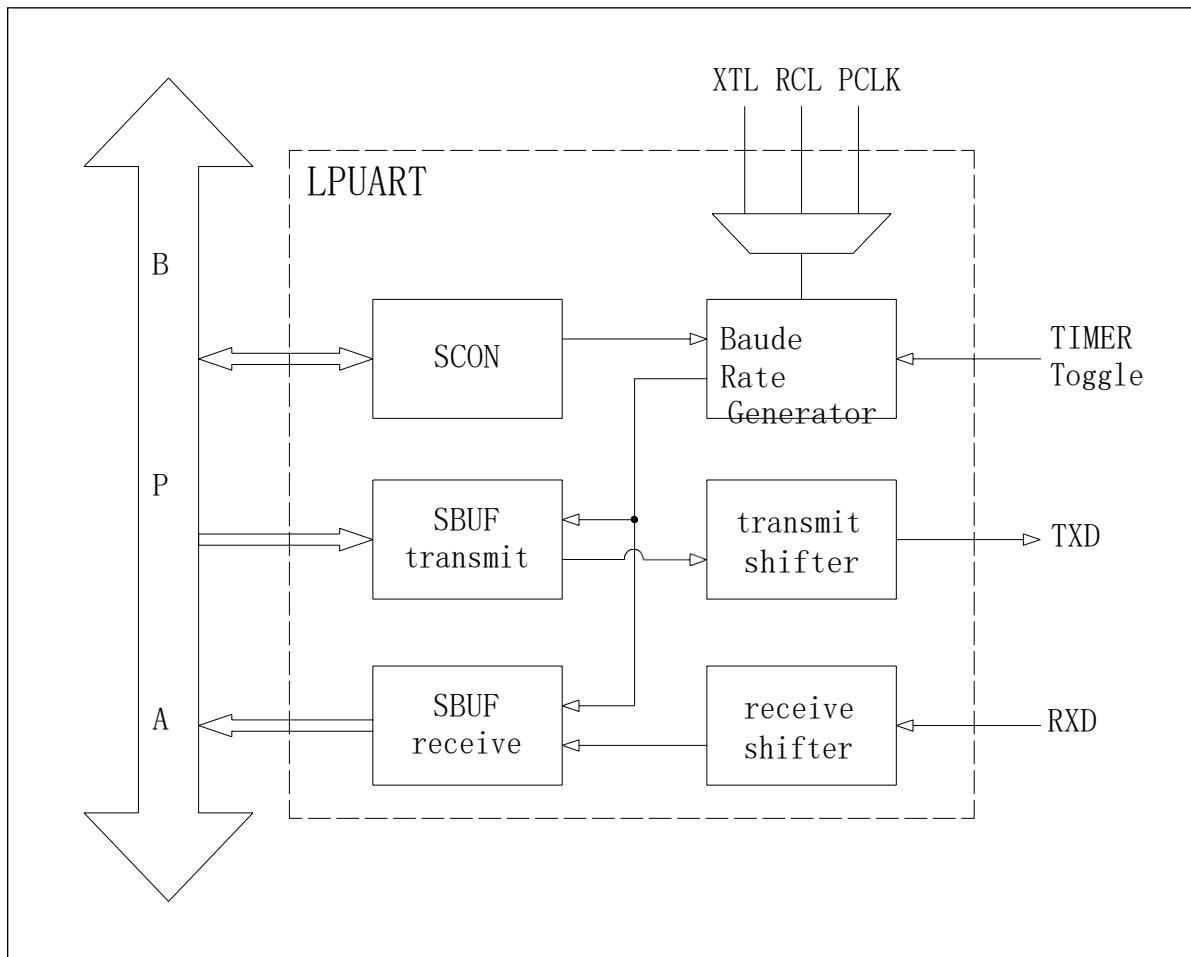


Figure 16-1 Block diagram of LPUART structure

16.3 Main Features

The LPUART module supports the following basic functions.

- Configure the clock PCLK
- Transmission clock SCLK (SCLK is selectable from XTL, RCL, and PCLK)
- Sending and receiving data in system low-power mode
- Full duplex transmission, half duplex transmission
- Programmable serial communication capability
 - Two character lengths: 8 bit, bit9
 - Mode0/1/2/3 Four transmission modes
- 16 Bit Baud Rate Counter
- Multi-machine communication
- Automatic address recognition

16.4 Function Description

16.4.1 Configuration clock and transmission clock

The LPUART module has two clocks: the configuration clock PCLK and the transmission clock SCLK.

- Configuring the clock

The configuration clock is used for register configuration of the LPUART module on the system APB bus and is fixed to PCLK.

- Transmission Clock

The transmission clock is used for LPUART data transceiver logic operation, with XTL, RCL and PCLK selectable. when SCLK

When the system enters DeepSleep, the LPUART can still send and receive data normally when the system is XTL or RCL.

16.4.2 Working mode

LPUART supports multiple operating modes: synchronous half-duplex mode, asynchronous full-duplex mode. By configuration

The value of `LPUARTx_SCON.SM` is used to obtain the desired operating mode.

LPUART has an additional LPMODE control bit than UART. When this position is "1", only Mode1/3 is supported.

The operating mode and the baud rate generation method will be changed.

16.4.2.1 Mode0~Mode3 Function Comparison

When LPMODE = 0.

SM can be selected by4 configuring LPUARTx_SCON.SM. The functions of each mode are compared as follows.

Working mode		Transfer bit width	Data Composition	Baud rate
Mode0	Synchronous mode Half Duplex	8bit	Data(8bit)	$BaudRate = \frac{f_{SCLK}}{12}$
Mode1	Asynchronous mode Full Duplex	10bit	Start (1bit) + Data(8bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{SCLK}}{32 * (65536 - TM)}$
Mode2	Asynchronous mode Full Duplex	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{SCLK}}{64}$
Mode3	Asynchronous mode Full Duplex	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{SCLK}}{32 * (65536 - TM)}$

Table 16-1Mode0/1/2/3Data Structure

When LPMODE = 1.

Mode2	Asynchronous mode Full Duplex	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{f_{SCLK}}{PreScale * 4}$
-------	----------------------------------	-------	--	--

Notes.

- Mode0 can only be used as a host to send LPUART synchronous shift clock, not as a slave to receive external input
LPUART synchronous shift clock.
- f_{SCLK} represents the current SCLK frequency.

-
- The definition of DBAUD is detailed in LPUARTx_SCON.
 - TM is the TIMER count value. Note that TIMER must be configured in Bit 16Auto Reload mode, and both the Count Register and Reload Register must be written to the TM value.
 - PreScale is the prescale factor.

16.4.2.2 Mode0 (synchronous mode, half-duplex) data sending and receiving instructions

SM is set to 0, LpUart works in Mode0, clock and data are input and output synchronously, baud rate is fixed to SCLK/12, data width is fixed to bit8, there is no start and end bit. The data is received and transmitted via the RXD pin; the synchronous shift clock is output from the TXD pin. Note that the TXD pin does not accept an input clock.

Mode0 operating mode is not supported when LPMODE=1.

发送数据时, 设置 LPUART_SCON.REN 为 0, 并将待发送的数据写入 SBUF 寄存器。

此时, 发送数据将从 RXD 输出(低位在先, 高位在后), 同步移位时钟从 TXD 输出。

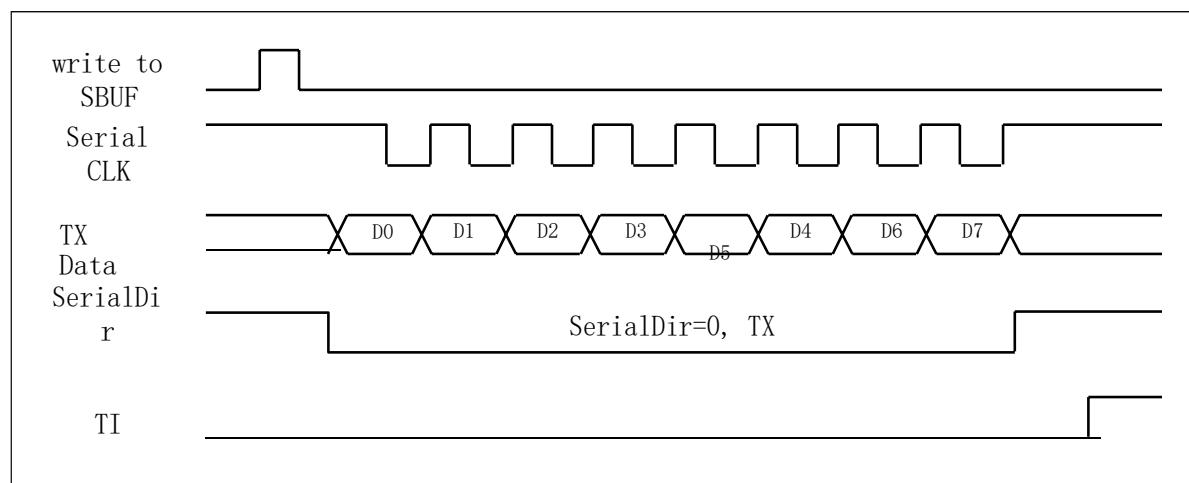


Figure 16-2 Mode0 Sending Data

接收数据时, 设置 LPUART_SCON.REN 为 1, 并将 LPUART_ISR.RI 位清零。当接收结束, 数据可从 LPUART_SBUF 寄存器读出。此时, 接收数据从 RXD 输入(低位在先, 高位在后), 同步移位时钟从 TXD 输出。

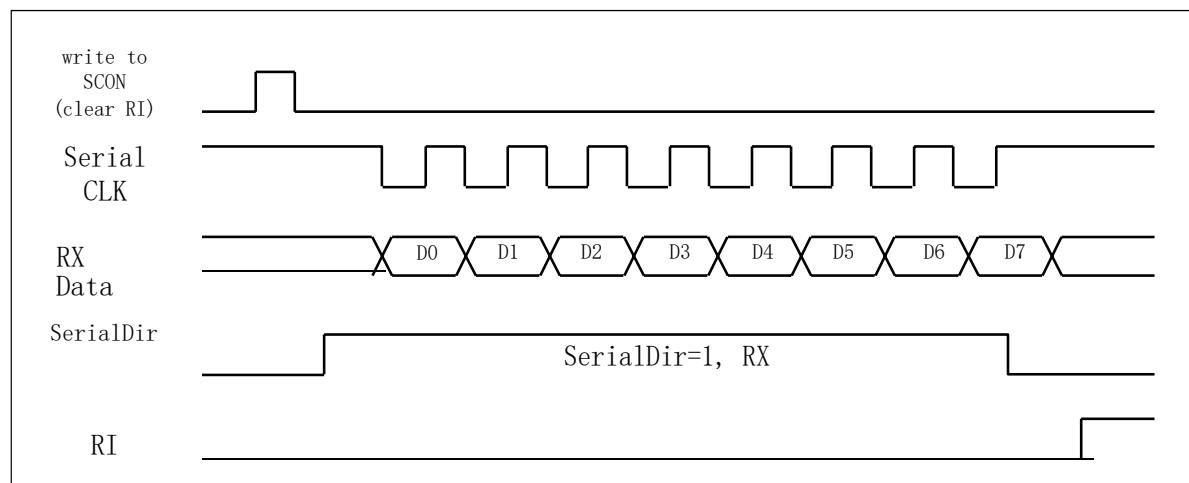


Figure 16-3 Mode0 Receiving Data

16.4.2.3 Model (asynchronous mode, full duplex) data sending and receiving

instruction operating in Mode1, transmit data is sent via TXD and receive data is

received via RXD. The data

It consists of a bit10: the start bit 0 "", followed by a 8data bit (low bit

first, high bit second) and finally the end bit "1".

In this mode, the baud rate of the LPUART is generated by the timer TIMER2 module and is programmable.

0Clear0 LPUART_SCON.SM #set1 LPUART_SCON.SM to enter Model operation mode.

When LPMODE=1, Model operation mode is supported, but the baud rate calculation method is changed, refer to the baud rate generation chapter for details.

When sending data, the sent data is written to LPUART_SBUF regardless of the value of LPUART_SCON.

register, the data is then shifted out of TXD (low bit first, high bit second)

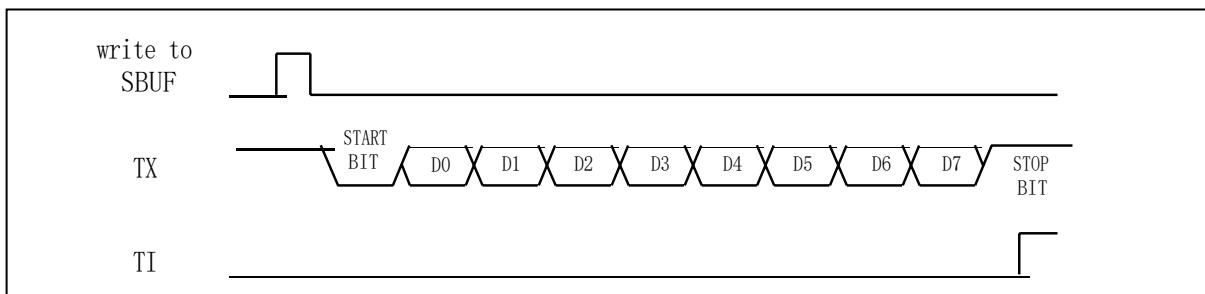
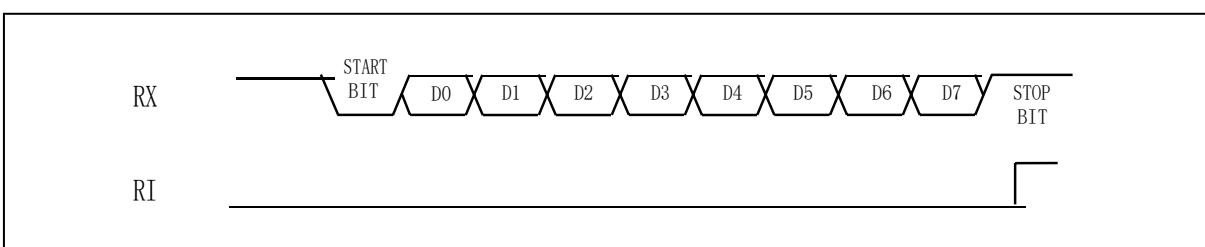
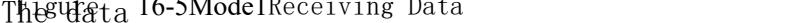


Figure 16-4Mode1 Sending Data

To receive data, you need to set the LPUART_SCON.REN position 1, and clear the LPUART_ISR.RI bit0 . Start to receive the data on RXD (low bit first, high bit second) when the reception is finished, you can read from LPUART_SBUF register.



16.4.2.3 Model (asynchronous mode, full duplex) data sending and receiving

instruction operating in Mode1, transmit data is sent via TXD and receive data is received via RXD. 

16.4.2.481 Mode2 (asynchronous mode, full duplex) data sending and receiving

instruction operating in Mode2, transmit data is sent via TXD and receive data is received via RXD. The data consists of 11 bits: a start bit "0", followed by 8 data bits, 1 a TB8 bit and an end bit. The extra TB8 bit is used in a multi-computer communication environment, when TB=81, it indicates that an address frame is being received; when TB=0, it indicates that a data frame is being received. This bit can also be used as a parity bit when multi-computer communication is not required. In this mode, the baud rate can be generated independently, without the need for external TIMER generation.

Set LPUART_SCON.SM1 clear LPUART_SCON.SM to enter Mode2 operation mode.

When LPMODE=1, Mode2 operation mode is not supported.

When sending data, the value of LPUART_SCON.REN is not relevant, and the sent data is written to

LPUART_SBUF register, the data will then be shifted out from TXD (low bit first, high bit second)

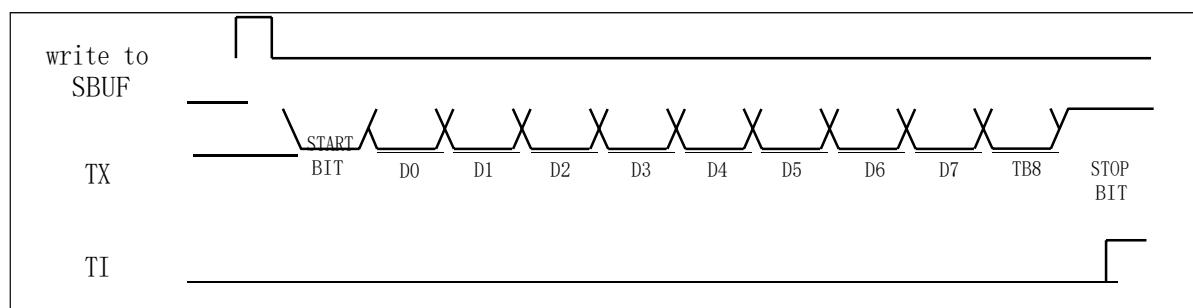


Figure 16-6Mode2 Sending Data

接收数据时，需将 LPUART_SCON.REN 位置 1，并将 LPUART_ISR.RI 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 LPUART_SBUF 寄存器读出。

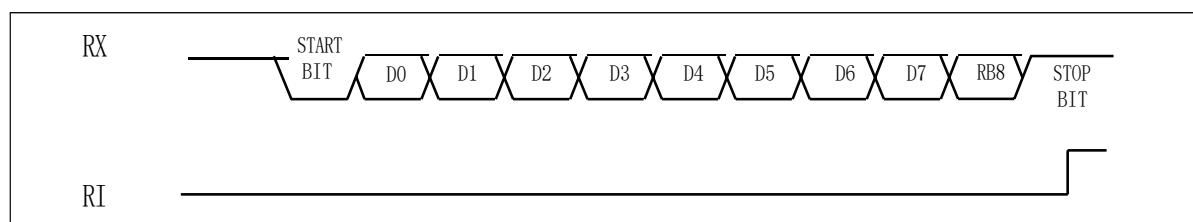


Figure 16-7Mode2 Receiving Data

16.4.2.482 Mode3 (asynchronous mode, full duplex) data sending and receiving

instrucMode3 has the same data format, transmission timing and operation as Mode2, the only difference is that Mode3's baud rate is generated by TIMER, not by the device itself as in Mode2. mode3's baud rate is programmable and the baud rate is generated in the same way as Mode1.

Set LPUART_SCON.SM0 to 1,LPUART_SCON.SM1 to 0,to enter Mode3 operation mode.

When LPMODE=1, Mode3 operation mode is supported. However, the baud rate calculation is changed, refer to the baud rate generation chapter for details.

When sending data, the value of LPUART_SCON.REN is not relevant, and the sent data is written to

LPUART_SBUF register, the data will then be shifted out from TXD (low bit first, high bit second),

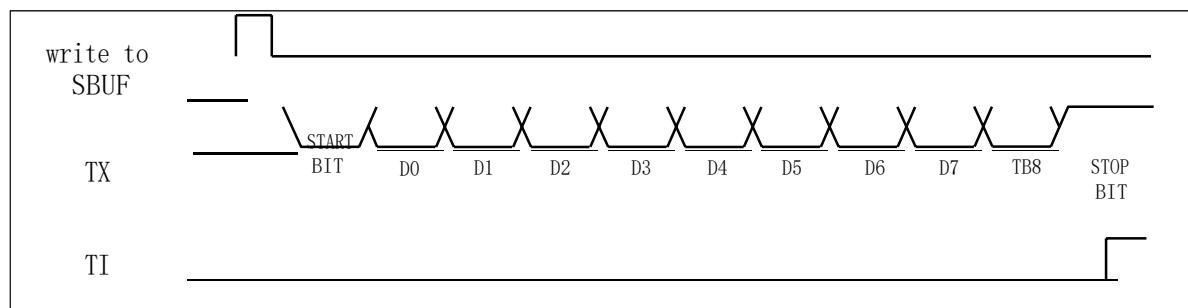


Figure 16-8Mode3 Sending Data

接收数据时，需将 LPUART_SCON.REN 位置 1，并将 LPUART_ISR.RI 位清 0。开始接收 RXD 上数据（低位在先，高位在后），当接收完毕，可以从 LPUART_SBUF 寄存器读出。

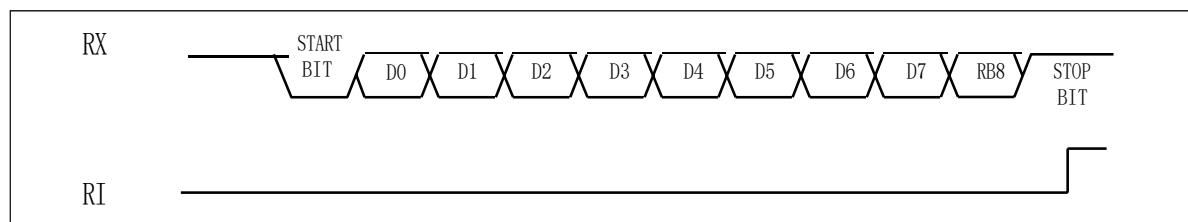


Figure 16-9Mode3 Receiving Data

16.4.3 Baud rate generation

LPMODE=0

Mode0~Mode3 The formula for generating the baud rate is shown below.

$$\text{Mode0 baud rate generation equation: } \text{BaudRate} = \frac{f_{SCLK}}{12}$$

$$\text{Mode1 Baud rate generation formula: } \text{BaudRate} = \frac{(DBAUD+1)f_{SCLK}}{32*(65536-TM)}$$

$$\text{Mode2 baud rate generation formula: } \text{BaudRate} = \frac{(DBAUD+1)f_{SCLK}}{64}$$

$$\text{Mode3 Baud rate generation formula: } \text{BaudRate} = \frac{(DBAUD+1)f_{SCLK}}{32*(65536-TM)}$$

LPMODE=1

When LPMODE=1, Mode0, Mode2 are not supported.

$$\text{Mode1, Mode3 baud rate generation formula: } \text{BaudRate} = \frac{f_{SCLK}}{\text{PreScale}*4}$$

Notes.

- f_{SCLK} represents the current SCLK frequency.
- The definition of DBAUD is detailed in LPUARTx_SCON.
- TM is the TIMER count value. Note that TIMER must be configured in Bit 16Auto Reload mode, and both the Count Register and Reload Register must be written to the TM value.
- PreScale is the prescale factor.

16.5 Frame error detection

When operating in Mode1/2/3, the LPUART has a frame error detection function where the hardware automatically detects whether the received frame data carries a valid Stop bit. If a valid Stop bit is not received within the expected time when receiving data, and thus there is a synchronization failure or excessive noise, the LPUART_ISR.FE position1 . The LPUART_ISR.FE bit is set1 by the hardware and cleared0 by the software. If the software is not cleared0 in time, the LPUART_ISR.FE flag will not be cleared0 for subsequent data received even with a valid Stop bit.

16.6 Multi-machine communication

Mode2/3 has a multi-computer communication function, for which the bits1 TB8/RB8 have been added to the frame format.

Set "1" to turn on the multi-computer communication bit.

When sending data after enabling the multi-computer communication bit, the host can distinguish whether the current frame is an address frame (LPUART_SCON.TB=81) or a data frame (LPUART_SCON.8TB=0) by LPUART_SCON.TB8.

- When it is a data frame, the frame data will not be stored into the LPUARTx_SBUF register of the slave, and the slave will not generate a receive interrupt.
- When the address frame is an address frame, the automatic address recognition function is enabled in the multi-machine communication, so that the slave can detect whether the received address matches its own address.
 - If the addresses match, the slave will set "1" to LPUARTx_ISR.RI and "1" to LPUARTx_SCON. RB8=1 and LPUARTx_ISR.RI=1, the slave software will clear the LPUARTx_SCON.SM2 bit to "0" and accept the data frame.
 - If the address does not match, the host is not addressing the slave and the slave hardware holds LPUARTx_RB8 RI is "0", the software keeps the LPUARTx_SCON.SM2 bit as "1", and the slave continues to be in the address listening state.

Notes.

- If necessary, you can also enable the multimachine communication bit under Mode1, where the TB8 bit is replaced by the stop bit. RC will be set to "1" when the slave receives a matching address frame and a valid stop bit.

16.7 Automatic address recognition

When the multi-machine communication bit is enabled (LPUART_SCON.SM2 is set to 1"), the automatic address recognition function will also be enabled. This function is implemented in hardware and allows the slave to detect the reception of each address frame, and if the address matches the slave address, the receiver will give the LPUART_ISR.RI receive flag. If the address does not match, the receiver will not give any receive flag.

16.7.1 Given address

The LPUART_SADDR register of LPUART device is used to indicate the given address of its own device, and the LPUART_SADEF register is the address mask, which can be used to define the irrelevant bits in the address. When one bit of LPUART_SADEF is "0", it means the address of that bit is irrelevant, which means that the address of that bit does not participate in the address matching process. These irrelevant bits increase the flexibility of addressing, allowing the host to address one or more slave devices at the same time. Note that the LPUART_SADEF register must be set to 8'hFF if a unique matching address is required.

GivenAddr SADDR & SADEF

16.7.2 Broadcast Address

The broadcast address is used to address all slave devices at the same time, typically the broadcast address is 8'hFF.

BroadCastAddr SADDR | SADEF

16.7.3 Examples

Suppose the LPUART_SADDR and LPUART_SADEF of a slave are configured as follows.

SADDR: 8'b01101001

SADEN: 8'b11111011

Then its given address and broadcast address are as follows.

Given: 8'b01101x01

Broadcast: 8'b11111x11

As can be seen, the host can address this slave with four addresses: 8'b01101001 and 8'b01101101 (given address) 8'b11111011 and 8'b11111111 (broadcast address).

16.8 Sender-sender cache

16.8.1 Receiving Cache

The LPUART receiver has a frame length (8/9bits) receive buffer, which means that when one frame of data is received, the data in the receive buffer will be held until the next frame of data is received by the stop bit, and

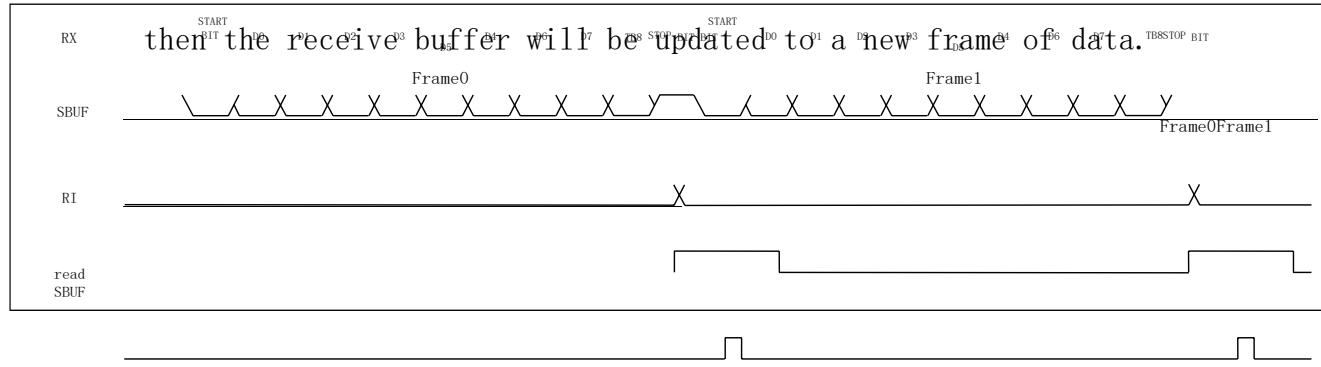


Figure 16-10 Receive Cache

16.8.2 Send cache

The LPUART transmitter has a frame length (8/9bits) transmit cache. When the transmit shift register is sending the current frame data, the CPU can

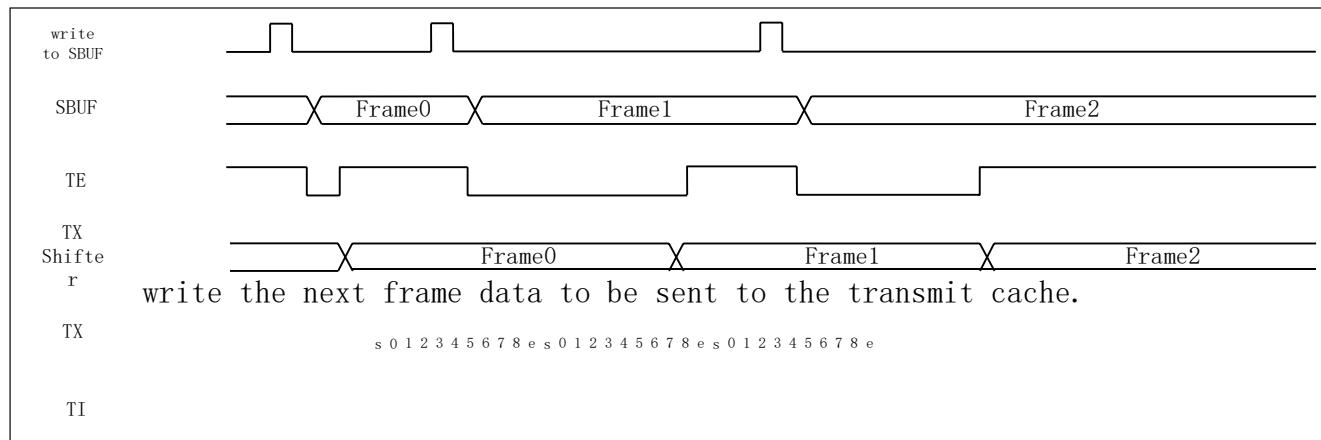


Figure 16-11 Send Cache

The LPUART module contains only one frame of the LPUART_ISR.TE register bit, which is the transmit buffer empty flag bit.

(8/9bits) transmit buffer, so when the LPUART_ISR.TE bit is "1", the

hardware will automatically block the software write operation to the LPUART_SBUF register until the LPUART_ISR.TE bit becomes "0" TE bit becomes "0". Before the software fills the transmit data into LPUART_SBUF register, it must judge the status of LPUART_ISR.TE bit "0" or "1", otherwise the transmit data will be lost.

16.9 Register

LPUART base address: 0x4000 0200

Register Name	Offset Address	Description
LPUART_SBUF	0x00	Data Register
LPUART_SCON	0x04	Control register
LPUART_SADDR	0x08	Address Register
LPUART_SADEN	0x0C	Address Mask Register
LPUART_ISR	0x10	Interrupt Flag Bit Register
LPUART_ICR	0x14	Interrupt flag bit clear register

16.9.1 Data register (LPUART_SBUF)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SBUF							
R								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	SBUF	<p>When sending data, when the sending data is written to this register; when receiving data, the data is read from this register when the data is received.</p> <p>Note that the value read to this register is actually the value in the RXBuffer, and the value written to this register is actually written to the TXShifter.</p>

16.9.2 Control register (LPUART_SCON)

Offset address: 0x04

Reset value: 0x0000 E000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRS	SCLKSEL	LPM ODE	DBA UD	TEEN	SM01	SM2	REN	TB8	RB8	TIEN	RIEN			
	RW	RW				RW	RW	RW	RW	RW	RW	RW			

position	Marker	Function Description
31:16	Reserved	
15:13	PRS	Transmission clock SCLK prescaling selection. 000: div128;001 : div64;010 : div32;.... ;110: div2;111: div1 PRS[2:0] is only valid when LPMODE=1; when LPMODE=0, PRS[2:0] will not prescale SCLK.
12:11	SCLKSEL	Transmission clock SCLK selection. 00: PCLK. 01: PCLK. 10: XTL. 11: RCL
10	LPMODE	Low power mode. 0: Normal operating mode. 1: Low-power operation mode
9	DBAUD	Double baud rate. 0: Single baud rate. 1: Double Baud Rate
8	TEEN	Transmit buffer air break enable. 0: disable;1 : enable
7:6	SM01	Operating mode;00 : mode0;01 : mode1; 10: mode2. 11: mode3

5	SM2	Multi-host communication. 0: disable, 1 : enable
---	-----	---

4	REN	Receive enable. mode0: 0 : send, 1 : receive Others: 0 ::send, 1 ::receive/send
3	TB8	Send TB8 bits
2	RB8	Receive RB8 bits
1	TIEN	Send completion interrupt enable. 0: disable; 1 : enable
0	RIEN	Receive completion interrupt enable. 0: disable; 1 : enable

16.9.3 Address register (LPUART_SADDR)

Offset address: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SADDR							
R								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	SADDR	Slave device address register

16.9.4 Address Mask Register (LPUART_SDEN)

Offset address: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SDEN							
R								RW							

position	Symbols	Function Description
31:8	Reserved	
7:0	SDEN	Slave device address mask register

16.9.5 Interrupt Flag Bit Register (LPUART_ISR)

Offset address: 0x10

Reset value: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TE		FE		TI		RI	
R								R		R		R		R	

pos iti on	Marker	Function Descrip tio n
31:4	Reserved	
3	TE	Send buffer air break flag bit, hardware set, hardware clear. Note: When the value of this bit is "0", the hardware automatically blocks the software from writing SBUF operations. 1:TE interrupt valid 0:TE interrupt is invalid
2	FE	Receive frame error flag bit, set in hardware, cleared in software 1:FE interrupt valid 0:FE interrupt invalid
1	TI	Send completion interrupt flag bit, set in hardware, cleared in software 1:TI interrupt valid 0:TI interrupt is invalid
0	RI	Receive completion interrupt flag bit, hardware set, software clear 1:RI interrupt valid 0:RI interrupt is invalid

16.9.6 Interrupt Flag Bit Clear Register (LPUART_ICR)

Offset address: 0x14

Reset value: 0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FE CLR				TI CLR		RI CLR	
R								W				W		W	

position	Symbols	Description
31:3	Reserved	
2	FECLR	Clear the receive frame error flag bit. Write 0clear, write 1invalid
1	TICLR	Clear the transmit complete interrupt flag bit. Write 0clear, write 1invalid
0	RICLR	Clear receive complete interrupt flag bit; write 0clear, write 1invalid

17 I2C bus (I2C)

17.1 Introduction

I2C is a two-wire bi-directional synchronous serial bus that uses a clock line and a data line to transfer information between two devices connected to the bus, providing a simple and efficient method for exchanging data between devices. Each device connected to the bus has a unique address, and any device can act as either a host or a slave, but only one host is allowed at any one time. The I2C logic can handle byte transfers autonomously. It keeps track of serial transfers and has a status register (I2C_STAT) reflects the status of the I2C bus controller and the I2C bus.

17.2 Main Features

The I2C controller supports the following features.

- Support host transmit/receive, slave transmit/receive four working modes
- Support Standard(100Kbps) / Fast(400Kbps) / High Speed(1Mbps) three operating rates
- Support Bit7 Addressing Function
- Support noise filtering function
- Broadcast address support
- Support interrupt status query function

17.3 Protocol Description

The I2C bus uses the "SCL" (serial clock bus) and "SDA" (serial data bus) of the connected device to transfer information. The host outputs the serial clock signal on the SCL line and the data is transferred on the SDA line, with each byte transferred (the highest bit MSB starts the transfer) than answer bit. One SCL clock pulse transmits one data bit.

17.3.1 Data transfer on the I2C bus

Usually the standard I2C transfer protocol consists of four parts: the start (S) or repeat start signal (Sr), the slave address and read/write bits,

the transfer data, and the stop signal (P).

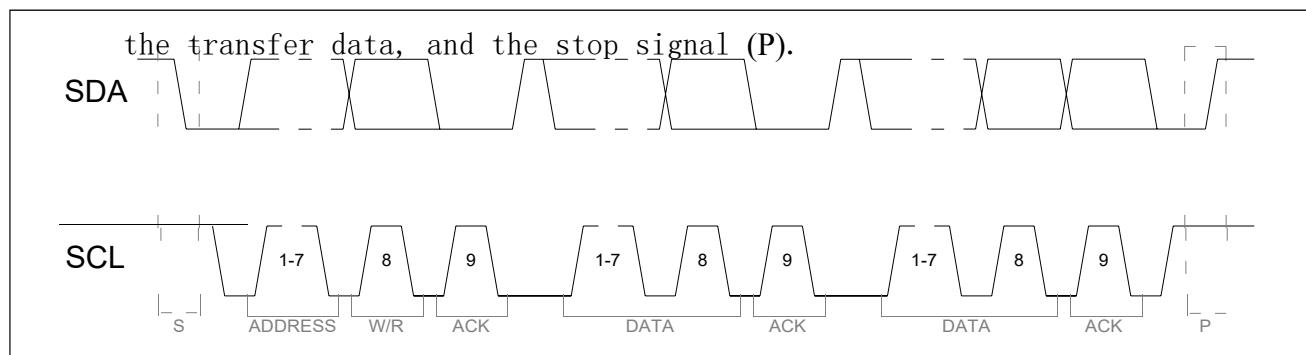


Figure 17-1I2CTransfer Protocol

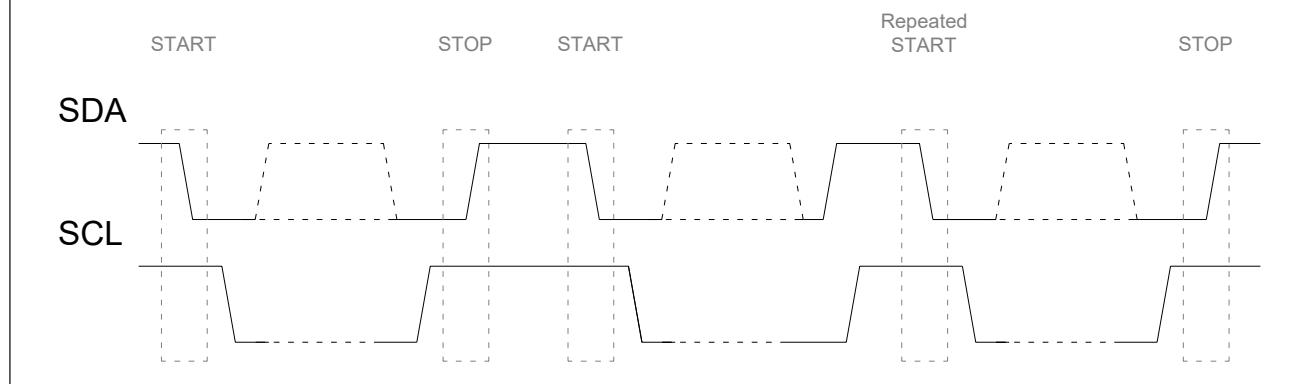
- Start signal, repeat start signal, stop signal

When the bus is idle (SCL and SDA lines are high at the same time) a high to low signal on the SDA line indicates that a start signal is generated on the bus.

A duplicate start signal is generated when there is no stop signal between two start signals. The master uses this method to communicate with another slave or the same slave in a different transmission direction (e.g., from writing to reading from the device) without releasing the bus.

When the SCL line is high, a low-to-high signal appears on the SDA line and is defined as a stop signal. The host sends a stop signal to the bus

to end the data transfer.



- Slave address and read/write bits

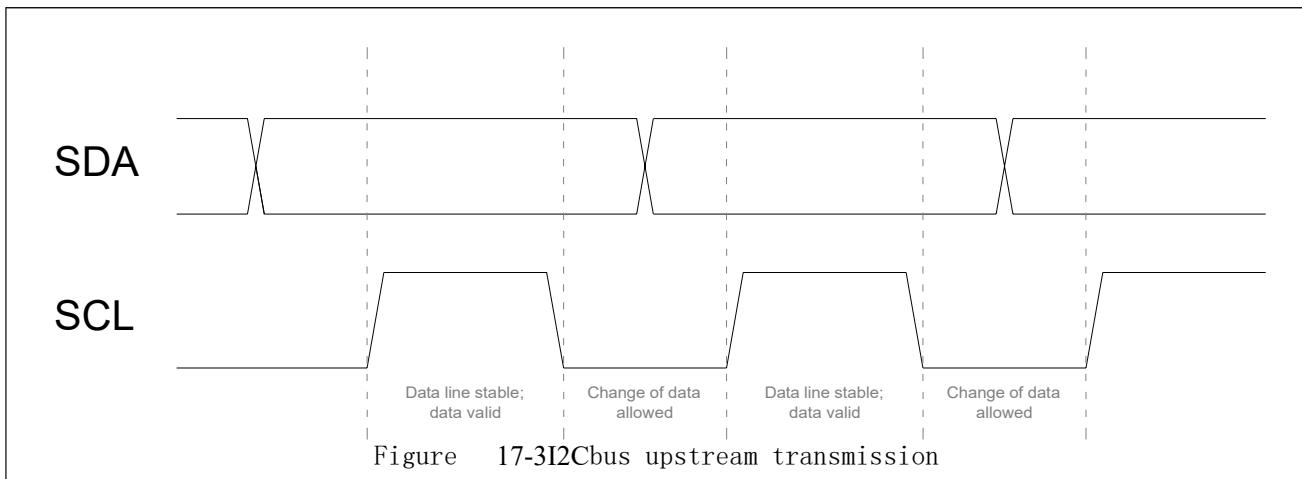
Figure 17-2STARTand STOP Conditions

When the start signal is generated, the host immediately transmits the first byte of data: the 7bit slave address + the read/write bit, which controls the direction of data transmission from the slave (0: write;1 : read) A slave addressed by the host will transmit the first byte of data by adding the first9

The SCL clock cycle sets SDA low as an answer.

- Transferring data

During data transfer, one SCL clock pulse transmits one data bit and the SDA line can only be changed when SCL is low.



17.3.2 Answer on the I2C bus

Each byte transmitted is followed by an answer bit. ACK is a low signal, and when the clock signal is high, the SDA remains low to indicate that the receiver has successfully received the data from the sender.

When the host is the transmitting device, if a no response signal (NACK) is generated on the slave, the host can generate a stop signal to exit the data transmission or generate a repeat start signal to start a new round of data transmission. When a no response signal (NACK) is generated on the slave when the host is the receiving device, the slave releases the SDA line, causing the host to generate a stop signal or a repeat start signal.

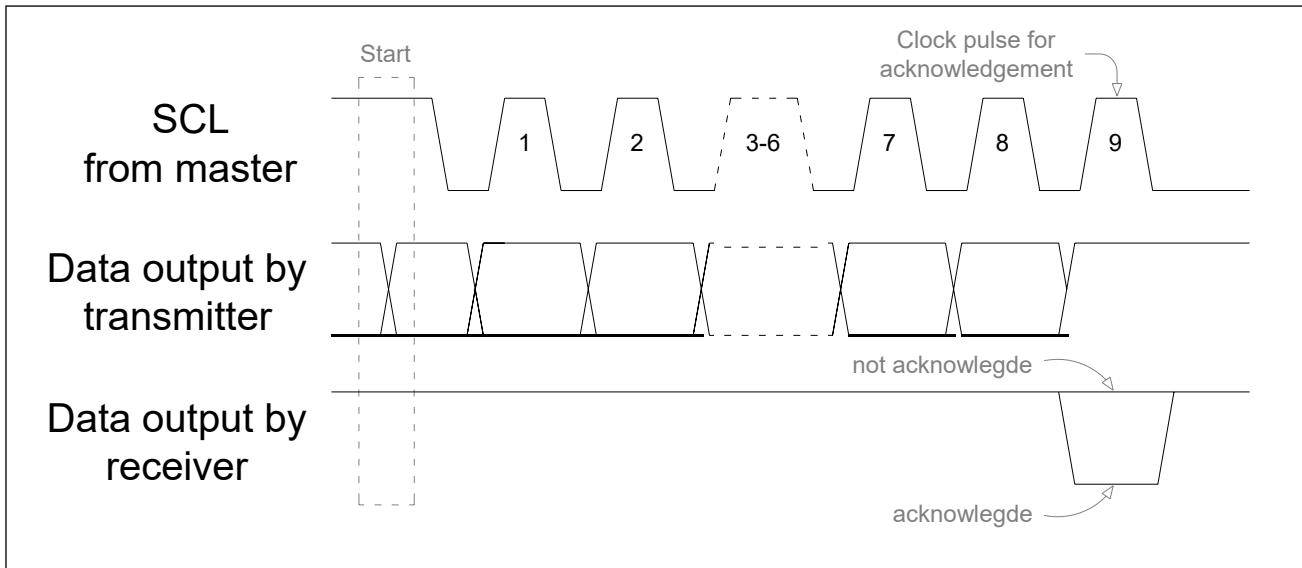


Figure 17-4 Answer Signal on I2CBus

17.3.3 Arbitration on the I2C bus

Arbitration on the I2C bus is divided into two parts: synchronization on the SCL line and arbitration on the SDA line

- Synchronization on the SCL line (clock synchronization)

Since the I2C bus has a line "with" logic function, the bus goes low whenever one node on the SCL line sends a low level. When all nodes are sending high, the bus will only behave high. Therefore, the clock low time is determined by the device with the longest clock level period, and the clock high time is determined by the device with the shortest clock high period. Due to this characteristic of I2C, when multiple hosts send clock signals at the same time, a uniform clock signal is represented on the bus. If the slave wants the host to slow down the transmission it can inform the host by pulling SCL low actively to extend its low time and wait when the host finds the level of SCL pulled low while preparing the next transmission until the slave finishes the operation and releases control of the SCL line.

- Arbitration on the SDA line

Arbitration on the SDA line is also due to the logical function of line "with" on the I2C bus. After sending data, the host decides whether to

exit the competition by comparing the data on the bus. A host that loses arbitration immediately switches to an unaddressed slave state to ensure that it can be addressed by the host that wins arbitration. The host that lost arbitration continues to output clock pulses (on SCL) until the current serial byte is sent. This principle ensures that the I2C bus does not lose data when multiple hosts attempt to control the bus.

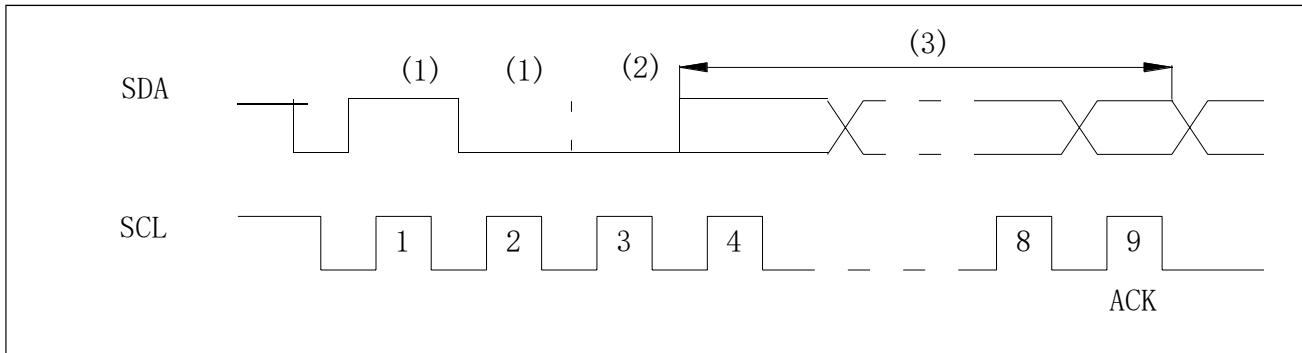


图 17-5 I2C 总线上的仲裁

- a) another device to send serial data.
- b) The other device undoes one of the logics1 sent by that I2C host by pulling the SDA low first (dashed line). Arbitration is lost and the I2C enters slave receive mode.
- c) At this point the I2C is in slave receive mode, but still generates clock pulses until the current byte has been sent. the I2C will not generate clock pulses for the next byte transfer. Once arbitration has been won, the data transfer on the SDA is initiated by the new host.

17.4 Function Description

The I2C bus uses dual wires to transfer information between devices connected to the bus "SCL" (serial clock line) and "SDA" (serial data line). Filtering logic protects data integrity by filtering out burrs on the data bus. Since only undirected ports are available, I2C components require the use of open-drain buffers to the pins. The I2C standard is a true multi-host bus with a conflict detection and arbitration mechanism. The I2C bus status can be queried in the status register.

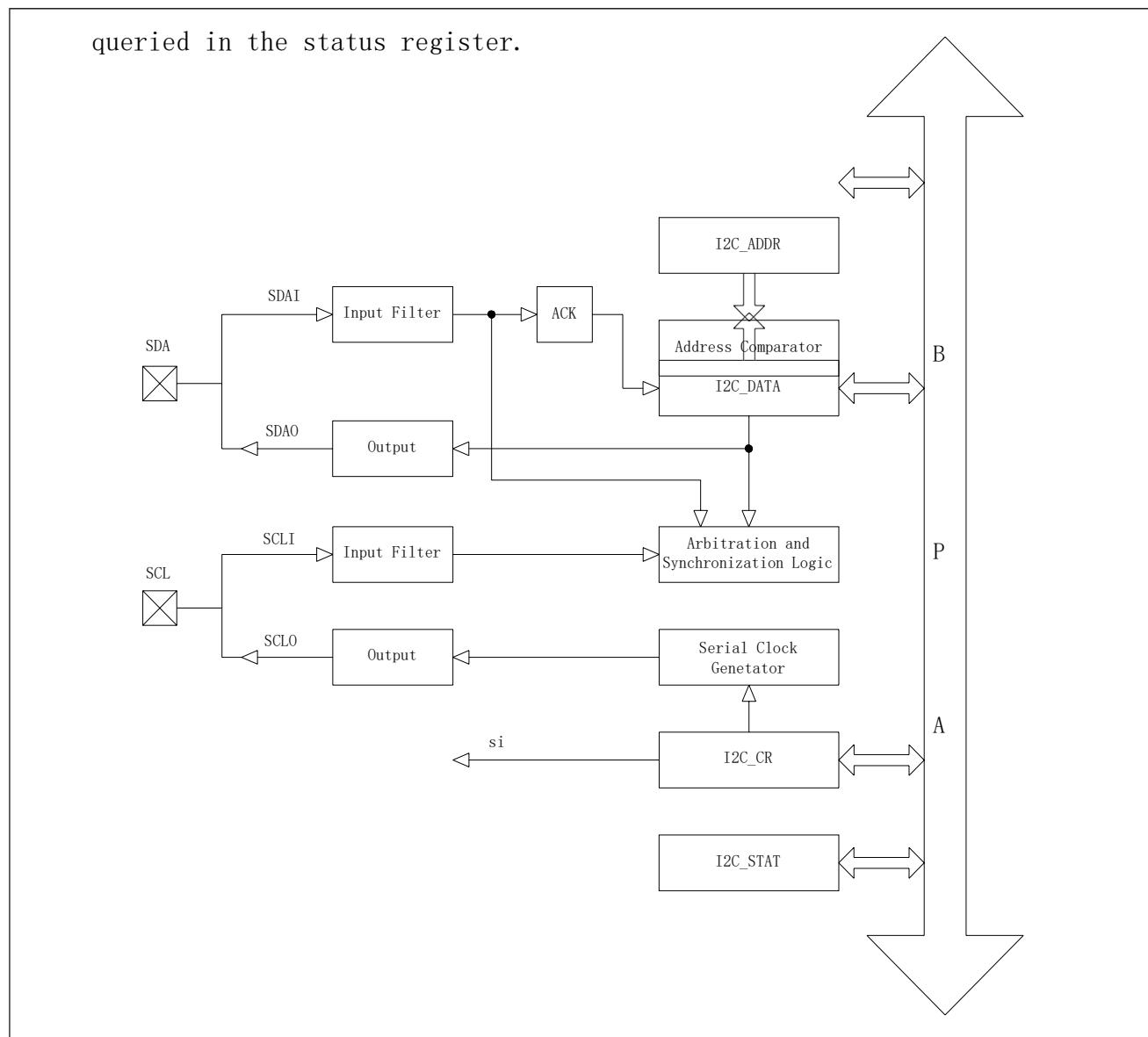


Figure 17-6 I2C Function Module Diagram

17.4.1 Serial Clock Generator

The serial clock generator uses a one-bit8 counter as the baud rate generator. The frequency relationship between the SCL signal and the PCLK signal is $F_{SCL} = F_{PCLK} / 8 / (I2C_TM.tm + 1)$, where $I2C_TM.tm$ should be greater than 0.

The following table shows the output frequency values of the SCL signal when the PCLK frequency is combined with $I2C_TM.tm$.

PCLK (KHz)	I2C_TM.tm						
	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

Table 17-1 I2Cclock signal baud rate

17.4.2 Input Filter

The input signal is synchronized with PCLK, and spike pulses below the PCLK cycle are filtered out.

If the value of $I2C_TM$ is less than or equal to 9, $I2C_CR.H1M$ should be set to 1 when this module is the host; if the value of $I2C_TM$ is greater than 9, $I2C_CR.H1M$ should be set to 0.

When this module is used as a slave, if the ratio of PCLK to SCL frequency is less than or equal to 30, $I2C_CR.H1M$ should be set to 1.

$H1M$ should be set to 0 if the ratio of PCLK to SCL frequency is greater than 10.

17.4.3 Address Comparator

The I2C comparator compares its own slave address with the received bit7 slave address. It can program its own slave address using `#I2C_ADDR` register. It compares the first received bit 8byte or broadcast address (0x00) with the "i2cadr" bit of the "I2C_ADDR" register. If either one is the same, the "I2C_CR" register

The "si" bit of the "si" device will be set and an interrupt request will be generated.

17.4.4 Answer flag bit

The "aa" flag bit of the "I2C_CR" register is the answer flag bit. When the "aa" bit is1, the I2C module returns the answer bit after receiving data, when the "aa" bit is0, the I2C module returns the non-answer bit after receiving data.

17.4.5 Interrupt generators

The "si" flag bit of the "I2C_CR" register is the interrupt flag bit. Whenever the value of the status register (I2C_STAT) changes (except to 0xF8) the "si" flag bit will be set1. When an interrupt is generated, the Status Register (I2C_STAT) is queried to find out the status of the I2C bus and to determine the actual source of the interrupt. The "si" flag bit must be cleared by software in order to proceed to the next step.

17.4.6 Working mode

The I2C component enables bidirectional data transfer of bits8 at rates up to 100Kbps in standard mode, 400Kbps in high speed mode, and 1Mbps in super high speed mode, and can operate in four modes: host transmit mode, host receive mode, slave receive mode, and slave transmit mode. There is also a special mode broadcast call mode, which operates in a similar way to the slave receive mode.

- Host sending mode

The host sends multiple bytes to the slave and the host generates the clock, so you need to fill in the set value in I2C_TM. I2C_CR.sta needs to be set in the host transmit mode1. If the bus is free, the host initiates a start bit START. if successful I2C_CR.si is set1 . Next, the slave address and the write bit (SLA+W) are written to I2C_DATA, the "si" bit is cleared and

SLA+W is issued on the bus.

When the host sends SLA+W and receives the slave answer bit ACK, "si" is set1. Next, the data is sent according to the user-defined format. After all data is sent, the I2C_CR.sto is set1, the "si" bit is cleared and the STOP signal is sent, or a repeat start signal can be sent for a new data transmission.

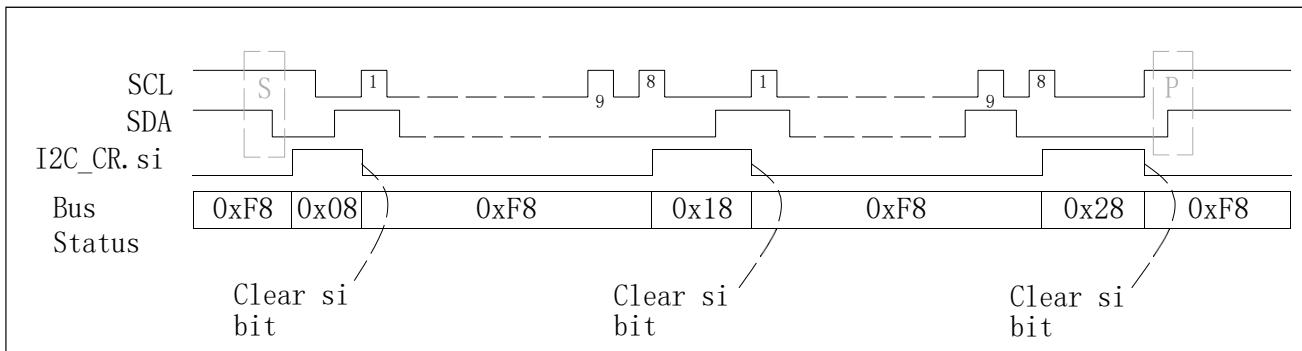


Figure 17-7 Master transmit mode data synchronization diagram

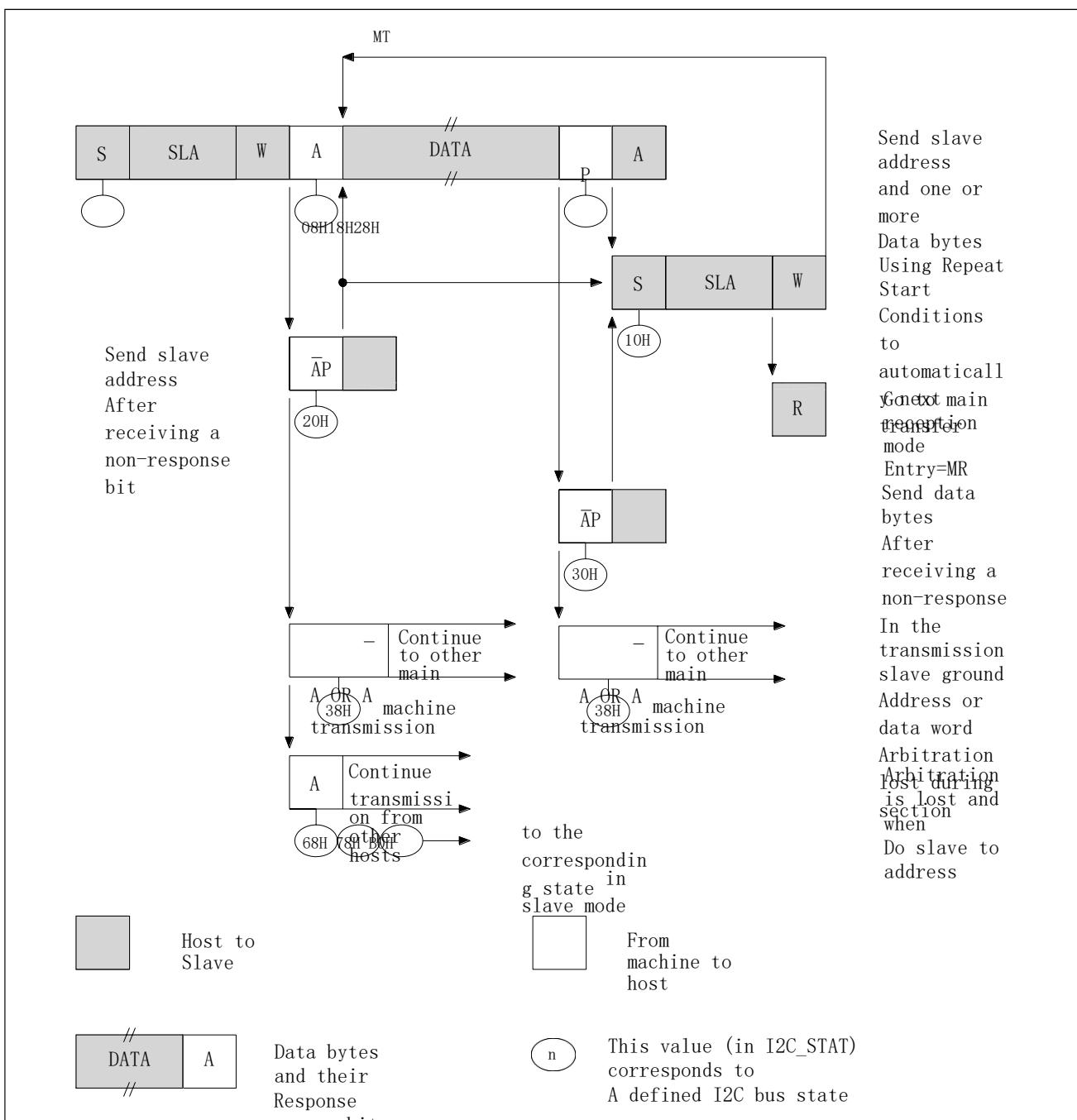


Figure 17-8 I₂C Host Transmit Status Diagram

- Host reception mode

Host receive mode, the slave transmits the data. The initialization settings are the same as for the master transmit mode. After the master sends the start bit, I2C_DATA should be written to the slave address and the "read bit" (SLA+R). After receiving the slave answer bit ACK, I2C_CR.si is set1. If I2C_CR.aa is set, the host returns the answer bit after receiving the data; if I2C_CR.aa is set1, the host does not return the I2C_CR.aa

SCL answer NACK after receiving the data.⁸ 1 8 P

SDA

I2C_CR.si

Bus Status

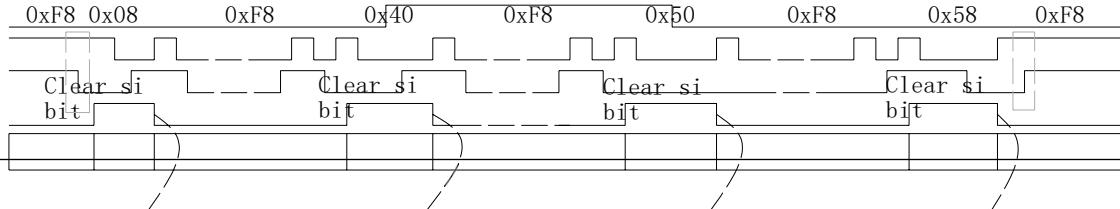


Figure 17-9 Master Receive Mode Data Synchronization Diagram

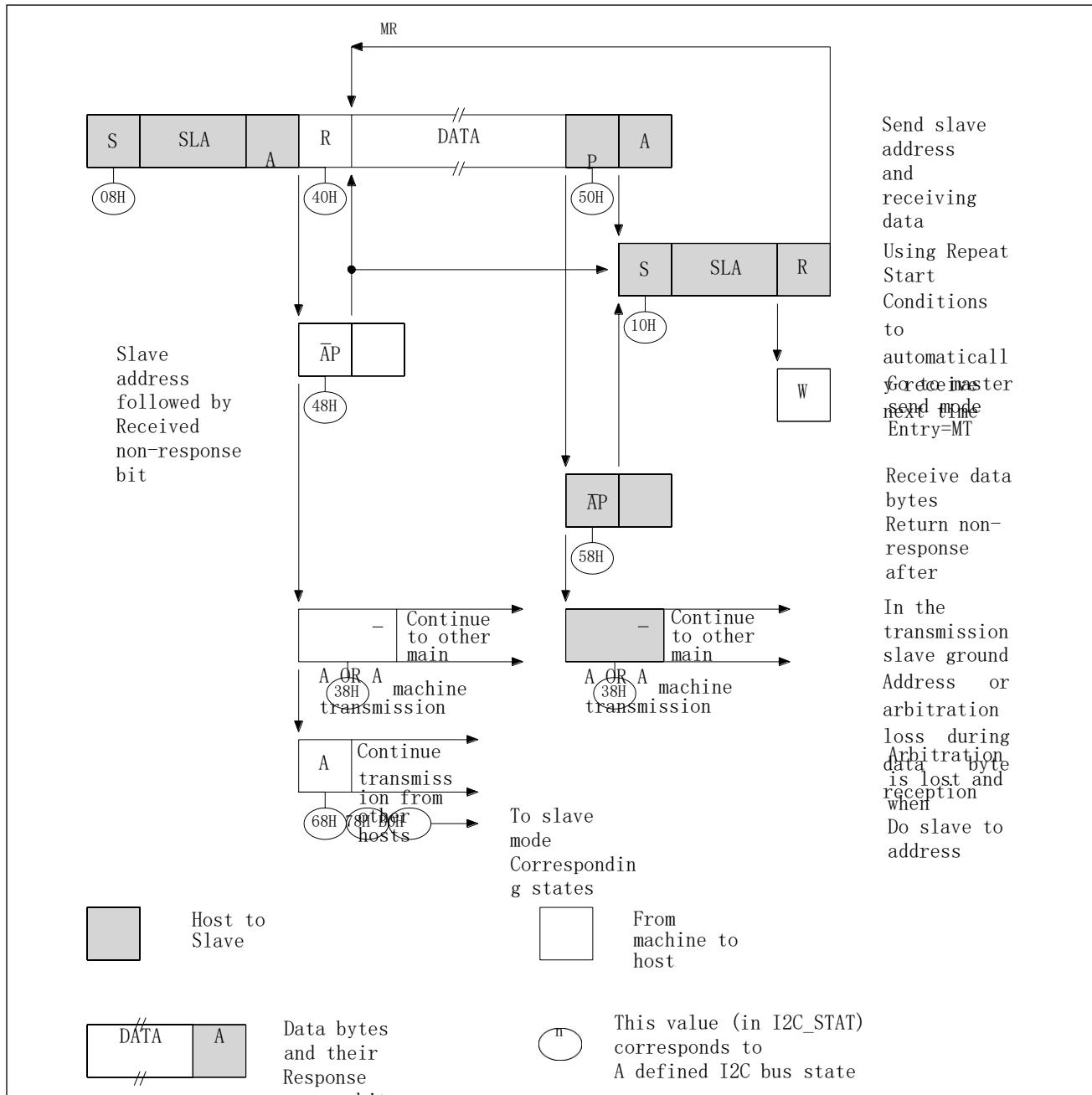


Figure 17-10 I2C Host Receive Status Diagram

- Slave receive mode

In slave receive mode, the slave receives the data from the host. Before the transmission starts, I2C_ADDR should be written to the slave's address and I2C_CR.aa is set to respond to the host's addressing. After the above initialization, the slave enters idle mode and waits for the "write" signal (SLA+W). If the host fails to arbitrate, it will also go directly to slave receive mode. When the slave is addressed by the "write" signal SLA+W, the "si" bit needs to be cleared in order to receive data from the host. If I2C_CR.aa=0 during transmission, the slave will return the no

answer bit NACK in the next byte and the slave will become an unaddressed slave, the contact with the host will be terminated and no more SCL S 1 9 8 1 9 8 P data will be received, and I2C_DATA will keep the previously received data.

I2C_CR.si Bus Slave address identification ~~can~~ be restored by setting "aa", which means Status that the "aa" bit temporarily separates the I2C module from the I2C bus.
Clear si Clear si bit bit

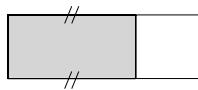
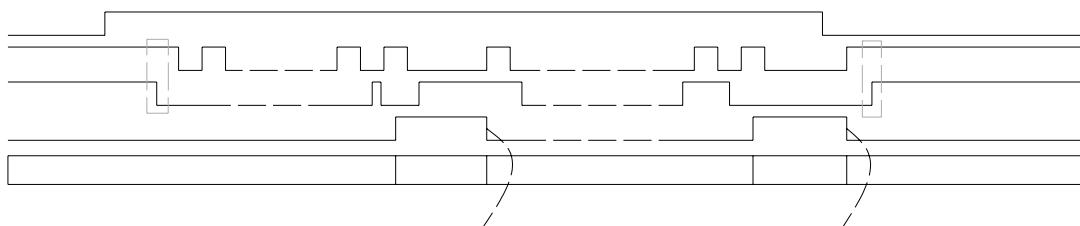


Figure 17-11 Data synchronization diagram from receive mode

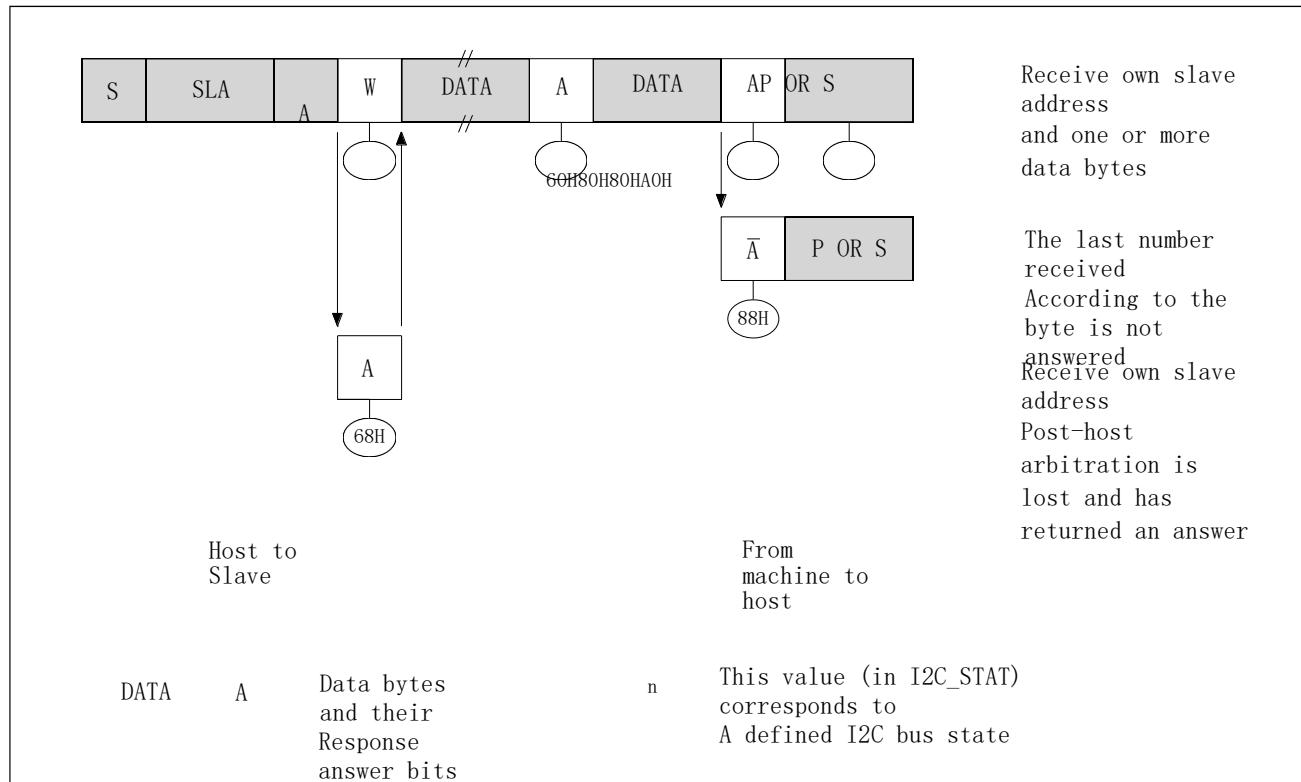


Figure 17-12 Slave Receive Status Diagram

- Slave transmit mode

In slave transmit mode, data is sent from the slave to the host. After initializing the I2C_ADDR and I2C_CR.aa values, the device waits until its address is addressed by the "read" signal (SLA+R). If the host fails to arbitrate, it can also enter slave transmit mode.

When the slave is addressed by the "read" signal SLA+R, "si" needs to be cleared to send data to the host. Normally the host will return an answer bit for each byte of data it receives.

If I2C_CR.aa is cleared during the transfer, the slave will send the last byte of data and send the full data1 in the next transfer and turn itself

into an unaddressed slave.

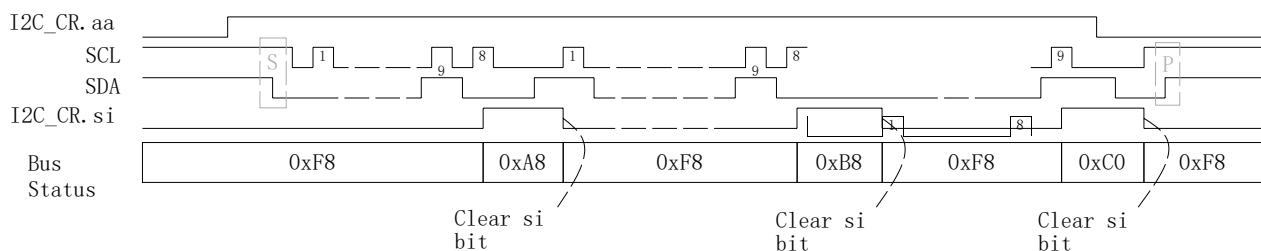


Figure 17-13 Data synchronization diagram from transmit mode

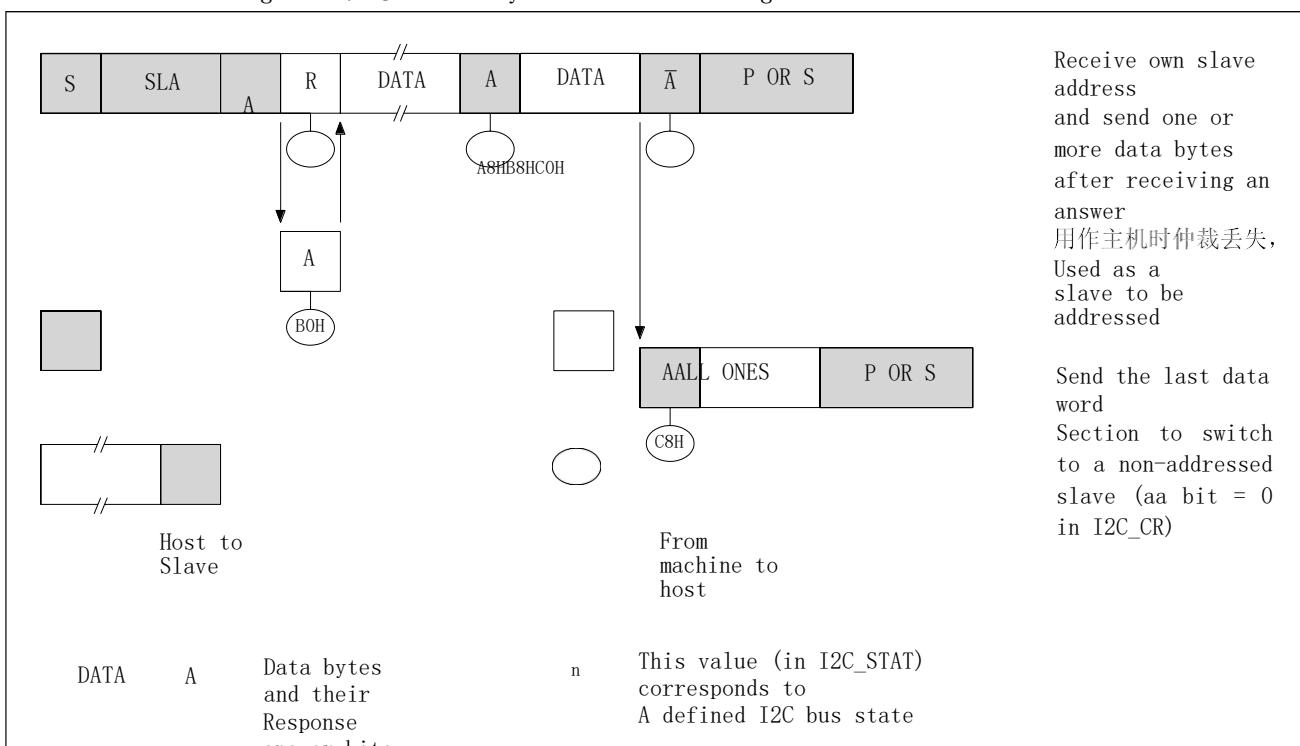


Figure 17-14I2Cslave transmit status diagram

- Broadcast call mode

Broadcast call mode is a special slave receive mode, addressed as 0x00, with slave address and read/write. When I2C_ADDR.GC and I2C_CR.aa are set1, the receive broadcast call mode is enabled. The I2C_STAT value in this mode is different from the normal slave receive mode I2C_STAT value.

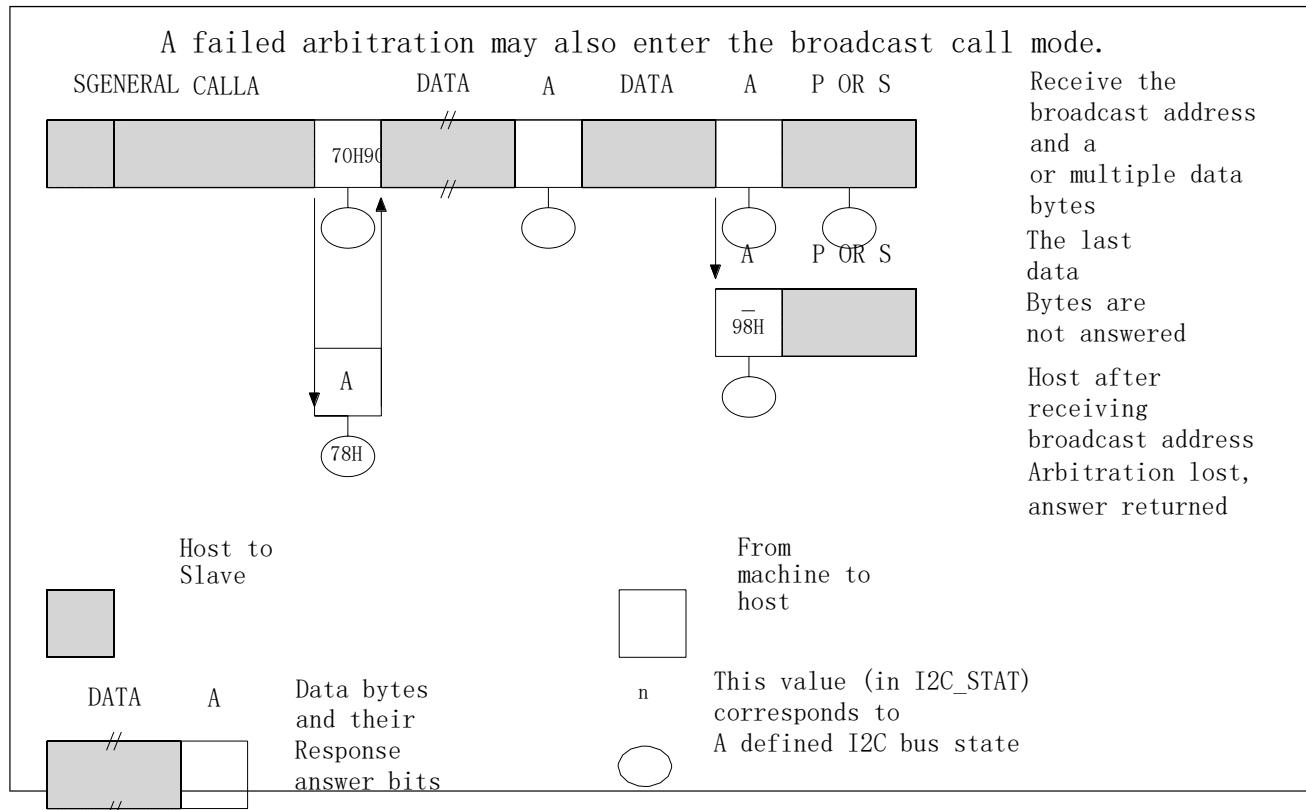


Figure 17-15 I2CBroadcast Call Status Diagram

17.4.7 Status code representation

There are two special states in the I2C status register: F8H and 00H.

F8H: This status code indicates that no relevant information is available, because the serial interrupt flag "si" is not yet set. This occurs between other states and when the I2C module has not yet started performing a serial transfer. 00H: This status code indicates that a bus error has occurred during an I2C serial transfer. A bus error is generated when a start or stop condition is present in an illegal position of the format

frameState	These illegal locations are address bytes, data bytes or answer bits during the serial transfer. Bus errors are also generated when external ription
Master sending mode	
when "s08H" set.	Sent start condition
10H	Duplicate start conditions have been sent
18H	SLA+W has been sent, ACK has been received
20H	SLA+W has been sent, non-ACK has been received
28H	Data in I2C_DATA has been sent, ACK has been received
30H	Data in I2C_DATA has been sent, non-ACK has been received
38H	Arbitration is lost when SLA+ reads, writes, or writes data bytes
Master receiver mode	
08H	Sent start condition
10H	Duplicate start conditions have been sent
38H	Arbitration lost in non-ACK
40H	SLA+R has been sent, ACK has been received
48H	SLA+R has been sent, non-ACK has been received
50H	Data bytes received, ACK returned
58H	Data bytes received, non-ACK returned
From reception mode	
60H	has received its own SLA+W, has returned ACK

68H	Master when in SLA+ read and write lost arbitration, has received its own SLA+W, has returned ACK
80H	The previous addressing uses its own slave address, has received data bytes, and has returned ACK
88H	The previous addressing uses its own slave address, has received data bytes, and has returned a non-ACK
A0H	Receive a stop condition or repeat a start condition when statically addressed
From sending mode	
A8H	has received its own SLA+R, has returned ACK
B0H	When the host when lost to arbitration, has received its own SLA+R, has returned ACK
B8H	Data sent, ACK received
C0H	Data bytes sent, non-ACK received
C8H	Loaded data bytes have been sent and ACKs have been received
Broadcast call mode	
70H	Broadcast address received (0x00) ACK returned
78H	Master when in SLA+ read/write loss arbitration, has received broadcast address, has returned ACK
90H	The previous addressing uses the broadcast address, has received data bytes, and has returned an ACK
98H	Previous addressing uses broadcast address, has received data bytes, has returned non-ACK
A0H	Receive a stop condition or repeat a start condition when statically addressed
Remaining Miscellaneous Status	
F8H	No relevant status information available, si=0
00H	Bus error during transmission, or external disturbance causing I2C to enter an undefined state

Table 17-2I2CStatus Code Expressions

17.5 Programming Examples

17.5.1 Host sending example

Step1: Mapping SCL, SDA to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring SCL, SDA pins to open-drain output mode.

Step2: Set PERI_CLKEN.I2C.tol1 enable the I2C module clock. Step3: Write 0, 1 to reset the I2C module in order to PERI_RESET. Step6: Set I2C_CR.ens to 1 enable the I2C module.

Step7: Set I2C_CR.sta to 1, and the bus tries to send the Start signal.

Step8: Wait for the I2C_CR.si to change to 1, and the Start signal has been sent to the bus.

Step9: Query I2C_STAT, if the register value is 0x08 or 0x10, continue to execute the next step, otherwise proceed to error processing.

Step10: Write SLA+W to I2C_DATA, set I2C_CR.sta to 0, set I2C_CR.si to 0, and send SLA+W.

Step11: Wait for I2C_CR.si to change to 1, SLA+W has been sent to the bus.

Step12: Query I2C_STAT, if the value of this register is 0x18, continue to execute the next step. Otherwise, proceed to error processing.

Step13: Write the data to be sent to I2C_DATA, set I2C_CR.si to 0, and send the data. Step14: Wait for I2C_CR.si to change to 1, and the data has been sent to the bus.

Step15: Query I2C_STAT, if the value of this register is 0x28, continue to execute the next step. Otherwise, proceed to error processing.

Step16: If the data to be sent is not completed, jump to Step13 to continue the execution.

Step17: Set I2C_CR.sto to 1, set I2C_CR.si to 0, and the bus tries to send Stop signal.

Step18: Wait for I2C_CR.si to change to 1, Stop signal has been sent to the bus.

17.5.2 Host reception example

Step1: Mapping SCLSDA to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring SCL, SDA pins to open-drain output mode.

Step2: Set PERI_CLKEN.I2C.tol1 enable the I2C module clock. Step3: Write0 ,1 to reset the I2C module in order to PERI_RESET. Step6: Set I2C_CR.ens to1 enable the I2C module.

Step7: Set I2C_CR.sta to 1, and the bus tries to send the Start signal.

Step8: Wait for the I2C_CR.si to change to 1, and the Start signal has been sent to the bus.

Step9: Query I2C_STAT, if the register value is 0x08 or 0x10, continue to execute the next step, otherwise proceed to error processing.

Step10: Write SLA+R to I2C_DATA, set I2C_CR.sta to 0, set I2C_CR.si to0 , send SLA+R.

Step11: Wait for I2C_CR.si to change to 1, SLA+R has been sent to the bus.

Step12: Query I2C_STAT, if the register value is 0x40, continue to execute the next step, otherwise proceed to error processing.

Step13: Set I2C_CR.aa to 1, enable the answer flag.

Step14: Set IC_CR.si to0 , the slave sends data, and the host sends ACK or NACK according to IC_CR.aa. Step15: Wait for I2C_CR.si to change to , and read the received data from I2C_DATA.

Step16: Query I2C_STAT, if the register value is 0x50 or 0x58, continue to execute the next step, otherwise proceed to error processing.

Step17: If the data to be received is only missing the last byte, set 2IC_CR. aa to 0, enable the non-response flag. Step18: If the data to be received is not completed, jump to Step14 to continue the execution.

Step19: Set I2C_CR.sto to 1, set I2C_CR.si to 0, and the bus tries to send Stop signal.

Step20: Wait for I2C_CR.si to change to 1, Stop signal has been sent to the bus.

17.5.3 Slave reception example

Step1: Mapping SCL/SDA to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring SCL, SDA pins to open-drain output mode.

Step2: Set PERI_CLKEN.I2C to 1 enable the I2C module clock. Step3: Write 0, 1 to reset the I2C module in turn to PERI_RESET.

Step5: Configure I2C_ADDR as the slave address.

Step6: Set I2C_CR.aa to 1, enabling the answer flag. Step7: Wait for I2C_CR.si to change to 1, addressed by SLA+W.

Step8: Query I2C_STAT, if the register value is 0x60, continue to execute the next step, otherwise proceed to error processing.

Step9: Set IC_CR.si to 0, the host sends data, and the slave returns ACK or NACK according to IC_CR.aa. Step10: Wait for I2C_CR.si to change to 1, and read the received data from I2C_DATA.

Step11: Query I2C_STAT, if the register value is 0x80, continue to execute the next step, otherwise, error processing

Step12: If the data to be received is not completed, jump to Step9 to continue the execution.

Step13: Set I2C_CR.aa to 0, set I2C_CR.si to 0.

17.5.4 Slave sending example

Step1: Mapping SCL/SDA to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring SCL, SDA pins to open-drain output mode.

Step2: Set PERI_CLKEN.I2C to 1 enable the I2C module clock. Step3: Write 0, 1 to reset the I2C module in turn to PERI_RESET.

Step5: Configure I2C_ADDR as the slave address. Step6: Set I2C_CR.aa to 1, enabling the answer flag. Step7: Wait for I2C_CR.si to change to 1, addressed by SLA+R.

Step8: Query I2C_STAT, if the value of this register is 0xA8, continue to execute the next step, otherwise proceed to error processing.

Step9: Write the data to be sent to I2C_DATA, set I2C_CR.si to 0, and send the data. Step10: Wait for I2C_CR.si to change to 1, and the data has been sent to the bus.

Step11: Query I2C_STAT, if the value of this register is 0xB8 or 0xC0, continue to execute the next step, otherwise proceed to error processing.

Step12: If the data to be sent is not completed, jump to Step9 to continue the execution.

Step13: Set I2C_CR.aa to 0, set I2C_CR.si to 0.

17.6 Register Description

Register List

I2C base address: 0x40000400

Offset	Register Name	Access	Register Description
0x00	I2C_TMRUN	RW	I2C baud rate counter enable register.
0x04	I2C_TM	RW	I2C baud rate counter configuration register.
0x08	I2C_CR	RW	I2C configuration register.
0x0c	I2C_DATA	RW	I2C data register.
0x10	I2C_ADDR	RW	I2C address register.
0x14	I2C_STAT	RO	I2C status register.

Table 17-3 Register List

17.6.1 I2C baud rate counter enable register (I2C_TMRUN)

Address Offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

position	Marker	Function Description
31: The1	Reserved	
0	tme	Baud rate counter enable . 0Disable 1- Enable

17.6.2 I2C baud rate counter configuration register (I2C_TM)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								tm							
								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	tm	tm:Baud rate counter configuration value. $F_{SCL} = F_{PCLK} / 8 / (tm + 1)$, where tm > 0

17.6.3 I2C configuration register (I2C_CR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ens	sta	sto	si	aa	Res	h1m	
								RW	RW	RW	RW	RW			RW

position	Marker	Function Description
31:7	Reserved	
6	ens	I2C module enable control 0 – Prohibition 1 – Enable
5	sta	I2C bus control 0 – No function 1 – Send START to the bus
4	sto	I2C bus control 0 – No function 1 – Send STOP to the bus
3	si	I2C interrupt flag Reads 1 that an I2C interrupt has occurred Write 0, I2C to perform a step-by-step operation
2	aa	Answer control bit 0 – Send NAK 1 – Send ACK
1	Reserved	
0	h1m	I2C filtering parameters configuration

		<p>0 – Advanced filtering for higher interference immunity</p> <p>1 – Simple filtering for faster communication rates</p> <p>Note: See the [Input Filters] section for details.</p>
--	--	---

17.6.4 I2C data register (I2C_DATA)

Address offset :

0x0c Reset value :

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

i2cdat

RW

position	Marker	Function Description
31:8	Reserved	
7:0	i2cdat	I2C Data Register In I2C send mode, write the data to be sent In I2C receive mode, read the received data

17.6.5 I2C address register (I2C_ADDR)

Address offset :

0x10 Reset value :

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								i2cadr							
Reserved								RW							

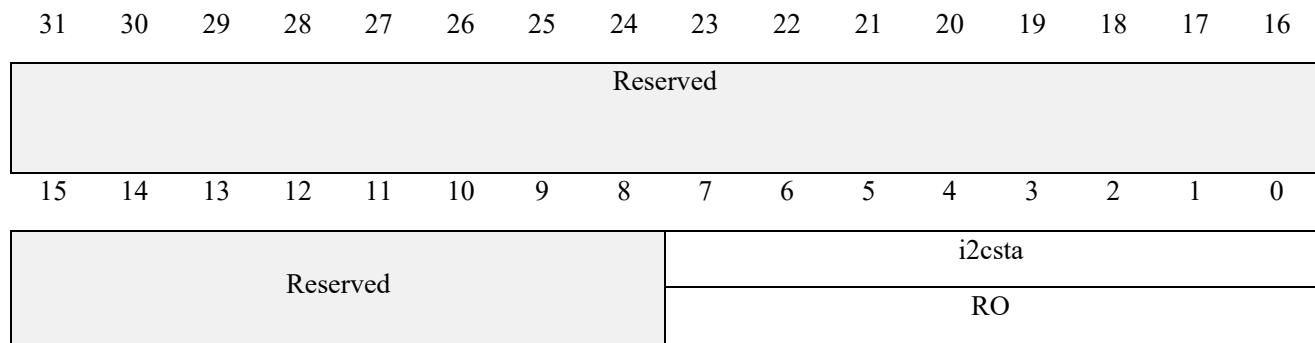
position	Marker	Function Description
31:8	Reserved	
7:1	i2cadr	I2C slave mode address.
0	GC	Broadcast Address Answer Enable 0 – Prohibition 1 – Enable

17.6.6 I2C Status Register (I2C_STAT)

Address offset :

0x14 Reset value :

0x00000000



position	Marker	Function Description
31:8	Reserved	
7:0	i2csta	I2C Status Register See the [Status Code Representation] section for the specific definition of status values

18 Serial Peripheral Interface (SPI)

18.1 SPI Introduction

The SPI interface is a synchronous serial data communication interface that operates in full duplex mode and communicates using the following 4pins: MISO, MOSI, SCK, and CS/SSN. When the SPI is acting as a master, the CS and SCK signals are output to control the communication process. When the SPI is used as a slave, communication is performed under the control of the SSN and SCK signals.

18.2 SPI Key Features

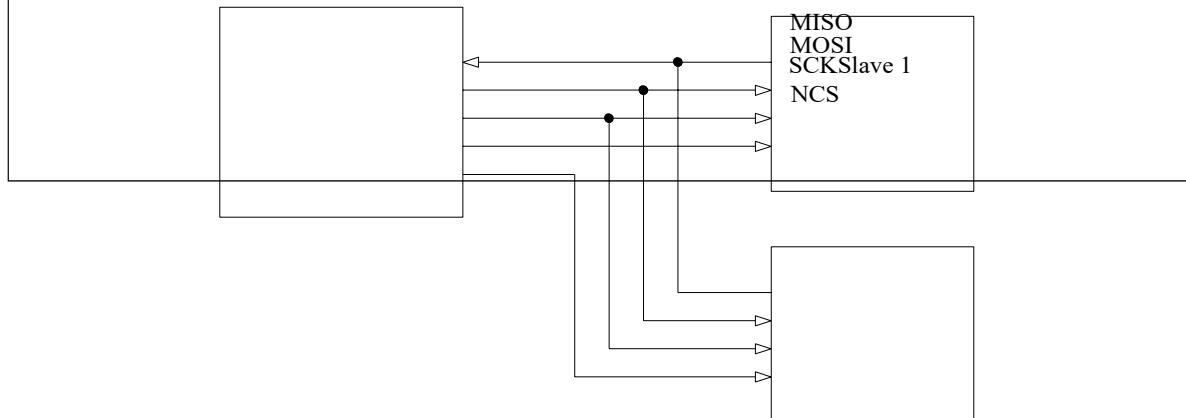
- Support SPI host mode, SPI slave mode
- Supports standard four-wire full-duplex communication
- Supports configuration of serial clock polarity and phase
- Host mode supports various communication rates
- Host mode with a maximum crossover factor of PCLK/2 and a maximum communication rate of 16M bps
- Slave mode with a maximum crossover factor of PCLK/4 and a maximum communication rate of 12M bps
- Fixed frame length in bits8, priority transmission MSB

18.3 SPI Function Description

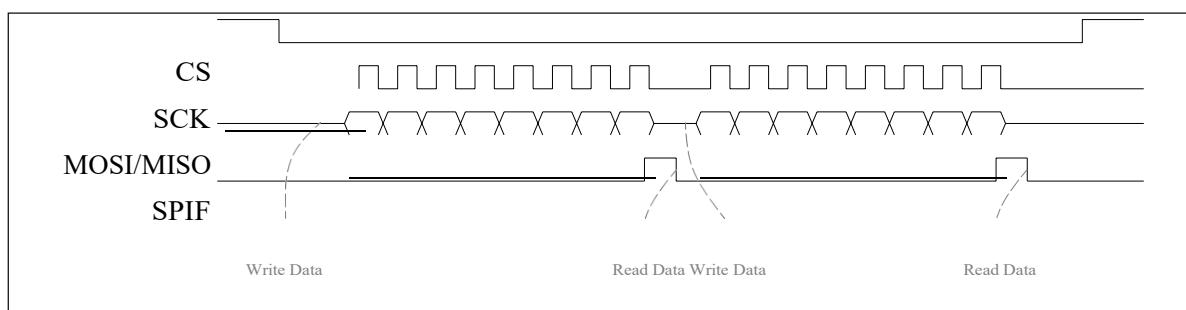
18.3.1 SPI Host Mode

The length of each data frame is fixed in bits8, and the first bit of the sent data is fixed in MSB. setting SPI_CR.mstr to 1, the SPI interface operates in host mode. The SPI transfer is initiated by writing data to the SPI_Data register; the SCK pin automatically generates the serial clock; at the serial clock edge, the data in the shift register is sent to the MOSI pin and the data on the MISO pin is received in the shift register. The output frequency of the SCK pin is controlled by SPI_CR[spr2:spr0], and its output frequency ranges from PCLK /2 to PCLK/128; the output level of the CS pin

is controlled by SPI_SSN.ssn, and the output level of the GPIO pin is controlled by SPI_SSN.ssn. The output level of CS pin is controlled by SPI_SSN.ssn, and the output level of GPIO pin is controlled by GPIO related registers. The typical application block diagram of host mode is shown below.



The host mode communication is schematically shown below, where CPOL=0 and CPHA=0.



18.3.2 SPI Slave Mode

The length of each data frame is fixed in bits8, and the first bit of the received data is fixed in MSB. set SPI_CR.mstr

is 0, the SPI interface operates in slave mode. In this mode, the SCK pin is used as an input pin and the serial clock comes from an external host; the SSN pin is used as an input pin and the chip select signal comes from an external host or is fixed low. A typical application block diagram for slave mode is shown below.

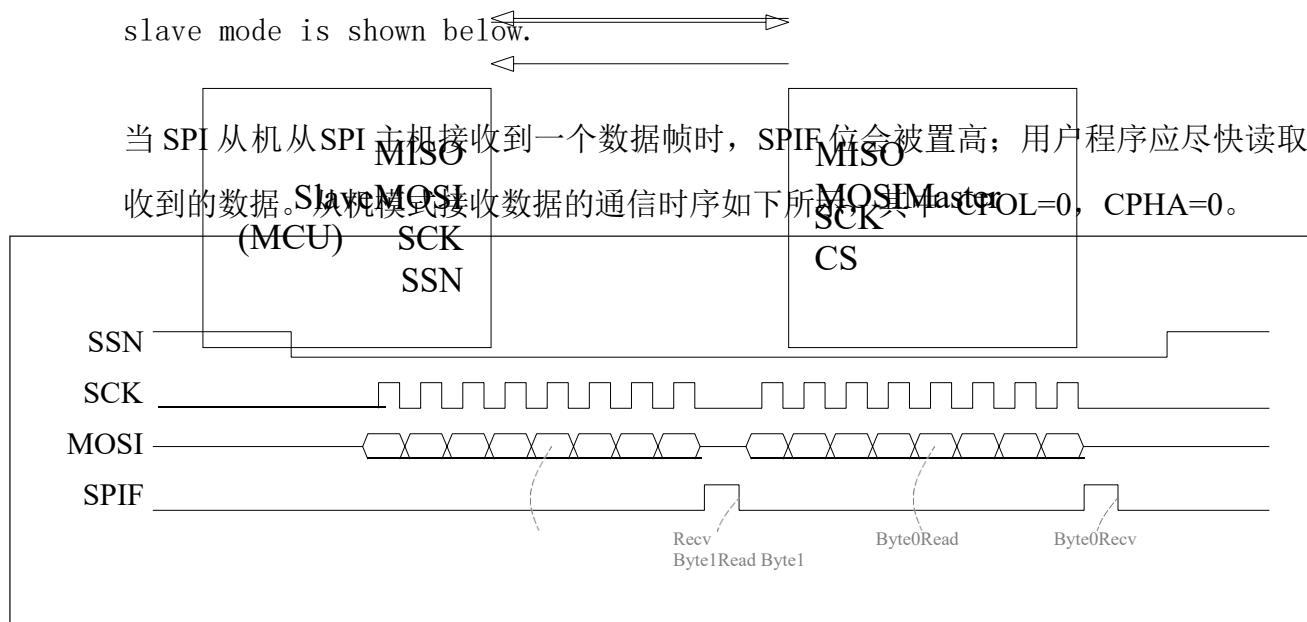


Figure 18-1 Slave reception diagram

当 SPI 从机需要发送数据到主机时，在主机拉低 NSS 之后应尽快向 SPI_DATA 寄存器中写入待发送的第一字节数据；每当查询到 SPIF 标志为 1 时，应尽快读取 SPI_DATA 以清除 SPIF 标志，并向 SPI_DATA 寄存器中写入待发送的后续数据。从机模式发送数据的通信时序如下所示，其中 CPOL=0, CPHA=0。

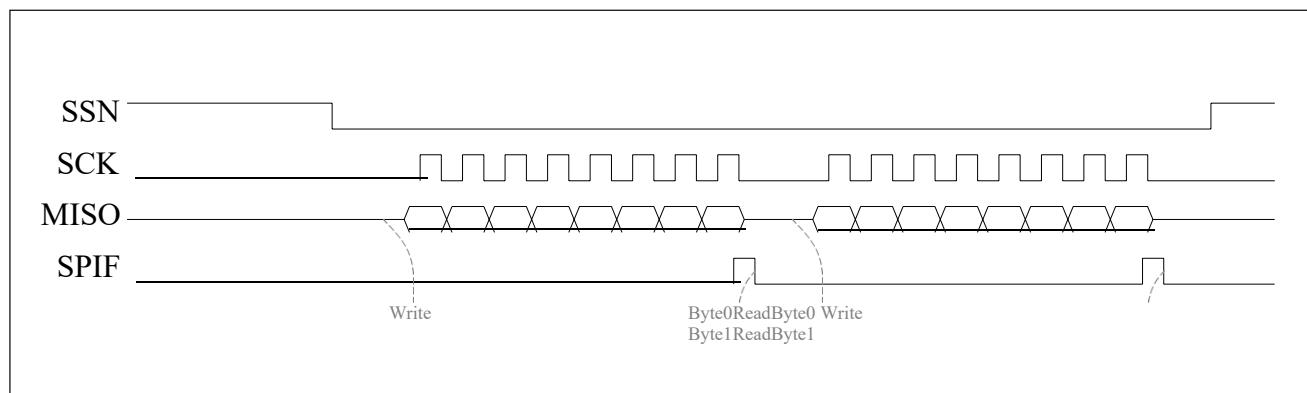


Figure 18-2 Slave transmitting diagram

18.3.3 SPI Data Frame Format

The SPI interface frame format depends on the configuration of the clock polarity bit CPOL and the clock phase bit CPHA.

When CPOL is 0, the SCK line idle state is low. When CPOL is 1, the SCK line idle state is high. When CPHA is 0, the data will be sampled when the first SCK clock transition signal jumps. When CPHA is 1, the data is sampled when the second SCK clock signal jumps.

The SPI interface host frame format is shown in the following figure.

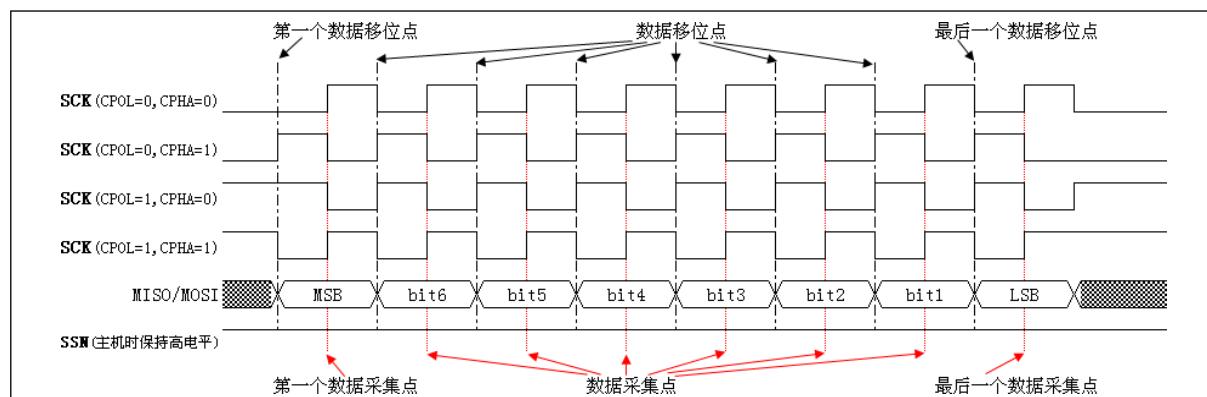


Figure 18-3 Host Mode Frame Format

The SPI interface slave frame format is shown in the following figure.

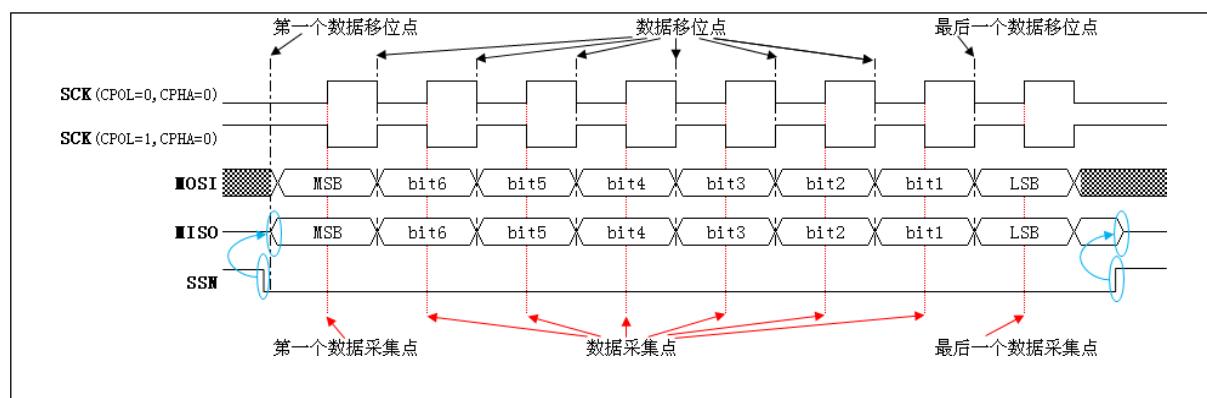


Figure 18-4 Data frame format when the slave CPHA is 0

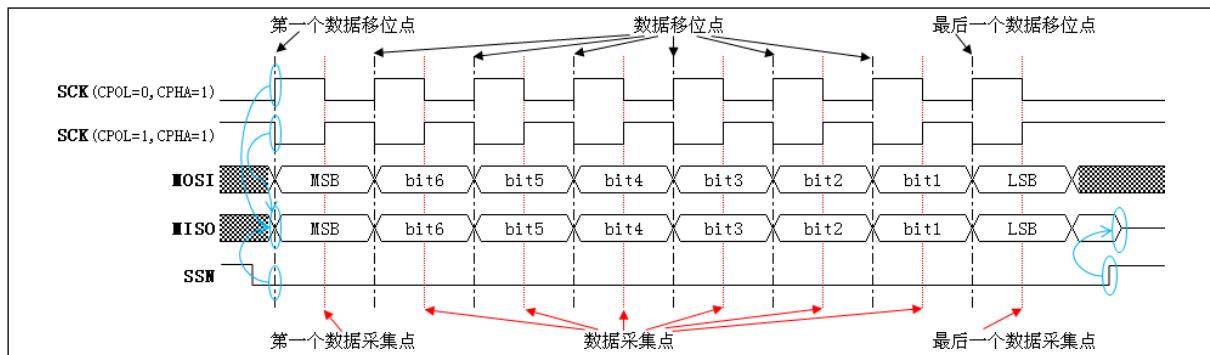


Figure 18-5 Data frame format when the slave CHPA is 1

18.3.4 SPI Status Flags and Interrupts

The SPI generates the following three status flags during operation, and their generation conditions and clearing methods are shown below.

- SPIF is set in hardware when the transfer of a data frame is completed. A read of the SPI_DATA register clears this flag bit.
- When the SPI is operating in host mode and the external SSN input is low, SPI_STAT.mdf is set by hardware, indicating that another SPI host is occupying the bus. When the SSN input is high, SPI_STAT.mdf is automatically cleared by hardware.
- When the SPI is operating in slave mode, if the SSN pin is pulled high while data transfer is in progress, the SPI_STAT.sserr will be set. This flag can be cleared by setting SPI_CR.spen to . If the SPI interrupt vector is allowed , an interrupt can be generated in either of the following cases.
 - SPI transfer is complete, i.e., SPI_STAT.SPIF is 1
 - SPI's host mode error, i.e. SPI_STAT.mdf is 1

18.3.5 SPI Multi-Computer System Configuration Instructions

- When the SPI module is used as a master and operates in a single-host system, the slave can be controlled via the SPI_CS pin or the GPIO pin. When the SPI_CS pin is selected as the slave's chip select signal0,1 set

SPI_SSN.ssn to select the slave and set SPI_SSN.ssn to release the slave.

When GPIO pin is selected as the slave's chip select signal, set the corresponding bit of GPIOx_OUT register to 0 to select the slave, and set the corresponding bit of GPIOx_OUT register to 1 to release the slave.

- When the SPI module is slave, configure the source of the SPI_SSN as needed (see GPIO Port Auxiliary Controller for details). When SSN is low, this slave is selected for communication; when SSN is high, this slave is in the not

Select the neutral state.

- When the SPI mode works in multi-master multi-slave, all the slave chip select signals are connected through the GPIO pins, and the master must also monitor the bus for occupancy by connecting the GPIO pins to the SSN signals of other masters. SSN is high; output low from GPIO0 to notify Master1 to release the SPI bus; output low from GPIO2 to select Slave1;

communicate with Slave1; output high from GPIO2 to release Slave1; output high from GPIO0 to release Master1.

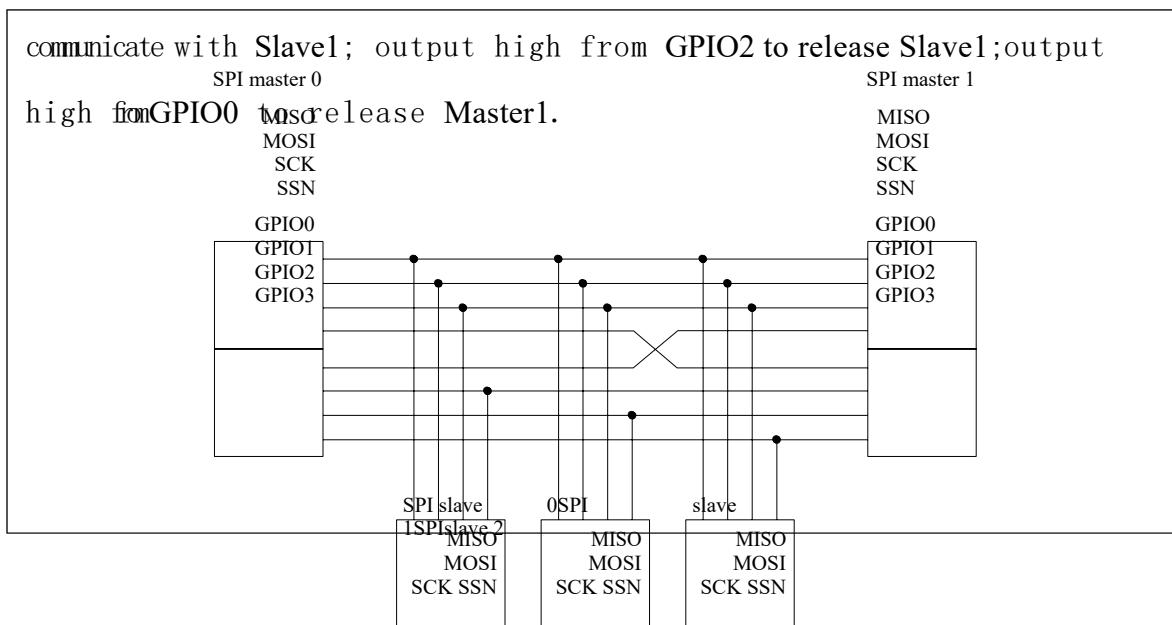


Figure 18-6 Schematic of SPI Multi-Master/Multi-Slave System

18.3.6 SPI Pin Configuration Description

SPI can maintain some or all of its functionality in some special pin configurations.

The details are listed in the following table (checkmark means the pin is configured and used, a blank means the pin is not configured)

	SPI_CS(main) / SPI_SSN(from)	SCK	MOSI	MISO	Function Description
Host Mode	✓	✓	✓	✓	General Configuration All hosts function normally
		✓	✓	✓	All hosts function normally
	✓	✓	✓		Host sending function is normal
	✓	✓		✓	Host reception function is normal
		✓	✓		Host sending function is normal
		✓		✓	Host reception function is normal
Slave Mode	✓	✓	✓	✓	General Configuration All slaves function normally
	✓	✓	✓		Slave reception function is normal
	✓	✓		✓	Slave sending function is normal
	✓ Fixed low level	✓	✓	✓	All slaves function normally

	✓ Fixed low level	✓	✓		Slave reception function is normal
	✓ Fixed low level	✓		✓	Slave sending function is normal

Table 18-1SPIPin Configuration Description Table

Caution.

- Cases not listed in the table are not supported at this time.
- In host mode, even if the SPI_CS chip select output is not used, you need to set SPI.SSN to before sending data 1and SPI.SSN to0 after sending data.
- When the slave mode and the chip select input is fixed low, SPI_CR.cpha=1 must be satisfied to maintain normal function.

18.4 SPI Programming Example

18.4.1 SPI Host Send Example

Step1: Mapping CS/SCK/MISO/MOSI to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring CS/SCK/MOSI pins as output mode and MISO pins as input mode. step2: Setting SPI_CR.mstr to 1, making SPI work in host mode.

Step3: Configure SPI_CR[spr2:spr0] to make the clock rate of SCK output meet the application requirements. step3: Configure SPI_CR.cpol and SPI_CR.cpha to make the data frame format meet the application requirements. step3: Set SPI_CR.spen to 1 to enable the SPI interface.

Step6: Set SPI_SSN.ssn to 0, so that the CS pin outputs low to select the slave. Step7: Write the data to be sent to SPI_DATA and wait for the SPIF to change to 1. Step8: Read SPI_DATA to clear the SPIF flag. Step9: If the data to be sent is not completed, skip to Step7 to continue the execution. Step10: Set SPI_SSN.ssn to 1, so that the CS pin outputs high to release the slave. Caution.

- GPIO can be used instead of CS to realize chip select output, which is mostly used in multi-computer communication system.
- SPI_SSN.ssn must be set to 0 during the transfer and SPI_SSN.ssn must be set to 1 after the transfer is completed.

18.4.2 SPI host reception example

Step1: Mapping CS/SCK/MISO/MOSI to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring CS/SCK/MOSI pins as output mode and MISO pins as input mode. step2: Setting SPI_CR.mstr to 1, making SPI work in host mode.

Step3: Configure SPI_CR[spr2:spr0] to make the clock rate of SCK output meet the application requirements. step3: Configure SPI_CR.cpol and SPI_CR.cpha to make the data frame format meet the application requirements. step3: Set SPI_CR.spen to enable the SPI interface.

Step6: Set SPI_SSN.ssn to0 , so that the CS pin outputs low to select the slave. Step7: Write any data to SPI_DATA to trigger the host to send SCK. Step8: Query and wait for SPI_STAT.SPIF to change to 1, which has received the data sent by the slave.

Step9: Read out the received data from SPI_DATA.

Step10: If the data to be received is not completed, skip to

Step7 to continue the execution. Step11: Set SPI_SSN.ssn to 1,

so that the CS pin outputs high to release the slave.

Attention.

- GPIO can be used instead of CS to realize chip select output, which is mostly used in multi-computer communication system.
- SPI_SSN.ssn must be set to 0 during the transfer and SPI_SSN.ssn must be set to 1 after the transfer is completed.

18.4.3 SPI Slave Send Example

Step1: Mapping SSN/SCK/MISO/MOSI to the required pins as described in the GPIO chapter Pin Digital Multiplexing Function; and configuring SSN/SCK/MOSI pins as input mode and MISO pins as output mode. see GPIO Port Auxiliary Controller for details of SSN pin source.

Step2: Set SPI_CR.mstr to 0, so that the SPI works in slave mode.

Step3: Configure SPI_CR.cpol and SPI_CR.cpha to make the data frame format meet the application requirements. step3: Set SPI_CR.spen to 1 to enable the SPI interface.

Step5: Query and wait for the SSN pin to pull low and the host to select the SPI slave. step6: Write the data to be sent to SPI_DATA and wait for SPIF to change to 1. step7: Read SPI_DATA to clear the SPIF flag.

Step8: When the SSN pin is low and the data to be sent is not yet completed, jump to Step6. Step9: The query waits for the SSN pin to pull high and the host releases the SPI slave.

18.4.4 SPI Slave Receive Example

the GPIO chapter Pin Digital Multiplexing Function; and configuring SSN/SCK/MOSI pins as input mode and MISO pins as output mode. see GPIO Port Auxiliary Controller for details of SSN pin source.

Step2: Set SPI_CR.mstr to 0, so that the SPI works in slave mode.

Step3: Configure SPI_CR.cpol and SPI_CR.cpha to make the data frame format meet the application requirements. step3: Set SPI_CR.spen to enable the SPI interface.

Step5: The query waits for the SSN pin to be pulled low and the host selects the SPI slave.

Step6: Query and wait for SPI_STAT.SPIF to change to 1, which has received the data sent by the host. **step7:** Read the received data from SPI_DATA.

Step8: If the data to be received is not completed, jump to **Step6** to continue the execution.

Step9: The query waits for the SSN pin to pull high and the host releases the SPI slave.

18.6 SPI Register Description

Register List

SPI base address: 0x40000800

Offset	Register Name	Access	Register Description
0x00	SPI_CR	RW	SPI Configuration Register
0x04	SPI_SSN	RW	SPI Chip Select Configuration Register
0x08	SPI_STAT	RO	SPI Status Register
0x0c	SPI_DATA	RW	SPI Data Register

Table 18-2SPIRegister List

18.6.1 SPI Configuration Register (SPI_CR)

Address Offset: 0x00

Reset value: 0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								spr2	spen	Res	mstr	cpol	cpha	spr1	spr0
								RW	RW		RW	RW	RW	RW	RW

position	Marker	Function Description
31:8	Reserved	
7	spr2	Baud rate selection bit 2 Reference spr0.
6	spen	SPI Module Enable Control 0 – Prohibition 1 – Enable
5	Reserved	
4	mstr	SPI Operating Mode Configuration 0 – Slave Mode 1 – Host Mode
3	cpol	SCK line idle state configuration 0 – Low Level 1 – High level
2	cpha	Clock phase configuration 0 – First edge 1 – Second edge
1	spr1	Baud rate selection bit 1 Reference spr0

0	spr0	Baud rate selection bit 0				
		spr2	spr1			spr0SCKRate
		0	0			0PCLK/2
		0	0			1PCLK/4
		0	1			0PCLK/8
		0	1			1PCLK/16
		1	0			0PCLK/32
		1	0			1PCLK/64
		1	1			0PCLK/128
		1	1			1Reserved

Table 18-3 Host mode baud rate selection

18.6.2 SPI Chip Select Configuration Register (SPI_SSN)

Address offset :

0x04 Reset value :

0x000000FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															ssn RW

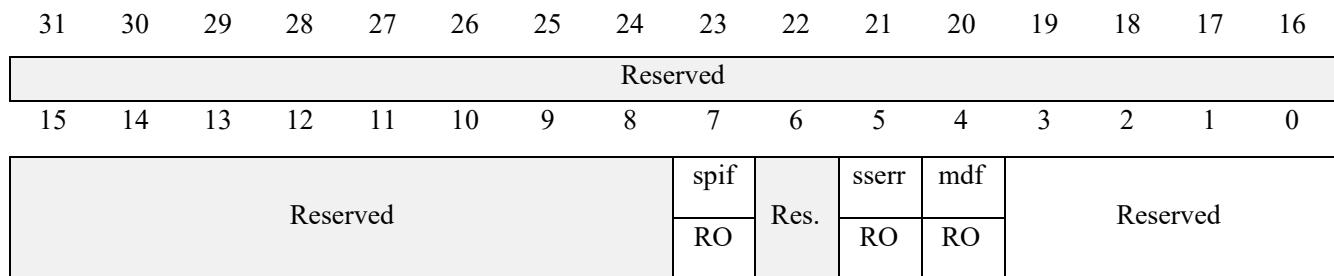
position	Marker	Function Description
31:1	Reserved	
0	ssn	SPI_CS output level configuration in host mode 0: SPI_CS port output low 1: SPI_CS port output high

18.6.3 SPI Status Register (SPI_STAT)

Address offset :

0x08 Reset value :

0x00000004



position	Marker	Function Description
31:8	Reserved	
7	spif	<p>Transmission completion flag</p> <p>1: SPI bus has completed a byte transfer</p> <p>0: SPI bus in transit</p>
6	Reserved	
5	sserr	Slave mode SSN error flag
4	mdf	<p>Conflict flag when in host mode</p> <p>1: SSN pin level is low</p> <p>0: SSN pin level is high</p>
3:0	Reserved	

18.6.4 SPI data register (**SPI_DATA**)

Address offset :

0x0c Reset value :

0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								spdat							
								RW							

position	Marker	Function Description
31:8	Reserved	
7:0	spdat	<p>Data Register</p> <p>In transmit mode, writing the bytes to be sent to this register; in receive mode, reading the received bytes from this register.</p>

19 Clock Calibration Module (CLKTRIM)

19.1 CLK_TRIM Introduction

The CLK_TRIM (Clock Trimming) module is a circuit dedicated to clock calibration/monitoring. In the calibration mode, a precise clock source is selected to calibrate an imprecise clock source, and the parameters of the imprecise clock source are repeatedly calibrated and adjusted until the frequency of the calibrated clock source meets the accuracy requirement. The count value in the calibration mode will have some error, but within the allowable accuracy error. In the monitoring mode to select a stable clock source to monitor the system operating clock, in the set monitoring period, monitor the system operating clock for failure and generate interruptions. In both calibration mode and monitoring mode, the required clock source must be initialized and enabled, please refer to Chapter 4 System Controller for the specific configuration process.

19.2 CLK_TRIM Main Features

CLK_TRIM supports the following features.

- Calibration Mode
- Monitoring Mode
- 32 Bit reference clock counter can be loaded with initial value
- 32 Bit to be calibrated clock counter configurable overflow value
- 6 Reference clock sources
- 4 Various clock sources to be calibrated
- Support interrupt method

19.3 CLK_TRIM Function Description

19.3.1 CLK_TRIM Calibration Mode

The calibration mode is mainly used to select an accurate clock source as a reference clock to calibrate an imprecise clock source to be calibrated.

The software is calibrated repeatedly in accordance with the following procedure to adjust the parameters of the clock source to be calibrated until the clock source to be calibrated meets the frequency accuracy requirements.

19.3.1.1 Operation process

1. Set the CLKTRIM_CR.refclk_sel register to select the reference clock.
2. Set the CLKTRIM_CR.calclk_sel register to select the clock being calibrated.
3. Set the CLKTRIM_REFCON.rcntval register to the calibration time.
4. Set the CLKTRIM_CR.IE register to enable interrupts.
5. Set CLKTRIM_CR.trim_start register to start calibration.
6. The reference clock counter and the clock counter to be calibrated start counting.
7. CLKTRIM_IFR.stop is set1 to trigger an interrupt when the reference clock counter is decremented from its initial value0.
8. The interrupt service subroutine determines that CLKTRIM_IFR.stop is 1, reads the register CLKTRIM_REFCNT and the value of CLKTRIM_CALCNT.
9. Clear CLKTRIM_CR.trim_start register to end calibration. Caution.
 - The calibration mode may overflow the clock counter before1 CLKTRIM_IFR.stop is set because the calibration time is set too long, and CLKTRIM_IFR.calcnt_of is1 set to trigger an interrupt. The interrupt service subroutine finds that CLKTRIM_IFR.calcnt_of is set 1 and clears the CLKTRIM_CR.trim_start register to end the calibration.

The calibration in this case cannot be performed correctly and the calibration time must be adjusted and recalibrated.

The specific steps are.

Set the CLKTRIM_REFCON .rcntval register to adjust the calibration time. Set the CLKTRIM_CR .trim_start register to restart the calibration.

19.3.2 CLK_TRIM Monitoring Mode

The monitoring mode is mainly used to select a stable clock source as the reference clock and monitor the abnormal status of the system operating clock at a set time period. Only external XTH clock or external XTL clock can be selected as the monitored clock in the monitoring mode.

19.3.2.1 Operation process

1. Set the CLKTRIM_CR.refclk_sel register to select the reference clock.
2. Set the CLKTRIM_CR.calclk_sel register to select the monitored clock.
3. Set the CLKTRIM_REFCON.rcntval register to the monitoring interval time.
4. Set the CLKTRIM_CALCON.ccntval register to the monitored clock counter overflow time.
5. Set the CLKTRIM_CR.mon_en register to enable the monitoring function.
6. Set the CLKTRIM_CR.IE register to enable interrupts.
7. Set CLKTRIM_CR.trim_start register to start monitoring.
8. The reference clock counter and the monitored clock counter start counting.
9. When the reference clock counter count reaches the monitoring interval time, determine whether the monitored clock counter overflows. If it overflows, it means the monitored clock is working normally. If there is no overflow, the monitored clock fails and CLKTRIM_IFR.xtal32k_fault/xtal32m_fault is set1 to trigger an interrupt.
10. Process the interrupt service subroutine and clear the interrupt flag bit CLKTRIM_IFR.xtal32k_fault/xtal32m_fault, clear CLKTRIM_CR.trim_start register to end monitoring.

19.4 CLK_TRIM Register Description

Register list

base address: 0x40001800

Offset	Register Name	Access	Register Description
0x00	CLKTRIM_CR	RW	Configuration register.
0x04	CLKTRIM_REFCON	RW	Reference counter initial value configuration register.
0x08	CLKTRIM_REFCNT	RO	Reference counter value register.
0x0c	CLKTRIM_CALCNT	RO	Calibrate the counter value register.
0x10	CLKTRIM_IFR	RO	Interrupt flag bit register.
0x14	CLKTRIM_ICLR	RW	Interrupt flag bit clear register
0x18	CLKTRIM_CALCON	RW	Calibration counter overflow value configuration register

Table 19-1 Register List

19.4.1 Configuration register (CLKTRIM_CR)

Offset address: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IE	mon_en	calclk_sel	refclk_sel	trim_start			
								RW	RW	RW	RW	RW			

position	Marker	Function Description
31:8	Reserved	
7	IE	Interrupt Enable Register0 - Disable 1 - Enable
6	mon_en	Monitor Mode Enable Register0 - Disable 1 - Enable
5:4	calclk_sel	Pending calibration/monitoring clock selection register -----00RCH -----01XTH -----10RCL -----11XTL
3:1	refclk_sel	Reference clock selection register ----- 000RCH ----- 001XTH ----- 010RCL ----- 011XTL ----- 100IRC10K ----- 101EXT_CLK_IN
0	trim_start	Calibration/Monitoring Start Register 0 - Stop 1 - Start

19.4.2 Reference counter initial value configuration register (CLKTRIM_REFCON)

Offset address: 0x04

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
rcntval[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rcntval[15:0]															
RW															

position	Marker	Function Description
31:0	rcntval	Reference counter initial value

19.4.3 Reference counter value register (CLKTRIM_REFCNT)

Offset address: 0x08

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
refcnt[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
refcnt[15:0]															
RO															

position	Marker	Function Description
31:0	refcnt	Reference counter value

19.4.4 Calibration counter value register (CLKTRIM_CALCNT)

Offset address: 0x0c

Reset value: 0x00000000

31302928272625242322212019181716

calcnt[31:16]
RO

1514131211109876543210

calcnt[15:0]
RO

position	Marker	Function Description
31:0	calcnt	Calibrate counter values

19.4.5 Interrupt Flag Bit Register (CLKTRIM_IFR)

Offset address: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
										xth_f ault	xtl_f ault	calcn t_of	stop		
										RO	RO	RO	RO		

position	Marker	Function Description
31:4	Reserved	
3	xth_fault	XTH Failure flag. CLKTRIM_ICLR.xth_fault_clr Write zero to clear this flag bit
2	xtl_fault	XTL failure flag. CLKTRIM_ICLR.xtl_fault_clr Write zero to clear this flag bit
1	calcnt_of	Calibration counter overflow flag. CLKTRIM_CR.start Write zero to clear this flag bit
0	stop	Reference counter stop flag. CLKTRIM_CR.start Write zero to clear this flag bit

19.4.6 Interrupt Flag Bit Clear Register (**CLKTRIM_ICLR**)

Offset address:

0x14 Reset

value: 0xf

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Reserved

xth_f	xtl_f	Reserved
ault_	ault_	
clr	clr	

RW RW

position	Marker	Function Description
31:4	Reserved	
3	xth_fault_clr	Clear XTH failure flag, write zero to clear.
2	xtl_fault_clr	Clear XTL failure flag, write zero clear.
1:0	Reserved	

19.4.7 Calibration counter overflow value configuration register **(CLKTRIM_CALCON)**

Offset address:

0x18 Reset value:

0xffff ffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ccntval[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ccntval[15:0]															
RW															

position	Marker	Function Description
31:0	ccntval	Calibration counter overflow value

20 Cyclic Redundancy Check (CRC)

20.1 Overview

The cyclic redundancy check (CRC) calculation unit takes a data stream or data block as input and generates an output number under the control of a generating polynomial. This output number is commonly used to verify the correctness and integrity of data transmission or storage. This module supports calculating CRC values and verifying CRC values.

20.2 Main Features

- An implementation standard: ISO/IEC13239
- One encoding method: CRC-16, $x^{16} + x^{12} + x^5 + 1$
- Three write bit widths: 8bit, 16bit, 32bit
- Two operating modes: CRC encoding mode, CRC checksum mode

20.3 Function Description

20.3.1 Working mode

This module supports two modes of operation: CRC encoding mode and CRC checksum mode.

CRC encoding mode is to input a certain amount of raw data to the CRC module and get the output value generated by the CRC module (`CRC_RESULT`). CRC checksum mode is to input a certain amount of raw data + CRC checksum value to the CRC module and verify that the raw data matches the CRC checksum value (`CRC_RESULT.FLAG`)

20.3.2 Coding method

This module supports CRC-16 encoding, which is calculated in bits16 and generates a polynomial of $x^{16} + x^{12} + x^5 + 1$.

20.3.3 Write Bit Width

The module supports three write bit widths: 8bit, 16bit, and 32bit. The writes of different bit widths need to comply with the principle of "consistent bit width, lower then higher", i.e., "each write must be written to a register with the same bit width as the current valid data, and the lower data must be written before the higher data. The writing of the lower bit data precedes the higher bit data.

The following shows the same sequence of data written using three bit widths, with the same output results.

- 8bit bit width writes: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16bit bit width write: 0x1100, 0x3322, 0x5544, 0x7766
- 32bit bit width write: 0x33221100, 0x77665544

20.4 Programming Examples

20.4.1 CRC-16 encoding mode

Step1 : Write 0xFFFF to CRC_RESULT to initialize CRC calculation.

Step2 : Write the raw data to be encoded to CRC_DATA register in 8bit, order, the write bit width can be selected

16bit, 32bit.

Step3 : Read CRC_RESULT.RESULT to get the CRC value.

20.4.2 CRC-16 test pattern

Step1 : Write 0xFFFF to CRC_RESULT to initialize CRC calculation.

Step2 : Write the encoded data sequence to CRC_DATA register in order, the writing bit width can be selected from 8bit, 16bit, 32bit.

Step3 : The value of CRC_RESULT.FLAG determines whether the encoded data sequence has been tampered with.

20.5 Register Description

20.5.1 Register List

Base address: 0x4002 0900

Register	Offset Address	Description
CRC_RESULT	0x04	CRC Result Register
CRC_DATA	0x80	CRC Data Register

20.5.2 Result register (CRC_RESULT)

Offset address: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															FLAG
															RO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT															
RW															

position	Symbols	Desc ription
16	FLAG	CRC Checksum Results 0: Current CRC checksum error 1: Current CRC checksum is correct
15:0	RESULT	CRC calculation results Read this register to get the result of CRC calculation Writing 0xFFFF to this register initializes the CRC calculation

20.5.3 Data register (CRC_DATA)

Offset address: 0x80

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
WO															

position	Symbols	Function Description
31:0	DATA	This register is used to write the data that needs to be calculated, and supports various write bit widths 8bit write method: * ((uint8_t *)0x40020980) = 0xXX 16bit write method: * ((uint16_t *)0x40020980) = 0xFFFF 32bit write method: * ((uint32_t *)0x40020980) = 0xFFFFFFFF

21 Analog to Digital Converters (ADC)

21.1 Module Introduction

External analog signals need to be converted to digital signals for further processing by the MCU. This series integrates a bit 12high-precision, high-rate successive approximation analog-to-digital converter (SAR ADC) module inside the MCU. It has the following features.

- 12 Bit conversion accuracy.
- 1Msps conversion speed (VCC>2.7V).
- 12 (a) Road conversion channels: 9one pin channel, built-in temperature sensor, built-in 1.2v reference voltage, 1/3 supply voltage.
- 4 (b) Various reference sources: supply voltage, ExRef pin, built-in 1.5v reference voltage, built-in 2.5v reference voltage.
- Voltage input range of the ADC: 0 to Vref.
- 3 (c) Various conversion modes: single conversion, continuous conversion, and cumulative conversion.
- Software configurable ADC conversion rate.
- Built-in signal amplifier for converting high-resistance signals.
- Supports on-chip peripherals to automatically trigger ADC conversion, effectively reducing chip power consumption and improving the real-time conversion.

21.2 ADC Block Diagram

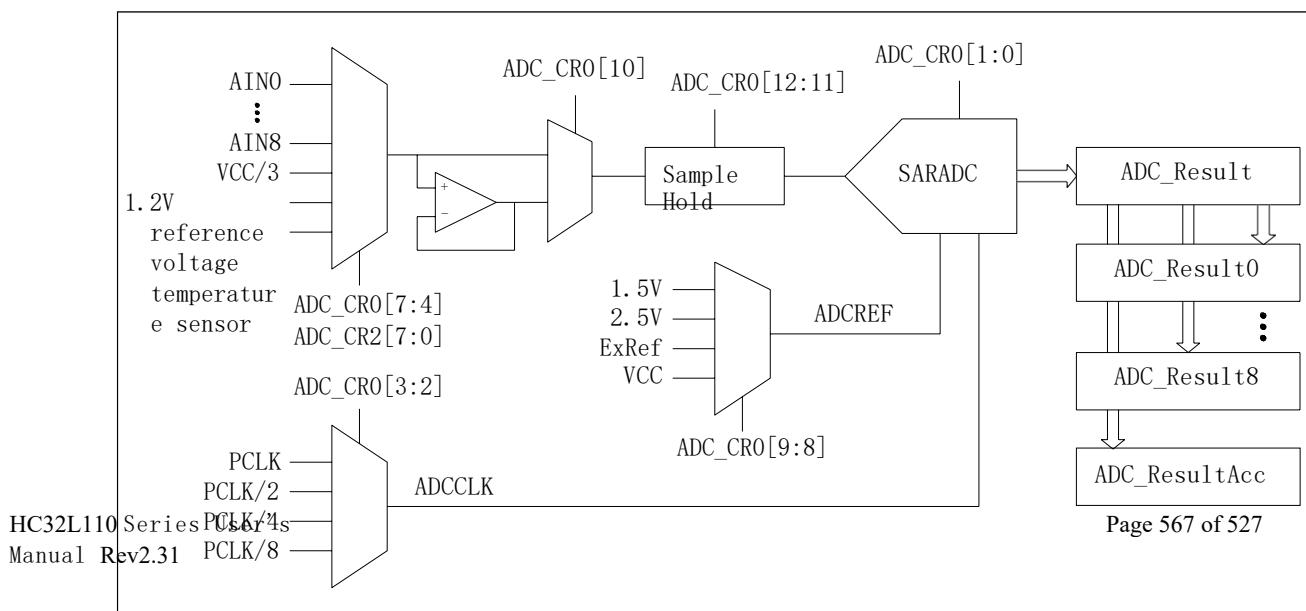


Figure 21-1ADC Schematic Block Diagram

21.3 Conversion Timing and Conversion Speed

The conversion timing of ADC is shown in the following diagram: a complete ADC conversion consists of a sampling process and a successive comparison process. The sampling process requires 4~12 AdcClk, which is configured by ADC_CR0.SAM; the successive comparison process requires one 16 AdcClk, so a total of 20~28 AdcClk are required for one ADC conversion.

The ADC conversion speed is measured in sps, which is the number of ADC conversions per second. the ADC conversion speed is calculated as the frequency of AdcClk / the number of AdcClk needed for one ADC conversion.

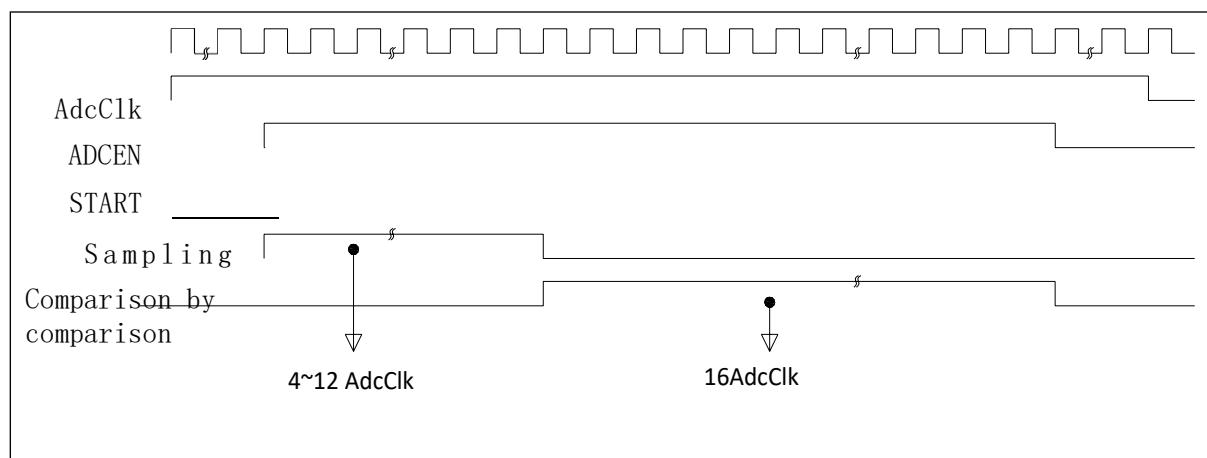


Figure 21-2 ADC Conversion Timing Diagram

The ADC conversion speed is related to the ADC reference voltage and VCC voltage, and the maximum conversion speed is shown in the following table.

ADC Reference Voltage	VCC Voltage	Maximum conversion speed	Maximum AdcClk Frequency
Internal 1.5V	1.8~5.5V	200Ksps	4MHz
Internal 2.5V	2.8~5.5V	200Ksps	4MHz
VCC / ExRef	1.8~2.4V	200Ksps	4MHz
VCC / ExRef	2.4~2.7V	500Ksps	16MHz
VCC / ExRef	2.7~5.5V	1Msps	24MHz

21.4 Single conversion mode

In single conversion mode, the ADC is started and only one conversion is performed for all 12ADC channels. This mode can be started either by setting the ADC_CR0.START bit or by setting the external trigger of ADC_CR1[9:0]. Once the ADC conversion of the selected channel is completed, the ADC_CR0.START bit is automatically cleared and the conversion result is stored in the ADC_result register.

Start ADC single conversion operation flow via START bit.

Step1: Configure the corresponding bits of P0ADS~P3ADS to configure the ADC channels to be converted as analog ports.

Step2: Set P3ADS.6 to 1, and configure the ADC external reference voltage pins as analog ports.

Note: This step can be skipped if the external reference voltage pin is not selected for the ADC reference voltage.

Step3: Set BGR_CR.BGR_EN to 1 enable the BGR module. Step4: Set ADC_CR0.ADCEN to 1 enable the ADC module. Step5: Delay 20uS and wait for the ADC and BGR module to finish starting. Step6: Set ADC_CR1.CT to 0 select the single conversion mode. Step7: Configure ADC_CR0. SREF to select the reference voltage of ADC.

Step8: Configure ADC_CR0.SAM and ADC_CR0.CLKSEL to set the conversion speed of ADC.

Step9: Configure ADC_CR0.SEL, select the channel to be converted. Step10: Set ADC_CR0.START to 1, start ADC single conversion.

Step11: Wait for ADC_CR0.START to change to 0, read ADC_result register to get ADC conversion result.

Step12: If you need to convert other channels, repeat Step9~Step11.

Step13: Set ADC_CR0.ADCEN and BGR_CR.BGR_EN to 0 turn off ADC module and BGR module.

Start ADC single conversion operation flow by external trigger.

Step1: Configure the corresponding bits of P0ADS~P3ADS to configure the ADC channels to be converted as analog ports.

Step2: Set P3ADS.6 to 1, and configure the ADC external reference voltage pins as analog ports.

Note: This step can be skipped if the external reference voltage pin is not selected for the ADC reference voltage.

Step3: Set `BGR_CR.BGR_EN` to 1 to enable the BGR module.

Step4: Set `ADC_CR0.ADCEN` to 1 to enable the ADC module. Step5: Delay 20 μ s and wait for the ADC and BGR module to finish. Step6: Set `ADC_CR1.CT` to 0 to select the single conversion mode. Step7: Set `ADC_HT` to 0x00. Step7: Set `ADC_HT` to 0x00.

Step8: Set `ADC_CR1.HtCmp` to 1 to enable ADC high threshold compare function.

Step9: Set `ADC_CR0.IE` to 1 to enable ADC interrupt.

Step10: Enable ADC interrupt in NVIC interrupt vector table. Step11: Configure `ADC_CR0.SREF` to select ADC reference voltage. Step12: Set `ADC_CR0`.

Step12: Configure `ADC_CR0.SAM` and `ADC_CR0.CLKSEL` to set the conversion speed of ADC.

Step13: Configure `ADC_CR0.SEL` to select the channel to be converted. Step14: Set `ADC_IFR` to 0x00 to clear the ADC interrupt flag.

Step15: Configure `ADC_CR1.TRIGS1` and `ADC_CR1.TRIGS0` to select the external trigger condition. Step16: The ADC module generates an interrupt when the external trigger condition triggers the ADC to complete the conversion. The user can set the interrupt in ADC

The `ADC_result` register is read in the interrupt service program to get the ADC conversion result.

Step17: If you need to convert other channels, repeat Step13~Step16.

Step18: Set `ADC_CR0.ADCEN` and `BGR_CR.BGR_EN` to 0 to turn off ADC module and BGR module.

21.5 Continuous conversion mode

In continuous conversion mode, the ADC can be started once to convert multiple channels in sequence; the convertible ADC channels are AIN0 to AIN7. The total number of ADC conversions is configured by ADCCR2.ADCCNT; the channels to be converted are configured by ADC_CR2[7:0]. This mode can be started either by setting the ADC_CR0.START bit or by setting the ADC_CR2[9:0] external trigger. After starting the continuous conversion, the ADC module converts the channels to be converted in AIN0~AIN7 in sequence until the total number of conversions is completed, and the ADC_IFR.CONT_INTF bit is automatically set after the ADC module completes the total number of conversions. If the total number of conversions is greater than the number of ADC channels to be converted, only the last conversion result is stored in the ADC_result0~ADC_result7 registers.

The following figure shows the 10 successive transitions of AIN0, AIN1, and AIN5. After the START is set through the register, the ADC internal state machine will convert AIN0, AIN1, and AIN5 in sequence until the ADCCNT count value

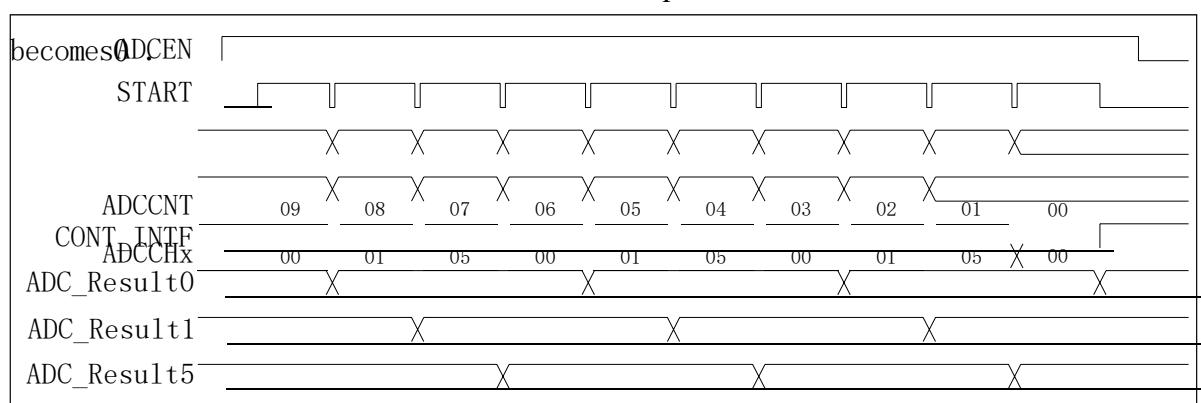


Figure 21-3 ADC Continuous Conversion Process Example

Start the ADC continuous conversion operation flow via the START bit.

Step1: Configure the corresponding bits of P0ADS~P3ADS to configure the ADC channels to be converted as analog ports.

Step2: Set P3ADS.6 to 1, and configure the ADC external reference voltage pins as analog ports.

Note: This step can be skipped if the external reference voltage pin is not selected for the ADC reference voltage.

Step3: Set BGR_CR.BGR_EN to enable the BGR module. step3: Set ADC_CR0.ADCEN to enable the ADC module. step3: Set BGR_CR.BGR_EN to enable the ADC module. step3: Set ADC_CR.ADCEN to enable the ADC module. step3: Set ADC_CR0.ADCEN to enable the ADC module. step3: Set BGR_CR.BGR_EN to enable the ADC module.

Step7: Set ADC_CR1[14:12] to 0 turn off the conversion result comparison function. Step8: Configure ADC_CR2.ADCCNT to select the total number of consecutive conversions. Step9: Configure ADC_CR0.SREF to select the reference voltage of ADC.

Step10: Configure ADC_CR0.SAM and ADC_CR0.CLKSEL to set the conversion speed of ADC.

Step11: Configure ADC_CR2[7:0] and select the channel to be converted.

Step12: Set ADC_ICLR.CONT_INTC to 0, clear ADC_IFR.CONT_INTF flag.

Step13: Set ADC_CR0.StateRst to 1 reset the continuous conversion state. Step14: Set ADC_CR0.START to 1 start the ADC continuous conversion.

Step15: Wait for ADC_IFR.CONT_INTF to become 1, read ADC_result0~ADC_result7 registers to get the conversion result of the corresponding channel.

Step16: If you need to convert other channels, repeat Step11~Step15.

Step17: Set ADC_CR0.ADCEN and BGR_CR.BGR_EN to 0 turn off ADC module and BGR module.

21.6 Continuous conversion accumulation mode

In continuous conversion accumulation mode, the ADC can be started once to convert multiple channels and accumulate the results of each conversion; the convertible ADC channels are AIN0~AIN7. The total number of ADC conversions is configured by ADCCR2.ADCCNT; the channels to be converted are configured by ADC_CR2[7:0]. This mode can be started either by setting the ADC_CR0.START bit or by setting the ADC_CR2[9:0] external trigger. After starting the continuous conversion, the ADC module converts the channels to be converted in AIN0~AIN7 in sequence until the total number of conversions is completed, and the ADC_IFR.CONT_INTF bit is automatically set1 after the ADC module completes the total number of conversions, and the cumulative value of the conversion results is stored in the ADC_result_acc register.

The following figure demonstrates the process of accumulating AIN0, AIN1, and AIN5 in successive 10 transitions. After 1 START is set through the register, the ADC internal state machine will convert AIN0, AIN1, and AIN5 in succession until the ADCCNT count value becomes 0. The ADC_result_acc

register is automatically totalized at the completion of each conversion.

The conversion results of AIN0, AIN1 and AIN5 given in the figure are 0x010, 0x020 and 0x040 in that order.

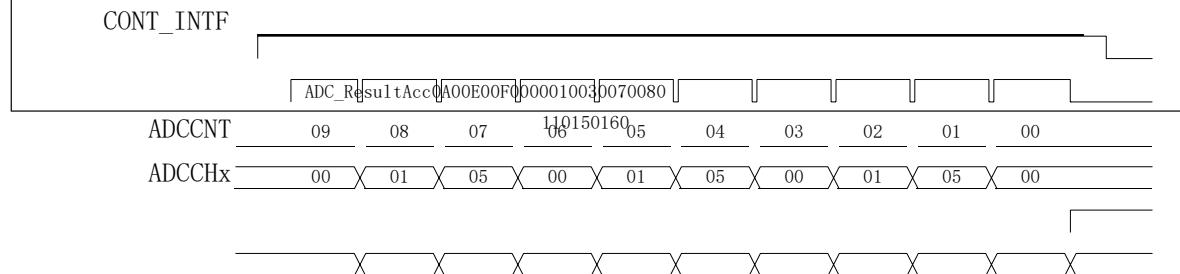


Figure 21-4ADCContinuous Conversion Accumulation Process Example

Start the ADC continuous conversion accumulation operation flow via the START bit.

Step1: Configure the corresponding bits of P0ADS~P3ADS to configure the HC32L110 Series User's Manual Rev2.31

ADC channels to be converted as analog ports.

Step2: Set P3ADS.6 to 1, and configure the ADC external reference voltage pins as analog ports.

Note: This step can be skipped if the external reference voltage pin is not selected for the ADC reference voltage.

Step3: Set BGR_CR.BGR_EN to 1 enable the BGR module.

Step4: Set ADC_CR0.ADCEN to 1 enable the ADC module.

Step5: Delay 20uS and wait for the ADC and BGR

module to finish starting. CT to , to select

continuous conversion mode. Step7: Set ADC_CR1[14:12]

to 0, to turn off the conversion result comparison

function.

Step8: Set ADC_CR1.RACC_EN to 1 enable the ADC conversion auto-accumulation function. Step9: Configure ADC_CR2.ADCCNT to select the total number of consecutive conversions. Step10: Configure ADC_CR0.

Step11: Configure ADC_CR0.SAM and ADC_CR0.CLKSEL to set the conversion speed of ADC.

Step12: Configure ADC_CR2[7:0] and select the channel to be converted.

Step13: Set ADC_ICLR.CONT_INTC to 0 clear ADC_IFR.CONT_INTF flag.

Step14: Set ADC_CR1.RACC_CLR to 01 clear ADC_result_acc register.

StateRst is to reset the continuous conversion state.

Step16: Set ADC_CR0.START to 1, start ADC continuous conversion.

Step17: Wait for ADC_IFR.CONT_INTF to become 1, read ADC_result_acc register to get the accumulative value of conversion result.

Step18: If you need to convert other channels, repeat Step12~Step17.

Step19: Set ADC_CR0.ADCEN and BGR_CR.BGR_EN to 0 turn off ADC module and BGR module.

21.7 Comparison of ADC conversion results

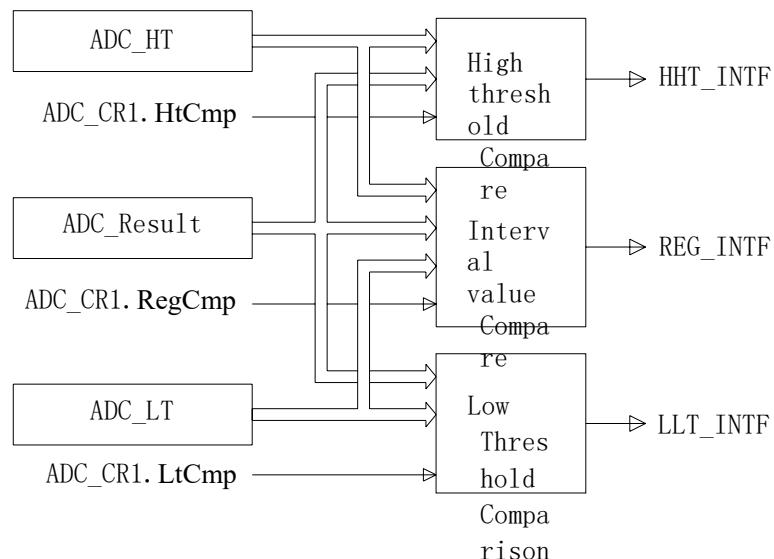
When the ADC conversion is completed, the ADC conversion result can be compared with the threshold value set by the user, supporting upper threshold comparison, lower threshold comparison, and interval value comparison. This function requires the corresponding control bits HtCmp, LtCmp, RegCmp to be set1. This function enables automatic monitoring of analog quantities until the ADC conversion result meets the user's expectation before generating an interrupt to request user program entry.

Upper threshold comparison: ADC_IFR.HHT_INTF when the ADC conversion result is within the [ADC_LT, 4095] interval

Set1; write to ADC_ICLR.HHT_INTC to 0clear ADC_IFR.HHT_INTF.

Lower threshold comparison: When ADC conversion result is within the interval [0,ADC_LT) then ADC_IFR.LLT_INTF is set1; write to ADC_ICLR.LLT_INTC then ADC_IFR.LLT_INTF is 0cleared.

Compare interval values: When the ADC conversion result is within the interval [ADC_LT , ADC_HT], ADC_IFR.REG_INTF is set1; write to ADC_ICLR.REG_INTC, ADC_IFR.REG_INTF is 0cleared.



21.8 ADC Interrupt

ADC interrupt requests are shown in the following table.

Interrupt source	Interruption flags	Interrupt Enable
ADC continuous conversion complete	ADC_IFR.CONT_INTF	ADC_CR0.IE
ADC conversion results in the interval value region	ADC_IFR.REG_INTF	
ADC conversion results in the upper threshold region	ADC_IFR.HHT_INTF	
ADC conversion result comparison lower threshold area	ADC_IFR.LLT_INTF	

21.9 Ambient temperature measurement using temperature sensors

The output voltage of the temperature sensor varies with the ambient temperature, so the corresponding ambient temperature can be calculated from the output voltage of the temperature sensor. When the output voltage of the temperature sensor is selected for the measurement channel of the ADC module, the ambient temperature can be measured.

The calculation formula is as follows.

$$\text{Ambient temperature} = +25 \times 0.0839V_{ref} \times (\text{AdcValue} - \text{Trim})$$

Where: Vref is the reference voltage of the current ADC module and takes the value of or 1.5 2.5.

AdcValue is the result of the ADC module measuring the output voltage of the temperature sensor, and the value is 0~4095. Trim is the 16bit calibration value, which needs to be read out from the Flash memory

during ADC ulation, Reference following table. Voltage	and Cailtis storage value storage address	as ilrate detailed in the Value Accuracy
Internal 1.5V	0x00100C34	5± ° C
Internal 2.5V	0x00100C36	5± ° C

An example calculation is as follows.

Conditions1 : Vref=2.5, AdcValue=0x7E5, Trim=0x76C.

Temperature1 : $25 + \frac{0.0839}{2.5} (0x7E5 - 0x76C) = 50^\circ C.$

Conditions2 : Vref=1.5, AdcValue=0x72D, Trim=0x76C.

Temperature3 : $25 + \frac{0.0839}{1.5} (0x72D - 0x76C) = 17^\circ C.$

Operating procedure for measuring ambient temperature via ADC.

Step1: Set BGR_CR.BGR_EN to3 enable the BGR module and temperature sensor module.

Step2: Set ADC_CR0.ADCEN to1 enable ADC module.

Step3: Delay 20uS and wait for ADC and BGR module to finish. Step4: Set ADC_CR1.CT to0 select single conversion mode.

Step5: Configure ADC_CR0.SREF to select the reference voltage of ADC as internal 1.5V or internal 2.5V. Step6: Configure ADC_CR0.SAM and ADC_CR0.CLKSEL to set the conversion speed of ADC. Step7: Set ADC_CR0. Step8: Set ADC_CR0.BUFEN to1 enable the input signal amplifier.

Step9: Set ADC_CR0.START to 1, start ADC single conversion.

Step10: Wait for ADC_CR0.START to change to 0, read ADC_result register to get ADC conversion result.

Step11: Set ADC_CR0.ADCEN and BGR_CR.BGR_EN to0 turn off ADC module and BGR module.

Step12: Read the temperature sensor calibration value and calculate the current ambient temperature according to the formula.

21.10 ADC Module Register

Base address 0x40002400

Register	Offset Address	Description
ADC_CR0	0x004	ADC Configuration Register0
ADC_CR1	0x008	ADC Configuration Register1
ADC_CR2	0x00C	ADC Configuration Register2
ADC_result0	0x030	ADC channel0 conversion results
ADC_result1	0x034	ADC channel1 conversion results
ADC_result2	0x038	ADC channel2 conversion results
ADC_result3	0x03C	ADC channel3 conversion results
ADC_result4	0x040	ADC channel4 conversion results
ADC_result5	0x044	ADC channel5 conversion results
ADC_result6	0x048	ADC channel6 conversion results
ADC_result7	0x04C	ADC channel7 conversion results
ADC_result8	0x050	ADC channel8 conversion results
ADC_result_acc	0x054	ADC conversion result summation value
ADC_HT	0x058	ADC Compare Upper Threshold
ADC_LT	0x05C	ADC Compare Lower Threshold
ADC_IFR	0x060	ADC interrupt flag register
ADC_ICLR	0x064	ADC Interrupt Clear Register
ADC_result	0x068	ADC Conversion Results

Table 21-1ADCRegisters

21.10.1 ADC Configuration Register0 (ADC_CR0)

Offset address 0x004

Reset value 0x000013F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
State Rst	IE	Res.	SAM	BUF EN	SREF	SEL			CLKSEL	STA RT	ADC EN				
R/W	R/W		R/W	R/W	R/W	R/W			R/W	R/W	R/W				

position	Marker	Function Description
31:16	Reserved	Reserved
15	StateRst	ADC continuous conversion state control1 : Reset ADC continuous conversion state 0: Invalid
14	IE	ADC interrupt control: enable interrupts 0: Disable interruption
13	Reserved	Reserved
12:11	SAM	ADC sample period selection 00:4 one sample period 01:6 sampling period 10:8 sampling period 11:12 sampling period
10	BUFEN	ADC input signal amplifier control 0: Turn off the amplifier, the external input signal is connected directly to the ADC. 1: Turn on the amplifier, the external input signal is amplified by the amplifier and connected to the ADC for high resistance signals.

9:8	SREF	ADC reference voltage selection 00: internal 1.5V 01: Internal 2.5V 10: External reference voltage ExRef (P3.6) 11: Power supply voltage
7:4	SEL	ADC conversion channel selection (single conversion mode) 0000 : Selects channel 0 input P2.4 0001: Selects channel1 input P2.6 0010: Selects channel2 input P3.2 0011: Selects channel3 input P3.3 0100: Selects channel4 input P3.4

		0101: Selects channel5 input P3.5 0110: Selects channel6 input P3.6 0111: Selects channel7 input P0.1 1000: Selects channel18 input P0.2 1001: VCC/3 Note: ADC_CR0.BUFEN must be1 1010Note: Built-in temperature sensor output voltage : ADC_CR0.BUFEN must be1 1011: Internal reference 1.2V output voltage Note: ADC_CR0.BUFEN must be1
3:2	CLKSEL	ADC clock selection 00: PCLK clock 01: PCLK clock2 division 10: PCLK clock4 division 11: PCLK clock8 division
1	START	ADC conversion control 1 : Start ADC conversion 0: Stop ADC conversion
0	ADCEN	ADC Enable Control 1: Enables ADC 0: Prohibit ADC

21.10.2 ADC Configuration Register1 (ADC_CR1)

Offset address 0x008

Reset value 0x00007000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RACC _CLR	RegCm p	HtCmp	LtCmp	HtCmp	CT	TRIGS1								TRIGS0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W								R/W	

position	Marker	Function Description
31:16	Reserved	Reserved
15	RACC_CLR	ADC conversion result accumulation register cleared to zero 1: no effect. 0: The ADC_result_acc register is cleared to zero. Note: This bit is read out0, so you need to pay special attention to the value of this bit when operating this register to prevent misoperation.
14	RegCmp	ADC interval comparison control: Enables interval comparison 0: Comparison of prohibited intervals
13	HtCmp	ADC high threshold comparison control: Enables high threshold comparison 0: prohibit high threshold comparisons
12	LtCmp	ADC low threshold comparison control: Enables low threshold comparison 0: Prohibit low threshold comparisons

11	RACC_EN	ADC conversion result auto-accumulation controll : Enables ADC conversion result auto-accumulation function 0: Disable the automatic accumulation of ADC conversion results
10	CT	ADC conversion mode selection1: continuous conversion mode 0: Single conversion mode
9:5	TRIGS1	ADC conversion auto-trigger selection 2: 00000: Disable auto-trigger ADC conversion 00001: Timer0 interrupt, automatically triggers ADC conversion 00010: Timer1 interrupt, automatic triggering of ADC conversion 00011: Timer2 interrupt, automatic triggering of ADC conversion 00100: LPTimer interrupt to automatically trigger ADC conversion 00101: Timer4 interrupt, automatically triggers ADC conversion

	<p>00110: Timer5 interrupt to automatically trigger ADC conversion 00111: Timer6 interrupt to automatically trigger ADC conversion 01000: UART0 interrupt, automatically trigger ADC conversion 01001: UART1 interrupt, automatically triggers ADC conversion 01010: LPUART interrupt to automatically trigger ADC conversion 01011: VCO interrupt, automatic triggering of ADC conversion 01100: VC1 interrupt to automatically trigger ADC conversion 01101: RTC interrupt to automatically trigger ADC conversion 01110: PCA interrupt to automatically trigger ADC conversion 01111: SPI interrupt to automatically trigger ADC conversion 10000: P01 interrupt, automatically trigger ADC conversion 10001: P02 interrupt to automatically trigger ADC conversion 10010: P03 interrupt to automatically trigger ADC conversion 10011: P14 interrupt to automatically trigger ADC conversion 10100: P15 interrupt to automatically trigger ADC conversion 10101: P23 interrupt, automatic triggering of ADC conversion 10110: P24 interrupt to automatically trigger ADC conversion 10111: P25 interrupt to automatically trigger ADC conversion 11000: P26 interrupt, automatically triggers ADC conversion 11001: P27 interrupt, automatic triggering of ADC conversion 11010: P31 interrupt, automatic triggering of ADC conversion 11011: P32 interrupt, automatic triggering of ADC conversion 11100: P33 interrupt, automatically triggers ADC conversion 11101: P34 interrupt, automatic triggering of ADC conversion 11110: P35 interrupt to automatically trigger ADC conversion 11111: P36 interrupt, automatic triggering of ADC conversion</p> <p>Note:</p> <ul style="list-style-type: none"> 1) TIM4/5/6 interrupts trigger the ADC auto-conversion, in addition to enabling the corresponding interrupts of TIM4/5/6, you also need to configure the Advanced Timer spread spectrum and the interrupt trigger selection register TIMX_CR to select the interrupt source that can trigger the ADC. 2) The ADC is triggered using the rising edge of each interrupt flag bit. If the trigger needs to be repeated, you need to clear the Interrupt flag. Do not enable interrupt enable for NVIC if you do not need to enter the interrupt service program.
--	---

4:0	TRIGS0	ADC conversion auto-trigger selection 2: 00000: Disable auto-trigger ADC conversion 00001: Select Timer0 interrupt to automatically trigger ADC conversion 00010: Select Timer1 interrupt to automatically trigger ADC conversion 00011: Select Timer2 interrupt to automatically trigger ADC conversion 00100: Select LPTimer interrupt to automatically trigger ADC conversion 00101: Select Timer4 interrupt to automatically trigger ADC conversion 00110: Select Timer5 interrupt to automatically trigger ADC conversion 00111: Select Timer6 interrupt to automatically trigger ADC conversion 01000: Select UART0 interrupt to automatically trigger ADC conversion
-----	--------	--

	<p>01001: Select UART1 interrupt to automatically trigger ADC conversion</p> <p>01010: Select LPUART interrupt to automatically trigger ADC conversion</p> <p>01011: Select VCO interrupt to automatically trigger ADC conversion</p> <p>01100: Select VC1 interrupt to automatically trigger ADC conversion</p> <p>01101: Select RTC interrupt to automatically trigger ADC conversion</p> <p>01110: Select PCA interrupt to automatically trigger ADC conversion</p> <p>01111: Select SPI interrupt to automatically trigger ADC conversion</p> <p>10000: Select P01 interrupt to automatically trigger ADC conversion</p> <p>10001: Select P02 interrupt to automatically trigger ADC conversion</p> <p>10010: Select P03 interrupt to automatically trigger ADC conversion</p> <p>10011: Select P14 interrupt to automatically trigger ADC conversion</p> <p>10100: Select P15 interrupt to automatically trigger ADC conversion</p> <p>10101: Select P23 interrupt to automatically trigger ADC conversion</p> <p>10110: Select P24 interrupt to automatically trigger ADC conversion</p> <p>10111: Select P25 interrupt to automatically trigger ADC conversion</p> <p>11000: Select P26 interrupt to automatically trigger ADC conversion</p> <p>11001: Select P27 interrupt to automatically trigger ADC conversion</p> <p>11010: Select P31 interrupt to automatically trigger ADC conversion</p> <p>11011: Select P32 interrupt to automatically trigger ADC conversion</p> <p>11100: Select P33 interrupt to automatically trigger ADC conversion</p> <p>11101: Select P34 interrupt to automatically trigger ADC conversion</p> <p>11110: Select P35 interrupt to automatically trigger ADC conversion</p> <p>11111: Select P36 interrupt to automatically trigger ADC conversion</p> <p>Note:</p> <ul style="list-style-type: none"> 1) TIM4/5/6 interrupts trigger the ADC auto-conversion, in addition to enabling the corresponding interrupts of TIM4/5/6, you also need to configure the Advanced Timer spreading and interrupt trigger selection register TIMX_CR to select the interrupt source that can trigger the ADC. 2) The ADC is triggered using the rising edge of each interrupt flag bit. If the trigger needs to be repeated, you need to clear the middle <p>The interrupt flag. Do not enable the NVIC interrupt enable if you do not need to enter the interrupt service program.</p>
--	---

21.10.3 ADC Configuration Register2 (ADC_CR2)

Offset address 0x00C

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCCNT								CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
R/W								R/W							

position	Marker	Function Description
31:16	Reserved	Reserved
15:8	ADCCNT	ADC continuous conversion number configuration0: continuous conversion 1times 1: Continuous conversion 2times 255: Continuous conversion 256times
7	CH7EN	ADC continuous conversion channel17 enable1: enable 0: Prohibition
6	CH6EN	ADC continuous conversion channel16 enable1: enable 0: Prohibition
5	CH5EN	ADC continuous conversion channel15 enable1: enable 0: Prohibition
4	CH4EN	ADC continuous conversion channel14 enable1: enable 0: Prohibition
3	CH3EN	ADC continuous conversion channel13 enable1: enable 0: Prohibition

2	CH2EN	ADC continuous conversion channel12 enable1: enable 0: Prohibition
1	CH1EN	ADC continuous conversion channel11 enable1: enable 0: Prohibition
0	CH0EN	ADC continuous conversion channel10 enable

		1: Enabled 0: Prohibition
--	--	------------------------------

21.10.4 ADC channel Conversion0 result (ADC_result0)

Offset address 0x030

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
result0															
Reserved				RO											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result0	ADC channel0 conversion results

21.10.5 ADC channel Conversion1 result (ADC_result1)

Offset address 0x034

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
result1															
Reserved				RO											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result1	ADC channel1 conversion results

21.10.6 ADC channel Conversion2 results (ADC_result2)

Offset address 0x038

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				result2											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result2	ADC channel12 conversion results

21.10.7 ADC channel Conversion3 results (ADC_result3)

Offset address 0x03C

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				result3											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result3	ADC channel13 conversion results

21.10.8 ADC channel Conversion4 results (ADC_result4)

Offset address 0x040

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				result4											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result4	ADC channel14 conversion results

21.10.9 ADC channel Conversion5 results (ADC_result5)

Offset address 0x044

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				result5											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result5	ADC channel15 conversion results

21.10.10 ADC channel Conversion6 results (ADC_result6)

Offset address 0x048

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	result6														

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result6	ADC channel6 conversion results

21.10.11 ADC channel Conversion7 results (ADC_result7)

Offset address 0x04C

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	result7														

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result7	ADC channel7 conversion results

21.10.12 ADC channel Conversion8 results (ADC_result8)

Offset address 0x050

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				result8											
RO				RO											

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result8	ADC channel18 conversion results

21.10.13 ADC conversion result accumulation value (ADC_result_acc)

Offset address 0x054

Reset value 0x00000000

31302928272625242322212019181716

Reserved												Result_acc[19:16]		
RO												RO		

Result_acc[15:0]														
RO														

1514131211109876543210

position	Marker	Function Description
31:20	Reserved	Reserved
19:0	Result_acc	ADC conversion of totalized values

21.10.14 ADC Compare Upper Threshold (ADC_HT)

Offset address 0x058

Reset value 0x00000FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HT											
R/W															

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	HT	ADC conversion result comparison upper threshold

21.10.15 ADC Compare Lower Threshold (ADC_LT)

Offset address 0x05C

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				LT											
R/W															

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	LT	ADC conversion result comparison lower threshold

21.10.16 ADC Interrupt Flag Register (ADC_IFR)

Offset address 0x060

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

position	Marker	Function Description
31:4	Reserved	Reserved
3	CONT_INTF	Continuous conversion completion mark 1: ADC continuous conversion completed 0: ADC continuous conversion not completed
2	REG_INTF	ADC conversion result comparison interval flag 1: The ADC conversion result lies within the interval [ADC_LT, ADC_HT]. 0: The ADC conversion result lies outside the interval [ADC_LT, ADC_HT].
1	HHT_INTF	ADC conversion result comparison upper threshold flag 1: The ADC conversion result lies within the interval [ADC_HT, 4095]. 0: The ADC conversion result lies outside the [ADC_HT, 4095] interval.
0	LLT_INTF	ADC conversion result comparison lower threshold flag 1: The ADC conversion result lies in the interval [0, ADC_LT) 0: The ADC conversion result lies outside the [0, ADC_LT) interval.

21.10.17 ADC Interrupt Clear Register (ADC_ICLR)

Offset address 0x064

Reset value 0x00000004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

position	Marker	Function Description
31:4	Reserved	Reserved
3	CONT_INTC	Write to 0clear the continuous conversion completion flag Write1 no effect
2	REG_INTC	Write to 0clear the ADC conversion result comparison interval flag Write1 no effect
1	HHT_INTC	Write 0clear ADC conversion result comparison upper threshold Write1 no effect
0	LLT_INTC	Write to 0clear ADC conversion result comparison lower threshold flag Write1 no effect

21.10.18 ADC result (ADC_result)

Offset address 0x068

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

position	Marker	Function Description
31:12	Reserved	Reserved
11:0	result	ADC conversion results

22 Analog Comparator (VC)

22.1 Analog Voltage Comparator VC Introduction

The analog voltage comparator VC compares the magnitude of two input analog voltages and outputs a high/low level according to the comparison result. When the voltage at the "+" input is higher than the voltage at the "-" input, the voltage comparator outputs a high level; when the voltage at the "+" input is lower than the voltage at the "-" input, the voltage comparator outputs a low level. When the voltage at the "+" input is lower than the voltage at the "-" input, the voltage comparator output is low. The internal integrated analog voltage comparator VC has the following characteristics.

- Support for voltage comparison functions.
- Internal VCC voltage divider support (voltage greater than 1.8V is required to use the divider source)
- Support for one 8external input port and the reference voltage from the on-chip BGR as input to the voltage comparator.
- Support three software configurable interrupt triggering methods: high level triggering/rising edge triggering/falling edge triggering.
- The output of the voltage comparator can be used as an input to the Base Timer and LPTimer gating ports.
- the output of the voltage comparator can be used as a brake input or capture input for the Advanced Timer.
- Support for operation in ultra-low power mode, where the interrupt output of the voltage comparator can wake up the chip from ultra-low power mode.
- Provides software-configurable filtering time to enhance the chip's immunity to interference.

22.2 Voltage comparator frame diagram

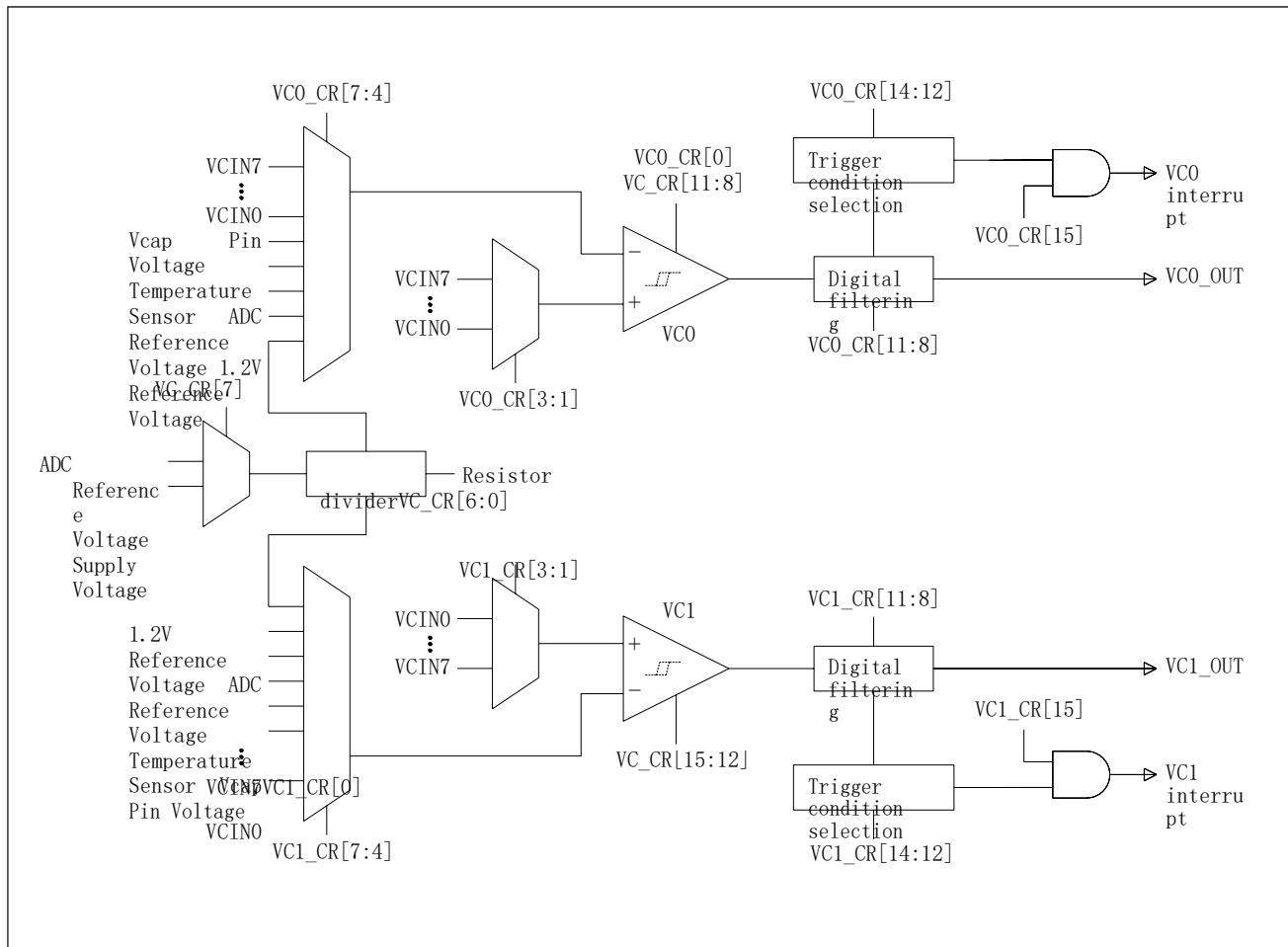


Figure 22-1 VCFramework Diagram

22.3 Set-up/response time

When using a voltage comparator, the time from VC enable or the change of the input voltage on both sides of the VC to the output of the correct result is determined by the **BIAS_SEL** control bit in the VC control register (**VC_CR**).

If the temperature sensor, 1.2V reference voltage, or ADC module reference voltage is selected as the end input to the comparator, the internal BGR module needs to be turned on. The start-up time of the internal BGR is about 20us, and the voltage comparator needs to wait for the internal BGR to stabilize before it can output properly.

22.4 Filtering time

In addition to the build/response time inherent in voltage comparators, users can set longer filter times to filter out system noise, such as high current noise when the motor stops.

The digitally filtered signal level can be read from register VC_IFR.VCx_Filter; when the GPIO function is configured as

When VCx_OUT, the digitally filtered signal can be output from GPIO for easy measurement.

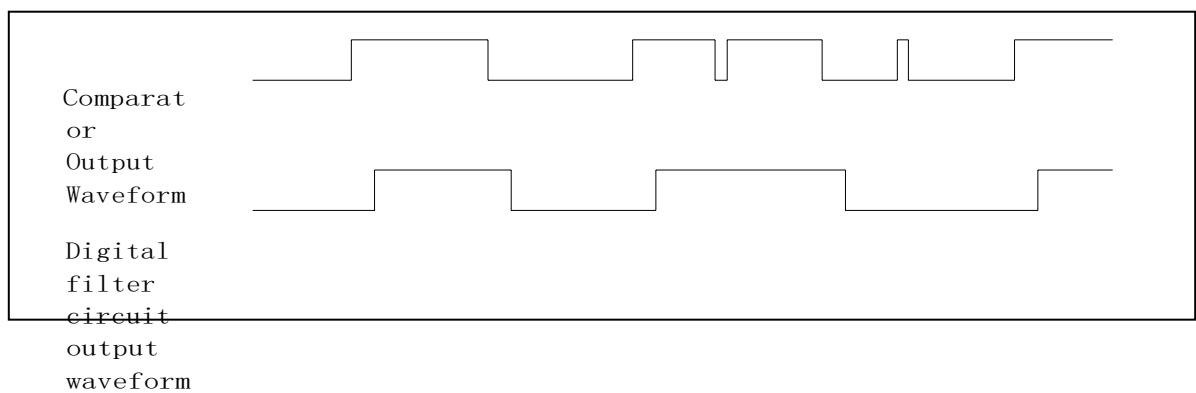


Figure 22-2VCFILTER Response Time

22.5 Hysteresis function

The hysteresis function can be selected for the voltage comparator. The diagram of the hysteresis function when enabled is as follows

22.4 Filtering time

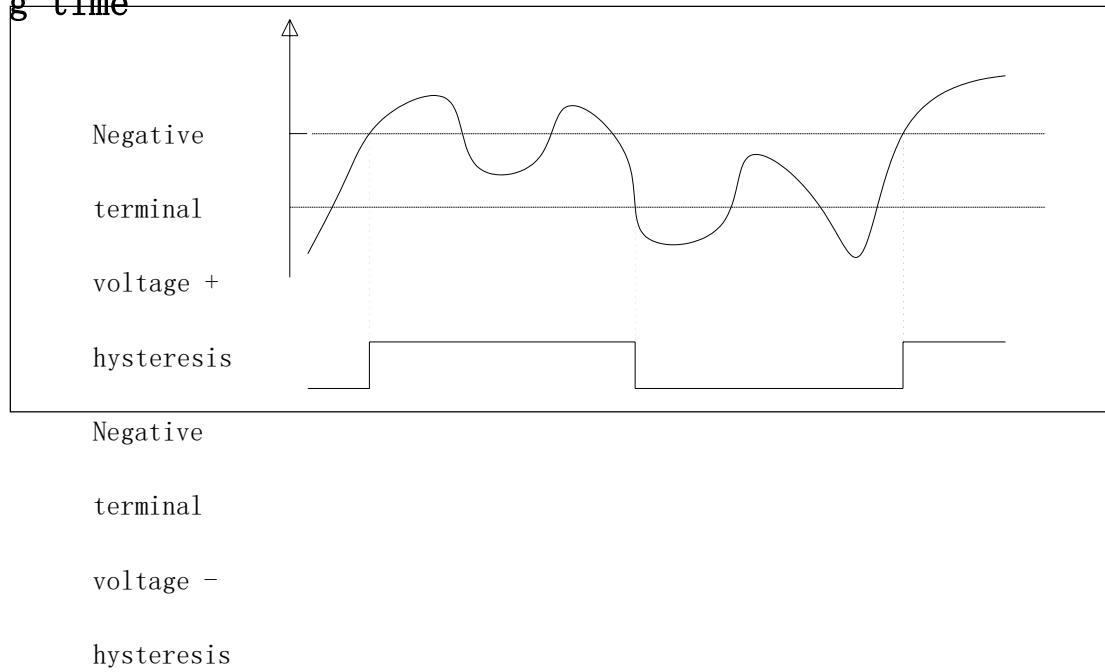


Figure 22-3VCHysteresis Function

22.6 VC Register

Base address 0x40002400

Register	Offset Address	Description
VC_CR	0x010	VC0/1 Configuration Register0
VC0_CR	0x014	VC0 Configuration Register
VC1_CR	0x018	VC1 Configuration Register
VC0_OUT_CFG	0x01C	VC0 Output Configuration Register
VC1_OUT_CFG	0x020	VC1 Output Configuration Register
VC_IFR	0x024	VC Interrupt Register

Table 22-1VCRegisters

22.6.1 VC Configuration Register (VC_CR)

Offset address 0x010

Reset value 0x00000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC1_HYS_SEL	VC1_BIAS_SEL	VC0_HYS_SEL	VC0_BIAS_SEL	VC_REF_2P5_SEL	VC_DI_V_EN	VC_DIV									
R/W	R/W	R/W	R/W	R/W	R/W	R/W									

position	Marker	Function Description
31:16	Reserved	Reserved
15:14	VC1_HYS_SEL	VC1 Hysteresis selection: 00:No lag 01:Hysteresis voltage about 10mV 10:Hysteresis voltage about 20mV 11:Hysteresis voltage about 30mV
13:12	VC1_BIAS_SEL	VC1 power consumption selection (the higher the power consumption, the faster the response) 00:300nA 01:1.2uA 10:10uA (need power supply voltage not less than 2.8V, need to open BGR manually) 11:20uA (need power supply voltage not less than 2.8V, need to manually open BGR) Note: BGR start time is about 30us
11:9	VC0_HYS_SEL	VC0 Hysteresis selection: 00:No lag 01:Hysteresis voltage about 10mV 10:Hysteresis voltage about 20mV 11:Hysteresis voltage about 30mV
9:8	VC0_BIAS_SEL	VC0 power consumption selection (the higher the power consumption, the faster the response) 00:300nA 01:1.2uA 10:10uA (need power supply voltage not less than 2.8V, need to open BGR manually) 11:20uA (need power supply voltage not less than 2.8V, need to manually open BGR) Note: BGR start time is about 30us

7	VC_REF2P5_SEL	VC_DIV Reference voltage Vref selection 0:VCC 1: Reference voltage selected by ADC_CR0.SREF
6	VC_DIV_EN	6Bit DAC Enable

		1: Enabled
5:0	VC_DIV	<p>6Bit DAC Configuration</p> <p>000000: 1/64 Vref</p> <p>000001:2/64 Vref</p> <p>000010:3/64 Vref</p> <p>000011:4/64 Vref</p> <p>...</p> <p>111110:63/64 Vref</p> <p>111111: Vref</p>

22.6.2 VC0 Configuration Register (VC0_CR)

Offset address 0x014

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time	FLTEN	n_sel	p_sel	EN							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							R/W

position	Marker	Function Description
31:16	Reserved	Reserved
15	IE	VC Interrupt Enable 1: enable;0 : disable
14	level	VC output signal high level triggers INT flag
13	rising	VC Output signal rising edge trigger INT flag
12	Falling	VC Output signal falling edge trigger INT flag
11:9	debounce_time	VC Output Filter Time Configuration 111: Filtering time is about 28.8ms 110: Filtering time is about 7.2ms 101: Filtering time is about 1.8ms 100: Filtering time of approximately 450us 011: Filtering time of approximately 112us 010: Filtering time of approximately 28us 001: Filtering time of approximately 14us 000: Filtering time of approximately 7us Note: The configuration of the filtering time is only valid when FLTEN=1.
8	FLTEN	1: Start VC filtering 0: VC without filtering
7:4	N_SEL	Voltage comparator "-" terminal input selection 0000: select channel input 0P2.3 0001: select channel input 1P2.5 0010: select channel input 2P3.2 0011: select channel input 3P3.3 0100: select channel input 4P3.4 0101: select channel input 5P3.5 0110: select channel input 6P3.6 0111: select channel input 7P0.1 1000: Resistor divider

		output voltage
--	--	----------------

		1001:Built-in temperature sensor output voltage 1010:Internal reference 1.2V output voltage 1011:Reference voltage of ADC module 1100:Voltage at VCAP pin
3:1	P_SEL	Voltage comparator "+" terminal input selection 000: select channel input 0P2.3 001: select channel input 1P2.5 010: select channel input 2P3.2 011: select channel input 3P3.3 100: select channel input 4P3.4 101: select channel input 5P3.5 110: select channel input 6P3.6 111: select channel input 7P0.1
0	EN	Voltage comparator enable 1: Enable voltage comparator 0: Turn off the voltage comparator

22.6.3 VC1 Configuration Register (VC1_CR)

Offset address 0x018

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	level	rising	falling	debounce_time	FLTEN		n_sel		p_sel		EN				
R/W	R/W	R/W	R/W	R/W	R/W		R/W		R/W		R/W		R/W		R/W

position	Marker	Function Description
31:16	Reserved	Reserved
15	IE	VC Interrupt Enable 1: enable;0 : disable
14	level	VC Output Signal Trigger Interrupt Selection 1: Enable high level trigger INT flag 0: Disable high level triggering INT flag
13	rising	VC Output Signal Trigger Interrupt Selection 1: Enabling rising edge triggering INT flag 0: Disable rising edge triggering INT flag
12	falling	VC Output signal trigger interrupt selection 1: Enable falling edge trigger INT flag 0: Disable falling edge triggering INT flag
11:9	debounce_time	VC Output Filter Time Configuration 111: Filtering time is about 28.8ms 110: Filtering time is about 7.2ms 101: Filtering time is about 1.8ms 100: Filtering time of approximately 450us 011: Filtering time of approximately 112us 010: Filtering time of approximately 28us 001: Filtering time of approximately 14us 000: Filtering time of approximately 7us Note: The configuration of the filtering time is only valid when FLTEN=1.
8	FLTEN	1: Start VC filtering 0: VC without filtering
7:4	N_SEL	Voltage comparator "-" terminal input selection 0000: select channel input 0P2.3 0001: select channel input 1P2.5 0010: select channel input 2P3.2

		0011: select channel input 3P3.3 0100: select channel input 4P3.4 0101: select channel input 5P3.5 0110: select channel input 6P3.6 0111: select channel input 7P0.1 1000: Resistor divider output voltage 1001: Built-in temperature sensor output voltage 1010: Internal reference 1.2V output voltage 1011: Reference voltage of ADC module 1100: Voltage at VCAP pin
3:1	P_SEL	Voltage comparator "+" terminal input selection 000: select channel input 0P2.3 001: select channel input 1P2.5 010: select channel input 2P3.2 011: select channel input 3P3.3 100: select channel input 4P3.4 101: select channel input 5P3.5 110: select channel input 6P3.6 111: select channel input 7P0.1
0	EN	Voltage comparator enable 1: Enable voltage comparator 0: Turn off the voltage comparator

22.6.4 VC0 Output Configuration Register (VC0_OUT_CFG)

Offset address 0x01C

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_T imer6	TIM5	INV_T imer5	TIM4	INV_T imer4	PCAE	PCAC	INV_ PCA	TM3E	LPTI	TIM2	TIM1	TIM0	INV_ Timer
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function Description
31:16	Reserved	Reserved
15	brake	VC0 as Advanced Timer Brake Control 1: Enabled;0 : Disabled.
14	TIM6	VC0 filter result output to TIM6 capture input enable 1: Enabled;0 : Disabled.
13	INV_TIM6	VC0 filter result output to TIM6 Reverse Enable 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
12	TIM5	VC0 filter result output to TIM5 capture input enable 1: Enabled;0 : Disabled.
11	INV_TIM5	VC0 filter result output to TIM5 Reverse Enable 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
10	TIM4	VC0 filter result output to TIM4 capture input enable 1: Enabled;0 : Disabled.
9	INV_TIM4	VC0 filter result output to TIM4 Reverse Enable 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
8	PCAE _{CI}	VC0 filter result output to PCA external clock enable control 1: Enabled;0 : Disabled.
7	PCACAP0	VC0 filter result output to PCA capture0 enable control 1: Enabled;0 : Disabled.
6	INV_PCA	VC0 filter result output negative to PCA 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
5	LPTIMECLK	VC0 filter Result output to LPTIMER external clock enable control 1: Enabled;0 : Disabled.
4	LPTIMG	VC0 filter result output to LPTIMER3 GATE enable control 1: Enabled;0 : Disabled.

3	TIM2G	VC0 filter result output to TIM2 GATE enable control 1: Enabled;0 : Disabled.
---	-------	--

2	TIM1G	VC0 filter result output to TIM1 GATE enable control 1: Enabled;0 : Disabled.
1	TIM0G	VC0 filter result output to TIM0 GATE enable control 1: Enabled;0 : Disabled.
0	INV_Timer	VC0 filter Resulting output negative to each TIM0/1/2, LPTimer 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.

22.6.5 VC1 Output Configuration Register (VC1_OUT_CFG)

Offset address 0x020

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
brake	TIM6	INV_Timer6	TIM5	INV_Timer5	TIM4	INV_Timer4	PCA	PCAC	INV_PCA	TM3E	LPTI_MG	TIM2G	TIM1G	TIM0G	INV_Timer
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

position	Marker	Function Description
31:16	Reserved	Reserved
15	brake	VC1 as Advanced Timer Brake Control 1: Enabled;0 : Disabled.
14	TIM6	VC1 filter result output to TIM6 capture input enable 1: Enabled;0 : Disabled.
13	INV_TIM6	VC1 filter result output to TIM6 Reverse Enable 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
12	TIM5	VC1 filter result output to TIM5 capture input enable 1: Enabled;0 : Disabled.
11	INV_TIM5	VC1 filter result output to TIM5 Reverse Enable 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
10	TIM4	VC1 filter result output to TIM4 capture input enable 1: Enabled;0 : Disabled.
9	INV_TIM4	VC1 filter result output to TIM4 Reverse Enable 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.
8	PCAECI	VC1 filter result output to PCA external clock enable control 1: Enabled;0 : Disabled.
7	PCACAP1	VC1 filter result output to PCA capture1 enable control 1: Enabled;0 : Disabled.
6	INV_PCA	VC1 filter result output negative to PCA 1: enable reverse;0 : disable reverse, input and VC output are in the same direction.
5	LPTIMECLK	VC1 filter Result output to LPTIMER external clock enable control 1: Enabled;0 : Disabled.
4	LPTIMG	VC1 filter result output to LPTIMER3 GATE enable control 1: Enabled;0 : Disabled.

3	TIM2G	VC1 filter result output to TIM2 GATE enable control 1: Enabled;0 : Disabled.
---	-------	--

2	TIM1G	VC1 filter result output to TIM1 GATE enable control 1: Enabled;0 : Disabled.
1	TIM0G	VC1 filter result output to TIM0 GATE enable control 1: Enabled;0 : Disabled.
0	INV_Timer	VC1 filter Resulting output negative to each TIM0/1/2, LPTimer 1: enable reverse;0 : disable reverse, input is in the same direction as VC output.

22.6.6 VC Interrupt Register (VC_IFR)

Offset address 0x024

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
												VC1_ Filter	VC0_ Filter	VC1_ INTF	VC0_ INTF
												RO	RO	R/W	R/W

position	Marker	Function Description
31:4	Reserved	Reserved
3	VC1_Filter	Status after VC1 Filter
2	VC0_Filter	Status after VC0 Filter
1	VC1_INTF	VC1 interrupt flag, 1 VC1 interrupt occurred; 0no interrupt occurred; write 0clear interrupt flag, write 1invalid
0	VC0_INTF	VC0 interrupt flag, 1 VC0 interrupt occurred; 0no interrupt occurred; write 0clear interrupt flag, write 1invalid

23 Low Voltage Detector (LVD)

23.1 LVD Introduction

LVD can be used to monitor the voltage of VCC and chip pins. When the result of the comparison between the monitored voltage and the LVD threshold meets the trigger condition, LVD generates an interrupt or reset signal, and the user can perform some emergency tasks based on this signal. LVD has the following characteristics.

- 4 Road monitoring sources, VCC, P03, P23, P25.
- 16 Order threshold voltage, flexible and versatile.
- 8 A combination of trigger conditions, high, rising and falling edges.
- 2 kind of trigger result, reset, interrupt.
- 8 Order filtering configuration to prevent false triggering.
- With hysteresis function, strong anti-interference.

23.2 LVD Block Diagram

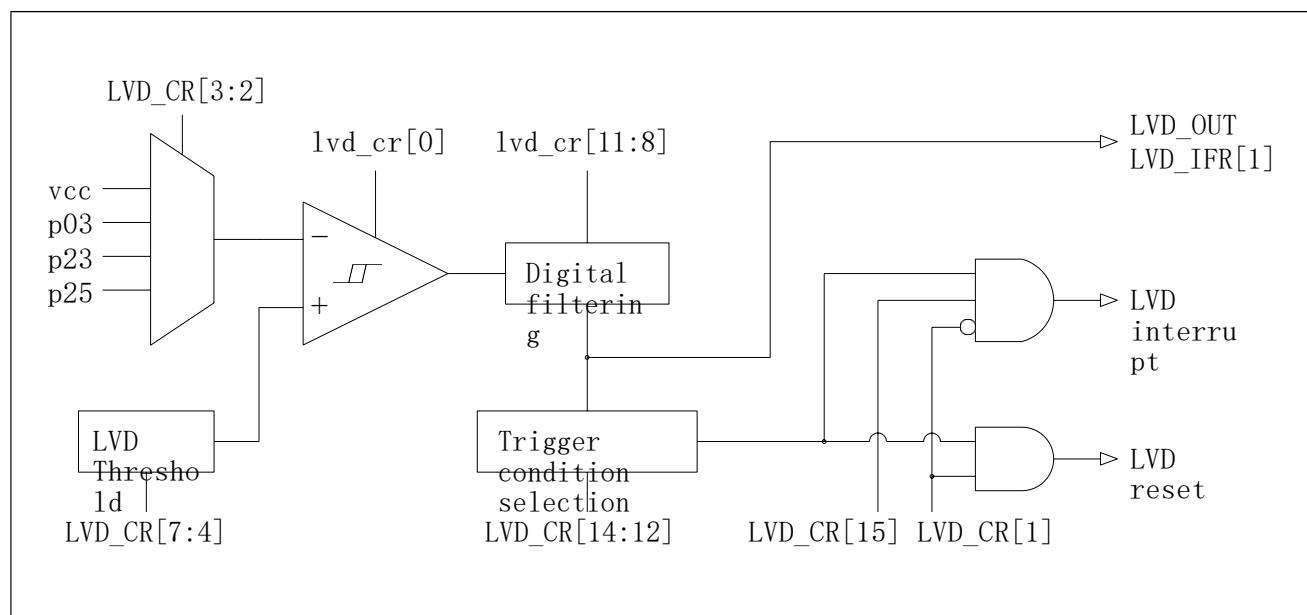


Figure 23-1LVD Block Diagram

23.3 Digital filtering

If the chip operates in a harsh environment, the output of the hysteresis comparator will have noise signals. If the digital filter module is enabled, any noise signal with pulse width less than LVD_CR.Debounce_time in the output waveform of the hysteresis comparator can be filtered out. If the digital filter module is disabled, the input and output signals of the digital filter module are the same.

The digitally filtered signal level can be read out from register LVD_IFR[1]; when GPIO function is configured as LVD_OUT, the digitally filtered signal can be output from GPIO for easy measurement.

The digital filter module is enabled and the filtering schematic is shown below.

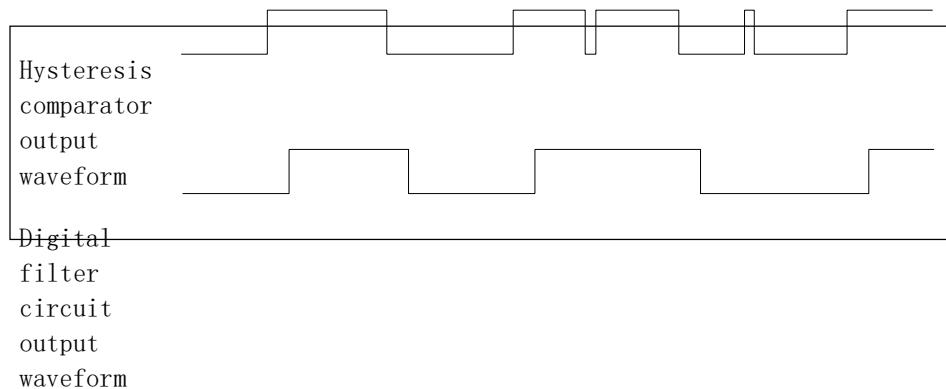


Figure 23-2LVDFilter Output

23.4 Hysteresis function

The LVD built-in voltage comparator has a hysteresis function, and its output signal will wait until the input signal is 20mV above or below the threshold voltage before flipping. The hysteresis function enhances the immunity of the chip, as shown in the figure below.

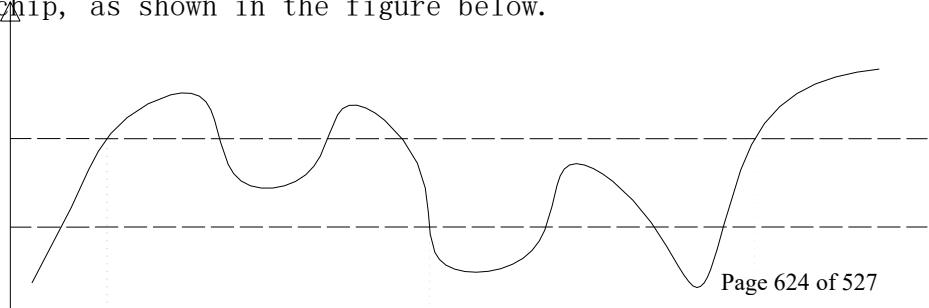




Figure 23-3LVDHysteresis Response

23.5 Configuration Example

23.5.1 LVD configured for low voltage reset

In this mode, the MCU is reset when the monitoring voltage falls below the threshold voltage. the configuration method is shown below.

Step1: Configure LVD_CR.Source_sel to select the voltage source to be monitored. step2: Configure LVD_CR.VTDS to select the threshold voltage of LVD. step3: Configure

LVD_CR.Debounce_time to select the LVD filtering time.

step4: Configure LVD_CR.FLTEN to enable the LVD filtering. LVD filtering.

Step5: Set LVD_CR.HTEN tol , select high level to trigger LVD action. Step6: Set LVD_CR.ACT to 11, select LVD action as reset.

23.5.2 LVD configured for voltage change interrupt

In this mode, an interrupt is generated when the monitoring voltage is above or below the threshold voltage. The configuration method is shown below.

Step1: Configure LVD_CR.Source_sel to select the voltage source to be monitored. step2: Configure LVD_CR.VTDS to select the threshold voltage of LVD. step3: Configure LVD_CR.Debounce_time to select the LVD filtering time.

step4: Configure LVD_CR.FLTEN to enable the LVD filtering. LVD filtering.

Step5: Set LVD_CR.RTEN and LVD_CR.FTEN tol select the level change to trigger

the LVD action.

Step6: Set LVD_CR.ACT to0 , to select LVD action as interrupt. Step7: Set LVD_CR.IE to1 , to enable LVD interrupt. Step8: Enable LVD interrupt in NVIC interrupt vector table. Step9: Set LVD_CR.LVDEN to 1, to enable LVD.

Step10: Write 0x00 to LVD_IFR in the interrupt service program to clear the interrupt flag.

23.6 LVD Register

Base address 0x40002400

Register	Offset Address	Description
LVD_CR	0x028	LVD Configuration Register
LVD_IFR	0x02C	LVD Interrupt Flag Register

Table 23-1LVDRegisters

23.6.1 LVD Configuration Register (LVD_CR)

Offset address 0x028

Reset value 0x000000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	HTEN	RTEN	FTEN	Debounce_time		FLT EN	VTDS			Source_sel	ACT	LVD EN			
R/W	R/W	R/W	R/W	R/W		R/W	R/W			R/W	R/W	R/W			

position	Marker	Function Description
31:16	Reserved	Reserved
15	IE	LVD interrupt enable1: enable. 0: Prohibited.
14	HTEN	High level trigger enable (monitored voltage is below the threshold voltage)1: enable. 0: Prohibited.
13	RTEN	Rising edge trigger enable (monitored voltage changes from above to below the threshold voltage)1: enable. 0: Prohibited.
12	FTEN	Falling edge trigger enable (the monitored voltage changes from below to above the threshold voltage)1: enable. 0: Prohibited.

11:9	Debounce_time	Digital filtering time configuration 111: Filtering time is about 28.8ms 110: Filtering time is about 7.2ms 101: Filtering time is about 1.8ms 100: Filtering time of approximately 450us
------	---------------	---

		011: Filtering time of approximately 112us 010: Filtering time of approximately 28us 001: Filtering time of approximately 14us 000: Filtering time of approximately 7us Note: The filtering time is only valid when FLTEN is 1
8	FLTEN	Digital filtering function configuration 1: Enables digital filtering 0: Disable digital filtering
7:4	VTDS	LVD monitoring voltage selection 1111: 3.3v 1110: 3.2v 1101: 3.1v 1100: 3.0v 1011: 2.9v 1010: 2.8v 1001: 2.7v 1000: 2.6v 0111: 2.5v 0110: 2.4v 0101: 2.3v 0100: 2.2v 0011: 2.1v 0010: 2.0v 0001: 1.9v 0000: 1.8v
3:2	Source_sel	LVD Monitoring Source Selection 11: P2.5 port input voltage 10: P2.3 port input voltage 01: P0.3 port input voltage 00: VCC supply voltage
1	ACT	LVD trigger action selection1: system reset 0: NVIC interrupts
0	LVDEN	LVD control1 : Enables LVD 0: LVD ban

23.6.2 LVD Interrupt Register (LVD_IFR)

Offset address 0x02C

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															LVD_Filter
Reserved															INTF
RO															R/W

position	Marker	Function Description
31:2	Reserved	Reserved
1	LVD_Filte	State after LVD Filter
0	INTF	LVD interrupt flag:1 : LVD interrupt occurred. 0: No interruptions have occurred. Write to 0clear the interrupt flag, write is invalid.

24 Simulate other registers

Base address 0x40002400

Register	Offset Address	Description
BGR_option	0x078	BGR control register

24.1 BGR Configuration Register (BGR_CR)

Offset address 0x000

Reset value 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																
															TS_EN	BGR_EN
															R/W	R/W

position	Marker	Function Description
31:2	Reserved	Reserved
1	TS_EN	<p>Internal temperature sensor enable 1: Activate internal temperature sensor 0: Disable internal temperature sensor Note: The TS requires a start-up time of approximately 20us to stabilize.</p>
0	BGR_EN	<p>BGR Enable Control 1: Enables BGR 0: BGR is prohibited Caution. 1) This register can be operated only when PERI_CLKEN.ADC is 2) The BGR can be used by other modules only after the BGR is stabilized, so the user should add the step of waiting for the BGR to be stabilized in the operation. 3) When using ADCs, BGR must be enabled. 4) When using VC, it is necessary to decide whether to enable BGR according to the configuration of VC register.</p>

25 SWD Debugging Interface

This series uses the ARM Cortex-M0+ core, which has a hardware debug module, SWD, to support complex debug operations. The hardware debug module allows the kernel to stop when fetching fingers (instruction breakpoints) or accessing data (data breakpoints). When the kernel is stopped, both the internal state of the kernel and the external state of the system can be queried in the IDE. After completing the query, the kernel and peripherals can be recovered and the program will continue to execute. When the HC32L110 microcontroller is connected to the debugger and starts debugging, the debugger will use the hardware debug module of the kernel to perform debugging operations.

Caution.

- SWD does not work in DeepSleep mode, please perform debugging operation in Active and Sleep mode.

25.1 SWD Debugging Add-On

This product uses the ARM Cortex-M0+ CPU, which includes hardware extensions for advanced debug functionality, so the debug features available in this product are consistent with the Cortex-M0+. The debug extensions allow the kernel to stop the kernel when fetching fingers (instruction breakpoints) or fetching access data (data breakpoints). When the kernel is stopped, the internal state of the kernel and the external state of the system can be queried. When the query is complete, the kernel and system are restored and program execution resumes.

The debug function is used when the debug host is connected to the MCU and debugging is performed.

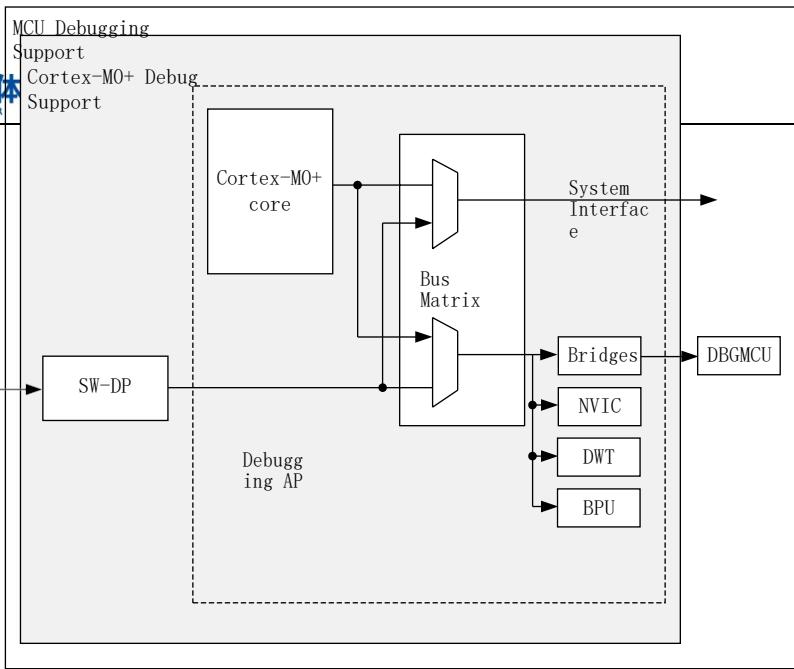


Figure 25-1 Debugging Support Block Diagram

The debug functionality built into the Cortex®-M0+ core is part of the ARM® CoreSight Design Suite.

The ARM® Cortex®-M0+ core provides integrated on-chip debug support. It includes.

- SW-DP: Serial line
- BPU: Breakpoint Unit
- DWT: Data

Observation Point

TriggerNote.

- For more information on the debugging features supported by the ARM® Cortex®-M0+ core, see the Cortex®- M0+ Technical Reference Manual.

25.2 ARM® Reference Documentation

- Cortex®-M0+ Technical Reference Manual (TRM)
Available from www.infocenter.arm.com.
- ARM® Debug Interface V5
- ARM® CoreSight Design Suite Version r1p1 Technical Reference Manual

25.3 Debug port pins

25.3.1 SWD Port Pins

The SWD interface of this series requires the use of the following 2pins.

SWD Port Name	Debugging functions	Pin Assignment
SWCLK	Serial Clock	P31
SWDIO	Serial data input/output	P27

25.3.2 SW-DP Pin Assignment

If the [Encryption Chip] option is enabled when burning the program, the SWD debugging function is disabled after power up. If the [Encryption Chip] option is not enabled when burning the program, both P27/P31 pins are initialized to be dedicated pins for debugger use after power on. User can set SYSCTRL1 . SWD_USE_IO register to disable the debug function of SWD pin ,

an [Encryption Chip] option will be released.	SWD USE IO be used as P27/P31 Function GPIO. The Configuration	GPIO. The
configuration and functions of SWD pin are summarized in the following table.	0	NA
Encryption	1	GPIO
No encryption	0	SWD
No encryption	1	GPIO

25.3.3 Internal pull-up on SWD pin

The reset state of the GPIO control register places the I/O in the equivalent state of

- SWDIO: Input pull-up

-
- SWCLK: Input pull-up

With built-in pull-up and pull-down resistors, there is no need to add external resistors.

25.4 SWD Port

25.4.1 Introduction to SWD Protocol

This synchronous serial protocol uses two pins.

- SWCLK: Clock from host to target
- SWDIO: Bidirectional

With this protocol, two register sets (DPACC register set and APACC register set) can be read and written at the same time. When transferring data, the LSB comes first.

For SWDIO bi-directional management, the line must be pulled up on the board (ARM® recommends k100). These pull-up resistors can be configured internally. No external pull-up resistors are required.

Each time the direction of SWDIO is changed in the protocol, a transition time is inserted when the line is neither host-driven nor target-driven. By default, this transition time is one bit, but can be adjusted by configuring the SWCLK frequency.

25.4.2 SWD Protocol Sequence

Each sequence consists of three phases.

1. Packet request sent by host (8bit)
2. Acknowledgement response sent by the target (3bits)
3. Data transmission phase sent by host or target (33bits)

position	Name	Description
0	Start	Must be for1
1	APnDP	0: DP access;1 : AP access
2	RnW	0: Write request;1 : Read request
4:3	A[3:2]	Address field of the DP or AP register
5	Parity Check	The first few bits of the unit parity check
6	Stop	0
7	Residency	Not driven by the host. Because of the presence of pull-ups, they must be read by the target as 1

See the Cortex®-M0+ TRM for a detailed description of the DPACC and APACC registers. packet requests are always followed by the conversion time (default 1 bit) when neither the host nor the target is driven.

position	Name	Description
0	ACK	001: FAULT 010: WAIT 100: OK

The ACK response must be followed by the transition time only when a READ transaction occurs or when a WAIT or FAULT acknowledgement is

position	Received.	Description
0:31	WDATA or RDATA	Writing or reading data
32	Parity Check	32 Single parity check for one data bit

The DATA transfer must be followed by the conversion time only when a READ transaction occurs.

25.4.3 SW-DP state machine (reset, idle state, ID code)

The state machine of the SW-DP has an internal ID code that is used to identify the SW-DP. This code is compliant with the JEP-106 standard. This ID code is the default ARM® code, set to **0xBB011477** (equivalent to 0Cortex®-M+) Caution.

- The SW-DP state machine does not work until the target reads this ID code.
- The SW-DP state machine is reset after a power-on reset or after the line has been high for more than one 50cycle.
- If the line is low for at least two cycles after the reset state, the SW-DP state machine is idle.
- After resetting the state, this state machine must first enter the idle state and then perform a read access to the DP-SW ID CODE register. Otherwise, the target will issue a FAULT acknowledge response on another transaction.

For more detailed information on the SW-DP state machine, see the *Cortex®-M0+ TRM* and *CoreSight Design Kit r1p0TRM*.

25.4.4 DP and AP read/write access

- Read access to the DP is not delayed: the target response can be sent immediately (if ACK=OK) or delayed (if ACK=WAIT)
- Delays the read access to the AP. This means that the result of the access will be returned on the next transfer. If the next access to be performed is not an AP access, the DP-RDBUFF register must be read to get the result.

- The DP-CTRL/STAT register is updated each time an AP read access or RDBUFF read request is made.
READOK flag to know if the AP read access was successful.
- The SW-DP has a write buffer (for DP or AP writes) so that write operations can be accepted even when other operations are still pending. If the write buffer is full, the target acknowledgement response is WAIT, except for IDCODE reads, CTRL/STAT reads, or ABORT writes, which are accepted even when the write buffer is full.
- Due to the presence of the asynchronous clock domains SWCLK and HCLK, two additional SWCLK cycles are required after the write operation (after the parity bit) to allow the write operation to take effect internally. These cycles should be applied when the line is driven low (idle state).

This is especially important when writing the CTRL/STAT registers to make a power-up request. Otherwise the next operation (operations that are valid until after the kernel is powered up) are executed immediately, which will result in a failure.

25.4.5 SW-DP Register

These registers are accessible when APnDP=0.

A[3:2]	RW	SELECT Register The CTRLSEL bit of the	Register	Note
00	Read		IDCODE	Manufacturer code set to default ARM® for Cortex®-M0+ Code. 0x0BB11477 (identifies SW-DP)
00	write		ABORT	
01	Read / Write	0	DP- CTRL/STAT	Objective. – Request system or commissioning power-up – Configure transport operations for AP access – Control comparison and verification operations – Read some status flags (overflow and power-up confirmation)

01	Read / Write	1	WIRE CONTROL	Used to configure physical serial port protocols (e.g., conversion time for continuous) (Time)
10	Read		READ RESEND	Allows recovery of read data from a corrupted debug software transfer. No need to repeat the original AP transfer.
10	write		SELECT	Word4 register window for selecting the current access port and activity
11	Read / Write		READ BUFFER	This read buffer is useful because AP access has been issued (provides the result of the read AP request when the next AP transaction is executed). This read buffer captures the data in the AP and is displayed as

The result of the previous read does not need to start a new operation.

25.4.6 SW-AP register

These registers are accessible when APnDP=1.

There are multiple AP registers, which are addressed in the following combinations.

- Shift value A[3:2]
 - Current value of DP SELECT register

Address	A[3:2] value	Description
0x0	00	Reserved, the reset value must be maintained.
0x4	01	DP CTRL/STAT register. Used for. – Request system or commissioning power-up – Configure transport operations for AP access – Control comparison and verification operations – Read some status flags (overflow and power-up confirmation)
0x8	10	DP SELECT Register: Used to select the current access port and the active word4 register window. – Bit 31:24: APSEL: select the current AP – Bit 23:8: Reserved – Bits 7:4: APBANKSEL: Selects the active word4 register window on the current AP – Bit 3:0: Reserved
0xC	11	DP RDBUFF register: used to get the final result after executing a series of operations through the debugger.

25.5 Kernel debugging

Debug the kernel through the kernel debug registers. Debug access to these registers through the debug access port. It consists of four registers.

Register	Description
DHCSR	32 Bit debug stop control and status registers This register provides information about the status of the processor, enables the kernel to enter the debug stop state and provides processor stepping capabilities.
DCRSR	17 Bit debug kernel register selector register. This register selects the processor registers that require read and write operations.
DCRDR	32 Bit debug kernel register data register. This register holds the data read and written between the register and the processor selected by the DCRSR (selector) register.
DEMCR	32 Bit debug exception and monitoring control registers. These registers are not reset on system reset. They can only be reset by a power-on reset. For more details see the Cortex®-M0+ TRM.

In order to put the kernel into the debug stop state immediately after a reset, it is necessary to.

- Enable bit 0 of the debug and exception monitoring control register (VC_CORRESET)
- Enable the debug stop control and status register bits (0C_DEBUGEN)

25.6 BPU (Breakpoint Unit)

The Cortex®-M0+ BPU implementation provides four breakpoint registers.

25.6.1 BPU Function

Processor breakpoints implement PC-based breakpoint functionality.

For more information on the BPU CoreSight identification registers and their address and access types, see the ARMv6-M ARM® and ARM® CoreSight Component Technical Reference Manual.

25.7 DWT (Data Watch Point)

The Cortex®-M0+ DWT implementation provides two watchpoint register sets.

25.7.1 DWT Function

The processor watchpoint implements data address and PC-based watchpoint functionality (i.e., PC sample registers) and supports comparator address masks, as described in ARMv6-M ARM®.

25.7.2 DWT Program Counter Sampling Register

The processor implementing the Data Watchpoint unit also implements the ARMv6-M optional DWT Program Counter Sampling Register (`DWT_PCSR`). This register allows the debugger to sample the PC periodically without stopping the processor. This can provide rough analysis. For more information, see ARMv6-M ARM®.

Cortex®-M0+ `DWT_PCSR` records the instructions that pass the condition codes and instructions as well as those that do not pass the condition codes.

25.8 MCU Debugging Gathering (DBG)

MCU debug components help the debugger provide support for.

- Low power consumption mode
- Timer, watchdog clock control during breakpoints

25.8.1 Debugging support for low-power modes

To enter low power mode, the command WFI or WFE must be executed.

The MCU supports several low-power modes that disable the CPU clock or reduce CPU power consumption.

The kernel does not allow to turn off FCLK or HCLK during the debug session, they must remain active because they are needed for debug connections during debugging. the MCU integrates special methods that allow the user to debug software in low-power mode.

25.8.2 Debugging support for timers, watchdogs

During the breakpoint, the timer and the watchdog's counter must be selected to behave in the following way.

- The counter continues to count when a breakpoint is generated. This is often required, for example, when a PWM controls a motor.
- The counter stops counting when a breakpoint is generated. This is required when used for watchdogs.

25.9 Debug mode module working state control (DEBUG_ACTIVE)

Reset value 0x00000FFF (This register setting works only in

SWD debug mode) Offset address: 0x038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		LPTIM	Res.	RTC	WDT	PCA	TIM6	TIM5	TIM4	LPTIM	TIM2	TIM1	TIM0		
RW		RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

position	Marker	Function Description
31:12	Reserved	Reserved
11	LPTIM	Timer3 counting function configuration during debugging 1: Pause Timer3 count function in SWD debug screen 0: Timer3 normal counting function in SWD debug interface
10	Reserved	Reserved
9	RTC	When debugging, the RTC counting function is configured 1: Suspend the RTC counting function in the SWD debug screen 0: RTC normal counting function under SWD debug interface
8	WDT	WDT counting function configuration during debugging 1: Suspend the WDT counting function in the SWD debug screen 0: WDT normal counting function under SWD debug interface
7	PCA	PCA counting function configuration during debugging 1: Suspend the PCA counting function in the SWD debug screen 0: PCA normal counting function under SWD debug interface
6	TIM6	Timer6 counting function configuration during debugging 1: Suspend the Timer count function in the SWD debug screen 0: Timer normal counting function in SWD debug interface
5	TIM5	Timer5 counting function configuration during debugging 1: Suspend the Timer count function in the SWD debug screen 0: Timer normal counting function in SWD debug interface
4	TIM4	Timer4 counting function configuration during debugging 1: Suspend the Timer count function in the SWD debug screen 0: Timer normal counting function in SWD debug interface
3	LPTIM	When debugging, the LpTimer counting function is configured 1: Suspend the Timer count function in the SWD debug screen 0: Timer normal counting function in SWD debug interface
2	TIM2	Timer2 count function configuration during debugging 1: Suspend the Timer count function in the SWD debug screen

		0: Timer normal counting function in SWD debug interface
1	TIM1	When debugging, the Timer1 counting function is configured 1: Suspend the Timer count function in the SWD debug screen 0: Timer normal counting function in SWD debug interface
0	TIM0	Timer0 count function configuration during debugging 1: Suspend the Timer count function in the SWD debug screen 0: Timer normal counting function in SWD debug interface

26 Electronic signature of devices

The electronic signature is stored in the system memory area of the flash memory module and can be read by the SWD or the CPU. It contains the chip identification information written at the factory and can be read by user firmware or external devices to automatically match different configurations of HC32Fxxx / HC32Lxxx microcontrollers.

26.1 Product Unique Identifier (UID) register (80bits)

Typical Application Scenarios for Unique Identifiers.

- Used as serial number
- Use UIDs as security keys when used in conjunction with software cryptographic primitives and protocols to improve the security of code in Flash before programming internal Flash
- Activating the safe bootstrap process, etc.

80 The bit unique device identifier provides a reference number that is unique for any device and any context. The user can never change these bits.80 Bitwise unique device identifiers can also be read in different ways such as single byte/half word/word and then concatenated using a custom algorithm.

Base address: 0x0010 0E74

Offset Address	Description	UID Bit(80bit)							
		7	6	5	4	3	2	1	0
0	X Coordinate on the wafer	UID[7:0]							
1	Rev ID	UID[15:8]							
2	Fixed value FFH	Reserved							
3	Fixed value FFH	Reserved							
4	Fixed value 00H	UID[23:16]							
5	Fixed value 00H	UID[31:24]							
6	Wafer Number	UID[39:32]							
7	Y Coordinate on the wafer	UID[47:40]							
8		UID[55:48]							
9		UID[63:56]							

10	Wafer Lot Number	UID[71:64]
11		UID[79:72]

26.2 Product Model Register

0x0010 0C60 ~ 0x0010 0C6F stores the ASCII code of the product model number.

If the product model number is less than one byte¹⁶ filled with 0x00.

Example: 48433324C3133364B38544100000000 The product model represented is HC32L136K8TA.

26.3 FLASH Capacity Register

Base address: 0x0010 0C70

31302928272625242322212019181716

FlashSize[31:16]
R

1514131211109876543210

FlashSize[15:0]
R

position	Marker	Function Description
31:0	FlashSize	The capacity of the product's built-in Flash, in bytes 0x00008000 means the Flash capacity is 32K Byte

26.4 RAM Capacity Register

Base address: 0x0010 0C74

31302928272625242322212019181716

RamSize[31:16]
R

1514131211109876543210

RamSize[15:0]
R

position	Marker	Function Description
31:0	RamSize	Capacity of the product's internal RAM in bytes 0x00000800 means the RAM capacity is 2K Byte

26.5 Number of Pins Register

Base address: 0x0010 0C7A

1514131211109876543210

PinCount[15:0]
R

position	Marker	Function Description
15:0	PinCount	Number of product pins, in units of only 0x0020 represents the number of product pins 32

27 Appendix ASysTickTimer

27.1 SysTick Timer Introduction

To support multitasking, the OS needs to perform context switching periodically, which requires hardware resources such as timers to interrupt program execution. The Cortex-M0 processor has a simple timer called SysTick that generates periodic interrupt requests.

The SysTick is a bit24 timer and counts down. When0 the timer count decreases, a programmable value is reloaded and a SysTick exception (exception number 15) is generated, which causes the execution of SysTick exception handling, a process that is part of the OS.

For systems that do not require an OS, the SysTick timer can also be used for other purposes, such as timing, timing or providing an interrupt source for tasks that require periodic execution. The generation of SysTick exceptions is controlled, and if exceptions are disabled, the SysTick timer can still be used in a polled way, such as checking the current count value or polling for overflow flags.

27.2 Set SysTick

Since both the overloaded and current values of the SysTick timer are undefined at reset time, the configuration of the SysTick needs to follow a certain procedure in order to prevent abnormal results.

Step1: Configure SysTick->CTRL. ENABLE to0 disable SysTick.

Step2: Configure SysTick->CTRL. CLKSOURCE and SYSTICK_CR[27:25] to select the clock source of SysTick.

Step3: Configure SysTick->LOAD and select the overflow period of SysTick.

Step4: Write any value to SysTick->VAL, clear SysTick->VAL and SysTick->CTRL.COUNTFLAG.

Step5: Configure SysTick->CTRL.TICKINT to enable SysTick interrupt. Step6: Configure SysTick->CTRL.ENABLE to enable SysTick. Step7: Read SysTick->CTRL in the interrupt service program to clear the overflow flag.
Note: The Systick overflow period is SysTick->LOAD+1. The configuration example is as follows.

Clock source	SysTick->LOAD	Overflow cycle
RCH4M	3999	1ms
XTL 32.768K	327	10.01ms

27.3 SysTick register

Address	Name	CMSIS Symbols	Full Name
0XE000E010	SYS_CSR	SysTick->CTRL	SysTick control and status registers
0XE000E014	SYS_RVR	SysTick->LOAD	SysTick Reload Value Register
0XE000E018	SYS_CVR	SysTick->VAL	SysTick current value register

27.3.1 SysTick Control and Status Register (CTRL)

position	Symbols	Function Description	Type	Reset value
31:17	Reserved	-	-	-
16	COUNTFLAG	Systick timer overflow flag1: Systick timer under overflow occurred. 0: Systick timer has not overflowed. Read this register to clear the COUNTFLAG flag	RO	0
15:3	Reserved	-	-	-
2	CLKSOURCE	SysTick clock source selection1: kernel clock (HCLK) 0: Reference clock, determined by SYSTICK_CR[27:25]	R/W	0
1	TICKINT	SysTick Interrupt Enable1: Enables interrupts 0: Disable interruption	R/W	0
0	ENABLE	SysTick Timer Enable1: Enables SysTick 0: Disable SysTick	R/W	0

27.3.2 SysTick Reload Register (LOAD)

position	Symbols	Function Description	Type	Reset value
31:24	Reserved	-	-	-
23:0	RELOAD	SysTick timer reload value	RW	-

27.3.3 SysTick Current Value Register (VAL)

position	Symbols	Function Description	Type	Reset value
31:24	Reserved	-	-	-
23:0	CURRENT	Read this register to get the current count value of SysTick timer Write any value to this register, clear this register and COUNTFLAG	RW	-

28 Appendix B Documentation Conventions

28.1 List of abbreviations related to registers

The following acronyms are used in the register descriptions.

RW read and write, software can read and write these bits.

RO is read-only, and the software can only read these bits.

WO write-only, the software can only write to this bit. Reading this bit will return invalid data.

W1 write only1, hardware auto-clear0, write invalid0

R0W1 software reads the bit as 0, and writes to clear the 1bit to zero. Writing has no effect on0 the value of this bit.

RW0 software can read and write this bit, write Invalid1, write Clear0

R1W0 software reads the bit as 1, and writes to clear the 0bit to zero. Writing has no effect on1 the value of this bit.

The RC software can read this bit. When this bit is read, it will be automatically cleared to zero. Writing a "0" has no effect on the value of this bit.

Res, Reserverd Reserved bits, must hold the reset value.

28.2 Glossary

This section provides a brief definition of the acronyms and abbreviations used in this document.

Word:32 Bit

data . half Word:16

Bit data. **byte:8 Bit**
data.

IAP(In-Application Programming) IAP means that the microcontroller's Flash can be reprogrammed during the user program run.

ICP(in-circuit programming) ICP means that the JTAG protocol can be used while the device is mounted on the user's application board.

SWD protocol or bootstrap program to program the microcontroller's Flash.

AHB: Advanced High

Performance Bus. **APB:**

Low Speed Peripheral

Bus. **DMA:** Direct

Memory Access. **TIM:**

Timer

Imprint & Contact

Version	Revision Date	Summary of Revisions
Rev1.0	2018/1/23	First edition of HC32L110 Series User's Manual is released.
Rev1.1	2018/5/4	Version update, corrected Flash data, revised 4chapter part description.
Rev1.2	2018/5/23	Modified the clock switching process and added PCA comparison capture function module settings.
Rev1.3	2018/11/1	Add 1chapter function module description, add 2chapter description pin configuration and function, add 9chapter FLASH operation description, modify chapter 28Chapter electrical characteristics parameters, add 29chapter and 31chapter.
Rev1.4	2018/11/26	Modify the name: UART2→LPUART. add "Note" to section 2.1and. 2.2
Rev1.5	2019/2/22	Correct the following data: ① ADC characteristics ② ESD characteristics ③ ECFLASH minimum value in memory characteristics ④ Increase package size ⑤ Add AVCC/AVSS to the pin configuration diagram.
Rev1.6	2019/7/5	Correct the following data: ① Correct the UID address ② Correct the programming mode ③ Update the QFN pin configuration diagram style.
Rev1.7	2019/12/13	Revise the following data: ① Typical application circuit diagram ② LVD block diagram ③ ADC characteristic unit ④ Voltage comparator frame diagram ⑤ Flag bit register (UARTx_ISR) ⑥ External clock source characteristics in XTH and XTL mapping and precautions ⑦ Update I2C bus (I2C), Serial Peripheral Interface (SPI), Universal Synchronous Asynchronous Transceiver (UART), and Low Power Synchronous Asynchronous Transceiver (LPUART) section description.
Rev1.8	2020/1/17	Correct the following data: ①Add CSP16 package ②Device electronic signature ③ Appendix A SysTick timer ④Cyclic redundancy check (CRC) ⑤ System control register1 (SYSCTRL1) add note ⑥ CR register (FLASH_CR) reset value (7) 0Bit of I2C configuration register (I2C_CR).
Rev1.9	2020/3/6	Fix the following data: ①Add attention item in programming mode ②17.5.1 and17.5.2 of Step8.
Rev2.0	2020/4/30	Correction of the following data: ① ADC characteristics to increase the accuracy of VCC/3 ② Correction of a pen error in the external clock source characteristics ③ Internal clock source characteristics Medium RCL oscillator accuracy.
Rev2.1	2020/5/29	Fix the following data: ①17.5 Add Step2 and Step3 in ②I2Cx is changed to I2C.
Rev2.2	2020/7/31	Fix the following data: ①Add TIM timer characteristics and communication interface section ②EFT level ③Internal in general operating conditions AHB/APB clock frequency ④ Correct pencil errors ⑤ Input characteristics in port characteristics - the values of viH and viL in ports P0,P1,P2,P3, RESET.
Rev2.3	2020/9/30	Correct the following data: ①7.8.4 Update the pen error ② Clock system description ③ RCH oscillator accuracy in internal clock source characteristics ④ viL and viH for RESETB pin characteristics ⑤ Add SPI characteristics.
Rev2.31	2020/12/31	Delete product features, pin configuration, package information, etc. (please refer to the latest datasheet for related information) and modify the declaration.



If you have any comments or suggestions in the process of purchase and use, please feel free to contact us.

Email: mcu@hdsc.com.cn

Website: <http://www.hdsc.com.cn/mcu.htm>

Correspondence Address: Room10 A, Building A,

No 1867. Zhongke Road, Pudong New Area, Shanghai,

P. R. China 201203

