



HC32L110 系列

32 位 ARM[®] Cortex[®]-M0+ 微控制器

参考手册

声 明

- 小华半导体有限公司（以下简称：“XHSC”）保留随时更改、更正、增强、修改小华半导体产品和/或本档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- 任何带有“®”或“™”标识的图形或字样是 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- 本通知中的信息取代并替换先前版本中的信息。

©2022 小华半导体有限公司 - 保留所有权利

目 录

声 明	2
目 录	3
简 介	23
1 系统结构.....	24
1.1 概述	24
1.2 系统地址划分.....	25
1.3 存储器和模块地址分配.....	27
2 工作模式.....	28
2.1 运行模式	30
2.2 休眠模式	31
2.3 深度休眠模式.....	33
3 系统控制器（SYSCTRL）	36
3.1 时钟源介绍	36
3.1.1 内部高速 RC 时钟 RCH	37
3.1.2 内部低速 RC 时钟 RCL.....	37
3.1.3 外部低速晶振时钟 XTL	38
3.1.4 外部高速晶振时钟 XTH.....	38
3.1.5 时钟启动过程.....	39
3.2 系统时钟切换.....	40
3.2.1 标准的时钟切换流程.....	40
3.2.2 从 RCH 切换到 XTL 示例	40
3.2.3 从 RCH 切换到 XTH 示例.....	41
3.2.4 从 RCL 切换到 XTH 示例	42
3.2.5 从 RCH 切换到 RCL 示例	43
3.2.6 从 RCL 切换到 RCH 示例	43
3.2.7 RCH 不同振荡频率间切换.....	44
3.3 时钟校准模块	45
3.4 中断唤醒控制	46
3.4.1 从深度休眠模式唤醒后执行中断服务程序的方法	46
3.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法	47
3.4.3 使用退出休眠特性	48
3.5 寄存器	49
3.5.1 系统控制寄存器 0（SYSCTRL0）	50
3.5.2 系统控制寄存器 1（SYSCTRL1）	52
3.5.3 系统控制寄存器 2（SYSCTRL2）	54
3.5.4 RCH 控制寄存器（RCH_CR）	55
3.5.5 振荡 XTH 控制寄存器（XTH_CR）	56
3.5.6 RCL 控制寄存器（RCL_CR）	57
3.5.7 XTL 控制寄存器（XTL_CR）	58
3.5.8 外围模块时钟控制寄存器（PERI_CLKEN）	59

3.5.9	SysTick 时钟控制 (SYSTICK_CR)	61
4	复位控制器 (RESET)	62
4.1	复位控制器介绍.....	62
4.1.1	上电下电复位 POR/BOR.....	63
4.1.2	外部复位引脚复位.....	63
4.1.3	WDT 复位.....	63
4.1.4	PCA 复位.....	63
4.1.5	LVD 低电压复位.....	63
4.1.6	Cortex-M0+ SYSRESETREQ 复位.....	63
4.1.7	Cortex-M0+ LOCKUP 复位.....	63
4.2	寄存器	65
4.2.1	复位标识寄存器 (Reset_flag)	65
4.2.2	外围模块复位控制寄存器 (PERI_RESET)	66
5	中断控制器 (NVIC)	68
5.1	概述	68
5.2	中断优先级	68
5.3	中断向量表	69
5.4	中断输入和挂起行为.....	70
5.5	中断等待	73
5.6	中断源	73
5.7	中断结构图	75
5.8	寄存器	77
5.8.1	中断使能设置寄存器 (SCS_SETENA)	77
5.8.2	中断使能清除寄存器 (SCS_CLRENA)	78
5.8.3	中断挂起状态设置寄存器 (SCS_SETPEND)	78
5.8.4	中断挂起状态清除寄存器 (SCS_CLRPEND)	79
5.8.5	中断优先级寄存器 (SCS_IPR0)	80
5.8.6	中断优先级寄存器 (SCS_IPR1)	81
5.8.7	中断优先级寄存器 (SCS_IPR2)	82
5.8.8	中断优先级寄存器 (SCS_IPR3)	83
5.8.9	中断优先级寄存器 (SCS_IPR4)	84
5.8.10	中断优先级寄存器 (SCS_IPR5)	85
5.8.11	中断优先级寄存器 (SCS_IPR6)	86
5.8.12	中断优先级寄存器 (SCS_IPR7)	87
5.8.13	中断屏蔽特殊寄存器 (SCS_PRIMASK)	88
5.9	软件基本操作.....	89
5.9.1	外部中断使能.....	89
5.9.2	NVIC 中断使能和清除使能	89
5.9.3	NVIC 中断挂起和清除挂起	89
5.9.4	NVIC 中断优先级	89
5.9.5	NVIC 中断屏蔽.....	90
6	端口控制器 (GPIO)	91

6.1	端口控制器简介.....	91
6.2	端口控制器主要特性.....	92
6.3	端口控制器功能描述.....	93
6.3.1	端口配置功能.....	93
6.3.2	端口的写入.....	95
6.3.3	端口的读取.....	96
6.3.4	端口复用功能.....	97
6.3.5	端口中断功能.....	98
6.4	端口配置操作.....	99
6.4.1	端口复用操作流程.....	99
6.4.2	端口中断操作流程.....	100
6.4.3	端口配置操作流程.....	101
6.5	端口控制器寄存器描述.....	102
6.5.1	端口 P0.....	105
6.5.1.1	端口 P01 功能配置寄存器(P01_SEL)	105
6.5.1.2	端口 P02 功能配置寄存器(P02_SEL)	106
6.5.1.3	端口 P03 功能配置寄存器(P03_SEL)	107
6.5.1.4	端口 P0 输入输出配置寄存器(P0DIR).....	108
6.5.1.5	端口 P0 输入值寄存器(P0IN)	109
6.5.1.6	端口 P0 输出值配置寄存器(P0OUT)	110
6.5.1.7	端口 P0 数模配置寄存器(P0ADS).....	111
6.5.1.8	端口 P0 驱动能力配置寄存器(P0DR).....	112
6.5.1.9	端口 P0 上拉使能配置寄存器(P0PU)	113
6.5.1.10	端口 P0 下拉使能配置寄存器(P0PD)	114
6.5.1.11	端口 P0 开漏输出配置寄存器(P0OD).....	115
6.5.1.12	端口 P0 高电平中断使能配置寄存器(P0HIE).....	116
6.5.1.13	端口 P0 低电平中断使能配置寄存器(P0LIE)	117
6.5.1.14	端口 P0 上升沿中断使能配置寄存器(P0RIE)	118
6.5.1.15	端口 P0 下降沿中断使能配置寄存器(P0FIE).....	119
6.5.1.16	端口 P0 中断状态寄存器(P0_STAT)	120
6.5.1.17	端口 P0 中断清除寄存器(P0_ICLR)	121
6.5.2	端口 P1.....	122
6.5.2.1	端口 P14 功能配置寄存器(P14_SEL)	122
6.5.2.2	端口 P15 功能配置寄存器(P15_SEL)	123
6.5.2.3	端口 P1 输入输出配置寄存器(P1DIR).....	124
6.5.2.4	端口 P1 输入值寄存器(P1IN)	125
6.5.2.5	端口 P1 输出值配置寄存器(P1OUT)	126
6.5.2.6	端口 P1 数模配置寄存器(P1ADS).....	127
6.5.2.7	端口 P1 驱动能力配置寄存器(P1DR).....	128
6.5.2.8	端口 P1 上拉使能配置寄存器(P1PU)	129
6.5.2.9	端口 P1 下拉使能配置寄存器(P1PD)	130
6.5.2.10	端口 P1 开漏输出配置寄存器(P1OD).....	131

6.5.2.11	端口 P1 高电平中断使能配置寄存器(P1HIE)	132
6.5.2.12	端口 P1 低电平中断使能配置寄存器(P1LIE)	133
6.5.2.13	端口 P1 上升沿中断使能配置寄存器(P1RIE)	134
6.5.2.14	端口 P1 下降沿中断使能配置寄存器(P1FIE)	135
6.5.2.15	端口 P1 中断状态寄存器(P1_STAT)	136
6.5.2.16	端口 P1 中断清除寄存器(P1_ICLR)	137
6.5.3	端口 P2	138
6.5.3.1	端口 P23 功能配置寄存器(P23_SEL)	138
6.5.3.2	端口 P24 功能配置寄存器(P24_SEL)	139
6.5.3.3	端口 P25 功能配置寄存器(P25_SEL)	140
6.5.3.4	端口 P26 功能配置寄存器(P26_SEL)	141
6.5.3.5	端口 P27 功能配置寄存器(P27_SEL)	142
6.5.3.6	端口 P2 输入输出配置寄存器(P2DIR)	143
6.5.3.7	端口 P2 输入值寄存器(P2IN)	144
6.5.3.8	端口 P2 输出值配置寄存器(P2OUT)	145
6.5.3.9	端口 P2 数模配置寄存器(P2ADS)	146
6.5.3.10	端口 P2 驱动能力配置寄存器(P2DR)	147
6.5.3.11	端口 P2 上拉使能配置寄存器(P2PU)	148
6.5.3.12	端口 P2 下拉使能配置寄存器(P2PD)	149
6.5.3.13	端口 P2 开漏输出配置寄存器(P2OD)	150
6.5.3.14	端口 P2 高电平中断使能配置寄存器(P2HIE)	151
6.5.3.15	端口 P2 低电平中断使能配置寄存器(P2LIE)	152
6.5.3.16	端口 P2 上升沿中断使能配置寄存器(P2RIE)	153
6.5.3.17	端口 P2 下降沿中断使能配置寄存器(P2FIE)	154
6.5.3.18	端口 P2 中断状态寄存器(P2_STAT)	155
6.5.3.19	端口 P2 中断清除寄存器(P2_ICLR)	156
6.5.4	端口 P3	157
6.5.4.1	端口 P31 功能配置寄存器(P31_SEL)	157
6.5.4.2	端口 P32 功能配置寄存器(P32_SEL)	158
6.5.4.3	端口 P33 功能配置寄存器(P33_SEL)	159
6.5.4.4	端口 P34 功能配置寄存器(P34_SEL)	160
6.5.4.5	端口 P35 功能配置寄存器(P35_SEL)	161
6.5.4.6	端口 P36 功能配置寄存器(P36_SEL)	162
6.5.4.7	端口 P3 输入输出配置寄存器(P3DIR)	163
6.5.4.8	端口 P3 输入值寄存器(P3IN)	165
6.5.4.9	端口 P3 输出值配置寄存器(P3OUT)	167
6.5.4.10	端口 P3 数模配置寄存器(P3ADS)	169
6.5.4.11	端口 P3 驱动能力配置寄存器(P3DR)	171
6.5.4.12	端口 P3 上拉使能配置寄存器(P3PU)	173
6.5.4.13	端口 P3 下拉使能配置寄存器(P3PD)	175
6.5.4.14	端口 P3 开漏输出配置寄存器(P3OD)	177
6.5.4.15	端口 P3 高电平中断使能配置寄存器(P3HIE)	179

6.5.4.16	端口 P3 低电平中断使能配置寄存器(P3LIE)	181
6.5.4.17	端口 P3 上升沿中断使能配置寄存器(P3RIE)	183
6.5.4.18	端口 P3 下降沿中断使能配置寄存器(P3FIE)	185
6.5.4.19	端口 P3 中断状态寄存器(P3_STAT)	187
6.5.4.20	端口 P3 中断清除寄存器(P3_ICLR)	189
6.5.5	端口辅助功能	191
6.5.5.1	端口辅助功能配置寄存器 1(GPIO_CTRL1)	191
6.5.5.2	端口辅助功能配置寄存器 2(GPIO_CTRL2)	193
6.5.5.3	端口辅助功能配置寄存器 3(GPIO_CTRL3)	195
6.5.5.4	端口辅助功能配置寄存器 4(GPIO_CTRL4)	197
7	FLASH 控制器 (FLASH)	199
7.1	概述	199
7.2	结构框图	199
7.3	功能描述	200
7.3.1	页擦除 (Sector Erase)	200
7.3.2	全片擦除 (Chip Erase)	201
7.3.3	写操作 (Program)	201
7.3.4	读操作	203
7.4	擦写时序	204
7.5	读等待周期	206
7.6	擦写保护	206
7.6.1	擦写保护位	206
7.6.2	PC 地址擦写保护	206
7.7	寄存器写保护	207
7.8	寄存器	208
7.8.1	TNVS 参数寄存器 (FLASH_TNVS)	208
7.8.2	TPGS 参数寄存器 (FLASH_TPGS)	209
7.8.3	TPROG 参数寄存器 (FLASH_TPROG)	209
7.8.4	TSERASE 寄存器 (FLASH_TSERASE)	210
7.8.5	TMERASE 参数寄存器 (FLASH_TMERASE)	210
7.8.6	TPRCV 参数寄存器 (FLASH_TPRCV)	211
7.8.7	TSRCV 参数寄存器 (FLASH_TSRCV)	211
7.8.8	TMRCV 参数寄存器 (FLASH_TMRCV)	212
7.8.9	CR 寄存器 (FLASH_CR)	212
7.8.10	IFR 寄存器 (FLASH_IFR)	213
7.8.11	ICLR 寄存器 (FLASH_ICLR)	213
7.8.12	BYPASS 寄存器 (FLASH_BYPASS)	214
7.8.13	SLOCK 寄存器 (FLASH_SLOCK)	215
8	RAM 控制器 (RAM)	216
8.1	概述	216
8.2	功能描述	216
8.3	寄存器	217

8.3.1	控制寄存器 (RAM_CR)	217
8.3.2	奇偶校验出错地址寄存器 (RAM_ERRADDR)	218
8.3.3	出错中断标志寄存器 (RAM_IFR)	218
8.3.4	出错中断标志清除寄存器 (RAM_ICLR)	219
9	基本定时器 (TIM0/1/2)	220
9.1	基本定时器简介	220
9.2	Base Timer 功能描述	221
9.2.1	计数功能	223
9.2.2	Buzzer 功能	224
9.3	Base Timer 互连	225
9.3.1	GATE 互联	225
9.3.2	Toggle 输出互联	225
9.4	Base Timer 寄存器描述	226
9.4.1	16 位模式重载寄存器 (TIMx_ARR)	226
9.4.2	16 位模式计数寄存器 (TIMx_CNT)	227
9.4.3	32 位模式计数寄存器 (TIMx_CNT32)	227
9.4.4	控制寄存器 (TIMx_CR)	228
9.4.5	中断标志寄存器 (TIMx_IFR)	229
9.4.6	中断标志清除寄存器 (TIMx_ICLR)	229
10	低功耗定时器 (LPTIM)	230
10.1	LPTimer 简介	230
10.2	LPTimer 功能描述	231
10.2.1	计数功能	232
10.2.2	定时功能	232
10.3	LPTimer 互连	233
10.3.1	GATE 互联	233
10.3.2	EXT 互联	233
10.3.3	Toggle 输出互联	233
10.4	LPTimer 寄存器描述	234
10.4.1	计数器计数值寄存器 (LPTIM_CNT)	235
10.4.2	重载寄存器 (LPTIM_ARR)	235
10.4.3	控制寄存器 (LPTIM_CR)	236
10.4.4	中断标志寄存器 (LPTIM_IFR)	237
10.4.5	中断标志清除寄存器 (LPTIM_ICLR)	237
11	可编程计数阵列 (PCA)	238
11.1	PCA 简介	238
11.2	PCA 功能描述	239
11.2.1	PCA 定时/计数器	239
11.2.2	PCA 捕获功能	241
11.2.3	PCA 比较功能	243
11.2.3.1	16 位软件计数模式	243
11.2.3.2	高速输出模式	244

11.2.3.3	PCA 模块 4 的 WDT 功能.....	245
11.2.3.4	PCA 8 位脉宽调制功能.....	247
11.3	PCA 模块与其他模块互连及控制.....	249
11.3.1	ECl 互连.....	249
11.3.2	PCACAP0.....	249
11.3.3	PCACAP1.....	249
11.3.4	PCACAP [4:2].....	249
11.4	PCA 寄存器描述.....	250
11.4.1	控制寄存器 (PCA_CCON).....	251
11.4.2	模式寄存器 (PCA_CMOD).....	252
11.4.3	计数寄存器 (PCA_CNT).....	253
11.4.4	中断清除寄存器 (PCA_ICLR).....	253
11.4.5	比较捕获模式寄存器 (PCA_CCAPM0~4).....	254
11.4.6	比较捕获数据寄存器高 8 位 (PCA_CCAP0~4H).....	255
11.4.7	比较捕获数据寄存器低 8 位 (PCA_CCAP0~4L).....	255
11.4.8	比较捕获 16 位寄存器 (PCA_CCAP0~4).....	256
11.4.9	比较高速输出标志寄存器 (PCA_CCAPO).....	256
12	高级定时器 (TIM4/5/6).....	257
12.1	高级定时器简介.....	257
12.2	Advanced Timer 功能描述.....	259
12.2.1	基本动作.....	259
12.2.1.1	基本波形模式.....	259
12.2.1.2	比较输出.....	260
12.2.1.3	捕获输入.....	261
12.2.2	时钟源选择.....	262
12.2.3	计数方向.....	263
12.2.3.1	锯齿波计数方向.....	263
12.2.3.2	三角波计数方向.....	263
12.2.4	数字滤波.....	264
12.2.5	软件同步.....	265
12.2.5.1	软件同步启动.....	265
12.2.5.2	软件同步停止.....	265
12.2.5.3	软件同步清零.....	265
12.2.6	硬件同步.....	267
12.2.6.1	硬件同步启动.....	267
12.2.6.2	硬件同步停止.....	267
12.2.6.3	硬件同步清零.....	267
12.2.6.4	硬件同步捕获输入.....	267
12.2.6.5	硬件同步计数.....	267
12.2.7	缓存功能.....	269
12.2.7.1	缓存传送时间点.....	270
12.2.7.2	通用周期基准值缓存传送时间点.....	270

12.2.7.3	通用比较基准值缓存传送时间点.....	270
12.2.7.4	捕获输入值缓存传送时间点.....	270
12.2.7.5	清零动作时缓存传送.....	270
12.2.8	通用 PWM 输出	271
12.2.8.1	PWM 展频输出	271
12.2.8.2	独立 PWM 输出	271
12.2.8.3	互补 PWM 输出	271
12.2.8.3.1	软件设定 GCMBR 互补 PWM 输出.....	272
12.2.8.3.2	硬件设定 GCMBR 互补 PWM 输出.....	273
12.2.8.4	多相 PWM 输出	274
12.2.9	正交编码计数	276
12.2.9.1	位置计数模式	276
12.2.9.1.1	基本计数.....	276
12.2.9.1.2	相位差计数.....	277
12.2.9.1.3	方向计数.....	278
12.2.9.2	公转模式	278
12.2.9.2.1	Z 相计数	278
12.2.9.2.2	位置溢出计数.....	279
12.2.9.2.3	混合计数.....	279
12.2.9.2.4	Z 相动作屏蔽	280
12.2.10	周期间隔响应.....	282
12.2.11	保护机制.....	283
12.2.12	中断说明.....	284
12.2.12.1	计数比较匹配中断.....	284
12.2.12.2	计数周期匹配中断.....	284
12.2.12.3	死区时间错误中断.....	284
12.2.13	刹车保护	285
12.2.13.1	端口刹车与软件刹车.....	285
12.2.13.2	低功耗模式自动刹车.....	285
12.2.13.3	输出电平同高同低刹车.....	286
12.2.13.4	VC 刹车	286
12.2.14	内部互连.....	287
12.2.14.1	中断触发输出	287
12.2.14.2	AOS 触发	287
12.2.14.3	端口触发 TRIGA-TRIGD.....	288
12.2.14.4	比较输出 VC 与 Advanced Timer 互连	289
12.2.14.5	UART 与 Advanced Timer 互连	289
12.3	寄存器描述	290
12.3.1	通用计数基准值寄存器 (TIMx_CNTER).....	292
12.3.2	通用周期基准值寄存器 (TIMx_PERAR)	292
12.3.3	通用周期缓存寄存器 (TIMx_PERBR)	293
12.3.4	通用比较基准值寄存器 (TIMx_GCMAR-GCMDR)	293

12.3.5	死区时间基准值寄存器 (TIMx_DTUAR- DTDAR)	294
12.3.6	通用控制寄存器 (TIMx_GCONR)	295
12.3.7	中断控制寄存器 (TIMx_ICONR)	297
12.3.8	端口控制寄存器 (TIMx_PCONR)	298
12.3.9	缓存控制寄存器 (TIMx_BCONR)	301
12.3.10	死区控制寄存器 (TIMx_DCONR)	302
12.3.11	滤波控制寄存器 (TIMx_FCONR)	303
12.3.12	有效周期寄存器 (TIMx_VPERR)	305
12.3.13	状态标志寄存器 (TIMx_STFLR)	306
12.3.14	硬件启动事件选择寄存器 (TIMx_HSTAR)	308
12.3.15	硬件停止事件选择寄存器 (TIMx_HSTPR)	310
12.3.16	硬件清零事件选择寄存器 (TIMx_HCELR)	312
12.3.17	硬件捕获 A 事件选择寄存器 (TIMx_HCPAR)	314
12.3.18	硬件捕获 B 事件选择寄存器 (TIMx_HCPBR)	316
12.3.19	硬件递加事件选择寄存器 (TIMx_HCUPR)	318
12.3.20	硬件递减事件选择寄存器 (TIMx_HCDOR)	320
12.3.21	软件同步启动寄存器 (TIMx_SSTAR)	322
12.3.22	软件同步停止寄存器 (TIMx_SSTPR)	323
12.3.23	软件同步清零寄存器 (TIMx_SCLRR)	324
12.3.24	中断标志寄存器 (TIMx_IFR)	325
12.3.25	中断标志清除寄存器 (TIMx_ICLR)	327
12.3.26	展频及中断触发选择 (TIMx_CR)	328
12.3.27	AOS 选择控制寄存器 (TIMx_AOSSR)	329
12.3.28	AOS 选择控制寄存器标志清除 (TIMx_AOSCL)	330
12.3.29	端口刹车控制寄存器 (TIMx_PTBKS)	331
12.3.30	端口触发控制寄存器 (TIMx_TTRIG)	332
12.3.31	AOS 触发控制寄存器 (TIMx_ITRIG)	333
12.3.32	端口刹车极性控制寄存器 (TIMx_PTBKP)	334
13	实时时钟 (RTC)	335
13.1	实时时钟简介.....	335
13.2	实时时钟功能描述.....	336
13.2.1	上电设定.....	336
13.2.2	RTC 计数开始设定	336
13.2.3	系统低功耗模式切换.....	336
13.2.4	读出计数寄存器.....	337
13.2.5	写入计数寄存器.....	337
13.2.6	闹钟设定.....	338
13.2.7	1Hz 输出.....	338
13.2.8	时钟误差补偿.....	339
13.3	RTC 中断.....	341
13.3.1	RTC 闹钟中断	341
13.3.2	RTC 周期中断	341

13.4	RTC 寄存器描述	342
13.4.1	控制寄存器 0 (RTC_CR0)	343
13.4.2	控制寄存器 1 (RTC_CR1)	345
13.4.3	秒计数寄存器 (RTC_SEC)	347
13.4.4	分计数寄存器 (RTC_MIN)	347
13.4.5	时计数寄存器 (RTC_HOUR)	348
13.4.6	日计数寄存器 (RTC_DAY)	350
13.4.7	周计数寄存器 (RTC_WEEK)	351
13.4.8	月计数寄存器 (RTC_MON)	352
13.4.9	年计数寄存器 (RTC_YEAR)	352
13.4.10	分闹钟寄存器 (RTC_ALMMIN)	353
13.4.11	时闹钟寄存器 (RTC_ALMHOUR)	353
13.4.12	周闹钟寄存器 (RTC_ALM WEEK)	354
13.4.13	时钟误差补偿寄存器 (RTC_COMPEN)	355
14	看门狗定时器 (WDT)	357
14.1	WDT 简介	357
14.2	WDT 功能描述	358
14.2.1	WDT 溢出后产生中断	358
14.2.2	WDT 溢出后产生复位	358
14.3	WDT 寄存器描述	359
14.3.1	WDT 清除控制寄存器 (WDT_RST)	359
14.3.2	WDT_CON 寄存器	360
15	通用同步异步收发器 (UART)	361
15.1	概述	361
15.2	结构框图	361
15.3	主要特性	362
15.4	功能描述	363
15.4.1	工作模式	363
15.4.1.1	Mode0~Mode3 功能对比	363
15.4.1.2	Mode0 (同步模式, 半双工)	363
15.4.1.3	Mode1 (异步模式, 全双工)	365
15.4.1.4	Mode2 (异步模式, 全双工)	366
15.4.1.5	Mode3 (异步模式, 全双工)	367
15.4.2	波特率生成	368
15.4.2.1	Mode1/Mode3 波特率设置示例	369
15.5	帧错误检测	372
15.6	多机通讯	374
15.7	自动地址识别	375
15.7.1	给定地址	375
15.7.1.1	广播地址	375
15.7.1.2	举例	375
15.8	收发端缓存	376

15.8.1	接收缓存.....	376
15.8.2	发送缓存.....	376
15.9	寄存器.....	377
15.9.1	数据寄存器 (UARTx_SBUF)	377
15.9.2	控制寄存器 (UARTx_SCON)	378
15.9.3	地址寄存器 (UARTx_SADDR)	379
15.9.4	地址掩码寄存器 (UARTx_SADEN)	379
15.9.5	标志位寄存器 (UARTx_ISR)	380
15.9.6	标志位清除寄存器 (UARTx_ICR)	381
16	低功耗同步异步收发器 (LPUART)	382
16.1	概述.....	382
16.2	结构框图.....	383
16.3	主要特性.....	384
16.4	功能描述.....	385
16.4.1	配置时钟和传输时钟.....	385
16.4.2	工作模式.....	385
16.4.2.1	Mode0~Mode3 功能对比.....	386
16.4.2.2	Mode0 (同步模式, 半双工) 数据收发说明.....	387
16.4.2.3	Mode1 (异步模式, 全双工) 数据收发说明.....	388
16.4.2.4	Mode2 (异步模式, 全双工) 数据收发说明.....	389
16.4.2.5	Mode3 (异步模式, 全双工) 数据收发说明.....	390
16.4.3	波特率生成.....	391
16.5	帧错误检测.....	391
16.6	多机通讯.....	392
16.7	自动地址识别.....	393
16.7.1	给定地址.....	393
16.7.2	广播地址.....	393
16.7.3	举例.....	393
16.8	收发端缓存.....	394
16.8.1	接收缓存.....	394
16.8.2	发送缓存.....	394
16.9	寄存器.....	395
16.9.1	数据寄存器 (LPUART_SBUF)	395
16.9.2	控制寄存器 (LPUART_SCON)	396
16.9.3	地址寄存器 (LPUART_SADDR)	398
16.9.4	地址掩码寄存器 (LPUART_SADEN)	398
16.9.5	中断标志位寄存器 (LPUART_ISR)	399
16.9.6	中断标志位清除寄存器 (LPUART_ICR)	400
17	I2C 总线 (I2C)	401
17.1	简介.....	401
17.2	主要特性.....	401
17.3	协议描述.....	401

17.3.1	I2C 总线上数据传输	402
17.3.2	I2C 总线上的应答	403
17.3.3	I2C 总线上的仲裁	404
17.4	功能描述	406
17.4.1	串行时钟发生器	407
17.4.2	输入滤波器	407
17.4.3	地址比较器	407
17.4.4	应答标志位	408
17.4.5	中断产生器	408
17.4.6	工作模式	408
17.4.7	状态码表述	415
17.5	编程示例	417
17.5.1	主机发送示例	417
17.5.2	主机接收示例	418
17.5.3	从机接收示例	419
17.5.4	从机发送示例	420
17.6	寄存器描述	421
17.6.1	I2C 波特率计数器使能寄存器(I2C_TMRUN)	421
17.6.2	I2C 波特率计数器配置寄存器(I2C_TM)	422
17.6.3	I2C 配置寄存器(I2C_CR)	423
17.6.4	I2C 数据寄存器(I2C_DATA)	425
17.6.5	I2C 地址寄存器(I2C_ADDR)	426
17.6.6	I2C 状态寄存器(I2C_STAT)	427
18	串行外设接口 (SPI)	428
18.1	SPI 简介	428
18.2	SPI 主要特性	428
18.3	SPI 功能描述	429
18.3.1	SPI 主机模式	429
18.3.2	SPI 从机模式	430
18.3.3	SPI 数据帧格式	431
18.3.4	SPI 状态标志及中断	432
18.3.5	SPI 多机系统配置说明	433
18.3.6	SPI 管脚配置说明	434
18.4	SPI 编程示例	435
18.4.1	SPI 主机发送示例	435
18.4.2	SPI 主机接收示例	435
18.4.3	SPI 从机发送示例	436
18.4.4	SPI 从机接收示例	436
18.6	SPI 寄存器描述	438
18.6.1	SPI 配置寄存器(SPI_CR)	439
18.6.2	SPI 片选配置寄存器(SPI_SSN)	441
18.6.3	SPI 状态寄存器(SPI_STAT)	442

18.6.4	SPI 数据寄存器(SPI_DATA)	443
19	时钟校准模块 (CLKTRIM)	444
19.1	CLK_TRIM 简介	444
19.2	CLK_TRIM 主要特性	444
19.3	CLK_TRIM 功能描述	445
19.3.1	CLK_TRIM 校准模式	445
19.3.1.1	操作流程	445
19.3.2	CLK_TRIM 监测模式	446
19.3.2.1	操作流程	446
19.4	CLK_TRIM 寄存器描述	447
19.4.1	配置寄存器(CLKTRIM_CR)	448
19.4.2	参考计数器初值配置寄存器(CLKTRIM_REFCON)	449
19.4.3	参考计数器值寄存器(CLKTRIM_REFCNT)	449
19.4.4	校准计数器值寄存器(CLKTRIM_CALCNT)	450
19.4.5	中断标志位寄存器(CLKTRIM_IFR)	451
19.4.6	中断标志位清除寄存器(CLKTRIM_ICLR)	452
19.4.7	校准计数器溢出值配置寄存器(CLKTRIM_CALCON)	453
20	循环冗余校验 (CRC)	454
20.1	概述	454
20.2	主要特性	454
20.3	功能描述	454
20.3.1	工作模式	454
20.3.2	编码方式	454
20.3.3	写入位宽	454
20.4	编程示例	455
20.4.1	CRC-16 编码模式	455
20.4.2	CRC-16 检验模式	455
20.5	寄存器描述	456
20.5.1	寄存器列表	456
20.5.2	结果寄存器 (CRC_RESULT)	457
20.5.3	数据寄存器 (CRC_DATA)	457
21	模数转换器 (ADC)	458
21.1	模块简介	458
21.2	ADC 框图	458
21.3	转换时序及转换速度	459
21.4	单次转换模式	460
21.5	连续转换模式	462
21.6	连续转换累加模式	464
21.7	ADC 转换结果比较	466
21.8	ADC 中断	467
21.9	使用温度传感器测量环境温度	467
21.10	ADC 模块寄存器	469

21.10.1	ADC 配置寄存器 0 (ADC_CR0)	470
21.10.2	ADC 配置寄存器 1 (ADC_CR1)	472
21.10.3	ADC 配置寄存器 2 (ADC_CR2)	475
21.10.4	ADC 通道 0 转换结果 (ADC_result0)	477
21.10.5	ADC 通道 1 转换结果 (ADC_result1)	477
21.10.6	ADC 通道 2 转换结果 (ADC_result2)	478
21.10.7	ADC 通道 3 转换结果 (ADC_result3)	478
21.10.8	ADC 通道 4 转换结果 (ADC_result4)	479
21.10.9	ADC 通道 5 转换结果 (ADC_result5)	479
21.10.10	ADC 通道 6 转换结果 (ADC_result6)	480
21.10.11	ADC 通道 7 转换结果 (ADC_result7)	480
21.10.12	ADC 通道 8 转换结果 (ADC_result8)	481
21.10.13	ADC 转换结果累加值 (ADC_result_acc)	481
21.10.14	ADC 比较上阈值 (ADC_HT)	482
21.10.15	ADC 比较下阈值 (ADC_LT)	482
21.10.16	ADC 中断标志寄存器 (ADC_IFR)	483
21.10.17	ADC 中断清除寄存器 (ADC_ICLR)	484
21.10.18	ADC 结果 (ADC_result)	484
22	模拟比较器 (VC)	485
22.1	模拟电压比较器 VC 简介	485
22.2	电压比较器框架图	486
22.3	建立/响应时间	486
22.4	滤波时间	487
22.5	迟滞功能	487
22.6	VC 寄存器	488
22.6.1	VC 配置寄存器 (VC_CR)	489
22.6.2	VC0 配置寄存器 (VC0_CR)	491
22.6.3	VC1 配置寄存器 (VC1_CR)	493
22.6.4	VC0 输出配置寄存器 (VC0_OUT_CFG)	495
22.6.5	VC1 输出配置寄存器 (VC1_OUT_CFG)	497
22.6.6	VC 中断寄存器 (VC_IFR)	499
23	低电压检测器 (LVD)	500
23.1	LVD 简介	500
23.2	LVD 框图	500
23.3	数字滤波	501
23.4	迟滞功能	501
23.5	配置示例	502
23.5.1	LVD 配置为低电压复位	502
23.5.2	LVD 配置为电压变化中断	502
23.6	LVD 寄存器	503
23.6.1	LVD 配置寄存器 (LVD_CR)	503
23.6.2	LVD 中断寄存器 (LVD_IFR)	505

24	模拟其它寄存器.....	506
24.1	BGR 配置寄存器 (BGR_CR)	506
25	SWD 调试接口.....	507
25.1	SWD 调试附加功能.....	507
25.2	ARM® 参考文档	508
25.3	调试端口引脚.....	509
25.3.1	SWD 端口引脚	509
25.3.2	SW-DP 引脚分配.....	509
25.3.3	SWD 引脚上的内部上拉	509
25.4	SWD 端口.....	510
25.4.1	SWD 协议简介	510
25.4.2	SWD 协议序列	510
25.4.3	SW-DP 状态机 (复位、空闲状态、ID 代码)	511
25.4.4	DP 和 AP 读/写访问.....	511
25.4.5	SW-DP 寄存器.....	512
25.4.6	SW-AP 寄存器.....	513
25.5	内核调试	514
25.6	BPU (断点单元)	514
25.6.1	BPU 功能	514
25.7	DWT (数据观察点)	515
25.7.1	DWT 功能.....	515
25.7.2	DWT 程序计数器采样寄存器.....	515
25.8	MCU 调试组件 (DBG)	516
25.8.1	对低功耗模式的调试支持	516
25.8.2	对定时器、看门狗的调试支持	516
25.9	调试模式模块工作状态控制 (DEBUG_ACTIVE)	517
26	器件电子签名.....	519
26.1	产品唯一身份标识 (UID) 寄存器 (80 位)	519
26.2	产品型号寄存器.....	520
26.3	FLASH 容量寄存器.....	520
26.4	RAM 容量寄存器	521
26.5	管脚数量寄存器.....	521
27	附录 A SysTick 定时器.....	522
27.1	SysTick 定时器简介	522
27.2	设置 SysTick.....	522
27.3	SysTick 寄存器	523
27.3.1	SysTick 控制和状态寄存器 (CTRL)	523
27.3.2	SysTick 重载寄存器 (LOAD)	523
27.3.3	SysTick 当前值寄存器 (VAL)	524
28	附录 B 文档约定	525
28.1	寄存器相关缩写词列表.....	525
28.2	词汇表	525

版本记录 & 联系方式526

表目录

表 2-1	运行模式下可运行模块图	30
表 2-2	休眠模式下可运行模块图	32
表 2-3	深度休眠模式下可运行模块图	34
表 5-1	Cortex-M0+ 处理器异常一览	68
表 5-2	外部中断与 NVIC 中断输入对应关系	74
表 6-1	端口状态真值表	94
表 6-2	端口复用表	97
表 7-1	不同频率下 FLASH 擦写时间参数	204
表 9-1	Base Timer 寄存器列表	226
表 10-1	LPTimer 寄存器列表	234
表 11-1	PCA 比较捕获功能模块设置	248
表 11-2	PCA 寄存器列表	250
表 12-1	Advanced Timer 基本特性	257
表 12-2	Advanced Timer 端口列表	257
表 12-3	AOS 源选择	288
表 12-4	端口触发选择	288
表 12-5	Advanced Timer 寄存器列表	291
表 13-1	RTC 的基本特性	335
表 13-2	RTC 寄存器列表	342
表 14-1	WDT 寄存器列表	359
表 15-1	Mode0/1/2/3 数据结构	363
表 16-1	Mode0/1/2/3 数据结构	386
表 17-1	I2C 时钟信号波特率	407
表 17-2	I2C 状态码表述	416
表 17-3	寄存器列表	421
表 18-1	SPI 管脚配置说明表	434
表 18-2	SPI 寄存器列表	438
表 18-3	主机模式波特率选择	440
表 19-1	寄存器列表	447
表 21-1	ADC 寄存器	469
表 22-1	VC 寄存器	488
表 23-1	LVD 寄存器	503

图目录

图 1-1	系统架构示意图	24
图 1-2	地址区域划分示意图	26
图 2-1	控制模式框图	28
图 3-1	时钟控制模块框图	37
图 3-2	晶振时钟启动示意图	39
图 3-3	时钟切换示意图	42
图 3-4	时钟校准原理图	45
图 4-1	复位来源示意图	62
图 5-1	只使用了高两位的优先级寄存器	68
图 5-2	中断向量表	69
图 5-3	中断激活和挂起状态	70
图 5-4	中断挂起状态被清除然后被重新确认	71
图 5-5	中断退出时若中断请求保持高电平就会引起中断处理的再次执行	71
图 5-6	中断处理中产生的中断挂起也可以被确认	72
图 5-7	中断结构图	75
图 6-1	端口电路示意图	95
图 6-2	AHB 总线端口随系统时钟的变化	95
图 6-3	读取端口引脚数据同步图	96
图 7-1	存储器 Sector 划分	199
图 9-1	Base Timer 框图	220
图 9-2	Timer 模式 1 框图	222
图 9-3	Timer 模式 2 框图	222
图 9-4	模式 1 时序图	223
图 9-5	模式 2 时序图（预分频设置为 2）	223
图 10-1	LPTimer 框图	230
图 10-2	LPTimer 模式 1	231
图 10-3	LPTimer 模式 2	232
图 11-1	PCA 整体框图	238
图 11-2	PCA 计数器框图	240
图 11-3	PCA 捕获功能框图	242
图 11-4	PCA 比较功能框图	244
图 11-5	PCA WDT 功能框图	245
图 11-6	PCA PWM 功能框图	247
图 11-7	PCA PWM 输出波形	248
图 12-1	Advanced Timer 框图	258
图 12-2	锯齿波波形（递加计数）	259
图 12-3	三角波波形	259
图 12-4	比较输出动作	260
图 12-5	捕获输入动作	261
图 12-6	捕获输入端口的滤波功能	264

图 12-7	软件同步动作	265
图 12-8	硬件同步动作	268
图 12-9	单缓存方式比较输出时序	269
图 12-10	PWM 展频输出示意图	271
图 12-11	CHA 输出 PWM 波	271
图 12-12	三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出	272
图 12-13	三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出 (对称死区)	273
图 12-14	6 相 PWM 波	274
图 12-15	三角波 A 模式时带死区时间三相互补 PWM 波输出	275
图 12-16	位置模式时基本计数动作	276
图 12-17	位置模式时相位差计数动作设定(1 倍)	277
图 12-18	位置模式时相位差计数动作设定(2 倍)	277
图 12-19	位置模式时相位差计数动作设定(4 倍)	277
图 12-20	位置模式时方向计数动作	278
图 12-21	公转模式时 Z 相计数动作	278
图 12-22	公转模式时位置计数器输出计数动作	279
图 12-23	公转模式时 Z 相计数和位置计数器输出混合计数动作	279
图 12-24	公转计数模式-混合计数 Z 相屏蔽动作例 1	280
图 12-25	公转计数模式-混合计数 Z 相屏蔽动作例 2	281
图 12-26	周期间隔有效请求信号动作	282
图 12-27	端口刹车与软件刹车示意图	285
图 12-28	输出同高同低刹车示意图	286
图 12-29	VC 刹车控制示意图	286
图 12-30	Timer4/5/6 中断选择	287
图 13-1	RTC 框图	335
图 14-1	WDT 整体框图	357
图 15-1	UART 结构框图	361
图 15-2	Mode0 发送数据	364
图 15-3	Mode0 接收数据	364
图 15-4	Mode1 发送数据	365
图 15-5	Mode1 接收数据	365
图 15-6	Mode2 发送数据	366
图 15-7	Mode2 接收数据	366
图 16-1	LPUART 结构框图	383
图 16-2	Mode0 发送数据	387
图 16-3	Mode0 接收数据	387
图 16-4	Mode1 发送数据	388
图 16-5	Mode1 接收数据	388
图 16-6	Mode2 发送数据	389
图 16-7	Mode2 接收数据	389
图 17-1	I2C 传输协议	402
图 17-2	START 和 STOP 条件	402

图 17-3	I2C 总线上位传输	403
图 17-4	I2C 总线上应答信号	404
图 17-5	I2C 总线上的仲裁	405
图 17-6	I2C 功能模块图	406
图 17-7	主发送模式数据同步图	409
图 17-8	I2C 主机发送状态图	409
图 17-9	主接收模式数据同步图	410
图 17-10	I2C 主机接收状态图	411
图 17-11	从接收模式数据同步图	412
图 17-12	从机接收状态图	412
图 17-13	从发送模式数据同步图	413
图 17-14	I2C 从机发送状态图	413
图 17-15	I2C 广播呼叫状态图	414
图 18-1	从机接收示意图	430
图 18-2	从机发送示意图	431
图 18-3	主机模式帧格式	431
图 18-4	从机 CPHA 为 0 时数据帧格式	432
图 18-5	从机 CHPA 为 1 时数据帧格式	432
图 18-6	SPI 多主机/多从机系统的示意图	433
图 21-1	ADC 示意框图	458
图 21-2	ADC 转换时序图	459
图 21-3	ADC 连续转换过程示例	462
图 21-4	ADC 连续转换累加过程示例	464
图 22-1	VC 框架图	486
图 22-2	VC 滤波响应时间	487
图 22-3	VC 迟滞功能	487
图 23-1	LVD 框图	500
图 23-2	LVD 滤波输出	501
图 23-3	LVD 迟滞响应	501

简介

HC32L110 系列是一款旨在延长便携式测量系统的电池使用寿命的超低功耗、Low Pin Count、宽电压工作范围的 MCU。集成 12 位 1Msps 高精度 SARADC 以及集成了比较器、多路 UART、SPI、I2C 等丰富的通讯外设，具有高整合度、高抗干扰、高可靠性和超低功耗的特点。本产品内核采用 Cortex-M0+ 内核，配合成熟的 Keil & IAR 调试开发软件，支持 C 语言及汇编语言，汇编指令。

超低功耗 MCU 典型应用

- 传感器应用，物联网应用
- 智能交通，智慧城市，智能家居
- 火警探头，智能门锁，无线监控等智能传感器应用
- 各种对于电池供电和对于功耗苛求的便携式设备等

关于本手册

本手册主要介绍芯片的功能、操作事项和使用方法。关于芯片的规格，请参阅对应的“数据手册”。

1 系统结构

1.1 概述

本产品系统由以下部分组成：

- 1 个 AHB 总线 Master:
 - Cortex-M0+
- 4 个 AHB 总线 Slaves:
 - FLASH 存储器
 - SRAM 存储器
 - AHB0, AHB to APB Bridge, 包含所有 APB 接口外设
 - AHB1, 包含所有 AHB 接口外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。如下图所示：

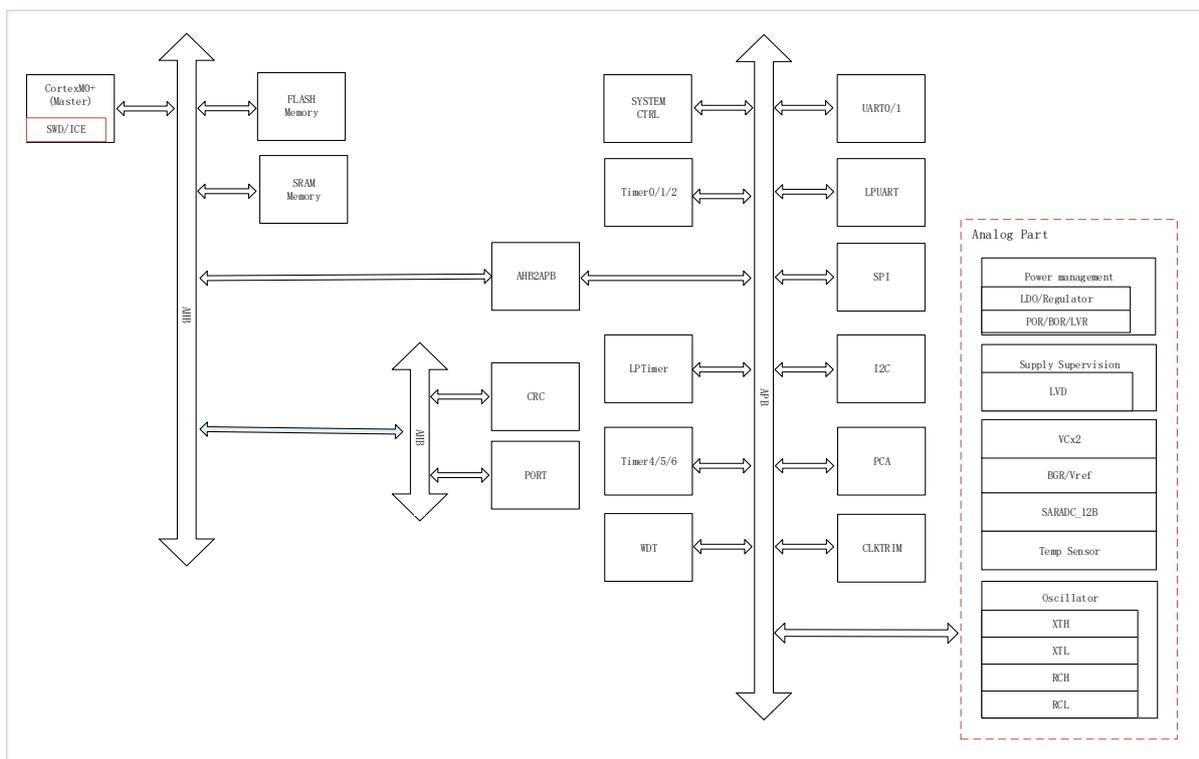
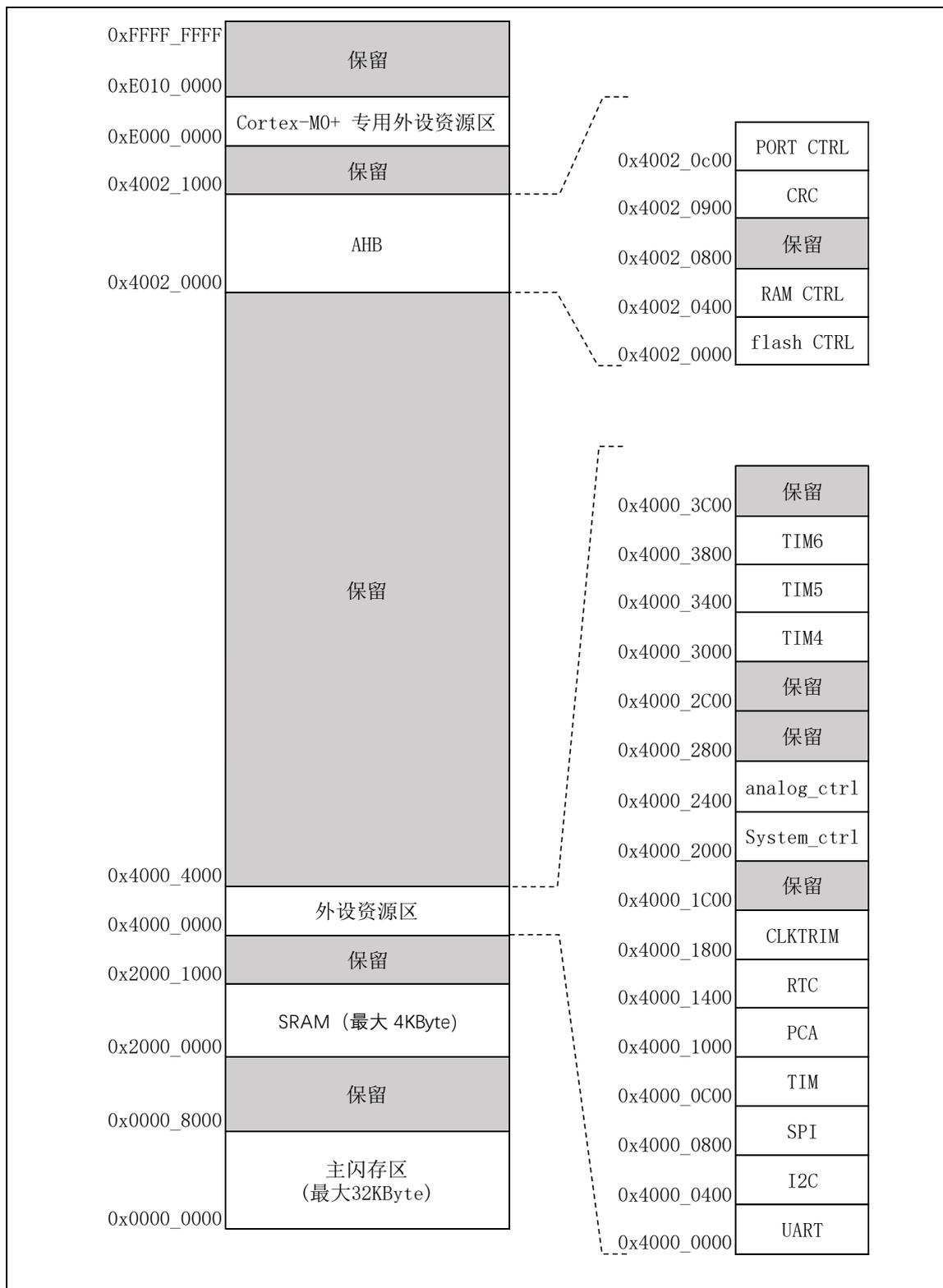


图 1-1 系统架构示意图

1.2 系统地址划分

整个 HC32L110 系统的地址区域划分，如下图所示：



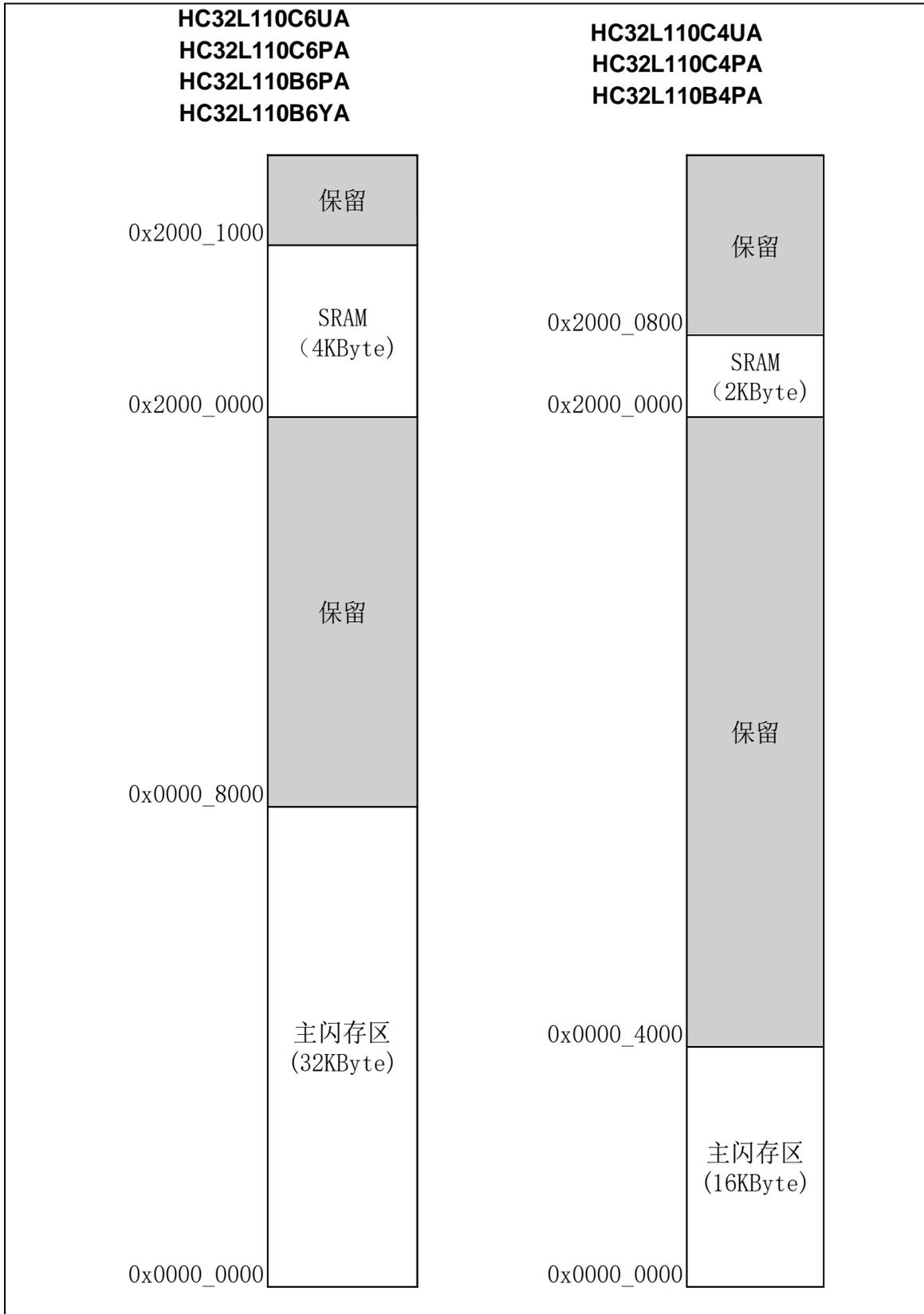


图 1-2 地址区域划分示意图

1.3 存储器 and 模块地址分配

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 工作模式

本产品的电源管理模块负责管理本产品各种工作模式之间的切换，以及控制各工作模式下的各功能模块的工作状态。本产品的工作电压（VCC）为 1.8 ~ 5.5V。

本产品有如下几个工作模式：

- （1）运行模式：CPU 运行，周边功能模块运行。
- （2）休眠模式：CPU 停止运行，周边功能模块运行。
- （3）深度休眠模式：CPU 停止运行，高速时钟停止运行。

从运行模式，通过执行软件程序，可进入其他低功耗模式。从其他各种低功耗模式，通过中断触发，可回到运行模式。

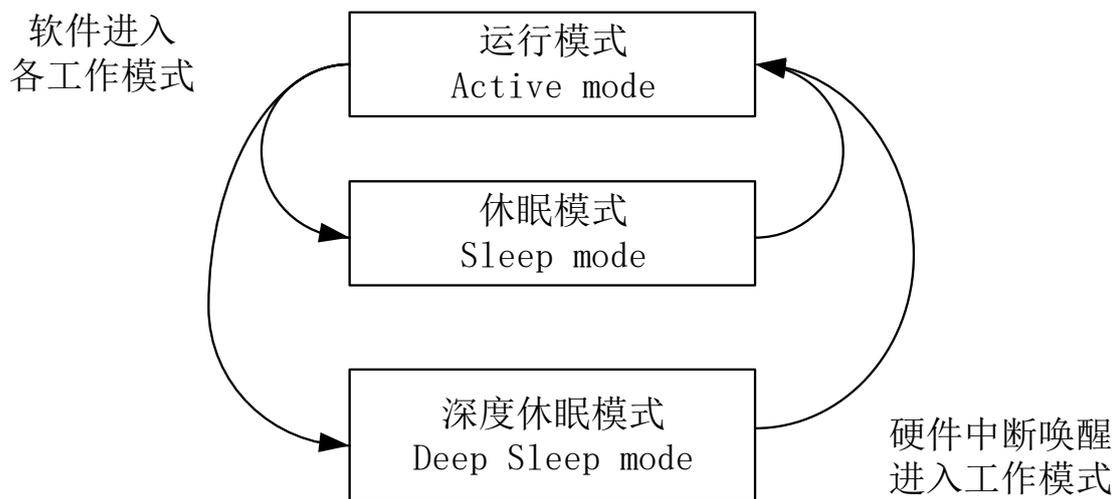


图 2-1 控制模式框图

在各模式下，CPU 可响应所有的中断类型。

	中断来源	运行模式	休眠模式	深度休眠模式
[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	✓	✓	✓

在各模式下，本产品可响应所有的复位类型。

	复位源	运行模式	休眠模式	深度休眠模式
[0]	上电掉电复位 POR	✓	✓	✓
[1]	外部 Reset Pin 复位	✓	✓	✓
[2]	LVD 复位	✓	✓	✓
[3]	WDT 复位	✓	✓	✓
[4]	PCA 复位	✓	✓	
[5]	Cortex-M0+ LOCKUP 硬件复位	✓		
[6]	Cortex-M0+ SYSRESETREQ 软件 复位	✓		

2.1 运行模式

本产品运行模式（Active Mode）：

在系统在电源上电复位后，或从各低功耗唤醒后，微控制器 MCU 处于运行状态。当 CPU 不需继续运行时，可以利用多种低功耗模式来节能，例如等待某个外部事件时。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

运行模式(Active 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

表 2-1 运行模式下可运行模块图

几种降低运行模式下芯片功耗的方法：

- 1) 在运行模式下，通过对预分频寄存器（SYSCLK0.AHB_CLK_DIV, SYSCLK0.APB_CLK_DIV）进行编程，可以降低任意一个系统时钟(HCLK, PCLK)的速度。进入休眠模式前，也可以利用预分频器来降低外设的时钟。
- 2) 在运行模式下，关闭不使用外设的时钟（PERI_CLKx）来减少功耗。
- 3) 在运行模式下，关闭不使用外设的时钟（PERI_CLKx）来减少功耗，并让系统进入休眠模式下更多地减少功耗，并在执行 WFI 指令前关闭不使用外设的时钟（PERI_CLKx）。
- 4) 使用低功耗模式代替休眠模式，因为本产品的唤醒时间极短（~4uS），亦可满足系统的实时响应的需求。

2.2 休眠模式

本产品休眠模式（Sleep Mode）

使用 WFI 指令可以进入休眠模式，休眠模式下，CPU 停止运行，但时钟模块，系统时钟，NVIC 中断处理以及周边的功能模块仍都可以工作。

系统进入休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。

如何进入休眠模式：

通过执行 WFI 指令进入休眠状态。根据 Cortex™-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择休眠模式进入机制：

SLEEP-NOW: 如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入休眠模式。

SLEEP-ON-EXIT: 如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

如何退出休眠模式：

如果执行 WFI 指令进入休眠模式，任意一个高优先级嵌套向量中断控制器响应的外设中断都能将系统从休眠模式唤醒。

使用注意：

- SLEEP-ON-EXIT 该位置 1，执行完中断自动进入 sleep，程序不需要写 __wfi()；
- SLEEP-ON-EXIT 该位清 0，main() 执行 __wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，执行 WFI 指令后进入 sleeping。等待后续中断触发。
- SLEEP-ON-EXIT 位不影响 __wfi() 指令的执行。SLEEP-ON-EXIT =0: main() 执行 wfi() 后进入 sleeping，中断触发且执行完中断程序返回 main() 后，继续往下执行；
- 若在中断中进入 sleep，只有优先级高于此中断的中断才能唤醒，先执行高优先级，再执行低优先级；优先级低于或等于此中断的中断不能唤醒。

休眠模式(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

灰色的模块在当前状态下不工作。

表 2-2 休眠模式下可运行模块图

2.3 深度休眠模式

本产品深度休眠模式（Deep Sleep Mode）

使用 SLEEPDEEP 配合 WFI 指令可以进入深度休眠模式，在深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟可配置是否运行，部分低功耗的周边模块可配置是否允许，NVIC 中断处理仍可以工作。

- 系统从高速时钟进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度休眠前的状态。
- 系统从低速时钟进入深度休眠模式，由于低速时钟不会自动关闭，保持运行，进入休眠模式。只有 ARM Cortex-M0+ 不运行，其他模块都运行。
- 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。
- 系统进入深度休眠状态，不会改变端口状态，在进入休眠前根据需要更改 IO 的状态为休眠下的状态。未使用的 IO 引脚及封装未引出的 IO 引脚均需要设置为输入并使能上拉。

如何进入深度休眠模式：

首先设置 Cortex-M0+ 系统控制寄存器中的 SLEEPDEEP 位，通过执行 WFI 指令进入休眠状态。根据 Cortex™-M0+ 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择深度休眠模式进入机制：

SLEEP-NOW: 如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入休眠模式。

SLEEP-ON-EXIT: 如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入休眠模式。

如何退出深度休眠模式：

如果执行 WFI 指令进入休眠模式，任意一个被嵌套向量中断控制器响应的外设中断（Deep Sleep 下可运行的 周边模块中断）都能将系统从休眠模式唤醒。

唤醒设置参考 3.4 中断唤醒控制 WIC。

深度休眠模式(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

灰色的模块在当前状态下不工作。

表 2-3 深度休眠模式下可运行模块图

系统控制寄存器(Cortex-M0+ 内核系统控制寄存器)

地址: 0xE000ED10

复位值: 0x0000 0000

位	标记	功能描述	读写
31:5	RESERVED	保留	
4	SEVONPEND	设置为1时, 每次新的中断挂起都会产生一个事件, 如果使用了WFE 休眠, 它可用于唤醒处理器	RW
3	RESERVED	保留	
2	SLEEPDEEP	设置为1时, 执行 WFI 进入深度休眠, 本产品进入 Deep sleep 模式 设置为0时, 执行 WFI 进入休眠, 本产品进入 sleep/Idle 模式	RW
1	SLEEPONEXIT	设置为1时, 当退出异常处理并返回程序线程时, 处理器自动进入休眠模式(WFI) 设置为0时, 该特性就会被自动禁止	RW
0	RESERVED	保留	

进入深度休眠后, 唤醒后系统时钟有两种选择, 默认使用进入深度休眠的时钟, 配置寄存器 SYSCTRL0.wakeup_byRCH 为 1 后不管进入深度休眠前是什么时钟, 唤醒后都使用内部高速时钟 RCH。如果使用外部晶体振荡这样设置可以加速唤醒系统。

3 系统控制器 (SYSCTRL)

3.1 时钟源介绍

时钟控制模块主要控制系统时钟以及外设时钟，可以配置不同的时钟源作为系统时钟、可以配置不同的系统时钟分频、可以启动或禁用外设时钟。另外为了确保振荡器精度，内部时钟均具有校准功能。

本产品支持以下四个不同的时钟源作为系统时钟：

- 内部高速 RC 时钟 RCH（输出频率为 4~24MHz）
- 内部低速 RC 时钟 RCL（38.4K 与 32.768K 可配置）
- 外部高速晶振时钟 XTH
- 外部低速晶振时钟 XTL

注意：

- 切换系统时钟的时钟源时，请严格按照操作步骤进行切换，详见 3.2 章节。
- XTL 可以不接晶振，直接从 P14 引脚输入 32.768KHz 的时钟信号。XTH 可以不接晶振，直接从 P01 引脚输入 4~32MHz 的时钟信号。

本产品还包含以下两个辅助时钟：

- 内部低速 10K 时钟；仅供看门狗和 CLKTRIM 模块使用。
- 内部 150K 时钟；仅供 LVD 和 VC 模块使用。

下图为本产品的时钟架构：

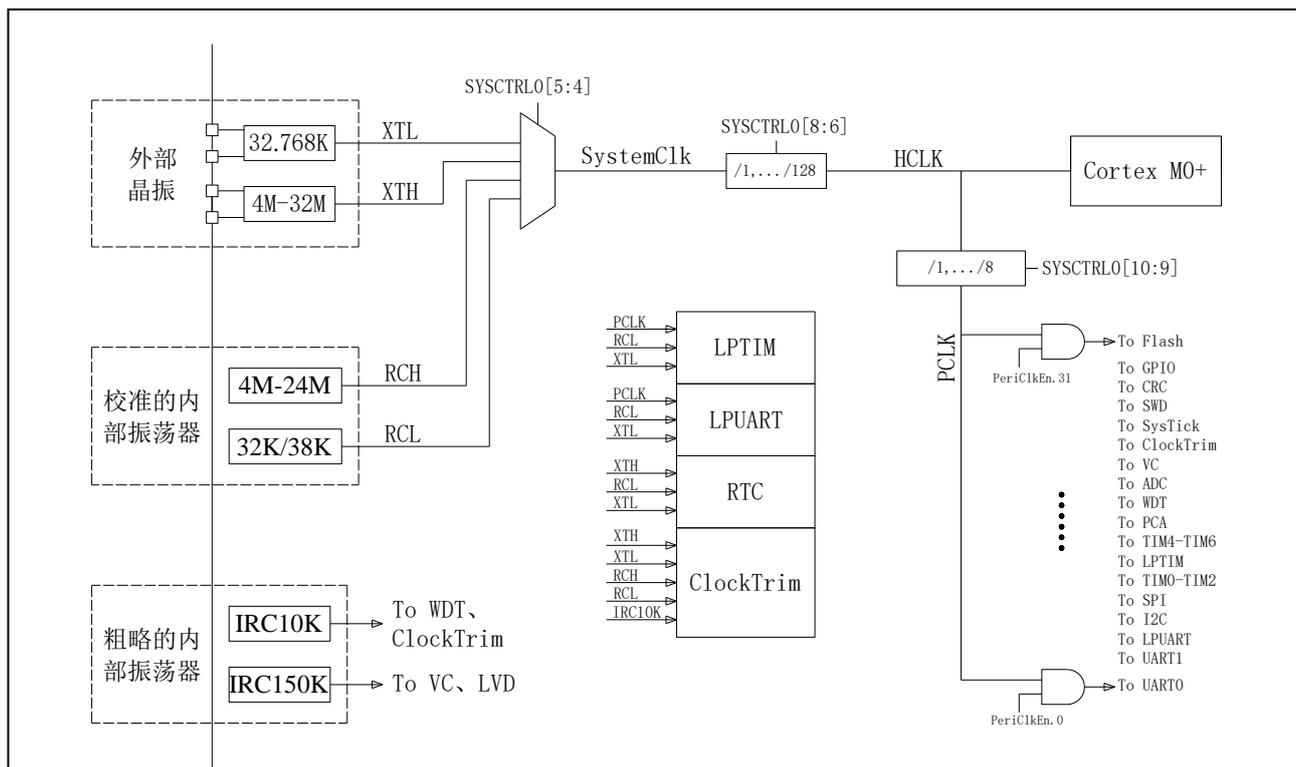


图 3-1 时钟控制模块框图

3.1.1 内部高速 RC 时钟 RCH

芯片上电或复位后的默认时钟源为频率为 4MHz 的内部高速时钟；当系统进入 Deep Sleep，此高速时钟会自动关闭。

更改寄存器 RCH_CR[10:0]的数值即可调整 RCH 的输出频率。寄存器数值每增加 1 则 RCH 的输出频率增加约 0.2%，总调整范围为 4~24MHz。

出厂时已预调好的 5 个频率 4MHz、8MHz、16MHz、22.12MHz、24MHz；如需其它频率请手动调整该寄存器的数值。

更改 RCH 输出频率需要按照特定的更改时序，详见系统时钟切换章节。

内部高速时钟从启动到稳定仅需 4us。为了在深度休眠模式下能快速响应中断，建议进入深度休眠模式前将系统时钟切换为 RCH。

3.1.2 内部低速 RC 时钟 RCL

内部低速时钟可以通过寄存器 RCL_CR[9:0]来选择其输出频率，可供选择的频率为 38.4KHz、32.768KHz。当系统进入 Deep Sleep，此低速时钟不会自动关闭，超低功耗外设模块可以选择 RCL 作为其时钟。

3.1.3 外部低速晶振时钟 XTL

外部低速晶振时钟需外接一个 32.768KHz 的低功耗晶振，具有超高精度以及超低功耗。当系统进入 Deep Sleep，此低速时钟不会自动关闭。超低功耗模式下工作的外设模块可以选择 XTL 作为其时钟。

XTL 也可以不接晶振，直接从 P14 引脚输入 32.768KHz 的时钟信号。从 P14 输入时钟信号的方法为：配置 P14 引脚为 GPIO 输入；设置 SYSCTRL1.EXTL_EN 为 1；设置 SYSCTRL0.XTL_EN 为 1。

注意：

- 晶体及其匹配器件需符合数据手册电气特性中低速外部时钟 XTL 的相关要求。

3.1.4 外部高速晶振时钟 XTH

外部高速晶振时钟需外接一个 4 MHz ~32MHz 的高速晶振。当系统进入 Deep Sleep，此高速时钟会自动关闭。

XTH 也可以不接晶振，直接从 P01 引脚输入 4MHz ~32MHz 的时钟信号。从 P01 输入时钟信号的方法为：配置 P01 引脚为 GPIO 输入；设置 SYSCTRL1.EXTH_EN 为 1；设置 SYSCTRL0.XTH_EN 为 1。

注意：

- 晶体及其匹配器件需符合数据手册电气特性中高速外部时钟 XTH 的相关要求。

3.1.5 时钟启动过程

上述四种时钟源都需要启动稳定时间，下图以外部 XTH 为例说明时钟的启动稳定过程。

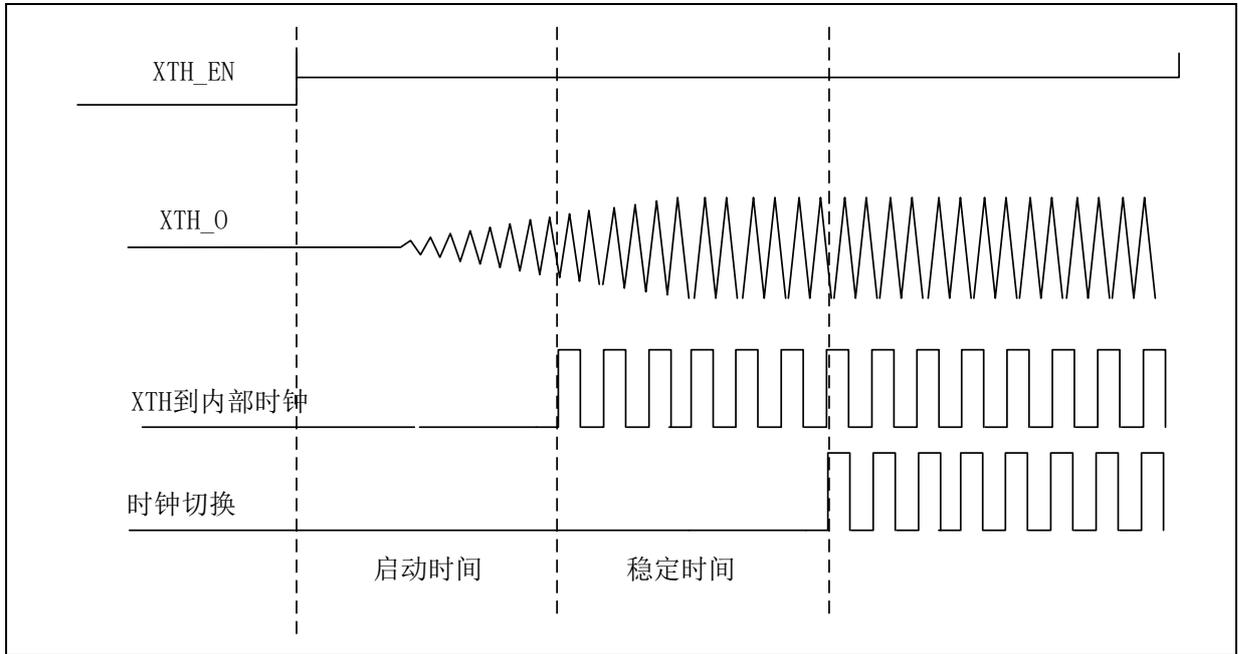


图 3-2 晶振时钟启动示意图

3.2 系统时钟切换

系统时钟的时钟源可通过 `SYSCTRL0[5:4]` 在 RCH、RCL、XTH、XTL 之间进行切换。时钟切换操作必须按照标准的时钟切换流程进行，否则可能出现异常。如果新的时钟源的频率大于 24MHz，则需要设置 `FLASH_CR.WAIT` 为 1。

注：改写 `FLASH_CR.WAIT` 的值，需要先向 `FLASH_BYPASS` 寄存器依次写入 `0x5A5A`、`0xA5A5` 然后再对 `FLASH_CR.WAIT` 进行赋值，详见 `FLASH` 控制器章节。

3.2.1 标准的时钟切换流程

操作流程如下：

Step1: 如新时钟源需要外部引脚，则将该引脚设置为适当的模式。

注：接外部晶振时需要模拟引脚；接外部时钟输入时需要 `GPIO` 输入并使能外部时钟输入。

Step2: 配置新时钟源的的振荡参数。

Step3: 使能新时钟源的振荡器。

Step4: 根据当前时钟源和新时钟源两者中较高的频率，按 `Flash` 控制器章节流程配置 `FLASH_CR.WAIT`。

Step5: 等待新时钟源输出稳定的频率。

Step6: 配置 `SYSCTRL0.Clk_sw4_sel`，选择系统时钟的来源为新时钟源。

Step7: 根据新时钟源的频率，按 `Flash` 控制器章节流程配置 `FLASH_CR.WAIT`。

Step8: 关闭不再使用的时钟源。

3.2.2 从 RCH 切换到 XTL 示例

操作流程如下：

Step1: 设置 `P1ADS.P1ADS4` 及 `P1ADS.P1ADS4` 为 1，配置 P14/P15 引脚为模拟端口。

Step2: 根据晶振特性，配置 `XTL_CR.Driver` 及 `XTL_CR.Startup`。

Step3: 向 `SYSCTRL2` 寄存器依次写入 `0x5A5A`、`0xA5A5`，使能寄存器改写。

Step4: 设置 `SYSCTRL0.XTL_EN` 为 1，使能晶振振荡电路。

Step5: 查询等待 `XTL_CR.Stable` 标志变为 1，晶振输出稳定时钟。

Step6: 向 `SYSCTRL2` 寄存器依次写入 `0x5A5A`、`0xA5A5`，使能寄存器改写。

Step7: 设置 SYSCTRL0. Clk_sw4_sel 为 3, 将系统时钟切换为 XTL。

Step8: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step9: 设置 SYSCTRL0. RCH_EN 为 0, 关闭 RCH 振荡器。

3.2.3 从 RCH 切换到 XTH 示例

操作流程如下:

Step1: 设置 P0ADS. P0ADS1 及 P0ADS. P0ADS1 为 1, 配置 P01/P02 引脚为模拟端口。

Step2: 根据晶振特性, 配置 XTH_CR.Driver。

Step3: 设置 XTH_CR.Startup 为 3, 选择最长的晶振稳定时间。

Step4: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step5: 设置 SYSCTRL0. XTH_EN 为 1, 使能晶振振荡电路。

Step6: 根据晶振频率, 配置 FLASH_CR.WAIT。

Step7: 查询等待 XTH_CR.Stable 标志变为 1 后, 软件延时 10ms 以上, 晶振输出稳定时钟。

Step8: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step9: 设置 SYSCTRL0. Clk_sw4_sel 为 1, 将系统时钟切换为 XTH。

Step10: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step11: 设置 SYSCTRL0. RCH_EN 为 0, 关闭 RCH 振荡器。

下图为时钟切换时序图：

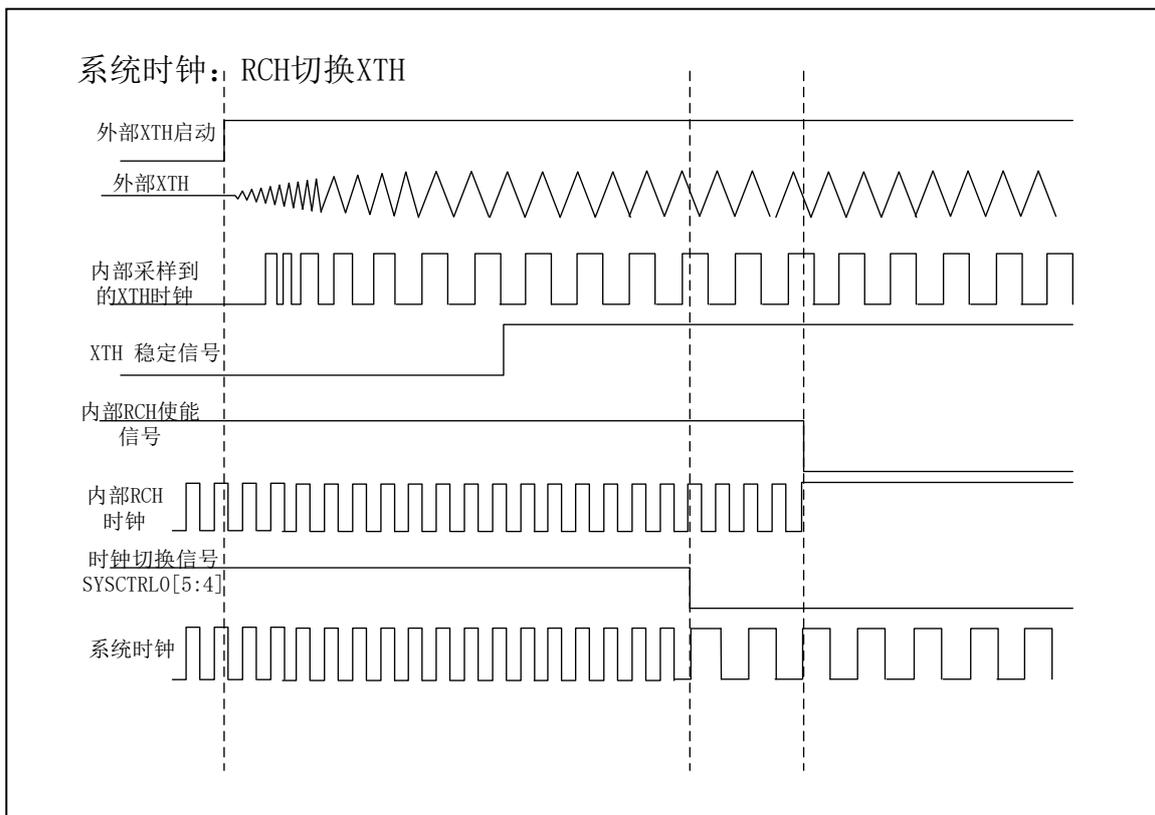


图 3-3 时钟切换示意图

3.2.4 从 RCL 切换到 XTH 示例

操作流程如下：

Step1: 设置 P0ADS.P0ADS1 及 P0ADS.P0ADS1 为 1，配置 P01/P02 引脚为模拟端口。

Step2: 根据晶振特性，配置 XTH_CR.Driver。

Step3: 设置 XTH_CR.Startup 为 3，选择最长的晶振稳定时间。

Step4: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step5: 设置 SYSCTRL0.XTH_EN 为 1，使能晶振振荡电路。

Step6: 根据晶振频率，配置 FLASH_CR.WAIT。

Step7: 查询等待 XTH_CR.Stable 标志变为 1 后，软件延时 10ms 以上，晶振输出稳定时钟。

Step8: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step9: 设置 SYSCTRL0.Clk_sw4_sel 为 1，将系统时钟切换为 XTH。

Step10: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step11: 设置 SYSCTRL0.RCL_EN 为 0, 关闭 RCL 振荡器。

3.2.5 从 RCH 切换到 RCL 示例

操作流程如下:

Step1: 配置 RCL_CR.TRIM 及 RCL_CR.Startup。

Step2: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step3: 设置 SYSCTRL0.RCL_EN 为 1, 使能 RCL 振荡电路。

Step4: 查询等待 RCL_CR.Stable 标志变为 1, RCL 输出稳定时钟。

Step5: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step6: 设置 SYSCTRL0.Clk_sw4_sel 为 2, 将系统时钟切换为 RCL。

Step7: 如需关闭 RCH 振荡器, 则执行后续操作: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写; 设置 SYSCTRL0.RCH_EN 为 0, 关闭 RCH 振荡器。

3.2.6 从 RCL 切换到 RCH 示例

操作流程如下:

Step1: 配置 RCH_CR.TRIM。

Step2: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step3: 设置 SYSCTRL0.RCH_EN 为 1, 使能 RCH 振荡电路。

Step4: 查询等待 RCH_CR.Stable 标志变为 1, RCH 输出稳定时钟。

Step5: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step6: 设置 SYSCTRL0.Clk_sw4_sel 为 0, 将系统时钟切换为 RCH。

Step7: 如需关闭 RCL 振荡器, 则执行后续操作: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写; 设置 SYSCTRL0.RCL_EN 为 0, 关闭 RCL 振荡器。

3.2.7 RCH 不同振荡频率间切换

RCH 不同振荡频率间切换具有两种方案。

方案 1:

Step1: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step2: 设置 SYSCTRL0.HCLK_PRIS 为 0x7。

Step3: 向上或向下逐级调整 RCH 的输出频率, 4M -> 8M -> 16M -> 24M/22.12M 或 24M/22.12M -> 16M -> 8M -> 4M。

Step4: 向 SYSCTRL2 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step5: 设置 SYSCTRL0.HCLK_PRIS 为 0x0。

从 4M 切换到 24M 的示例代码如下所示:

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

方案 2:

Step1: 将系统时钟切换为 RCL, 参见 3.2.5 从 RCH 切换到 RCL 示例。

Step2: 更改 RCH_CR.TRIM 值, 更改 RCH 振荡的频率。

Step3: 将系统时钟切换为 RCH, 参见 3.2.6 从 RCL 内部低速切换到 RCH 示例。

3.3 时钟校准模块

本产品内嵌时钟校准电路，如下图所示，系统时钟的四个源均可以相互校准，当选择好参考时钟以及被校准时钟后，设置寄存器 REFCNT 值，置位 cali.start 启动时钟校准电路，此时两个 32 位计数器（递增、递减）同时工作，当递减计数器等于 0 时，cali.finish 被置位，表明校准结束，此时软件可以读取 CALCNT 值，这样就很容易得到参考时钟与被校准时钟之间的频率关系。

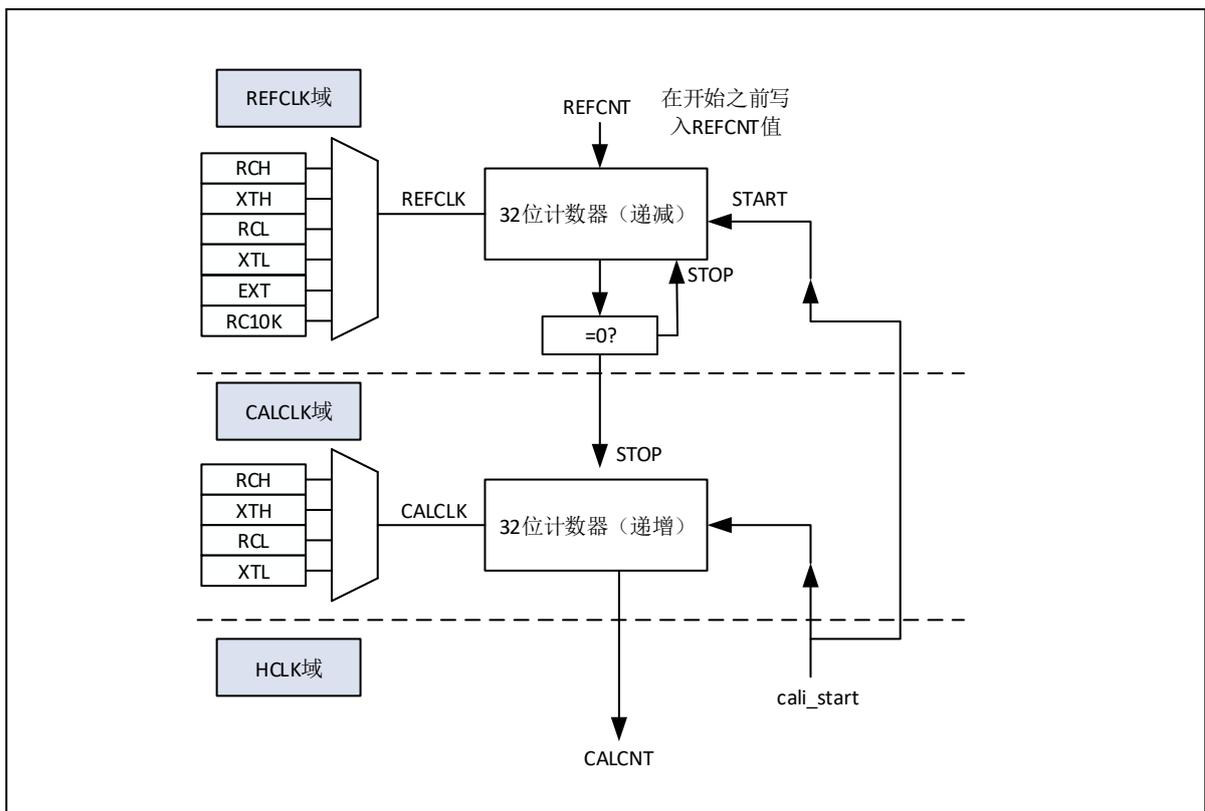


图 3-4 时钟校准原理图

3.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠状态时，会停止执行指令。在休眠状态下发生了中断请求（更高优先级）且需要处理时，处理器就会被唤醒。

休眠状态下的处理器收到中断请求时的行为如下表所示：

PRIMASK 状态	WFI 行为	唤醒	ISR 执行
0	IRQ 优先级 > 当前等级	Y	Y
0	IRQ 优先级 \leq 当前等级	N	N
1	IRQ 优先级 > 当前等级	Y	N
1	IRQ 优先级 \leq 当前等级	N	N

3.4.1 从深度休眠模式唤醒后执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 执行 WFI 指令以进入深度休眠模式
5. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务程序

例程：

```

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

3.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 PRIMASK 为 1
4. 设置 SCB->SCR.SLEEPDEEP 为 1
5. 执行 WFI 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令
7. 清除中断标志，清除中断挂起状态

例程：

```
__asm("CPSID I"); //Set PRIMASK
SCB_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 使用退出休眠特性

退出休眠（sleep-on-exit）非常适合中断驱动的应用程序。当该特性使能时，只要完成异常处理并且返回到了线程模式，处理器就会进入休眠模式。利用退出休眠特性，处理器可以尽可能多的处于休眠模式。

Cortex-M0 利用退出休眠特性进入休眠，这种情况同执行完异常退出后立即执行 WFI 的效果差不多。不过，为了下次进入异常时，不用再进行压栈操作，处理器不会执行出栈的过程。

1. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 设置 SCB->SCR.SLEEPONEXIT 为 1
5. 执行 WFI 指令以进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序
7. 退出中断服务时自动进入休眠模式

例程：

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

3.5 寄存器

基地址 0x40002000

寄存器	偏移地址	描述
SYSCTRL0	0x000	系统控制寄存器0
SYSCTRL1	0x004	系统控制寄存器1
SYSCTRL2	0x008	系统控制寄存器2
RCH_CR	0x00C	RCH 控制寄存器
XTH_CR	0x010	XTH 控制寄存器
RCL_CR	0x014	RCL 控制寄存器
XTL_CR	0x018	XTL 控制寄存器
PERI_CLKEN	0x020	外围模块时钟控制寄存器
SYSTICK_CR	0x034	Systick 时钟控制

表 3-1 系统控制寄存器表格

3.5.1 系统控制寄存器 0 (SYSCTRL0)

偏移地址: 0x000

复位值: 0x0000 0001

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
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		

位	标记	功能描述
31:16	Reserved	保留
15	wakeup_ byRCH	1: 从 Deep Sleep 唤醒后, system clock来源为RCH, 原时钟继续使能。 0: 从 Deep Sleep 唤醒后, 不改变system clock来源。
14:11	Reserved	保留
10:9	PCLK_PRS	PCLK分频选择 00: HCLK 01: HCLK/2 10: HCLK/4 11: HCLK/8
8:6	HCLK_PRS	HCLK分频选择 000: SystemClk 001: SystemClk/2 010: SystemClk/4 011: SystemClk/8 100: SystemClk/16 101: SystemClk/32 110: SystemClk/64 111: SystemClk/128
5:4	Clk_sw4_sel	系统时钟来源选择 00: 内部高速时钟 RCH 01: 外部高速晶振 XTH 10: 内部低速时钟 RCL 11: 外部低速晶振 XTL
3	XTL_EN	外部低速晶振 XTL 使能控制 0: 关闭 1: 使能 注: 需要将P14、P15设置成模拟端口。

2	RCL_EN	内部低速时钟 RCL 使能控制 0: 关闭 1: 使能
1	XTH_EN	外部高速晶振 XTH 使能控制 0: 关闭 1: 使能 注: 当系统进入DeepSleep, 此高速时钟会自动关闭。
0	RCH_EN	内部高速时钟 RCH 使能信号。 0: 关闭 1: 使能 注: 当系统进入DeepSleep, 此高速时钟会自动关闭。

注意:

- 每次改写 SYSCTRL0, SYSCTRL1 的值, 均需要先对 SYSCTRL2 依次写入 0x5A5、0xA5A5。这样的步骤可有效防止对 SYSCTRL0, SYSCTRL1 寄存器的误操作。

3.5.2 系统控制寄存器 1 (SYSCTRL1)

偏移地址: 0x004

复位值: 0x00000008

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				RTC_FREQ_ADJUST		SWD_ UIO	RES_ UIO	LOCK _EN	RTC_ LPW	Res	XTL_alwa yson	EXTL _EN	EXTH _EN	Re s	
				R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W		

位	标记	功能描述
31:12	Reserved	
11:9	RTC_FREQ_ADJUST	RTC 高速时钟补偿时钟频率选择 000 4M; 001 6M; 010 8M; 011 12M 100 16M; 101 20M; 110 24M; 111 32M;
8	SWD_USE_IO	SWD 端口功能配置 0: SWD 端口 1: GPIO 端口
7	RESET_USE_IO	RESET 端口功能配置 0: RESET 端口 1: GPIO 端口
6	LOCKUP_EN	Cortex-M0+ LockUp 功能配置 0: 关闭 1: 使能 注: 使能后, CPU读到无效指令时会复位MCU, 可增强系统可靠性。
5	RTC_LPW	RTC 模块低功耗控制 1: 低功耗模式使能 0: 低功耗模式禁能 注: 使能后, RTC 模块进低功耗状态, 其寄存器不可以读写。
4	Reserved	
3	XTL_ALWAYS_ON	XTL 高级使能控制 1: SYSCTRL0.XTL_EN 只可置位。 0: SYSCTRL0.XTL_EN 可置位可清零。
2	EXTL_EN	外部 XTL 时钟输入控制 1: XTL 输出时钟从 P14 输入。 0: XTL 输出时钟由晶振产生。 注: 使用 P14 输入时钟时, 需设置 SYSCTRL0.XTL_EN 为 1。

1	EXTH_EN	<p>外部 XTH 输入控制</p> <p>1: XTH 输出时钟从 P01输入。</p> <p>0: XTH 输出时钟由晶振产生。</p> <p>注: 使用P01 输入时钟时, 需设置SYSCTRL0.XTH_EN为1。</p>
0	Reserved	

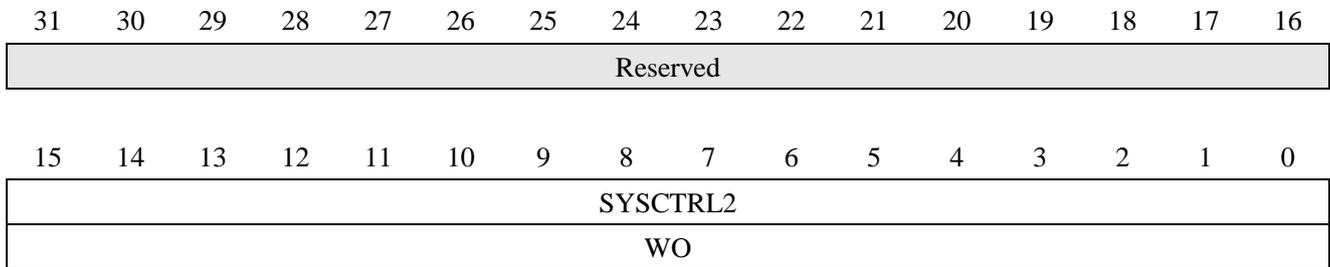
注意:

- 每次改写 SYSCTRL0, SYSCTRL1 的值, 均需要先对 SYSCTRL2 依次写入 0x5A5、0xA5A5。这样的步骤可有效防止对 SYSCTRL0, SYSCTRL1 寄存器的误操作。

3.5.3 系统控制寄存器 2 (SYSCTRL2)

偏移地址: 0x008

复位值: 0x00000000



位	标记	功能描述
31:16	Reserved	
15:0	SYSCTRL2	寄存器 SYSCTL0, SYSCTRL1 保护系列控制寄存器, 对 SYSCTRL2 先写 0x5A5A, 再写 0xA5A5, 启动对于寄存器 SYSCTL0, SYSCTRL1 的写操作, 只要对寄存器 SYSCTL0, SYSCTRL1 写操作了, 这个保护位自动回复保护状态, 需要重新写入系列打开保护。

3.5.4 RCH 控制寄存器 (RCH_CR)

偏移地址: 0x00C

复位值: 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										
				RO	R/W										

位	标记	功能描述
31:12	Reserved	
11	stable	内部高速时钟 RCH 稳定标志位。 1: 代表 RCH 已经稳定, 可以被内部电路使用。 0: 代表 RCH 未稳定, 不可以被内部电路使用。
10:0	TRIM	时钟频率调整, 更改该寄存器的数值即可调整RCH的输出频率。寄存器数值每增加1则RCH的输出频率增加约0.2%, 总调整范围为4~24MHz。 Flash中已保存了5组频率的校准值, 将Flash内的校准值读出并写入 RCH_CR.TRIM即可获得精准的频率。 24M校准值地址: 0x00100C00 - 0x00100C01 22.12M校准值地址: 0x00100C02 - 0x00100C03 16M校准值地址: 0x00100C04 - 0x00100C05 8M校准值地址: 0x00100C06 - 0x00100C07 4M校准值地址: 0x00100C08 - 0x00100C09 RCH输出频率更改需要按照特定的时序, 详见系统时钟切换3.2.7章节。

3.5.5 振荡 XTH 控制寄存器 (XTH_CR)

偏移地址: 0x010

复位值: 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				

位	标记	功能描述
31:6	Reserved	
6	stable	外部高速时钟 XTH 稳定标志位。 1: 代表 XTH 已经稳定, 可以被内部电路使用。 0: 代表 XTH 未稳定, 不可以被内部电路使用。 注意: 为了增加系统可靠性, 在查询到该标志后, 需要软件延时10ms以上, 才可将系统时钟切换为XTH。
5:4	Startup	外部高速时钟 XTH 稳定时间选择 00: 256个周期; 01: 1024个周期; 10: 4096个周期; 11: 16384个周期; 注: 强烈建议将 XTH 的稳定时间设置为11。如果 XTH 稳定时间不足, 在进行时钟切换时或从深度休眠唤醒时, 系统不能稳定工作。
3:0	Driver	3:2 Freq 选择晶振的工作频率 11: 24M~32M 10: 16M~24M 01: 8M~16M 00: 4M~8M 1:0 Driver 选择晶振的驱动能力 11: 最强驱动能力 10: 默认驱动能力(推荐值) 01: 弱驱动能力 00: 最弱驱动能力 注: 需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大; 驱动能力越弱则功耗越小。

3.5.6 RCL 控制寄存器 (RCL_CR)

偏移地址: 0x014

复位值: 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										

位	标记	功能描述
31:13	Reserved	
12	stable	内部低速时钟 RCL 稳定标志位。 1: 代表 RCL 已经稳定, 可以被内部电路使用。 0: 代表 RCL 未稳定, 不可以被内部电路使用。
11:10	Startup	内部低速时钟 RCL 稳定时间选择 11: 256个周期; 10: 64个周期; 01: 16个周期; 00: 4个周期;
9:0	TRIM	内部低速时钟频率调整, Flash中保存了2组频率的校准值。 将Flash内的校准值读出并写入RCL_CR.TRIM即可获得精准的频率。 38.4K校准值地址: 0x00100C20 - 0x00100C21 32.768K校准值地址: 0x00100C22 - 0x00100C23

3.5.7 XTL 控制寄存器 (XTL_CR)

偏移地址: 0x018

复位值: 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			

位	标记	功能描述
31:6	Reserved	
6	stable	外部低速晶振 XTL 稳定标志位。 1: 代表 XTL 已经稳定, 可以被内部电路使用。 0: 代表 XTL 未稳定, 不可以被内部电路使用。
5:4	Startup	外部低速晶振 XTL 稳定时间选择 00: 256 个周期; 01: 1024 个周期; 10: 4096 个周期; 11: 16384 个周期;
3:0	Driver	外部低速晶振 XTL 驱动选择 1111: 最大驱动 0000: 最小驱动 3:2 Amp_control XTL 振荡幅度的细微调整。 11: 最大振幅 10: 较大振幅 (推荐值) 01: 正常振幅 00: 最小振幅 1:0 Driver 外部晶振驱动能力选择 11: 最强驱动能力 10: 较强驱动能力 01: 默认驱动能力 (推荐值) 00: 最弱驱动能力, 注: 需要根据晶振特性、负载电容以及电路板的寄生参数选择适当的驱动能力。驱动能力越大则功耗越大; 驱动能力越弱则功耗越小。00: 最弱驱动能力

3.5.8 外围模块时钟控制寄存器（PERI_CLKEN）

复位值: 0xC080_0000

偏移地址: 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 TIM	LP TIM	BASE TIM	Res.	SPI	Res.	I2C	Res.	LPUART	UART1	UART0
R/W	R/W				R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W

位	标记	功能描述
31	flash	Flash 控制器模块时钟使能，操作 flash 寄存器时需要使能该时钟 1: 使能; 0: 关闭 注: 从 flash 中执行程序此位不受影响。
30:29	Res.	保留位
28	GPIO	GPIO 模块时钟使能。 1: 使能; 0: 关闭
27	Res.	保留位
26	CRC	CRC 模块时钟使能。 1: 使能; 0: 关闭
25	Res.	保留位
24	TICK	SysTick定时器参考时钟使能。 1: 使能; 0: 关闭
23:22	Res.	保留位
21	Trim	CLKTRIM模块时钟使能。 1: 使能; 0: 关闭
20	RTC	RTC 模块时钟使能。 1: 使能; 0: 关闭
19:18	Res.	保留位
17	VC	VC, LVD, 模块时钟使能。 1: 使能; 0: 关闭
16	ADC	ADC 模块时钟使能。 1: 使能; 0: 关闭
15	WDT	WDT 模块时钟使能。 1: 使能; 0: 关闭

14	PCA	PCA 模块时钟使能。 1: 使能; 0: 关闭
13:11	Res.	保留位
10	ADVTIME	Timer456 模块时钟使能。 1: 使能; 0: 关闭
9	LPTIM	LPTimer 模块时钟使能。 1: 使能; 0: 关闭
8	BASETIM	Timer012 模块时钟使能。 1: 使能; 0: 关闭
7	Res.	保留位
6	SPI	SPI 模块时钟使能。 1: 使能; 0: 关闭
5	Res.	保留位
4	I2C	I2C 模块时钟使能。 1: 使能; 0: 关闭
3	Res.	保留位
2	LPUART	LPUART 模块时钟使能。 1: 使能; 0: 关闭
1	UART1	UART1 模块时钟使能。 1: 使能; 0: 关闭
0	UART0	UART0 模块时钟使能。 1: 使能; 0: 关闭

3.5.9 SysTick 时钟控制 (SYSTICK_CR)

复位值: 0x0100_0147

偏移地址: 0x034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CLK_SEL	NO REF	SKEW	STCALIB[23: 16]								
				R/W	R/W	R/W	R/W								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STCALIB[15: 0]															
R/W															

位	标记	功能描述
31-28	Reserved	
27-26	CLK_SEL	SysTick 外部参考时钟选择 00: 外部低速时钟XTL 01: 内部低速时钟RCL 10: 系统时钟8分频SystemClk/8 11: 外部高速时钟XTH
25	NOREF	SysTick 时钟源选择 1: 内部高频时钟HCLK 0: 外部参考时钟, 由SYSTICK_CR[27:26]选择 注: 使用外部参考时钟时, 参考时钟频率不允许高于HCLK
24	SKEW	STCALIB 精度指示 1: STCALIB 值代表粗略的 10ms 0: STCALIB 值代表精确的 10ms
23:0	STCALIB	参考时钟为XTL时, 10毫秒校准值

4 复位控制器（RESET）

4.1 复位控制器介绍

本产品具有 7 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序计数器 PC 会被复位指向 00000000。

- POR/BOR 复位（VCC 域及 Vcore 域）
- 外部 Reset PAD 复位
- WDT 复位
- PCA 复位
- LVD 复位
- Cortex-M0+ SYSRESETREQ 软件复位
- Cortex-M0+ LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示。复位标志均由硬件置位，需要用户软件清零。

芯片复位时，如果查询到 Reset_flag.POR15V 或 Reset_flag.POR5V 为 1 则为上电复位。

上电复位时用户程序应当将寄存器 Reset_flag 清零，则下一次复位时可通过 Reset_flag 的相关比特判断复位来源。

下图描述各区域的复位来源。

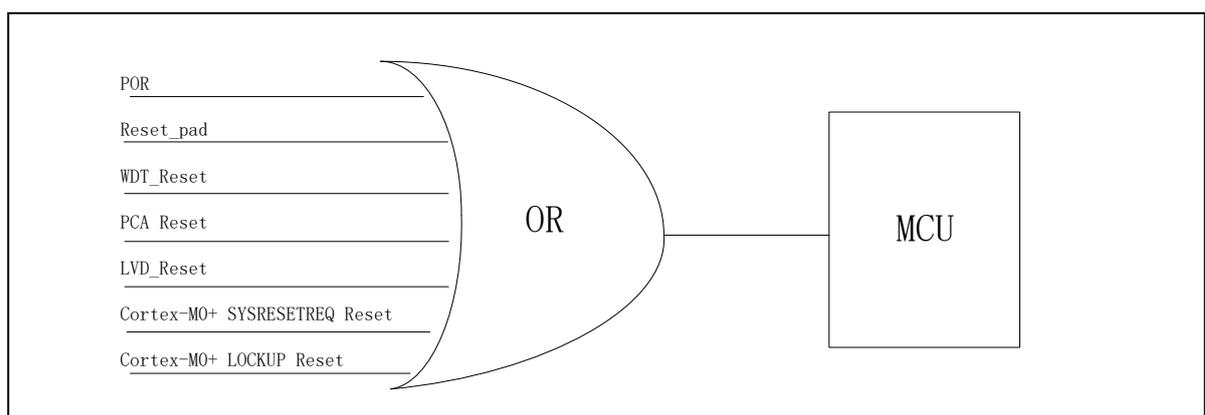


图 4-1 复位来源示意图

4.1.1 上电下电复位 POR/BOR

本产品有两个供电区域：**VCC 区域**、**Vcore 区域**。所有的模拟模块及 IO 工作于 **VCC 区域**；其它模块工作于 **Vcore 区域**。

VCC 区域上电时，当 **VCC** 电压低于 **POR** 阈值电压时（典型值为 1.65V），会产生 **POR5V** 信号；**VCC 区域**下电时，当 **VCC** 电压低于 **BOR** 阈值电压时（典型值为 1.5V），会产生 **POR5V** 信号。

Vcore 区域上电时，当 **Vcore** 电压低于 **POR** 阈值电压时，会产生 **POR15V** 信号；**Vcore 区域**下电时，当 **Vcore** 电压低于 **BOR** 阈值电压时，会产生 **POR15V** 信号。

POR5V 信号和 **POR15V** 信号均会将芯片的寄存器复位到初始化状态。

4.1.2 外部复位引脚复位

当外部复位引脚检测到低电平时会产生一个系统复位。该复位引脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于 20us（典型值）的毛刺信号，因此，加到复位引脚上的低电平信号必须大于 20us，才能确保芯片可靠复位。

4.1.3 WDT 复位

看门狗复位，请参看 **WDT** 一章说明。

4.1.4 PCA 复位

PCA 复位，请参看 **PCA** 一章说明。

4.1.5 LVD 低电压复位

LVD 复位，请参考 **LVD** 一章说明。

4.1.6 Cortex-M0+ SYSRESETREQ 复位

Cortex-M0+ 软件复位

4.1.7 Cortex-M0+ LOCKUP 复位

当 **Cortex-M0+** 遇到严重的异常时，它会将自己的 **PC** 指针停在当前地址处，并锁死自

已，并在几个时钟周期延时之后复位整个 CORE 区域。

4.2.2 外围模块复位控制寄存器（PERI_RESET）

复位值: 0xD7B3C757

地址: 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
		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				TIM	TIM	TIM		R/W		R/W		RT	R/W	R/W
					R/W	R/W	R/W						R/W	R/W	R/W

位	标记	功能描述
31:29	Res.	保留位
28	GPIO	GPIO 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
27	Res.	保留位
26	CRC	CRC 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
25	Res.	保留位
24	TICK	SYSTICK 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
23:22	Res.	保留位
21	TRIM	CLKTRIM 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
20	RTC	RTC 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
19-18	Res.	保留位
17	VC	VC, LVD, 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
16	ADC	ADC 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
15	Res.	保留位
14	PCA	PCA 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
13:11	Res.	保留位
10	ADVTIM	Timer456 模块复位使能。 1: 正常工作; 0: 模块处于复位状态

9	LPTIM	LPTimer 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
8	BASETIM	Timer012 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
7	Res.	保留位
6	SPI	SPI 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
5	Res.	保留位
4	I2C	I2C 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
3	Res.	保留位
2	LPUART	LPUART 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
1	UART1	UART1 模块复位使能。 1: 正常工作; 0: 模块处于复位状态
0	UART0	UART0 模块复位使能。 1: 正常工作; 0: 模块处于复位状态

5 中断控制器 (NVIC)

5.1 概述

Cortex-M0+ 处理器内置了嵌套向量中断控制器(NVIC),支持最多 32 个中断请求(IRQ)输入, 以及 1 个不可屏蔽中断 (NMI) 输入 (在本产品系统中并未使用)。另外, 处理器还支持多个内部异常。

每个异常源都有一个单独的异常编号, 每种异常类型都有对应的优先级, 有些异常的优先级是固定的, 而有些则是可编程的。具体如下表所示:

异常编号	异常类型	优先级	描述
1	复位	-3 (最高)	复位
2	NMI	-2	不可屏蔽中断 (在本系统中没有使用)
3	硬件错误	-1	错误处理异常
4-10	保留	NA	...
11	SVC	可编程	通过SVC指令调用管理程序
12-13	保留	NA	...
14	PendSV	可编程	系统服务的可挂起请求
15	SysTick	可编程	SysTick定时器
16	中断#0	可编程	外部中断#0
17	中断#1	可编程	外部中断#1
...
47	中断#31	可编程	外部中断#31

表 5-1 Cortex-M0+ 处理器异常一览

本章节只对处理器的 32 个外部中断请求 (中断#0 到中断#31) 做详细介绍, 处理器内部异常的具体情况可参考其他相关文档。同时, 本章节只讨论处理器内核中 NVIC 的中断处理机制, 外设模块自身的中断产生机制不在这里展开讨论。

5.2 中断优先级

每一个外部中断都对应一个优先级寄存器, 每个优先级都是 2 位宽, 并且使用中断优先级寄存器的最高两位, 每个寄存器占 1 个字节 (8 位)。在这个设定下, 可以使用的优先级为 0x00 (最高)、0x40、0x80 和 0xc0 (最低)。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
已使用		未使用, 读数为 0					

图 5-1 只使用了高两位的优先级寄存器

如果处理器已经在运行另外一个中断处理，而新中断的优先级大于正在执行的，这时就会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。

如果处理器正在运行的另外一个中断处理的优先级相同或者更高，新的中断将会等待并且进入挂起状态。挂起的中断将会一直等到当前中断等级改变，例如，当前运行的中断处理完成返回后，当前优先级降低到了比挂起中断还要小。

如果两个中断同时发生，并且它们的优先级相同，中断编号较小的中断将会首先执行。例如，如果中断#0 和中断#1 使能且具有相同的优先级，在它们同时被触发时，中断#0 会首先执行。

5.3 中断向量表

当 Cortex-M0+处理器要处理中断服务请求时，它需要首先确定异常处理的起始地址，所需的信息叫做向量表，如图 5-2 所示。向量表存储在存储器空间的开始位置，包含了系统中可用异常（中断）的异常（中断）向量，以及主栈指针（MSP）的初始值。

存储器地址		异常编号
0x0000004C	中断#3向量	19
0x00000048	中断#2向量	18
0x00000044	中断#1向量	17
0x00000040	中断#0向量	16
0x0000003C	SysTick向量	15
0x00000038	PendSV向量	14
0x00000034	未使用	13
0x00000030	未使用	12
0x0000002C	SVC向量	11
0x00000028	未使用	10
0x00000024	未使用	9
0x00000020	未使用	8
0x0000001C	未使用	7
0x00000018	未使用	6
0x00000014	未使用	5
0x00000010	未使用	4
0x0000000C	硬件错误异常	3
0x00000008	NMI向量	2
0x00000004	复位向量	1
0x00000000	MSP初始值	0

图 5-2 中断向量表

其中，中断向量的存储顺序同中断编号一致，由于每个向量都是 1 个字（4 字节），中断向量的地址为中断编号乘 4，每个中断向量都是中断处理的起始地址。

5.4 中断输入和挂起行为

Cortex-M0+ 处理器的 NVIC 模块中，每一个中断输入都对应着一个挂起状态寄存器，且每个寄存器只有 1 位，用于保存中断请求，而不管这个请求有没有得到确认。当处理器开始处理这个中断时，硬件将会自动清除挂起状态位。

本系统的外设使用电平触发中断输出，当中断事件发生时，由于外设连接到了 NVIC 上，中断信号会得到确认。在处理器执行中断服务并且清除外设的中断信号以前，该信号会保持高电平。在 NVIC 内部，当检测到有中断发生时，该中断的挂起状态会被置位，当处理器接收该中断并且开始执行中断服务程序后，挂起状态就会被清除。该过程如图 5-3 所示：

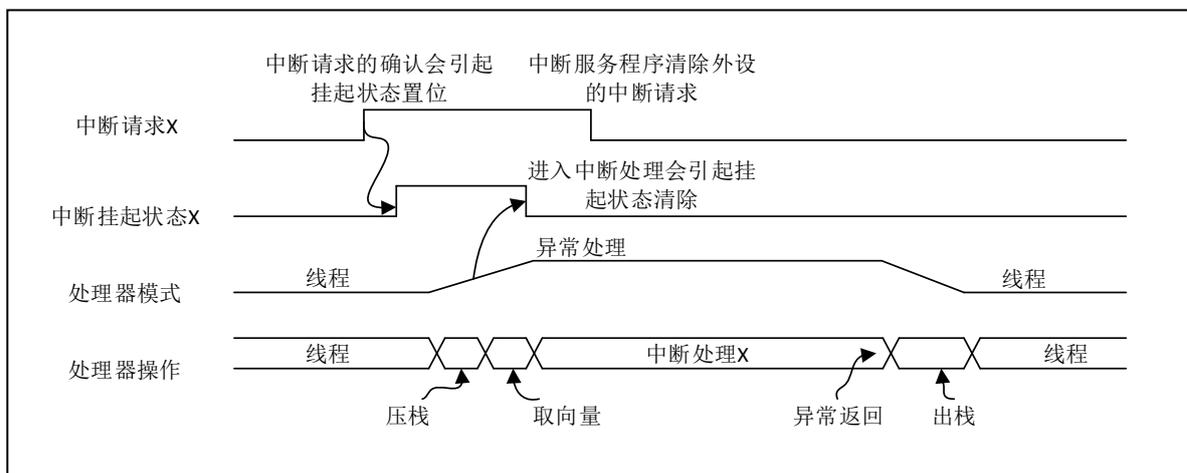


图 5-3 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除了，这样处理器会忽略掉本次请求，并且不会执行中断处理。可以通过写 NVIC_CLRPEND 寄存器来清除中断挂起状态，这种处理在设置外设时非常有用，因为在设置以前，该外设可能已经产生了一个中断请求。

如果在软件清除挂起状态时，外设仍然保持着中断请求，挂起状态还会立即生成。该过程如图 5-4 所示：

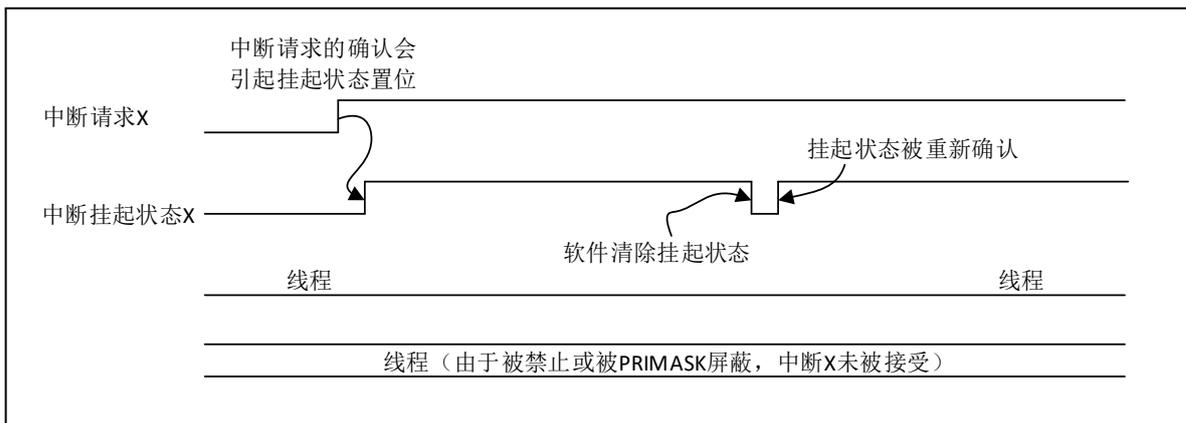


图 5-4 中断挂起状态被清除然后被重新确认

如果外设产生的中断请求在异常处理时没有被清除，异常返回后挂起状态就会被又一次激活，这样中断服务程序会再次执行。该过程如图 5-5 所示：

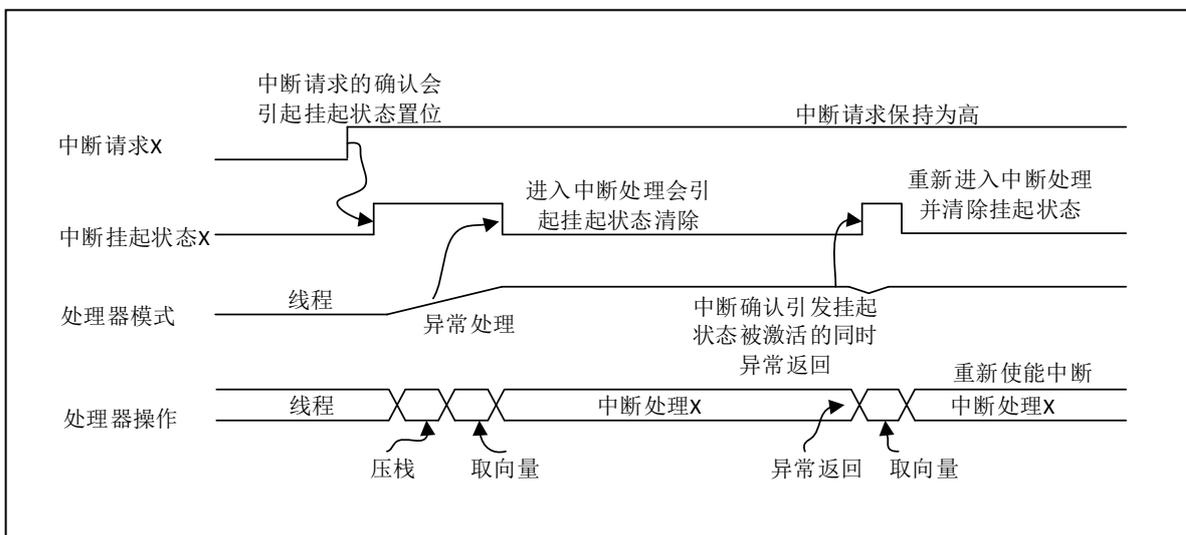


图 5-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

如果在终端服务程序执行的过程中产生外设中断请求，该请求会被当作新的中断请求，并且在本次中断退出后，还会引起中断服务程序的再次执行。该过程如图 5-6 所示：

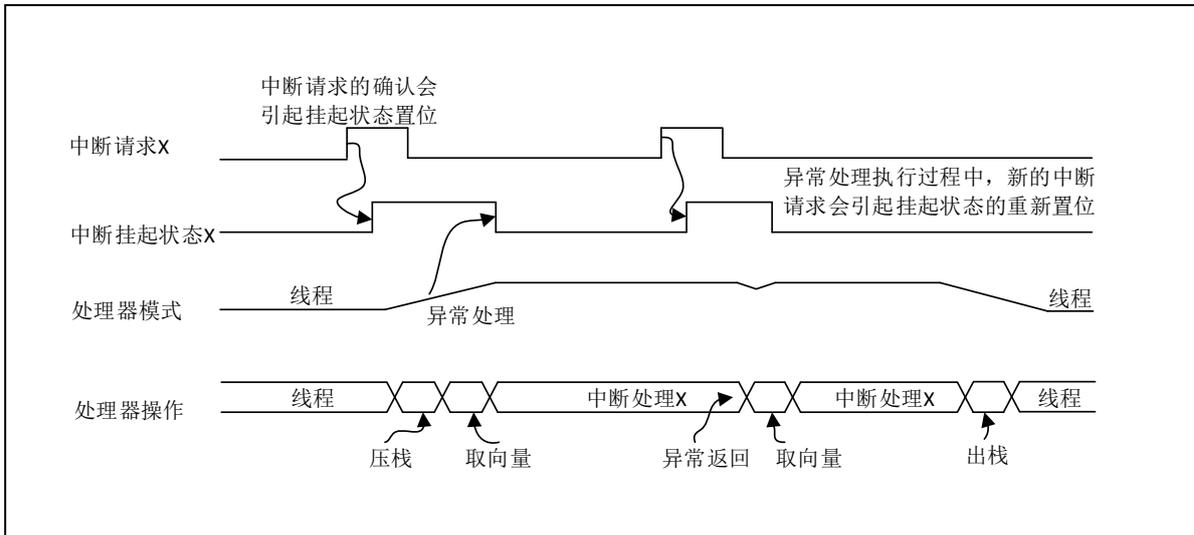


图 5-6 中断处理中产生的中断挂起也可以被确认

5.5 中断等待

通常情况下，NVIC 的中断等待时间为 16 个周期。这个等待时间从中断确认的处理器时钟周期开始，一直到中断处理开始执行结束。计算中断等待需要具备以下前提：

- 该中断使能并且没有被 SCS_PRIMASK 或是其他正在执行的异常处理所屏蔽。
- 存储器系统没有任何等待状态，在中断处理、压栈、取向量或者中断处理开始时的取指都会用到总线传输，如果存储器系统需要等待，那么发生总线传输时产生的等待状态则可能使得中断延迟。

下面的几种情况可能会导致不同的中断等待：

- 中断的末尾连锁，如果中断返回时产生了另外一个中断请求，处理器就会跳过出栈和压栈过程，这样就减少了中断等待时间。
- 延迟到达，如果中断发生时，另外一个低优先级的中断正在进行压栈处理，由于延迟到达机制的存在，高优先级的中断会首先执行，这样也会导致高优先级中断的等待时间减小。

5.6 中断源

因为 Cortex-M0+ 处理器的 NVIC 支持最多 32 个外部中断，而在本系统中，外部中断源大于 32 个，所以部分外部中断复用在同一个 NVIC 中断输入上，并且 NMI（不可屏蔽中断）并没有使用。本系统所有外部中断源和 NVIC 中断输入的对应关系如下表所示：

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#0	PORT0	v	v	v
中断#1	PORT1	v	v	v
中断#2	PORT2	v	v	v
中断#3	PORT3	v	v	v
中断#4	Reserved	-	-	-
中断#5	Reserved	-	-	-
中断#6	UART0	v	v	-
中断#7	UART1	v	v	-
中断#8	LPUART	v	v	v
中断#9	Reserved	-	-	-
中断#10	SPI	v	v	-
中断#11	Reserved	-	-	-

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#12	I2C	v	v	-
中断#13	Reserved	-	-	-
中断#14	TIM0	v	v	-
中断#15	TIM1	v	v	-
中断#16	TIM2	v	v	-
中断#17	LPTIM	v	v	v
中断#18	TIM4	v	v	-
中断#19	TIM5	v	v	-
中断#20	TIM6		v	-
中断#21	PCA	v	v	-
中断#22	WDT	v	v	v
中断#23	RTC	v	v	v
中断#24	ADC	v	v	-
中断#25	Reserved	-	-	-
中断#26	VC0	v	v	v
中断#27	VC1	v	v	v
中断#28	LVD	v	v	v
中断#29	Reserved	-	-	-
中断#30	EFCTRL/RAMCTRL	v	v	-
中断#31	CLK_TRIM	v	v	v

表 5-2 外部中断与 NVIC 中断输入对应关系

注意：

- 由于某些模块中断被复用于同一个 IRQ 中断源，当 CPU 进入该中断操作时，必须先判断是哪个模块产生的中断，再进行相对应的中断操作。

5.7 中断结构图

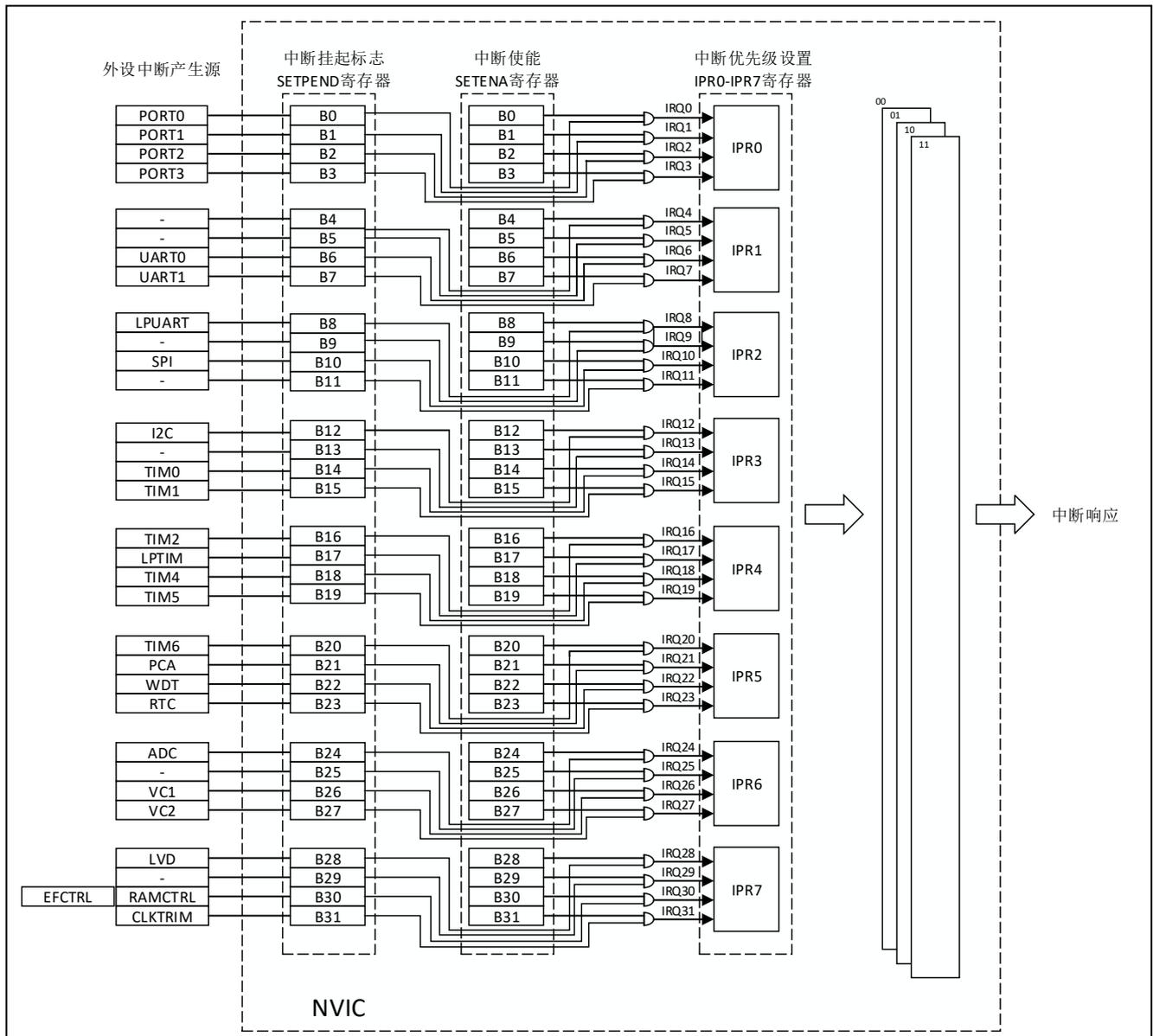


图 5-7 中断结构图

本系统的中断结构框图如图 5-7 所示。需要注意几点：

- 外设中断源各自的中断使能没有在图中标出，这里只包括外设中断产生之后的中断信号逻辑框图。
- IRQ30 有 2 个外设中断输入复用，必须分别读取这 2 个外设的中断标志位来判定是哪个外设的中断。
- 如果外设中断源有高电平产生，无论 NVIC 中断使能寄存器 SCS_SETENA 置位与否，中断挂起寄存器 SCS_SEPEND 都会被置位，表示相应的外设中断源有中断产生。

- 只有中断使能寄存器 SCS_SETENA 置位，相应的中断 IRQ 才会给处理器相应，执行相应的中断程序。
- 在中断程序中必须清除外设中断源高电平中断信号，中断挂起寄存器 SCS_SETPEND 由硬件自动清除。
- 中断优先级寄存器 SCS_IPR0- SCS_IPR7 设置了 32 个中断源的优先级，00 优先级最高，11 优先级最低。当优先级相同时，优先级由中断编号决定，编号越小优先级越高。

5.8 寄存器

基地址：0xE000 E000

寄存器	偏移地址	描述
SCS_SETENA	0x100	中断请求使能寄存器
SCS_CLRENA	0x180	中断请求清除使能寄存器
SCS_SETPEND	0x200	中断设置挂起寄存器
SCS_CLRPEND	0x280	中断清除挂起寄存器
SCS_IPR0	0x400	中断#0-中断#3优先级寄存器
SCS_IPR1	0x404	中断#4-中断#7优先级寄存器
SCS_IPR2	0x408	中断#8-中断#11优先级寄存器
SCS_IPR3	0x40C	中断#12-中断#15优先级寄存器
SCS_IPR4	0x410	中断#16-中断#19优先级寄存器
SCS_IPR5	0x414	中断#20-中断#23优先级寄存器
SCS_IPR6	0x418	中断#24-中断#27优先级寄存器
SCS_IPR7	0x41C	中断#28-中断#31优先级寄存器
SCS_PRIMASK	-	中断屏蔽特殊寄存器

5.8.1 中断使能设置寄存器 (SCS_SETENA)

偏移地址：0x100

复位值：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															

位	标记	功能描述
31:0	SETENA [31:0]	设置使能中断#0到中断#31；写“1”置位，写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

5.8.2 中断使能清除寄存器 (SCS_CLRENA)

偏移地址: 0x180

复位值: 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															

位	标记	描述
31:0	CLRENA	清除使能中断#0到中断#31; 写“1”清零, 写“0”无效

5.8.3 中断挂起状态设置寄存器 (SCS_SETPEND)

偏移地址: 0x200

复位值: 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															

位	标记	功能描述
31:0	SETPEND	设置中断#0到中断#31的挂起状态; 写“1”置位, 写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

5.8.4 中断挂起状态清除寄存器 (SCS_CLRPEND)

偏移地址: 0x280

复位值: 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															

位	标记	描述
31:0	CLRPEND	清除中断#0到中断#31的挂起状态; 写“1”清零, 写“0”无效 [0]:IRQ0 [1]:IRQ1 [2]:IRQ2 [31]:IRQ31

5.8.5 中断优先级寄存器 (SCS_IPR0)

偏移地址: 0x400

复位值: 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															

位	标记	功能描述
31:0	IPR0[31:0]	中断#0到中断#3的优先级: [31:30]: 中断#3 的优先级 [23:22]: 中断#2 的优先级 [15:14]: 中断#1 的优先级 [7:6]: 中断#0 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.6 中断优先级寄存器 (SCS_IPR1)

偏移地址: 0x404

复位值: 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															

位	标记	功能描述
31:0	IPR1[31:0]	中断#4到中断#7的优先级: [31:30]: 中断#7 的优先级 [23:22]: 中断#6 的优先级 [15:14]: 中断#5 的优先级 [7:6]: 中断#4 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.7 中断优先级寄存器 (SCS_IPR2)

偏移地址: 0x408

复位值: 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															

位	标记	功能描述
31:0	IPR2[31:0]	中断#8到中断#11的优先级; [31:30]: 中断#11 的优先级 [23:22]: 中断#10 的优先级 [15:14]: 中断#9 的优先级 [7:6]: 中断#8 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.8 中断优先级寄存器 (SCS_IPR3)

偏移地址: 0x40C

复位值: 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															

位	标记	功能描述
31:0	IPR3[31:0]	中断#12到中断#15的优先级; [31:30]: 中断#15 的优先级 [23:22]: 中断#14 的优先级 [15:14]: 中断#13 的优先级 [7:6]: 中断#12 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.9 中断优先级寄存器 (SCS_IPR4)

偏移地址: 0x410

复位值: 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															

位	标记	功能描述
31:0	IPR4[31:0]	中断#16到中断#19的优先级; [31:30]: 中断#19 的优先级 [23:22]: 中断#18 的优先级 [15:14]: 中断#17 的优先级 [7:6]: 中断#16 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.10 中断优先级寄存器 (SCS_IPR5)

偏移地址: 0x414

复位值: 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															

位	标记	功能描述
31:0	IPR5[31:0]	中断#20到中断#23的优先级; [31:30]: 中断#23 的优先级 [23:22]: 中断#22 的优先级 [15:14]: 中断#21 的优先级 [7:6]: 中断#20 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.11 中断优先级寄存器 (SCS_IPR6)

偏移地址: 0x418

复位值: 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															

位	标记	功能描述
31:0	IPR6[31:0]	中断#24到中断#27的优先级; [31:30]: 中断#27 的优先级 [23:22]: 中断#26 的优先级 [15:14]: 中断#25 的优先级 [7:6]: 中断#24 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.12 中断优先级寄存器 (SCS_IPR7)

偏移地址: 0x41C

复位值: 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															

位	标记	功能描述
31:0	IPR7[31:0]	中断#28到中断#31的优先级; [31:30]: 中断#31 的优先级 [23:22]: 中断#30 的优先级 [15:14]: 中断#29 的优先级 [7:6]: 中断#28 的优先级 其中, 00 优先级最高, 11 优先级最低

5.8.13 中断屏蔽特殊寄存器 (SCS_PRIMASK)

偏移地址: ---

复位值: 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															PRIMASK
R															RW

SCS_PRIMASK															
复位值: 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															PRIMASK
RO															RW
BIT	符号	描述													
31:1	Reserved														
0	PRIMASK	置位后，除了 NMI 和硬件错误异常外的其他中断都会被屏蔽掉 清零后，所有异常和中断都不会被屏蔽掉 该特殊寄存器需要通过 MSR 和 MRS 特殊寄存器操作指令访问，也可以用改变处理器状态指令 CPS 访问。在处理对时间敏感的应用时，需要操作 PRIMASK 寄存器。													

5.9 软件基本操作

5.9.1 外部中断使能

在每一个外设模块内部都有各自的中断使能寄存器，在需要进行中断操作时，必须首先打开外设自己的中断使能。该使能位的操作没有在本章节讨论，请参考外设模块各自的章节描述。

5.9.2 NVIC 中断使能和清除使能

Cortex-M0+ 处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零使能位。这样就有了 32 位的中断使能寄存器 `SCS_SETENA` 和 32 位的清零使能寄存器 `SCS_CLRENA`。如果想使能某一个中断，则对 `SCS_SETENA` 寄存器的相应位置 1。如果想清零某一个中断，则对 `SCS_CLRENA` 寄存器的相应位置 1。

注意，这里提到的中断使能仅仅是针对处理器 NVIC 而言的，每个外设的中断生成与否，是由外设的中断控制寄存器决定，与 `SCS_SETENA` 和 `SCS_CLRENA` 无关。

5.9.3 NVIC 中断挂起和清除挂起

如果一个中断发生了，却无法立即处理，这个中断请求将会被挂起。挂起状态保存在一个寄存器中，如果处理器的当前优先级还没有降低到可以处理挂起的请求，并且没有手动清除挂起状态，该状态将会一直保持合法。

当处理器开始进入中断处理，硬件会自动引起挂起状态的清除。

可以通过操作中断设置挂起 `SCS_SETPEND` 和中断清除挂起 `SCS_CLRPEND` 这两个寄存器来访问或修改中断挂起状态。中断挂起状态寄存器允许使用软件来触发中断。

5.9.4 NVIC 中断优先级

设置 `SCS_IPR0-SCS_IPR7` 寄存器决定 `SCS_IRQ0-SCS_IRQ32` 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优先级，这种情况的结果不可预知，并且不被 Cortex-M0+ 处理器支持。

5.9.5 NVIC 中断屏蔽

有些对时间敏感的应用，需要在一段较短的时间内禁止所有中断，可以利用中断屏蔽寄存器 `SCS_PRIMASK` 实现。特殊寄存器 `SCS_PRIMASK` 只有 1 位有效，并且在复位后默认为 0。该寄存器为 0 时，所有的中断和异常都处于允许状态；而设为 1 后，只有 NMI（本系统不支持）和硬件错误异常处于使能。实际上，当 `SCS_PRIMASK` 设置为 1 后，处理器的当前优先级就降到了 0（可和值的最高优先级）。

可以通过多种方法编程 `SCS_PRIMASK` 寄存器，使用汇编语言，可以利用 `MSR` 指令在设置和清除 `SCS_PRIMASK` 寄存器。若使用 C 语言以及 CMSIS 设备驱动库，用户可以使用以下函数来设置和清除 `PRIMASK`。

```
void __enable_irq(void); //清除 PRIMASK  
void __disable_irq(void); //设置 PRIMASK
```

6 端口控制器 (GPIO)

6.1 端口控制器简介

本产品有 16 个数字通用输入输出端口 P01-P03, P14-P15, P23-P27, P31-P36 以及 1 个数字通用输入端口 P00。模拟信号 ADC/VC/LVD 的输入输出信号、各功能模块 (如 SPI, UART, I2C, Timer 等) 的输入输出信号以及测试调试功能的输入输出信号都可以和数字端口复用。

每个端口都可以配置成内部上拉(pull up)/下拉(pull down)的输入, 高阻输入 (floating input), 推挽输出(CMOS output), 开漏输出(open drain output), 两档驱动能力输出。为防止芯片被异常复位时, 外部器件产生异常动作, 芯片复位后端口配置为高阻输入。但为了避免高阻输入而产生的漏电, 用户要在芯片启动之后对端口进行相应的配置 (配置成内部上拉/下拉输入或者输出)。

数字端口被配置成模拟端口后, 数字功能被隔离, 不能输出数字“1”和“0”, CPU 读取端口的结果为“0”。

所有端口都可以提供外部中断, 并且每个中断都可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发 4 种类型, 查询 Px_STAT[n] 的中断标志位即可查出相应的中断触发端口。另外, 每个端口的中断都可以把芯片从休眠模式/深度休眠模式唤醒到工作模式。

6.2 端口控制器主要特性

端口控制器支持以下特性：

- 端口复用功能
 - 模拟功能引脚复用
 - 调试引脚复用
 - 数字通用引脚复用
 - 数字功能引脚复用
- 配置功能
 - 支持上拉/下拉
 - 低驱动/高驱动
 - 推挽输出
 - 开漏输出
- 外部中断源
 - 高电平/低电平
 - 上升沿/下降沿
- 支持工作模式/休眠模式/深度休眠模式下中断

6.3 端口控制器功能描述

6.3.1 端口配置功能

每一个端口都可以根据系统需求通过配置寄存器(PxADS)把端口配置成模拟端口或数字端口。当配置为数字端口时，还可以配置相应的寄存器实现以下特性：

1. 内部上拉 (PxPU) /下拉 (PxPD)

上拉寄存器 (PxPU)、下拉寄存器 (PxPD) 分别对应端口上拉使能和端口下拉使能，当对应位为 ‘1’ 时，设置对应位引脚上拉/下拉使能，为 ‘0’ 时，禁止对应位引脚上拉/下拉。

2. 两档驱动输出(PxDR)

能够通过 PxDR 寄存器改变驱动能力，PxDR 为 ‘1’ 时为低驱动能力，PxDR 为 ‘0’ 时为高驱动能力。

3. 开漏输出 (PxOD)

通过 PxOD 寄存器设置管脚输出状态。当 PxOD 为 ‘1’ 时，端口开漏输出使能，为 ‘0’ 时，端口开漏输出禁止。开漏引脚不连接外部的上拉电阻时，只能输出低电平，如果需要同时具备输出高电平的功能时，则需要上拉电阻。

4. 方向选择 (PxDIR)

用于设置端口引脚的方向。PxDIR 为 ‘0’ 的时候端口为输出，PxDIR 为 ‘1’ 的时候端口为输入。

5. 输出高低电平选择(PxOUT)，可通过 AHB 总线访问

当端口引脚配置为输出时，如果 PxOUT 为 ‘1’，端口引脚输出为高电平，若配置成开漏输出，则需外部上拉电阻来拉高。如果 PxOUT 为 ‘0’，则输出低电平。

6. 输入电平状态 (PxIN) 可通过 AHB 总线访问

通过读取 PxIN 寄存器能够获得同步后的管脚电平，PxIN 为 ‘1’ 时为高电平，PxIN 为 ‘0’ 时为低电平

注：以上特性在配置为模拟端口时无效。

端口的状态和寄存器配置的关系如下表:

IO 状态	IO 方向	PxADS	PxDIR	PxOUT	PxIN	PxPU	PxPD	PxOD	PxDR	Px_SEL
模拟	输入/输出	1	W	W	0	W	W	W	W	W
浮空	输入	0	1	W	X	0	0	W	W	0
下拉	输入	0	1	W	0	0	1	W	W	0
上拉	输入	0	1	W	1	1	0	W	W	0
上拉	输入	0	1	W	1	1	1	W	W	0
1	输入	0	1	W	1	W	W	W	W	0
0	输入	0	1	W	0	W	W	W	W	0
1	输出	0	0	1	1	W	W	0	W	0
0	输出	0	0	0	0	W	W	0	W	0
1	输出	0	0	W	1	W	W	0	W	0
0	输出	0	0	W	0	W	W	0	W	0
(SET)1/(CLR)0	输出	0	0	W	(SET)1/(CLR)0	W	W	0	W	0
0	输出	0	0	0	0	W	W	1	W	0
Z	输出	0	0	1	X	0	0	1	W	0
0	输出	0	0	1	0	0	1	1	W	0
1	输出	0	0	1	1	1	0	1	W	0
1	输出	0	0	1	1	1	1	1	W	0

注: 0 - Logic low 1 - Logic high W - Whatever 0 or 1 X - unknow state Z - high impedance

表 6-1 端口状态真值表

端口电路结构如下图所示：

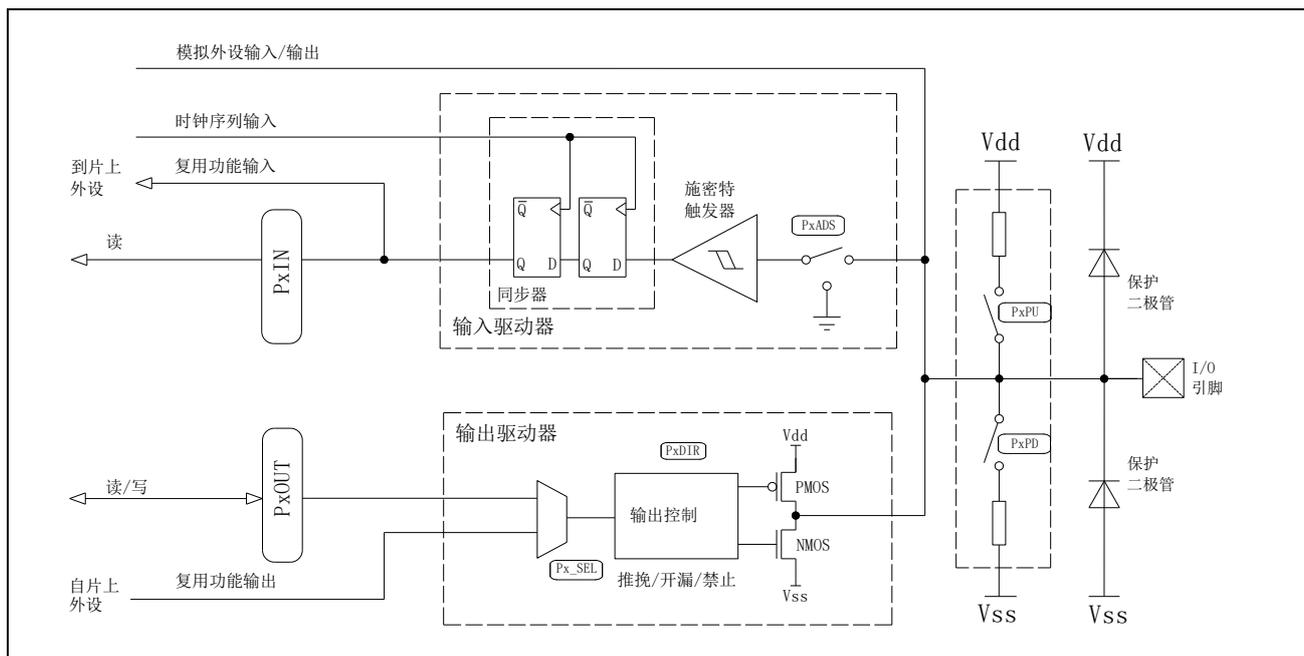


图 6-1 端口电路示意图

6.3.2 端口的写入

端口输入值/输出值寄存器（PxIN/PxOUT）支持 AHB 总线读写。对于 AHB 总线，系统时钟（HCLK）对其处理周期与其他总线并不相同，每两个 HCLK 周期，IO 翻转一次。下图为 AHB 总线端口翻转的最快时序：

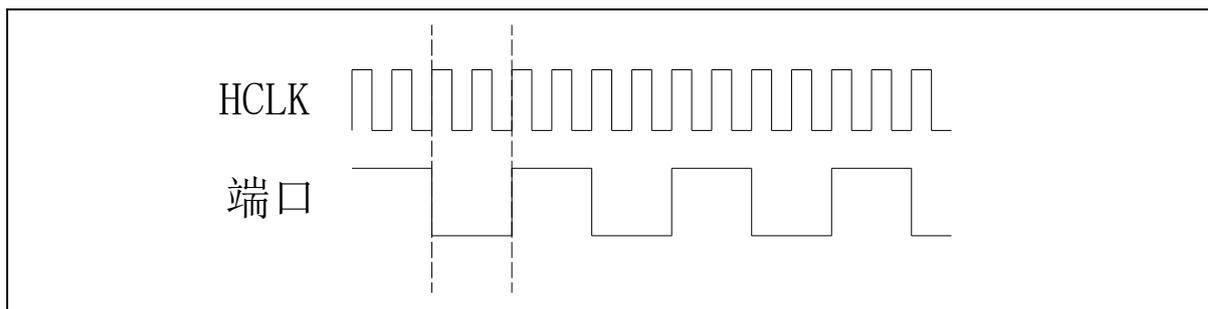


图 6-2 AHB 总线端口随系统时钟的变化

6.3.3 端口的读取

每一个端口可以通过读取 PxIN 寄存器来获得端口引脚电平。PxIN 寄存器的各个位与其前面的锁存器组成了一个同步器，从而避免了系统时钟状态发生改变时在短时间内引脚电平变化而造成的信号不稳定，但是同时也引入了延迟。读取端口引脚数据的同步图如下：

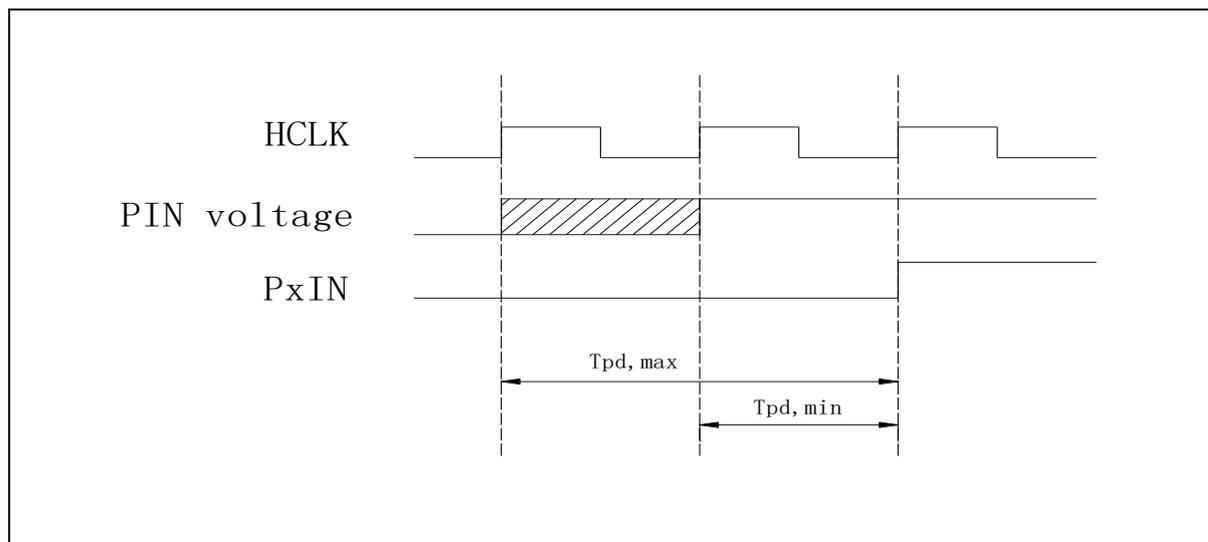


图 6-3 读取端口引脚数据同步图

在系统时钟上升沿之后的时钟时期，管脚电平信号会锁存在内部寄存器，如阴影部分所示，在下一次系统时钟上升沿之后，稳定的管脚电平信号能够被读取。之后再系统时钟上升沿时，数据锁存到 PxIN 寄存器中。信号装换延迟 T_{pd} 为 1-2 个系统时钟。

注意：

– 未连接引脚的处理：

在使用过程中如果有未接引脚，为避免在其他数字输入使能模式时引脚没有确定的电平造成悬空消耗电流，推荐赋予引脚一个确定的电平。

6.3.4 端口复用功能

端口复用是端口控制器的主要功能之一。通过配置寄存器，可以灵活的把端口配置成模拟端口/测试调试端口/数字通用端口/数字功能端口。

PxADS 寄存器用于数字端口/模拟端口切换，当 PxADS 为 ‘1’ 时，端口配置成模拟端口，此时数字功能被隔离，不能输出数字“1”和“0”，CPU 读取端口的结果为“0”。当 PxADS 为 ‘0’ 时，端口配置成数字端口，此时通过配置 Px_sel 寄存器实现数字通用端口/数字功能端口切换，每一个端口都可以独立配置成系统所需的功能端口。测试调试端口的配置信息请参考相关章节。

模拟		数字							
PxADS=1		PxADS=0							
		Px_sel=0	Px_sel=1	Px_sel=2	Px_sel=3	Px_sel=4	Px_sel=5	Px_sel=6	Px_sel=7
AIN4	VCIN4	P34	PCA_CH0	LPUART_TXD	TIM5_CHA	TIM0_EXT	TIM4_CHA	RTC_1Hz	TIM1_TOG
AIN5	VCIN5	P35	UART1_TXD	TIM6_CHB	UART0_TXD	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	I2C_SCL
AIN7 XTHI	VCIN7	P01	UART0_RXD	I2C_SDA	UART1_TXD	TIM0_TOG	TIM5_CHB	SPI_SCK	TIM2_EXT
AIN8 XTHO		P02	UART0_TXD	I2C_SCL	UART1_RXD	TIM0_TOGN	TIM6_CHA	SPI_CS	TIM2_GATE
LV DIN1		P03	PCA_CH3	SPI_CS	TIM6_CHB	LPTIM_EXT	RTC_1Hz	PCA_ECI	VC0_OUT
XTLO		P15	I2C_SDA	TIM2_TOG	TIM4_CHB	LPTIM_GATE	SPI_SCK	UART0_RXD	LVD_OUT
XTLI		P14	I2C_SCL	TIM2_TOGN	PCA_ECI	ADC_RDY	SPI_CS	UART0_TXD	NC
LV DIN2	VCIN0	P23	TIM6_CHA	TIM4_CHB	TIM4_CHA	PCA_CH0	SPI_MISO	UART1_TXD	IR_OUT
AIN0		P24	TIM4_CHB	TIM5_CHB	HCLK_OUT	PCA_CH1	SPI_MOSI	UART1_RXD	VC1_OUT
LV DIN3	VCIN1	P25	SPI_SCK	PCA_CH0	TIM5_CHA	LVD_OUT	LPUART_RXD	I2C_SDA	TIM1_GATE
AIN1		P26	SPI_MOSI	TIM4_CHA	TIM5_CHB	PCA_CH2	LPUART_TXD	I2C_SCL	TIM1_EXT
		P27/SWDIO	SPI_MISO	TIM5_CHA	TIM6_CHA	PCA_CH3	UART0_RXD	RCH_OUT	XTH_OUT
		P31/SWCLK	LPTIM_TOG	PCA_ECI	PCLK_OUT	VC0_OUT	UART0_TXD	RCL_OUT	HCLK_OUT
AIN2	VCIN2	P32	LPTIM_TOGN	PCA_CH2	TIM6_CHB	VC1_OUT	UART1_TXD	PCA_CH4	RTC_1Hz
AIN3	VCIN3	P33	LPUART_RXD	PCA_CH1	TIM5_CHB	PCA_ECI	UART1_RXD	XTL_OUT	TIM1_TOGN
		P00 Reset							

表 6-2 端口复用表

6.3.5 端口中断功能

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，分别对应的中断使能寄存器为高电平中断使能寄存器/低电平中断使能寄存器/上升沿中断使能寄存器/下降沿中断使能寄存器。

当中断触发时，通过查询中断状态寄存器就可以判断是哪一个端口触发了中断，通过清零中断清除寄存器就可以清除对应的中断状态标志位。

6.4 端口配置操作

6.4.1 端口复用操作流程

端口复用配置为模拟端口

Step1: 设置寄存器 Px_ADS 为 1

端口复用配置为数字通用端口

- a) 设置寄存器 Px_ADS 为 0
- b) 设置寄存器 Px_sel 为 0
- c) 设置寄存器 PxDIR 为 1: 端口方向为输入, CPU 可以读取端口的状态 PxIN
- d) 设置寄存器 PxDIR 为 0: 端口方向为输出
- e) 设置寄存器 PxOUT 为 1: 端口输出高电平
- f) 设置寄存器 PxOUT 为 0: 端口输出低电平

端口复用配置为数字功能端口

- a) 设置寄存器 Px_ADS 为 0
- b) 设置寄存器 Px_sel 为 1~7 (根据系统需求, 参考端口复用表)
- c) 设置寄存器 PxDIR(根据系统需求)

端口复用配置为调试测试端口

参考测试调试相关章节。

端口复用配置为红外输出信号

端口 P23 可以配置为频率为 38K 的红外输出信号。

- a) 设置寄存器 P23_ADS 为 0
- b) 设置寄存器 P23_sel 为 7
- c) 设置寄存器 P2DIR[3] 为 0: 端口方向为输出
- d) 设置寄存器 GPIO_CTRL1 的 bit14 选择红外信号输出极性
- e) 设置寄存器 P2OUT[3] 门控红外信号的输出

6.4.2 端口中断操作流程

高电平中断

- a) 设置寄存器 Px_ADS 为 0
- b) 设置寄存器 Px_sel 为 0
- c) 设置寄存器 PxDIR 为 1
- d) 设置寄存器 PxHIE 为 1
- e) 中断触发后读取中断状态寄存器 Px_STAT
- f) 设置寄存器 Px_ICLR 为 0 清除中断状态寄存器 Px_STAT

低电平中断

- a) 设置寄存器 Px_ADS 为 0
- b) 设置寄存器 Px_sel 为 0
- c) 设置寄存器 PxDIR 为 1
- d) 设置寄存器 PxLIE 为 1
- e) 中断触发后读取中断状态寄存器 Px_STAT
- f) 设置寄存器 Px_ICLR 为 0 清除中断状态寄存器 Px_STAT

上升沿中断

- a) 设置寄存器 Px_ADS 为 0
- b) 设置寄存器 Px_sel 为 0
- c) 设置寄存器 PxDIR 为 1
- d) 设置寄存器 PxRIE 为 1
- e) 中断触发后读取中断状态寄存器 Px_STAT
- f) 设置寄存器 Px_ICLR 为 0 清除中断状态寄存器 Px_STAT

下降沿中断

- a) 设置寄存器 Px_ADS 为 0
- b) 设置寄存器 Px_sel 为 0
- c) 设置寄存器 PxDIR 为 1
- d) 设置寄存器 PxFIE 为 1
- e) 中断触发后读取中断状态寄存器 Px_STAT

- f) 设置寄存器 Px_ICLR 为 0 清除中断状态寄存器 Px_STAT

6.4.3 端口配置操作流程

上拉使能

- a) 设置寄存器 PxPU 为 1

下拉使能

- a) 设置寄存器 PxPU 为 0
- b) 设置寄存器 PxPD 为 1

高驱动能力

- a) 设置寄存器 PxDR 为 0

开漏输出

- a) 设置寄存器 PxOD 为 1

6.5 端口控制器寄存器描述

寄存器列表

基地址：0x40020C00

偏移量	寄存器名称	访问	寄存器描述
0x04	P01_SEL	RW	端口 P01 功能配置寄存器
0x08	P02_SEL	RW	端口 P02 功能配置寄存器
0x0c	P03_SEL	RW	端口 P03 功能配置寄存器
0x50	P14_SEL	RW	端口 P14 功能配置寄存器
0x54	P15_SEL	RW	端口 P15 功能配置寄存器
0x8c	P23_SEL	RW	端口 P23 功能配置寄存器
0x90	P24_SEL	RW	端口 P24 功能配置寄存器
0x94	P25_SEL	RW	端口 P25 功能配置寄存器
0x98	P26_SEL	RW	端口 P26 功能配置寄存器
0x9c	P27_SEL	RW	端口 P27 功能配置寄存器
0xc4	P31_SEL	RW	端口 P31 功能配置寄存器
0xc8	P32_SEL	RW	端口 P32 功能配置寄存器
0xcc	P33_SEL	RW	端口 P33 功能配置寄存器
0xd0	P34_SEL	RW	端口 P34 功能配置寄存器
0xd4	P35_SEL	RW	端口 P35 功能配置寄存器
0xd8	P36_SEL	RW	端口 P36 功能配置寄存器
0x100	P0DIR	RW	端口 P0 输入输出配置寄存器
0x104	P0IN	RO	端口 P0 输入值寄存器
0x108	P0OUT	RW	端口 P0 输出值配置寄存器
0x10c	P0ADS	RW	端口 P0 数模配置寄存器
0x11c	P0DR	RW	端口 P0 驱动能力配置寄存器
0x120	P0PU	RW	端口 P0 上拉使能配置寄存器
0x124	P0PD	RW	端口 P0 下拉使能配置寄存器
0x12c	P0OD	RW	端口 P0 开漏输出配置寄存器

0x130	POHIE	RW	端口 P0 高电平中断使能配置寄存器
0x134	POLIE	RW	端口 P0 低电平中断使能配置寄存器
0x138	PORIE	RW	端口 P0 上升沿中断使能配置寄存器
0x13c	POFIE	RW	端口 P0 下降沿中断使能配置寄存器
0x200	PO_STAT	RO	端口 P0 中断状态寄存器
0x210	PO_ICLR	RW	端口 P0 中断清除寄存器
0x140	P1DIR	RW	端口 P1 输入输出配置寄存器
0x144	P1IN	RO	端口 P1 输入值寄存器
0x148	P1OUT	RW	端口 P1 输出值配置寄存器
0x14c	P1ADS	RW	端口 P1 数模配置寄存器
0x15c	P1DR	RW	端口 P1 驱动能力配置寄存器
0x160	P1PU	RW	端口 P1 上拉使能配置寄存器
0x164	P1PD	RW	端口 P1 下拉使能配置寄存器
0x16c	P1OD	RW	端口 P1 开漏输出配置寄存器
0x170	P1HIE	RW	端口 P1 高电平中断使能配置寄存器
0x174	P1LIE	RW	端口 P1 低电平中断使能配置寄存器
0x178	P1RIE	RW	端口 P1 上升沿中断使能配置寄存器
0x17c	P1FIE	RW	端口 P1 下降沿中断使能配置寄存器
0x240	P1_STAT	RO	端口 P1 中断状态寄存器
0x250	P1_ICLR	RW	端口 P1 中断清除寄存器
0x180	P2DIR	RW	端口 P2 输入输出配置寄存器
0x184	P2IN	RO	端口 P2 输入值寄存器
0x188	P2OUT	RW	端口 P2 输出值配置寄存器
0x18c	P2ADS	RW	端口 P2 数模配置寄存器
0x19c	P2DR	RW	端口 P2 驱动能力配置寄存器
0x1a0	P2PU	RW	端口 P2 上拉使能配置寄存器
0x1a4	P2PD	RW	端口 P2 下拉使能配置寄存器

0x1ac	P2OD	RW	端口 P2 开漏输出配置寄存器
0x1b0	P2HIE	RW	端口 P2 高电平中断使能配置寄存器
0x1b4	P2LIE	RW	端口 P2 低电平中断使能配置寄存器
0x1b8	P2RIE	RW	端口 P2 上升沿中断使能配置寄存器
0x1bc	P2FIE	RW	端口 P2 下降沿中断使能配置寄存器
0x280	P2_STAT	RO	端口 P2 中断状态寄存器
0x290	P2_ICLR	RW	端口 P2 中断清除寄存器
0x1c0	P3DIR	RW	端口 P3 输入输出配置寄存器
0x1c4	P3IN	RO	端口 P3 输入值寄存器
0x1c8	P3OUT	RW	端口 P3 输出值配置寄存器
0x1cc	P3ADS	RW	端口 P3 数模配置寄存器
0x1dc	P3DR	RW	端口 P3 驱动能力配置寄存器
0x1e0	P3PU	RW	端口 P3 上拉使能配置寄存器
0x1e4	P3PD	RW	端口 P3 下拉使能配置寄存器
0x1ec	P3OD	RW	端口 P3 开漏输出配置寄存器
0x1f0	P3HIE	RW	端口 P3 高电平中断使能配置寄存器
0x1f4	P3LIE	RW	端口 P3 低电平中断使能配置寄存器
0x1f8	P3RIE	RW	端口 P3 上升沿中断使能配置寄存器
0x1fc	P3FIE	RW	端口 P3 下降沿中断使能配置寄存器
0x2c0	P3_STAT	RO	端口 P3 中断状态寄存器
0x2d0	P3_ICLR	RW	端口 P3 中断清除寄存器
0x304	GPIO_CTRL1	RW	端口辅助功能配置寄存器 1
0x308	GPIO_CTRL2	RW	端口辅助功能配置寄存器 2
0x30c	GPIO_CTRL3	RW	端口辅助功能配置寄存器 3
0x310	GPIO_CTRL4	RW	端口辅助功能配置寄存器 4

6.5.1 端口 P0

6.5.1.1 端口 P01 功能配置寄存器(P01_SEL)

偏移地址: 0x04

复位值: 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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P01_sel	端口 P01 功能选择. 000: GPIO P01 001: UART0_RXD UART0 模块 RXD 信号 010: I2C_SDA I2C 模块数据信号 011: UART1_TXD UART1 模块 TXD 信号 100: TIM0_TOG Timer0 模块翻转信号 101: TIM5_CHB Advanced Timer 模块通道 1 B 信号 110: SPI_SCK SPI 模块时钟信号 111: TIM2_EXT Timer2 模块外部时钟输入信号

6.5.1.2 端口 P02 功能配置寄存器(P02_SEL)

偏移地址：0x08

复位值：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		

位	标记	功能描述
31:3	Reserved	
2:0	P02_sel	端口 P02 功能选择. 000: GPIO P02 001: UART0_TXD UART0 模块 TXD 信号 010: I2C_SCL I2C 模块时钟信号 011: UART1_RXD UART1 模块 RXD 信号 100: TIM0_TOGN Timer0 模块翻转信号的反向信号 101: TIM6_CHA Advanced Timer 模块通道 2 A 信号 110: SPI_CS SPI 模块主机模式片选信号 111: TIM2_GATE Timer2 模块门控信号

6.5.1.3 端口 P03 功能配置寄存器(P03_SEL)

偏移地址：0x0C

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P03_sel	端口 P03 功能选择. 000: GPIO P03 001: PCA_CH3 PCA 模块通道 3 捕获/比较信号 010: SPI_CS SPI 模块主机模式片选信号 011: TIM6_CHB Advanced Timer 模块通道 2 B 信号 100: LPTIM_EXT Timer3 模块外部时钟输入信号 101: RTC_1Hz RTC 模块 1Hz 输出信号 110: PCA_ECI PCA 模块外部时钟输入信号 111: VC0_OUT VC0 模块输出

6.5.1.4 端口 P0 输入输出配置寄存器(P0DIR)

偏移地址：0x100

复位值：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	P0D	P0D	Res
												IR3	IR2	IR1	
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0DIR3	端口 P03 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
2	P0DIR2	端口 P02 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
1	P0DIR1	端口 P01 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
0	Reserved	

6.5.1.5 端口 P0 输入值寄存器(P0IN)

偏移地址：0x104

复位值：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												P0I	P0I	P0I	P0I
												N3	N2	N1	N0
												RO	RO	RO	RO

位	标记	功能描述
31:4	Reserved	
3	P0IN3	端口 P03 输入值寄存器 1: 输入为高电平 0: 输入为低电平
2	P0IN2	端口 P02 输入值寄存器 1: 输入为高电平 0: 输入为低电平
1	P0IN1	端口 P01 输入值寄存器 1: 输入为高电平 0: 输入为低电平
0	P0IN0	端口 P00 输入值寄存器 1: 输入为高电平 0: 输入为低电平

6.5.1.6 端口 P0 输出值配置寄存器(P0OUT)

偏移地址：0x108

复位值：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												P0O	P0O	P0O	Res
												UT3	UT2	UT1	
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0OUT3	端口 P03 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
2	P0OUT2	端口 P02 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
1	P0OUT1	端口 P01 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
0	Reserved	

6.5.1.7 端口 P0 数模配置寄存器(P0ADS)

偏移地址：0x10C

复位值：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												P0A	P0A	P0A	Res
												DS3	DS2	DS1	
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0ADS3	端口 P03 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
2	P0ADS2	端口 P02 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
1	P0ADS1	端口 P01 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
0	Reserved	

6.5.1.8 端口 P0 驱动能力配置寄存器(P0DR)

偏移地址：0x11C

复位值：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												P0D	P0D	P0D	Res
												R3	R2	R1	
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0DR3	端口 P03 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
2	P0DR2	端口 P02 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
1	P0DR1	端口 P01 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
0	Reserved	

6.5.1.9 端口 P0 上拉使能配置寄存器(P0PU)

偏移地址：0x120

复位值：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												P0P U3	P0P U2	P0P U1	Res
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0PU3	端口 P03 上拉使能配置寄存器 1: 使能 0: 禁止
2	P0PU2	端口 P02 上拉使能配置寄存器 1: 使能 0: 禁止
1	P0PU1	端口 P01 上拉使能配置寄存器 1: 使能 0: 禁止
0	Reserved	

6.5.1.10 端口 P0 下拉使能配置寄存器(P0PD)

偏移地址：0x124

复位值：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												POP	POP	POP	Res
												D3	D2	D1	
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0PD3	端口 P03 下拉使能配置寄存器 1: 使能 0: 禁止
2	P0PD2	端口 P02 下拉使能配置寄存器 1: 使能 0: 禁止
1	P0PD1	端口 P01 下拉使能配置寄存器 1: 使能 0: 禁止
0	Reserved	

6.5.1.11 端口 P0 开漏输出配置寄存器(P0OD)

偏移地址：0x12C

复位值：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												P0O	P0O	P0O	Res
												D3	D2	D1	
												RW	RW	RW	

位	标记	功能描述
31:4	Reserved	
3	P0OD3	端口 P03 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
2	P0OD2	端口 P02 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
1	P0OD1	端口 P01 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
0	Reserved	

6.5.1.12 端口 P0 高电平中断使能配置寄存器(POHIE)

偏移地址：0x130

复位值：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												POH	POH	POH	POH
Reserved												IE3	IE2	IE1	IE0
Reserved												RW	RW	RW	RW

位	标记	功能描述
31:4	Reserved	
3	POHIE3	端口 P03 高电平中断使能配置寄存器 1: 使能 0: 禁止
2	POHIE2	端口 P02 高电平中断使能配置寄存器 1: 使能 0: 禁止
1	POHIE1	端口 P01 高电平中断使能配置寄存器 1: 使能 0: 禁止
0	POHIE0	端口 P00 高电平中断使能配置寄存器 1: 使能 0: 禁止

6.5.1.13 端口 P0 低电平中断使能配置寄存器(POLIE)

偏移地址：0x134

复位值：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												POL	POL	POL	POL
												IE3	IE2	IE1	IE0
												RW	RW	RW	RW

位	标记	功能描述
31:4	Reserved	
3	POLIE3	端口 P03 低电平中断使能配置寄存器 1: 使能 0: 禁止
2	POLIE2	端口 P02 低电平中断使能配置寄存器 1: 使能 0: 禁止
1	POLIE1	端口 P01 低电平中断使能配置寄存器 1: 使能 0: 禁止
0	POLIE0	端口 P00 低电平中断使能配置寄存器 1: 使能 0: 禁止

6.5.1.14 端口 P0 上升沿中断使能配置寄存器(P0RIE)

偏移地址：0x138

复位值：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												P0R	P0R	P0R	P0R
Reserved												IE3	IE2	IE1	IE0
Reserved												RW	RW	RW	RW

位	标记	功能描述
31:4	Reserved	
3	P0RIE3	端口 P03 上升沿中断使能配置寄存器 1: 使能 0: 禁止
2	P0RIE2	端口 P02 上升沿中断使能配置寄存器 1: 使能 0: 禁止
1	P0RIE1	端口 P01 上升沿中断使能配置寄存器 1: 使能 0: 禁止
0	P0RIE0	端口 P00 上升沿中断使能配置寄存器 1: 使能 0: 禁止

6.5.1.15 端口 P0 下降沿中断使能配置寄存器(P0FIE)

偏移地址：0x13C

复位值：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												P0F	P0F	P0F	P0F
Reserved												IE3	IE2	IE1	IE0
Reserved												RW	RW	RW	RW

位	标记	功能描述
31:4	Reserved	
3	P0FIE3	端口 P03 下降沿中断使能配置寄存器 1: 使能 0: 禁止
2	P0FIE2	端口 P02 下降沿中断使能配置寄存器 1: 使能 0: 禁止
1	P0FIE1	端口 P01 下降沿中断使能配置寄存器 1: 使能 0: 禁止
0	P0FIE0	端口 P00 下降沿中断使能配置寄存器 1: 使能 0: 禁止

6.5.1.16 端口 P0 中断状态寄存器(P0_STAT)

偏移地址：0x200

复位值：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												POS	POS	POS	POS
Reserved												TA3	TA2	TA1	TA0
Reserved												RO	RO	RO	RO

位	标记	功能描述
31:4	Reserved	
3	POSTA3	端口 P03 中断状态寄存器 1: 中断触发 0: 无中断触发
2	POSTA2	端口 P02 中断状态寄存器 1: 中断触发 0: 无中断触发
1	POSTA1	端口 P01 中断状态寄存器 1: 中断触发 0: 无中断触发
0	POSTA0	端口 P00 中断状态寄存器 1: 中断触发 0: 无中断触发

6.5.1.17 端口 P0 中断清除寄存器(P0_ICLR)

偏移地址：0x210

复位值：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												P0C	P0C	P0C	P0C
												LR3	LR2	LR1	LR0
												RW	RW	RW	RW

位	标记	功能描述
31:4	Reserved	
3	P0CLR3	端口 P03 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
2	P0CLR2	端口 P02 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
1	P0CLR1	端口 P01 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
0	P0CLR0	端口 P00 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位

6.5.2 端口 P1

6.5.2.1 端口 P14 功能配置寄存器(P14_SEL)

偏移地址: 0x50

复位值: 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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P14_sel	端口 P14 功能选择. 000: GPIO P14 001: I2C_SCL I2C 模块时钟信号 010: TIM2_TOGN Timer2 模块翻转信号的反向信号 011: PCA_ECI PCA 模块外部时钟输入信号 100: ADC_RDY ADC 模块 RDY 信号 101: SPI_CS SPI 模块主机模式片选信号 110: UART0_TXD UART0 模块 TXD 信号 111: NC

6.5.2.2 端口 P15 功能配置寄存器(P15_SEL)

偏移地址：0x54

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P15_sel	端口 P15 功能选择. 000: GPIO P15 001: I2C_SDA I2C 模块数据信号 010: TIM2_TOG Timer2 模块翻转信号 011: TIM4_CHB Advanced Timer 模块通道 0 B 信号 100: LPTIM_GATE Timer3 模块门控信号 101: SPI_SCK SPI 模块时钟信号 110: UART0_RXD UART0 模块 RXD 信号 111: LVD_OUT LVD 模块输出信号

6.5.2.3 端口 P1 输入输出配置寄存器(P1DIR)

偏移地址：0x140

复位值：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										P1D	P1D	Reserved			
Reserved										IR5	IR4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1DIR5	端口 P15 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
4	P1DIR4	端口 P14 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
3:0	Reserved	

6.5.2.4 端口 P1 输入值寄存器(P1IN)

偏移地址：0x144

复位值：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										P1I	P1I	Reserved			
Reserved										N5	N4	Reserved			
Reserved										RO	RO	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1IN5	端口 P15 输入值寄存器 1: 输入为高电平 0: 输入为低电平
4	P1IN4	端口 P14 输入值寄存器 1: 输入为高电平 0: 输入为低电平
3:0	Reserved	

6.5.2.5 端口 P1 输出值配置寄存器(P1OUT)

偏移地址：0x148

复位值：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	P1O	Reserved			
Reserved										UT5	UT4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1OUT5	端口 P15 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
4	P1OUT4	端口 P14 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
3:0	Reserved	

6.5.2.6 端口 P1 数模配置寄存器(P1ADS)

偏移地址：0x14C

复位值：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	P1A	Reserved			
Reserved										DS5	DS4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1ADS5	端口 P15 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
4	P1ADS4	端口 P14 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
3:0	Reserved	

6.5.2.7 端口 P1 驱动能力配置寄存器(P1DR)

偏移地址：0x15C

复位值：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	P1D	Reserved			
Reserved										R5	R4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1DR5	端口 P15 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
4	P1DR4	端口 P14 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
3:0	Reserved	

6.5.2.8 端口 P1 上拉使能配置寄存器(P1PU)

偏移地址：0x160

复位值：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	P1P	Reserved			
										U5	U4				
										RW	RW				

位	标记	功能描述
31:6	Reserved	
5	P1PU5	端口 P15 上拉使能配置寄存器 1: 使能 0: 禁止
4	P1PU4	端口 P14 上拉使能配置寄存器 1: 使能 0: 禁止
3:0	Reserved	

6.5.2.9 端口 P1 下拉使能配置寄存器(P1PD)

偏移地址：0x164

复位值：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	P1P	Reserved			
Reserved										D5	D4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1PD5	端口 P15 下拉使能配置寄存器 1: 使能 0: 禁止
4	P1PD4	端口 P14 下拉使能配置寄存器 1: 使能 0: 禁止
3:0	Reserved	

6.5.2.10 端口 P1 开漏输出配置寄存器(P1OD)

偏移地址：0x16C

复位值：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	P1O	Reserved			
Reserved										D5	D4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1OD5	端口 P15 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
4	P1OD4	端口 P14 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
3:0	Reserved	

6.5.2.11 端口 P1 高电平中断使能配置寄存器(P1HIE)

偏移地址：0x170

复位值：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				

位	标记	功能描述
31:6	Reserved	
5	P1HIE5	端口 P15 高电平中断使能配置寄存器 1: 使能 0: 禁止
4	P1HIE4	端口 P14 高电平中断使能配置寄存器 1: 使能 0: 禁止
3:0	Reserved	

6.5.2.12 端口 P1 低电平中断使能配置寄存器(P1LIE)

偏移地址：0x174

复位值：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 IE5 RW	P1L IE4 RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1LIE5	端口 P15 低电平中断使能配置寄存器 1: 使能 0: 禁止
4	P1LIE4	端口 P14 低电平中断使能配置寄存器 1: 使能 0: 禁止
3:0	Reserved	

6.5.2.13 端口 P1 上升沿中断使能配置寄存器(P1RIE)

偏移地址：0x178

复位值：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			
Reserved										IE5	IE4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1RIE5	端口 P15 上升沿中断使能配置寄存器 1: 使能 0: 禁止
4	P1RIE4	端口 P14 上升沿中断使能配置寄存器 1: 使能 0: 禁止
3:0	Reserved	

6.5.2.14 端口 P1 下降沿中断使能配置寄存器(P1FIE)

偏移地址：0x17C

复位值：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	P1F	Reserved			
										IE5	IE4				
										RW	RW				

位	标记	功能描述
31:6	Reserved	
5	P1FIE5	端口 P15 下降沿中断使能配置寄存器 1: 使能 0: 禁止
4	P1FIE4	端口 P14 下降沿中断使能配置寄存器 1: 使能 0: 禁止
3:0	Reserved	

6.5.2.15 端口 P1 中断状态寄存器(P1_STAT)

偏移地址：0x240

复位值：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	P1S	Reserved			
Reserved										TA5	TA4	Reserved			
Reserved										RO	RO	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1STA5	端口 P15 中断状态寄存器 1: 中断触发 0: 无中断触发
4	P1STA4	端口 P14 中断状态寄存器 1: 中断触发 0: 无中断触发
3:0	Reserved	

6.5.2.16 端口 P1 中断清除寄存器(P1_ICLR)

偏移地址: 0x250

复位值: 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										PIC	PIC	Reserved			
Reserved										LR5	LR4	Reserved			
Reserved										RW	RW	Reserved			

位	标记	功能描述
31:6	Reserved	
5	P1CLR5	端口 P15 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
4	P1CLR4	端口 P14 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
3:0	Reserved	

6.5.3 端口 P2

6.5.3.1 端口 P23 功能配置寄存器(P23_SEL)

偏移地址: 0x8C

复位值: 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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P23_sel	端口 P23 功能选择. 000: GPIO P23 001: TIM6_CHA Advanced Timer 模块通道 2 A 信号 010: TIM4_CHB Advanced Timer 模块通道 0 B 信号 011: TIM4_CHA Advanced Timer 模块通道 0 A 信号 100: PCA_CH0 PCA 模块通道 0 捕获/比较信号 101: SPI_MOSI SPI 模块主机输入从机输出数据信号 110: UART1_TXD UART1 模块 TXD 信号 111: IR_OUT 红外输出信号

6.5.3.2 端口 P24 功能配置寄存器(P24_SEL)

偏移地址：0x90

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P24_sel	端口 P24 功能选择. 000: GPIO P24 001: TIM4_CHB Advanced Timer 模块通道 0 B 信号 010: TIM5_CHB Advanced Timer 模块通道 1 B 信号 011: HCLK_OUT AHB 总线时钟输出信号 100: PCA_CH1 PCA 模块通道 1 捕获/比较信号 101: SPI_MOSI SPI 模块主机输出从机输入数据信号 110: UART1_RXD UART1 模块 RXD 信号 111: VC1_OUT VC1 模块输出

6.5.3.3 端口 P25 功能配置寄存器(P25_SEL)

偏移地址：0x94

复位值：0x0000 0000

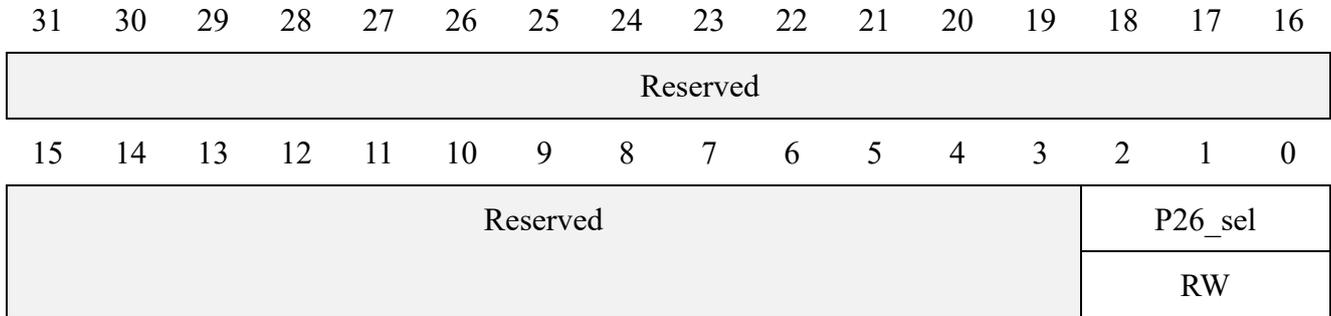
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													P25_sel		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P25_sel	端口 P25 功能选择. 000: GPIO P25 001: SPI_SCK SPI 模块时钟信号 010: PCA_CH0 PCA 模块通道 0 捕获/比较信号 011: TIM5_CHA Advanced Timer 模块通道 1 A 信号 100: LVD_OUT LVD 模块输出信号 101: LPUART_RXD LPUART 模块 RXD 信号 110: I2C_SDA I2C 模块数据信号 111: TIM1_GATE Timer1 模块门控信号

6.5.3.4 端口 P26 功能配置寄存器(P26_SEL)

偏移地址：0x98

复位值：0x0000 0000



位	标记	功能描述
31:3	Reserved	
2:0	P26_sel	端口 P26 功能选择. 000: GPIO P26 001: SPI_MOSI SPI 模块主机输出从机输入数据信号 010: TIM4_CHA Advanced Timer 模块通道 0 A 信号 011: TIM5_CHB Advanced Timer 模块通道 1 B 信号 100: PCA_CH2 PCA 模块通道 2 捕获/比较信号 101: LPUART_TXD LPUART 模块 TXD 信号 110: I2C_SCL I2C 模块时钟信号 111: TIM1_EXT Timer1 模块外部时钟输入信号

6.5.3.5 端口 P27 功能配置寄存器(P27_SEL)

偏移地址：0x9C

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P27_sel	端口 P27 功能选择. 000: GPIO P27 001: SPI_MOSI SPI 模块主机输入从机输出数据信号 010: TIM5_CHA Advanced Timer 模块通道 1 A 信号 011: TIM6_CHA Advanced Timer 模块通道 2 A 信号 100: PCA_CH3 PCA 模块通道 3 捕获/比较信号 101: UART0_RXD UART0 模块 RXD 信号 110: RCH_OUT 内部 24M RC 时钟输出信号 111: XTH_OUT 外部 32M 晶振输出信号

6.5.3.6 端口 P2 输入输出配置寄存器(P2DIR)

偏移地址：0x180

复位值：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			
Reserved								IR7	IR6	IR5	IR4	IR3	Reserved			
Reserved								RW	RW	RW	RW	RW	Reserved			

位	标记	功能描述
31:8	Reserved	
7	P2DIR7	端口 P27 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
6	P2DIR6	端口 P26 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
5	P2DIR5	端口 P25 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
4	P2DIR4	端口 P24 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
3	P2DIR3	端口 P23 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
2:0	Reserved	

6.5.3.7 端口 P2 输入值寄存器(P2IN)

偏移地址：0x184

复位值：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								P2I	P2I	P2I	P2I	P2I	Reserved		
								N7	N6	N5	N4	N3			
								RO	RO	RO	RO	RO			

位	标记	功能描述
31:8	Reserved	
7	P2IN7	端口 P27 输入值寄存器 1: 输入为高电平 0: 输入为低电平
6	P2IN6	端口 P26 输入值寄存器 1: 输入为高电平 0: 输入为低电平
5	P2IN5	端口 P25 输入值寄存器 1: 输入为高电平 0: 输入为低电平
4	P2IN4	端口 P24 输入值寄存器 1: 输入为高电平 0: 输入为低电平
3	P2IN3	端口 P23 输入值寄存器 1: 输入为高电平 0: 输入为低电平
2:0	Reserved	

6.5.3.8 端口 P2 输出值配置寄存器(P2OUT)

偏移地址：0x188

复位值：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								P2O	P2O	P2O	P2O	P2O	Reserved			
								UT7	UT6	UT5	UT4	UT3				
								RW	RW	RW	RW	RW				

位	标记	功能描述
31:8	Reserved	
7	P2OUT7	端口 P27 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
6	P2OUT6	端口 P26 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
5	P2OUT5	端口 P25 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
4	P2OUT4	端口 P24 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
3	P2OUT3	端口 P23 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
2:0	Reserved	

6.5.3.9 端口 P2 数模配置寄存器(P2ADS)

偏移地址：0x18C

复位值：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								P2A DS7	P2A DS6	P2A DS5	P2A DS4	P2A DS3	Reserved			
								RW	RW	RW	RW	RW				

位	标记	功能描述
31:8	Reserved	
7	P2ADS7	端口 P27 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
6	P2ADS6	端口 P26 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
5	P2ADS5	端口 P25 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
4	P2ADS4	端口 P24 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
3	P2ADS3	端口 P23 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
2:0	Reserved	

6.5.3.10 端口 P2 驱动能力配置寄存器(P2DR)

偏移地址：0x19C

复位值：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								P2D	P2D	P2D	P2D	P2D	Reserved			
Reserved								R7	R6	R5	R4	R3	Reserved			
Reserved								RW	RW	RW	RW	RW	Reserved			

位	标记	功能描述
31:8	Reserved	
7	P2DR7	端口 P27 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
6	P2DR6	端口 P26 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
5	P2DR5	端口 P25 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
4	P2DR4	端口 P24 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
3	P2DR3	端口 P23 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
2:0	Reserved	

6.5.3.11 端口 P2 上拉使能配置寄存器(P2PU)

偏移地址: 0x1A0

复位值: 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								P2P	P2P	P2P	P2P	P2P	Reserved			
								U7	U6	U5	U4	U3				
								RW	RW	RW	RW	RW				

位	标记	功能描述
31:8	Reserved	
7	P2PU7	端口 P27 上拉使能配置寄存器 1: 使能 0: 禁止
6	P2PU6	端口 P26 上拉使能配置寄存器 1: 使能 0: 禁止
5	P2PU5	端口 P25 上拉使能配置寄存器 1: 使能 0: 禁止
4	P2PU4	端口 P24 上拉使能配置寄存器 1: 使能 0: 禁止
3	P2PU3	端口 P23 上拉使能配置寄存器 1: 使能 0: 禁止
2:0	Reserved	

6.5.3.12 端口 P2 下拉使能配置寄存器(P2PD)

偏移地址：0x1A4

复位值：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								P2P	P2P	P2P	P2P	P2P	Reserved			
								D7	D6	D5	D4	D3				
								RW	RW	RW	RW	RW				

位	标记	功能描述
31:8	Reserved	
7	P2PD7	端口 P27 下拉使能配置寄存器 1: 使能 0: 禁止
6	P2PD6	端口 P26 下拉使能配置寄存器 1: 使能 0: 禁止
5	P2PD5	端口 P25 下拉使能配置寄存器 1: 使能 0: 禁止
4	P2PD4	端口 P24 下拉使能配置寄存器 1: 使能 0: 禁止
3	P2PD3	端口 P23 下拉使能配置寄存器 1: 使能 0: 禁止
2:0	Reserved	

6.5.3.13 端口 P2 开漏输出配置寄存器(P2OD)

偏移地址：0x1AC

复位值：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								P2O	P2O	P2O	P2O	P2O	Reserved		
Reserved								D7	D6	D5	D4	D3			
Reserved								RW	RW	RW	RW	RW			

位	标记	功能描述
31:8	Reserved	
7	P2OD7	端口 P27 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
6	P2OD6	端口 P26 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
5	P2OD5	端口 P25 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
4	P2OD4	端口 P24 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
3	P2OD3	端口 P23 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
2:0	Reserved	

6.5.3.14 端口 P2 高电平中断使能配置寄存器(P2HIE)

偏移地址：0x1B0

复位值：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								P2H	P2H	P2H	P2H	P2H	Reserved		
								IE7	IE6	IE5	IE4	IE3			
								RW	RW	RW	RW	RW			

位	标记	功能描述
31:8	Reserved	
7	P2HIE7	端口 P27 高电平中断使能配置寄存器 1: 使能 0: 禁止
6	P2HIE6	端口 P26 高电平中断使能配置寄存器 1: 使能 0: 禁止
5	P2HIE5	端口 P25 高电平中断使能配置寄存器 1: 使能 0: 禁止
4	P2HIE4	端口 P24 高电平中断使能配置寄存器 1: 使能 0: 禁止
3	P2HIE3	端口 P23 高电平中断使能配置寄存器 1: 使能 0: 禁止
2:0	Reserved	

6.5.3.15 端口 P2 低电平中断使能配置寄存器(P2LIE)

偏移地址：0x1B4

复位值：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								P2L	P2L	P2L	P2L	P2L	Reserved		
								IE7	IE6	IE5	IE4	IE3			
								RW	RW	RW	RW	RW			

位	标记	功能描述
31:8	Reserved	
7	P2LIE7	端口 P27 低电平中断使能配置寄存器 1: 使能 0: 禁止
6	P2LIE6	端口 P26 低电平中断使能配置寄存器 1: 使能 0: 禁止
5	P2LIE5	端口 P25 低电平中断使能配置寄存器 1: 使能 0: 禁止
4	P2LIE4	端口 P24 低电平中断使能配置寄存器 1: 使能 0: 禁止
3	P2LIE3	端口 P23 低电平中断使能配置寄存器 1: 使能 0: 禁止
2:0	Reserved	

6.5.3.16 端口 P2 上升沿中断使能配置寄存器(P2RIE)

偏移地址：0x1B8

复位值：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								P2R	P2R	P2R	P2R	P2R	Reserved		
								IE7	IE6	IE5	IE4	IE3			
								RW	RW	RW	RW	RW			

位	标记	功能描述
31:8	Reserved	
7	P2RIE7	端口 P27 上升沿中断使能配置寄存器 1: 使能 0: 禁止
6	P2RIE6	端口 P26 上升沿中断使能配置寄存器 1: 使能 0: 禁止
5	P2RIE5	端口 P25 上升沿中断使能配置寄存器 1: 使能 0: 禁止
4	P2RIE4	端口 P24 上升沿中断使能配置寄存器 1: 使能 0: 禁止
3	P2RIE3	端口 P23 上升沿中断使能配置寄存器 1: 使能 0: 禁止
2:0	Reserved	

6.5.3.17 端口 P2 下降沿中断使能配置寄存器(P2FIE)

偏移地址：0x1BC

复位值：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								P2F	P2F	P2F	P2F	P2F	Reserved		
								IE7	IE6	IE5	IE4	IE3			
								RW	RW	RW	RW	RW			

位	标记	功能描述
31:8	Reserved	
7	P2FIE7	端口 P27 下降沿中断使能配置寄存器 1: 使能 0: 禁止
6	P2FIE6	端口 P26 下降沿中断使能配置寄存器 1: 使能 0: 禁止
5	P2FIE5	端口 P25 下降沿中断使能配置寄存器 1: 使能 0: 禁止
4	P2FIE4	端口 P24 下降沿中断使能配置寄存器 1: 使能 0: 禁止
3	P2FIE3	端口 P23 下降沿中断使能配置寄存器 1: 使能 0: 禁止
2:0	Reserved	

6.5.3.18 端口 P2 中断状态寄存器(P2_STAT)

偏移地址：0x280

复位值：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								P2S	P2S	P2S	P2S	P2S	Reserved		
Reserved								TA7	TA6	TA5	TA4	TA3	Reserved		
Reserved								RO	RO	RO	RO	RO	Reserved		

位	标记	功能描述
31:8	Reserved	
7	P2STA7	端口 P27 中断状态寄存器 1: 中断触发 0: 无中断触发
6	P2STA6	端口 P26 中断状态寄存器 1: 中断触发 0: 无中断触发
5	P2STA5	端口 P25 中断状态寄存器 1: 中断触发 0: 无中断触发
4	P2STA4	端口 P24 中断状态寄存器 1: 中断触发 0: 无中断触发
3	P2STA3	端口 P23 中断状态寄存器 1: 中断触发 0: 无中断触发
2:0	Reserved	

6.5.3.19 端口 P2 中断清除寄存器(P2_ICLR)

偏移地址：0x290

复位值：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								P2C	P2C	P2C	P2C	P2C	Reserved		
Reserved								LR7	LR6	LR5	LR4	LR3	Reserved		
Reserved								RW	RW	RW	RW	RW	Reserved		

位	标记	功能描述
31:8	Reserved	
7	P2CLR7	端口 P27 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
6	P2CLR6	端口 P26 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
5	P2CLR5	端口 P25 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
4	P2CLR4	端口 P24 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
3	P2CLR3	端口 P23 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
2:0	Reserved	

6.5.4 端口 P3

6.5.4.1 端口 P31 功能配置寄存器(P31_SEL)

偏移地址: 0xC4

复位值: 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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P31_sel	端口 P31 功能选择. 000: GPIO P31 001: LPTIM_TOG Timer3 模块翻转信号 010: PCA_ECI PCA 模块外部时钟输入信号 011: PCLK_OUT APB 总线时钟输出信号 100: VC0_OUT VC0 模块输出 101: UART0_TXD UART0 模块 TXD 信号 110: RCL_OUT 内部 38K RC 时钟输出信号 111: HCLK_OUT AHB 总线时钟输出信号

6.5.4.2 端口 P32 功能配置寄存器(P32_SEL)

偏移地址：0xC8

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P32_sel	端口 P32 功能选择. 000: GPIO P32 001: LPTIM_TOGN Timer3 模块翻转信号的反向信号 010: PCA_CH2 PCA 模块通道 2 捕获/比较信号 011: TIM6_CHB Advanced Timer 模块通道 2 B 信号 100: VC1_OUT VC1 模块输出 101: UART1_TXD UART1 模块 TXD 信号 110: PCA_CH4 PCA 模块通道 4 捕获/比较信号 111: RTC_1Hz RTC 模块 1Hz 输出信号

6.5.4.3 端口 P33 功能配置寄存器(P33_SEL)

偏移地址：0xCC

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P33_sel	端口 P33 功能选择. 000: GPIO P33 001: LPUART_RXD LPUART 模块 RXD 信号 010: PCA_CH1 PCA 模块通道 1 捕获/比较信号 011: TIM5_CHB Advanced Timer 模块通道 1 B 信号 100: PCA_ECI PCA 模块外部时钟输入信号 101: UART1_RXD UART1 模块 RXD 信号 110: XTL_OUT 外部 32K 晶振输出信号 111: TIM1_TOGN Timer1 模块翻转信号的反向信号

6.5.4.4 端口 P34 功能配置寄存器(P34_SEL)

偏移地址: 0xD0

复位值: 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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P34_sel	端口 P34 功能选择. 000: GPIO P34 001: PCA_CH0 PCA 模块通道 0 捕获/比较信号 010: LPUART_TXD LPUART 模块 TXD 信号 011: TIM5_CHA Advanced Timer 模块通道 1 A 信号 100: TIM0_EXT Timer0 模块外部时钟输入信号 101: TIM4_CHA Advanced Timer 模块通道 0 A 信号 110: RTC_1Hz RTC 模块 1Hz 输出信号 111: TIM1_TOG Timer1 模块翻转信号

6.5.4.5 端口 P35 功能配置寄存器(P35_SEL)

偏移地址：0xD4

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P35_sel	端口 P35 功能选择. 000: GPIO P35 001: UART1_TXD UART1 模块 TXD 信号 010: TIM6_CHB Advanced Timer 模块通道 2 B 信号 011: UART0_TXD UART0 模块 TXD 信号 100: TIM0_GATE Timer0 模块门控信号 101: TIM4_CHB Advanced Timer 模块通道 0 B 信号 110: SPI_MOSI SPI 模块主机输入从机输出数据信号 111: I2C_SDA I2C 模块数据信号

6.5.4.6 端口 P36 功能配置寄存器(P36_SEL)

偏移地址：0xD8

复位值：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		
Reserved													RW		

位	标记	功能描述
31:3	Reserved	
2:0	P36_sel	端口 P36 功能选择. 000: GPIO P36 001: UART1_RXD UART1 模块 RXD 信号 010: TIM6_CHA Advanced Time r 模块通道 2 A 信号 011: UART0_RXD UART0 模块 RXD 信号 100: PCA_CH4 PCA 模块通道 4 捕获/比较信号 101: TIM5_CHA Advanced Timer 模块通道 1 A 信号 110: SPI_MOSI SPI 模块主机输出从机输入数据信号 111: I2C_SCL I2C 模块时钟信号

6.5.4.7 端口 P3 输入输出配置寄存器(P3DIR)

偏移地址：0x1C0

复位值：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	P3D	P3D	P3D	P3D	P3D	Res
									IR6	IR5	IR4	IR3	IR2	IR1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3DIR6	端口 P36 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
5	P3DIR5	端口 P35 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
4	P3DIR4	端口 P34 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
3	P3DIR3	端口 P33 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
2	P3DIR2	端口 P32 输入输出配置寄存器 1: 配置成输入 0: 配置成输出
1	P3DIR1	端口 P31 输入输出配置寄存器

		1: 配置成输入 0: 配置成输出
0	Reserved	

6.5.4.8 端口 P3 输入值寄存器(P3IN)

偏移地址：0x1C4

复位值：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	

位	标记	功能描述
31:7	Reserved	
6	P3IN6	端口 P36 输入值寄存器 1: 输入为高电平 0: 输入为低电平
5	P3IN5	端口 P35 输入值寄存器 1: 输入为高电平 0: 输入为低电平
4	P3IN4	端口 P34 输入值寄存器 1: 输入为高电平 0: 输入为低电平
3	P3IN3	端口 P33 输入值寄存器 1: 输入为高电平 0: 输入为低电平
2	P3IN2	端口 P32 输入值寄存器 1: 输入为高电平 0: 输入为低电平
1	P3IN1	端口 P31 输入值寄存器

		1: 输入为高电平 0: 输入为低电平
0	Reserved	

6.5.4.9 端口 P3 输出值配置寄存器(P3OUT)

偏移地址：0x1C8

复位值：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	P3O	P3O	P3O	P3O	P3O	Res
										UT6	UT5	UT4	UT3	UT2	UT1	
										RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3OUT6	端口 P36 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
5	P3OUT5	端口 P35 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
4	P3OUT4	端口 P34 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
3	P3OUT3	端口 P33 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
2	P3OUT2	端口 P32 输出值配置寄存器 1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
1	P3OUT1	端口 P31 输出值配置寄存器

		1: 输出高电平。如果配置成开漏输出，则需外部上拉电阻来拉高。 0: 输出低电平。
0	Reserved	

6.5.4.10 端口 P3 数模配置寄存器(P3ADS)

偏移地址：0x1CC

复位值：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	P3A	P3A	P3A	P3A	P3A	Res
										DS6	DS5	DS4	DS3	DS2	DS1	
										RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3ADS6	端口 P36 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
5	P3ADS5	端口 P35 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
4	P3ADS4	端口 P34 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
3	P3ADS3	端口 P33 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
2	P3ADS2	端口 P32 数模配置寄存器 1: 配置为模拟端口 0: 配置为数字端口
1	P3ADS1	端口 P31 数模配置寄存器

		1: 配置为模拟端口 0: 配置为数字端口
0	Reserved	

6.5.4.11 端口 P3 驱动能力配置寄存器(P3DR)

偏移地址：0x1DC

复位值：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	P3D	P3D	P3D	P3D	P3D	Res
									R6	R5	R4	R3	R2	R1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3DR6	端口 P36 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
5	P3DR5	端口 P35 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
4	P3DR4	端口 P34 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
3	P3DR3	端口 P33 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
2	P3DR2	端口 P32 驱动能力配置寄存器 1: 低驱动能力 0: 高驱动能力
1	P3DR1	端口 P31 驱动能力配置寄存器

		1: 低驱动能力 0: 高驱动能力
0	Reserved	

6.5.4.12 端口 P3 上拉使能配置寄存器(P3PU)

偏移地址：0x1E0

复位值：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	P3P	P3P	P3P	P3P	P3P	Res
									U6	U5	U4	U3	U2	U1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3PU6	端口 P36 上拉使能配置寄存器 1: 使能 0: 禁止
5	P3PU5	端口 P35 上拉使能配置寄存器 1: 使能 0: 禁止
4	P3PU4	端口 P34 上拉使能配置寄存器 1: 使能 0: 禁止
3	P3PU3	端口 P33 上拉使能配置寄存器 1: 使能 0: 禁止
2	P3PU2	端口 P32 上拉使能配置寄存器 1: 使能 0: 禁止
1	P3PU1	端口 P31 上拉使能配置寄存器

		1: 使能 0: 禁止
0	Reserved	

6.5.4.13 端口 P3 下拉使能配置寄存器(P3PD)

偏移地址：0x1E4

复位值：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	P3P	P3P	P3P	P3P	P3P	Res
									D6	D5	D4	D3	D2	D1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3PD6	端口 P36 下拉使能配置寄存器 1: 使能 0: 禁止
5	P3PD5	端口 P35 下拉使能配置寄存器 1: 使能 0: 禁止
4	P3PD4	端口 P34 下拉使能配置寄存器 1: 使能 0: 禁止
3	P3PD3	端口 P33 下拉使能配置寄存器 1: 使能 0: 禁止
2	P3PD2	端口 P32 下拉使能配置寄存器 1: 使能 0: 禁止
1	P3PD1	端口 P31 下拉使能配置寄存器

		1: 使能 0: 禁止
0	Reserved	

6.5.4.14 端口 P3 开漏输出配置寄存器(P3OD)

偏移地址：0x1EC

复位值：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	P3O	P3O	P3O	P3O	P3O	Res
									D6	D5	D4	D3	D2	D1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3OD6	端口 P36 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
5	P3OD5	端口 P35 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
4	P3OD4	端口 P34 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
3	P3OD3	端口 P33 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
2	P3OD2	端口 P32 开漏输出配置寄存器 1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
1	P3OD1	端口 P31 开漏输出配置寄存器

		1: 将端口输出方式设置为开漏输出 0: 将端口输出方式设置为推挽输出
0	Reserved	

6.5.4.15 端口 P3 高电平中断使能配置寄存器(P3HIE)

偏移地址：0x1F0

复位值：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	

位	标记	功能描述
31:7	Reserved	
6	P3HIE6	端口 P36 高电平中断使能配置寄存器 1: 使能 0: 禁止
5	P3HIE5	端口 P35 高电平中断使能配置寄存器 1: 使能 0: 禁止
4	P3HIE4	端口 P34 高电平中断使能配置寄存器 1: 使能 0: 禁止
3	P3HIE3	端口 P33 高电平中断使能配置寄存器 1: 使能 0: 禁止
2	P3HIE2	端口 P32 高电平中断使能配置寄存器 1: 使能 0: 禁止
1	P3HIE1	端口 P31 高电平中断使能配置寄存器

		1: 使能 0: 禁止
0	Reserved	

6.5.4.16 端口 P3 低电平中断使能配置寄存器(P3LIE)

偏移地址：0x1F4

复位值：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	

位	标记	功能描述
31:7	Reserved	
6	P3LIE6	端口 P36 低电平中断使能配置寄存器 1: 使能 0: 禁止
5	P3LIE5	端口 P35 低电平中断使能配置寄存器 1: 使能 0: 禁止
4	P3LIE4	端口 P34 低电平中断使能配置寄存器 1: 使能 0: 禁止
3	P3LIE3	端口 P33 低电平中断使能配置寄存器 1: 使能 0: 禁止
2	P3LIE2	端口 P32 低电平中断使能配置寄存器 1: 使能 0: 禁止
1	P3LIE1	端口 P31 低电平中断使能配置寄存器

		1: 使能 0: 禁止
0	Reserved	

6.5.4.17 端口 P3 上升沿中断使能配置寄存器(P3RIE)

偏移地址：0x1F8

复位值：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	P3R	P3R	P3R	P3R	P3R	Res
									IE6	IE5	IE4	IE3	IE2	IE1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3RIE6	端口 P36 上升沿中断使能配置寄存器 1: 使能 0: 禁止
5	P3RIE5	端口 P35 上升沿中断使能配置寄存器 1: 使能 0: 禁止
4	P3RIE4	端口 P34 上升沿中断使能配置寄存器 1: 使能 0: 禁止
3	P3RIE3	端口 P33 上升沿中断使能配置寄存器 1: 使能 0: 禁止
2	P3RIE2	端口 P32 上升沿中断使能配置寄存器 1: 使能 0: 禁止
1	P3RIE1	端口 P31 上升沿中断使能配置寄存器

		1: 使能 0: 禁止
0	Reserved	

6.5.4.18 端口 P3 下降沿中断使能配置寄存器(P3FIE)

偏移地址：0x1FC

复位值：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	

位	标记	功能描述
31:7	Reserved	
6	P3FIE6	端口 P36 下降沿中断使能配置寄存器 1: 使能 0: 禁止
5	P3FIE5	端口 P35 下降沿中断使能配置寄存器 1: 使能 0: 禁止
4	P3FIE4	端口 P34 下降沿中断使能配置寄存器 1: 使能 0: 禁止
3	P3FIE3	端口 P33 下降沿中断使能配置寄存器 1: 使能 0: 禁止
2	P3FIE2	端口 P32 下降沿中断使能配置寄存器 1: 使能 0: 禁止
1	P3FIE1	端口 P31 下降沿中断使能配置寄存器

		1: 使能 0: 禁止
0	Reserved	

6.5.4.19 端口 P3 中断状态寄存器(P3_STAT)

偏移地址：0x2C0

复位值：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	

位	标记	功能描述
31:7	Reserved	
6	P3STA6	端口 P36 中断状态寄存器 1: 中断触发 0: 无中断触发
5	P3STA5	端口 P35 中断状态寄存器 1: 中断触发 0: 无中断触发
4	P3STA4	端口 P34 中断状态寄存器 1: 中断触发 0: 无中断触发
3	P3STA3	端口 P33 中断状态寄存器 1: 中断触发 0: 无中断触发
2	P3STA2	端口 P32 中断状态寄存器 1: 中断触发 0: 无中断触发
1	P3STA1	端口 P31 中断状态寄存器

		1: 中断触发 0: 无中断触发
0	Reserved	

6.5.4.20 端口 P3 中断清除寄存器(P3_ICLR)

偏移地址：0x2D0

复位值：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	P3C	P3C	P3C	P3C	P3C	Res
									LR6	LR5	LR4	LR3	LR2	LR1	
									RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:7	Reserved	
6	P3CLR6	端口 P36 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
5	P3CLR5	端口 P35 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
4	P3CLR4	端口 P34 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
3	P3CLR3	端口 P33 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
2	P3CLR2	端口 P32 中断清除寄存器 1: 保留中断标志位 0: 清除中断标志位
1	P3CLR1	端口 P31 中断清除寄存器

		1: 保留中断标志位 0: 清除中断标志位
0	Reserved	

6.5.5 端口辅助功能

6.5.5.1 端口辅助功能配置寄存器 1(GPIO_CTRL1)

偏移地址: 0x304

复位值: 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	hclk	pclk	hclk_sel	pclk_sel	ssn_sel			ext_clk_sel						
	ol	_en	_en												
	RW	RW	RW	RW	RW	RW			RW						

位	标记	功能描述
31:15	Reserved	
14	ir_pol	IR 输出极性选择. 0 – 正向输出 1 – 反向输出
13	hclk_en	hclk 输出控制. 0 – 输出 0 1 – 输出 hclk
12	pclk_en	pclk 输出门控. 0 – 输出 0 1 – 输出 pclk
11:10	hclk_sel	hclk 输出分频选择 00: hclk 01: hclk/2 10: hclk/4 11: hclk/8

9:8	pclk_sel	pclk 输出分频选择 00: hclk 01: hclk/2 10: hclk/4 11: hclk/8
7:4	ssn_sel	SPI SSN 信号来源选择 0000: 高电平 1000: P23 0001: P35 1001: P24 0010: P36 1010: P25 0011: P01 1011: P26 0100: P02 1100: P27 0101: P03 1101: P31 0110: P15 1110: P32 0111: P14 1111: P33
3:0	ext_clk_sel	外部时钟信号来源选择 0000: 高电平 1000: P23 0001: P35 1001: P24 0010: P36 1010: P25 0011: P01 1011: P26 0100: P02 1100: P27 0101: P03 1101: P31 0110: P15 1110: P32 0111: P14 1111: P33

6.5.5.2 端口辅助功能配置寄存器 2(GPIO_CTRL2)

偏移地址：0x308

复位值：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	pca_cap3	pca_cap2	pca_cap1	pca_cap0						
						_sel	_sel	_sel	_sel	_sel						
						RW	RW	RW	RW	RW						

位	标记	功能描述
31:10	Reserved	
9:8	pca_cap4_sel	PCA 捕获通道 4 信号来源选择 00: PCA_CH4 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
7:6	pca_cap3_sel	PCA 捕获通道 3 信号来源选择 00: PCA_CH3 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
5:4	pca_cap2_sel	PCA 捕获通道 2 信号来源选择 00: PCA_CH2 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
3:2	pca_cap1_sel	PCA 捕获通道 1 信号来源选择

		00: PCA_CH1 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
1:0	pca_cap0_sel	PCA 捕获通道 0 信号来源选择 00: PCA_CH0 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD

6.5.5.3 端口辅助功能配置寄存器 3(GPIO_CTRL3)

偏移地址：0x30C

复位值：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						

位	标记	功能描述
31:12	Reserved	
11:10	tm6_a_sel	Timer6 A 通道信号来源选择 00: TIM6_CHA 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
9:8	tm5_a_sel	Timer5 A 通道信号来源选择 00: TIM5_CHA 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
7:6	tm4_a_sel	Timer4 A 通道信号来源选择 00: TIM4_CHA 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
5:4	tm6_b_sel	Timer6 B 通道信号来源选择 00: TIM6_CHB

		01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
3:2	tm5_b_sel	Timer5 B 通道信号来源选择 00: TIM5_CHB 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
1:0	tm4_b_sel	Timer4 B 通道信号来源选择 00: TIM4_CHB 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD

6.5.5.4 端口辅助功能配置寄存器 4(GPIO_CTRL4)

偏移地址：0x310

复位值：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_	tm2_gate_	tm1_gate_	tm0_gate_				
								sel	sel	sel	sel				
								RW	RW	RW	RW				

位	标记	功能描述
31:8	Reserved	
7:6	tm3_gate_sel	Timer3 门控输入信号来源选择 00: LPTIM_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
5:4	tm2_gate_sel	Timer2 门控输入信号来源选择 00: TIM2_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
3:2	tm1_gate_sel	Timer1 门控输入信号来源选择 00: TIM1_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD
1:0	tm0_gate_sel	Timer0 门控输入信号来源选择

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

7 FLASH 控制器 (FLASH)

7.1 概述

本设备包含一颗 32kByte 容量的 FLASH 存储器，共划分为 64 个 Sector，每个 Sector 容量为 512Byte。本模块支持对该存储器的擦除、编程以及读取操作。此外，本模块支持对 FLASH 存储器擦写的保护，以及控制寄存器的写保护。

7.2 结构框图

地址范围	Sector 序号		地址范围	Sector 序号
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

图 7-1 存储器 Sector 划分

7.3 功能描述

本控制器支持三种位宽模式 FLASH 读写：Byte（8bits）、Half-word（16bits）、Word（32bits）。注意，Byte 操作的地址必须按 Byte 对齐，Half-word 操作的目标地址必须按 Half-word 对齐（地址的最低比特为 0），Word 操作的地址必须按 Word 对齐（地址最低两比特为 0）。如果读写操作的地址没有按照上述方式进行对齐，系统会进入 Hard Fault 出错中断。

7.3.1 页擦除（Sector Erase）

页擦除每次可以擦除用户指定的一个页（Sector）。擦除操作完成后，页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH_CR.BUSY 变为 0）；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH_CR.BUSY 变为 0）。

页（Sector）擦除操作步骤如下：

Step1: 配置 FLASH 擦写参数，详见擦写时序章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 2，设置 Flash 操作模式为 Sector 擦除。

Step4: 检查 FLASH_CR.OP 是否为 2，如不为 2 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除该 Sector 的擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8: 对待擦除的 Sector 内的任意地址进行写入任意数据，触发 Sector 擦除。

例： * ((unsigned char *) 0x00000200) = 0x00。

Step9: 等待 FLASH_CR.BUSY 变为 0，Sector 擦除操作完成。

Step10: 如需擦除其它 Sector，重复 Step5 – Step9。

7.3.2 全片擦除 (Chip Erase)

全片擦除可以一次性擦除全部的页 (Sector)。擦除操作完成后, 所有页 (Sector) 内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行, 则 CPU 会停止取指, 硬件自动等待该操作完成 (FLASH_CR. BUSY 变为 0); 如果该擦除操作是从 RAM 内执行, 则 CPU 不会停止取指, 用户软件应等待该操作完成 (FLASH_CR. BUSY 变为 0)。

全片擦除操作步骤如下:

Step1: 配置 FLASH 擦写参数, 详见擦写时序章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 3, 设置 Flash 操作模式为 Chip 擦除。

Step4: 检查 FLASH_CR.OP 是否为 3, 如不为 3 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step6: 设置 FLASH_SLOCK 为 0xFFFF, 去除所有 Sector 的擦写保护。

Step7: 检查 FLASH_SLOCK 是否为 0xFFFF, 如不为 0xFFFF 则跳转到 Step5。

Step8: 对待擦除的 Chip 内的任意地址进行写操作, 触发 Chip 擦除。

例: * ((unsigned char *) 0x00000000) = 0x00。

Step9: 等待 FLASH_CR. BUSY 变为 0, Chip 擦除操作完成。

7.3.3 写操作 (Program)

写操作只能将 FLASH 内的比特数据由 1 写成 0, 故写入数据前应确保待写入的地址内的数据为 0xFF。支持写入 3 种数据长度: Byte (8bits)、Half-word (16bits)、Word (32bits), 写入的数据以小端模式存放在 FLASH 中, 即低地址存放数据的低字节。如果写操作是从 FLASH 内执行, 则 CPU 会停止取指, 硬件自动等待该操作完成 (FLASH_CR. BUSY 变为 0); 如果该写操作是从 RAM 内执行, 则 CPU 不会停止取指, 用户软件应等待该操作完成 (FLASH_CR. BUSY 变为 0)。

Byte 写操作步骤如下:

Step1: 配置 FLASH 擦写参数, 详见擦写时序章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 1, 设置 Flash 操作模式为写入。

Step4: 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5。

Step8: 对待写入的目标地址进行 Byte 写操作，触发写入操作。

例: $* ((\text{unsigned char } *) 0x00001231) = 0x5A$ 。

Step9: 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

Step10: 如需写 Byte 到已去除擦写保护的其它地址，重复 Step8 – Step9。

Half-word 写操作步骤如下：

Step1: 配置 FLASH 擦写时间，详见擦写时序章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入。

Step4: 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5

Step8: 对待写入的目标地址进行 Half-word 写操作，触发写入操作。

例: $* ((\text{unsigned short } *) 0x00001232) = 0xABCD$ 。

Step9: 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

Step10: 如需写 Half-word 到已去除擦写保护的其它地址，重复 Step8 – Step9。

Word 写操作步骤如下：

Step1: 配置 FLASH 擦写参数，详见擦写时序章节。

Step2: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3: 配置 FLASH_CR.OP 为 1，设置 Flash 操作模式为写入。

Step4: 检查 FLASH_CR.OP 是否为 1，如不为 1 则跳转到 Step2。

Step5: 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step6: 设置 FLASH_SLOCK 相应的比特为 1，去除擦写保护。

Step7: 检查 FLASH_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step5

Step8: 对待写入的目标地址进行 Word 写操作，触发写入操作。

例: $* ((\text{unsigned long } *) 0x00001234) = 0x55667788。$

Step9: 等待 FLASH_CR.BUSY 变为 0，写入操作完成。

Step10: 如需写 Word 到已去除擦写保护的其它地址，重复 Step8 – Step9。

7.3.4 读操作

支持读出 3 种数据长度：Byte（8bits）、Half-word（16bits）、Word（32bits），读出的数据为小端模式，即低地址存放数据的低字节。读操作无需操作步骤，任何时刻都可以读出 FLASH 内的数据。

Byte 读操作示例: $\text{temp} = * ((\text{unsigned char } *) 0x00001231)$

Half-word 读操作示例 $\text{temp} = * ((\text{unsigned int } *) 0x00001232)$

Word 读操作示例 $\text{temp} = * ((\text{unsigned long } *) 0x00001234)$

7.4 擦写时序

FLASH 存储器对擦除和编程操作的控制信号具有严格的时间要求，控制信号的时序不合格会造成擦除操作和编程操作失败。上电时默认装载了 HCLK 为 4MHz 时的擦写参数；如果对 Flash 进行擦写时的 HCLK 频率不为 4MHz，则用户程序应装载 HCLK 频率所对应的擦写参数。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。如果 HCLK 由默认 4MHz 升高为 8MHz，则上述 FLASH_Tx 寄存器的值应该设为默认值的 2 倍，即保持当前 $T_{sysclk} * FLASH_Tx$ 的结果与默认值相等即可。

以下表格列出了不同频率下相应的 FLASH 擦写时序参数：

	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

表 7-1 不同频率下 FLASH 擦写时间参数

配置系统频率为 8MHz 时擦写参数的操作步骤如下：

- Step1:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step2:** 向 FLASH_TNVS 寄存器写入 0x40，如读出的该寄存器的值不为 0x40，则跳转到上一步。
- Step3:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step4:** 向 FLASH_TPGS 寄存器写入 0x2E，如读出的该寄存器的值不为 0x2E，则跳转到上一步。

- Step5:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step6:** 向 FLASH_TPROG 寄存器写入 0x36，如读出的该寄存器的值不为 0x36，则跳转到上一步。
- Step7:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step8:** 向 FLASH_TSERASE 寄存器写入 0x8CA0，如读出的该寄存器的值不为 0x8CA0，则跳转到上一步。
- Step9:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step10:** 向 FLASH_TMERASE 寄存器写入 0x445C0，如读出的该寄存器的值不为 0x445C0，则跳转到上一步。
- Step11:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step12:** 向 FLASH_TPRCV 寄存器写入 0x30，如读出的该寄存器的值不为 0x30，则跳转到上一步。
- Step13:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step14:** 向 FLASH_TSRCV 寄存器写入 0x1E0，如读出的该寄存器的值不为 0x1E0，则跳转到上一步。
- Step15:** 向 FLASH_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step16:** 向 FLASH_TMRCV 寄存器写入 0x7D0，如读出的该寄存器的值不为 0x7D0，则跳转到上一步。

7.5 读等待周期

本设备内置的 FLASH 支持的最快取指频率为 24MHz。当 HCLK 频率超过 24MHz 时，必须为 CPU 取指插入等待周期，即设置 FLASH_CR.WAIT 为 1。当插入等待周期时，CPU 每两个周期才读取一次 FLASH 存储器中的指令码。

7.6 擦写保护

7.6.1 擦写保护位

整个 32kByte FLASH 存储器被划分为 64 个 Sector，每 4 个 Sector 共用一个擦写保护位。当 Sector 被保护时，对该 Sector 进行的擦写操作均无效并产生报警标志位和中断信号。当 FLASH 存储器中的任意 Sector 被保护时，对该 FLASH 的 Chip 擦写无效，并产生报警标志位和中断信号。

7.6.2 PC 地址擦写保护

CPU 在 FLASH 中运行程序时，如果当前 PC 指针正好落在待擦写的 Sector 地址范围之内，那么该擦写操作无效并产生报警标志位和中断信号。。

7.7 寄存器写保护

本模块的重要控制器屏蔽普通的写操作，必须用写序列方式才能修改。

需要通过写序列方式才能更改的寄存器如下所示：

FLASH_TNVS、FLASH_TPGS、FLASH_TPROG、FLASH_TSERASE、FLASH_TMERASE、
FLASH_TPRCV、FLASH_TSRCV、FLASH_TMRCV、FLASH_CR、FLASH_SLOCK。

不需要通过写序列方式即可更改的寄存器如下所示：

FLASH_ICLR、FLASH_BYPASS。

通过写序列方式修改寄存器值的具体操作步骤如下所示：

Step1: 向 FLASH_BYPASS 寄存器写入 0x5A5A。

Step2: 向 FLASH_BYPASS 寄存器写入 0xA5A5。

Step3: 对待修改的寄存器写入目标值。

Step4: 验证待修改的寄存器的当前值是否与目标值相同，如不相同则跳转到 Step1。

Step5: 执行其它操作。

注意：

- 写 0x5a5a 和写 0xa5a5 这两步操作之间不可插入任何写操作，不可以被中断打断，否则该 Bypass 序列会失效，需要重新写入 0x5a5a-0xa5a5 序列。

7.8 寄存器

基地址：0x4002 0000

寄存器	偏移地址	描述
FLASH_TNVS	0x00	Tnvs 时间参数
FLASH_TPGS	0x04	Tpgs 时间参数
FLASH_TPROG	0x08	Tprog 时间参数
FLASH_TSERASE	0x0C	Tserase 时间参数
FLASH_TMERASE	0x10	Tmerase 时间参数
FLASH_TPRCV	0x14	Tprcv 时间参数
FLASH_TSRCV	0x18	Tsrcv 时间参数
FLASH_TMRCV	0x1C	Tmrcv 时间参数
FLASH_CR	0x20	控制寄存器
FLASH_IFR	0x24	中断标志寄存器
FLASH_ICLR	0x28	中断标志清除寄存器
FLASH_BYPASS	0x2C	0x5a5a-0xa5a5 Bypass 序列寄存器
FLASH_SLOCK	0x30	Sector 擦写保护寄存器

7.8.1 TNVS 参数寄存器 (FLASH_TNVS)

偏移地址：0x00

复位值：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							

位	标记	功能描述
31:9	Reserved	
8:0	TNVS	计算公式：TNVS = 8*HCLK，HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例：TNVS = 8*4 = 32。

7.8.2 TPGS 参数寄存器 (FLASH_TPGS)

偏移地址: 0x04

复位值: 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							

位	标记	功能描述
31:8	RESERVED	
7:0	TPGS	计算公式: $TPGS = 5.75 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TPGS = 5.75 * 4 = 23$ 。

7.8.3 TPROG 参数寄存器 (FLASH_TPROG)

偏移地址: 0x08

复位值: 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							

位	标记	功能描述
31:8	Reserved	
7:0	TPROG	计算公式: $TPROG = 6.75 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TPROG = 6.75 * 4 = 27$ 。

7.8.4 TSERASE 寄存器 (FLASH_TSERASE)

偏移地址: 0x0C

复位值: 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															

位	标记	功能描述
31:18	Reserved	
17:0	TSERASE	计算公式: $TSERASE = 4500 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TSERASE = 4500 * 4 = 18000$ 。

7.8.5 TMERASE 参数寄存器 (FLASH_TMERASE)

偏移地址: 0x10

复位值: 0x000222E0

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

位	标记	功能描述
31:21	Reserved	
20:0	TMERASE	计算公式: $TMERASE = 35000 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TMERASE = 35000 * 4 = 140000$ 。

7.8.6 TPRCV 参数寄存器 (FLASH_TPRCV)

偏移地址: 0x14

复位值: 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											

位	标记	功能描述
31:12	Reserved	
11:0	TPRCV	计算公式: $TPRCV = 6 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TPRCV = 6 * 4 = 24$ 。

7.8.7 TSRCV 参数寄存器 (FLASH_TSRCV)

偏移地址: 0x18

复位值: 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											

位	标记	功能描述
31:12	Reserved	
11:0	TSRCV	计算公式: $TSRCV = 60 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TSRCV = 60 * 4 = 240$ 。

7.8.8 TMRCV 参数寄存器 (FLASH_TMRCV)

偏移地址: 0x1C

复位值: 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												

位	标记	功能描述
31:13	RESERVED	
12:0	TMRCV	计算公式: $TMRCV = 250 * HCLK$, HCLK的单位为MHz。修改该寄存器值的方法详见7.7。 4MHz示例: $TMRCV = 250 * 4 = 1000$ 。

7.8.9 CR 寄存器 (FLASH_CR)

偏移地址: 0x20

复位值: 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	RO	RW	RW	

位	标记	功能描述
31:7	Reserved	
6:5	IE	IE[6]: FLASH 擦写被保护地址中断使能; 0: 不使能; 1: 使能 IE[5]: FLASH 擦写 PC 值中断使能; 0: 不使能; 1: 使能
4	BUSY	空闲/忙标志位; 0: 空闲状态; 1: 忙状态;
3	Reserved	
2	WAIT	读FLASH 等待周期; 0: 0等待周期; 1: 1个等待周期
1:0	OP	FLASH 操作; 00: read; 01: program; 10: sector erase; 11: chip erase
修改该寄存器值的方法详见7.7。		

7.8.10 IFR 寄存器 (FLASH_IFR)

偏移地址: 0x24

复位值: 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

位	标记	描述
31:2	Reserved	
1	IF1	擦写保护报警中断标志位
0	IF0	擦写 PC 地址报警中断标志位

7.8.11 ICLR 寄存器 (FLASH_ICLR)

偏移地址: 0x28

复位值: 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														ICLR 1	ICLR 0
R														RW0	RW0

位	标记	功能描述
31:4	Reserved	
3:2	Reserved	写无效, 读为0x3
1	ICLR1	清除保护报警中断标志位; 写0清除; 写1无效;
0	ICLR0	清除 PC 地址报警中断标志位; 写0清除; 写1无效;

7.8.12 BYPASS 寄存器 (FLASH_BYPASS)

偏移地址: 0x2C

复位值: 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															

位	标记	描述
31:16	Reserved	
15:0	BYSEQ	在修改本模块寄存器之前, 必须对 BYSEQ[15:0] 寄存器写入 0x5a5a-0xa5a5 序列。每次写入该 Bypass 序列后, 只可以修改一次寄存器。如需再次修改寄存器, 必须再次输入 0x5a5a-0xa5a5 序列。详见7.7。

7.8.13 SLOCK 寄存器 (FLASH_SLOCK)

偏移地址: 0x30

复位值: 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															

位	标记	描述
31:16	Reserved	
15:0	SLOCK	Sector 擦写保护位; 0: 不允许擦写; 1: 允许擦写 SLOCK[0] 对应: Sector0-1-2-3 SLOCK[1] 对应: Sector4-5-6-7 SLOCK[2] 对应: Sector8-9-10-11 SLOCK[3] 对应: Sector12-13-14-15 SLOCK[15] 对应: Sector60-61-62-63

8 RAM 控制器 (RAM)

8.1 概述

本产品包含一块容量为 2K/4K byte 的静态 SRAM, 支持 byte(8bits)、half-word(16bits)、word(32bits) 三种读写操作。可在 HCLK 频率下进行读写操作, 无须等待周期。此外, 本控制器还支持奇偶校验, 可对每个 byte 的 SRAM 数据进行奇偶校验, 并产生奇偶校验出错中断。

8.2 功能描述

读写位宽

本控制器支持 Byte (8bits)、Half-word (16bits)、Word (32bits) 三种位宽的读写操作。Byte 操作的地址必须按 Byte 对齐, Half-word 操作的目标地址必须按 Half-word 对齐 (地址最低位为 0), Word 操作的地址必须按 Word 对齐 (地址最低两位为 0)。如果读写操作的目标地址没有按照位宽规定对齐, 该操作无效, 并且系统会产生 Hard Fault 出错中断。

奇偶校验

本控制器支持 SRAM 数据的奇偶校验。在复位后, 该功能默认关闭。当打开奇偶校验功能后, 当对 SRAM 写数据时, 对每个 Byte 的数据做奇偶校验, 并把 1bit 校验值和 8bits 数据一起存入 SRAM 中。当对 SRAM 读数据时, 控制器会读取 8bits 数据和 1bit 校验值, 并做奇偶校验, 如果校验出错, 则置位奇偶校验出错标志位, 在中断使能情况下, 会产生出错中断。

注意:

- 在奇偶校验使能时, 在读取 SRAM 数据前, 必须初始化 SRAM, 否则可能会误触发奇偶校验报警标志位或者中断。

8.3 寄存器

基地址：0x4002 0400

寄存器	偏移地址	描述
RAM_CR	0x00	控制寄存器
RAM_ERRADDR	0x04	出错地址寄存器
RAM_IFR	0x08	出错中断标志寄存器
RAM_ICLR	0x0C	出错中断标志清除寄存器

8.3.1 控制寄存器（RAM_CR）

偏移地址：0x00

复位值：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
R														RW	RW

位	标记	功能描述
31:2	RESERVED	
1	IE	出错报警中断使能信号；1：使能报警中断，0：关闭报警中断；
0	CHKEN	奇偶校验使能信号；1：使能奇偶校验，0：关闭奇偶校验；

8.3.2 奇偶校验出错地址寄存器 (RAM_ERRADDR)

偏移地址: 0x04

复位值: 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											

位	标记	功能描述
31:12	Reserved	
11:0	ERRADDR	12bits 奇偶校验出错 byte 地址

8.3.3 出错中断标志寄存器 (RAM_IFR)

偏移地址: 0x08

复位值: 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

位	标记	功能描述
31:1	RESERVED	
0	ERR	奇偶校验出错标志位

8.3.4 出错中断标志清除寄存器 (RAM_ICLR)

偏移地址: 0x0C

复位值: 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															ERR CLR
R															RW0

位	标记	功能描述
31:1	Reserved	
0	ERRCLR	出错中断标志清除位; 写1: 无效, 写0: 清零

9 基本定时器 (TIM0/1/2)

9.1 基本定时器简介

基本定时器包含三个定时器 Timer0/1/2。Timer0/1/2 功能完全相同。Timer0/1/2 是同步定时/计数器，可以作为 16 位自动重载功能的定时/计数器，也可以作为 32 位无重载功能的定时/计数器。Timer0/1/2 可以对外部脉冲进行计数或者实现系统定时。

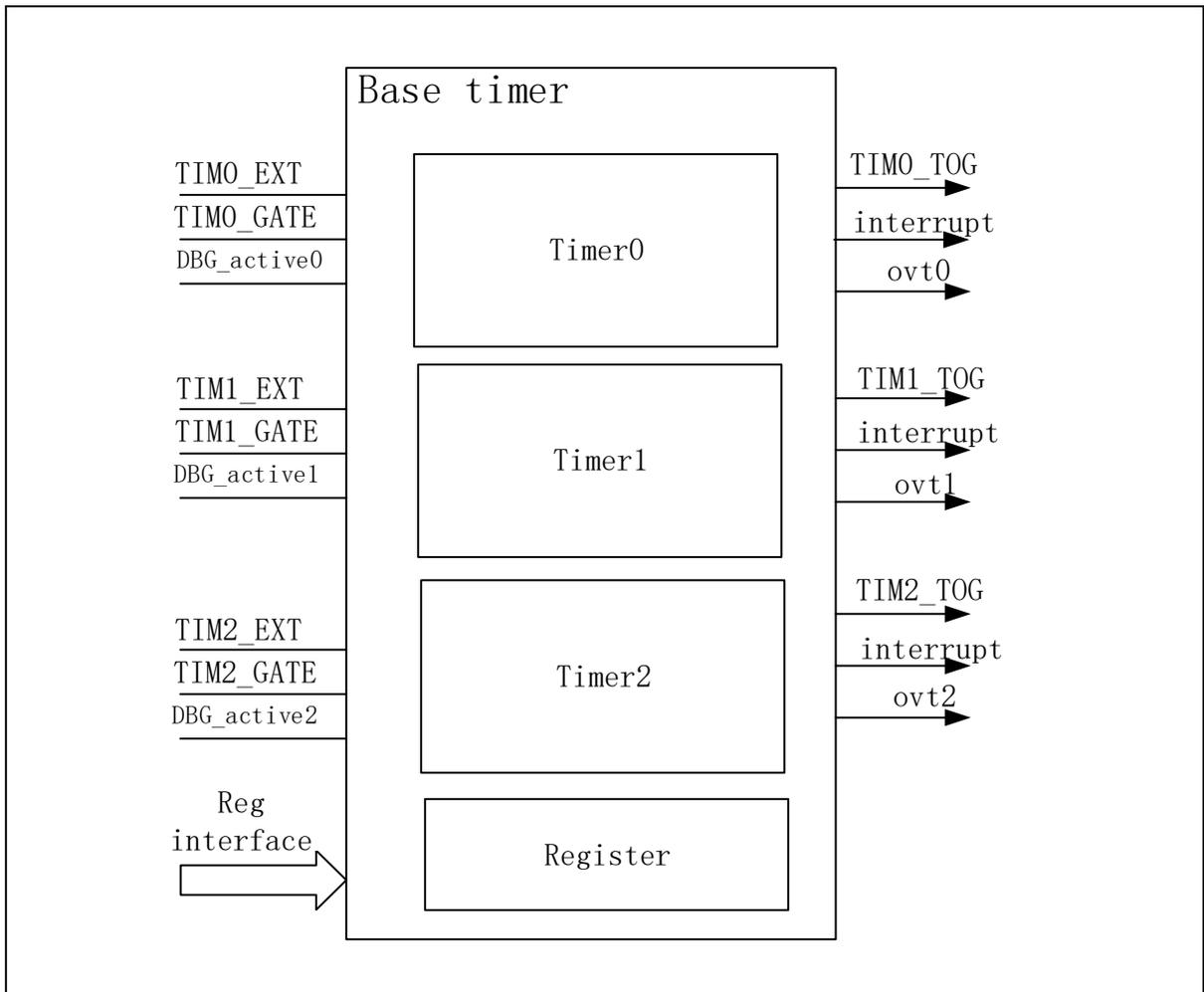


图 9-1 Base Timer 框图

9.2 Base Timer 功能描述

Timer0/1/2 每个定时/计数器都有独立的控制启动信号，以及外部输入时钟，门控信号。

Timer0/1/2 使用 EXT, GATE 来进行计数功能, EXT 用于计数器的外部输入时钟信号, GATE 用于有效电平计数使能信号。当门控功能使能后，当且仅当外部输入 GATE 电平有效时，计数器才会计数，否则计数器处于保持状态。门控使能使用 CR.gate 控制。默认门控功能关闭。门控电平选择使用 CR.GATE_P 控制。默认高电平为门控有效电平；设置为 1 后门控低电平为有效电平。

TIM0/1/2 使用 PCLK, GATE 来进行定时功能, PCLK 用于定时器的内部输入时钟信号, GATE 可用于有效电平定时使能信号。当门控功能使能后，当且仅当外部输入 GATE 电平有效时，定时器才会计数，否则定时器处于定时计数器停止状态。门控使能使用 CR.Gate 控制。默认门控功能关闭。门控电平选择使用 CR.Gate_P 控制。默认高电平为门控有效电平；设置为 1 后门控有效电平是低电平。定时功能可以配置预除频。CR.PRS 控制分频比。

PRS [2:0]	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	256

TIM0/1/2 的定时/计数器支持两种工作模式，通过设置定时器控制寄存器（CR）中 MD 选择工作模式。模式 1 为 32 位自由计数模式。模式 2 是 16 位重载模式。

32 位自由计数模式，计数到最大 0xFFFFFFFF 溢出后产生中断，定时/计数器变为 0X00000000;然后继续计数；16 位重载模式，计数到最大 0xFFFF 后溢出，产生中断，定时/计数器的值被装载为 ARR 的值，然后继续向上计数。

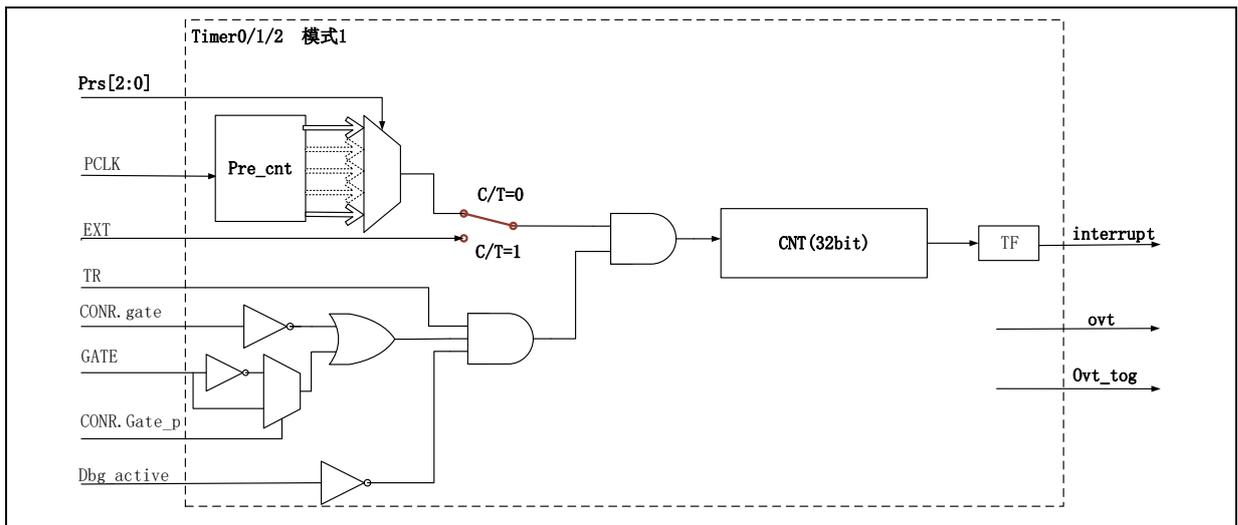


图 9-2 Timer 模式 1 框图

在模式 2 重载模式下，定时时间设的小的情况下需要考虑软件处理速度，否则中断会来不及处理而造成中断丢失。

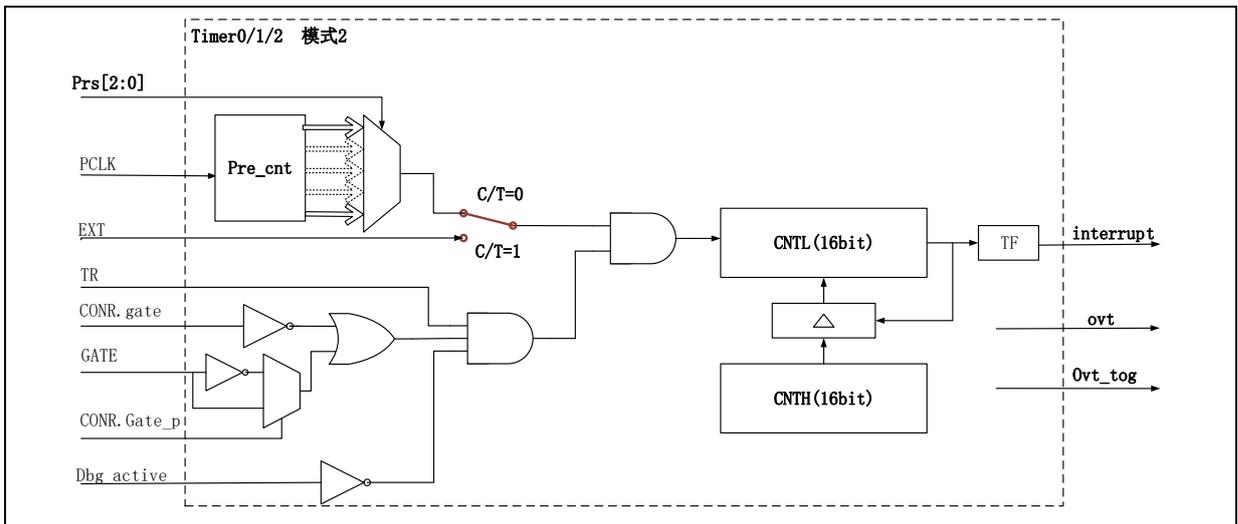


图 9-3 Timer 模式 2 框图

当设置对应定时器 TR 为 1 后定时器开始运行。模式 1 为 32 位定时/计数器，启动后从寄存器设定的初值 T 开始计数，计数到最大后产生溢出中断。然后从 0 继续计数。模式 2 为 16 位重载定时/计数器，启动后从寄存器初值 CNT 开始向上计数，计数到最大 0xFFFF 后产生中断后重载寄存器 ARR 的值到计数器 CNT 中，继续向上计数。

9.2.1 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 PCLK 采样，因此外部输入时钟频率不能超过系统的 Pclk 时钟。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

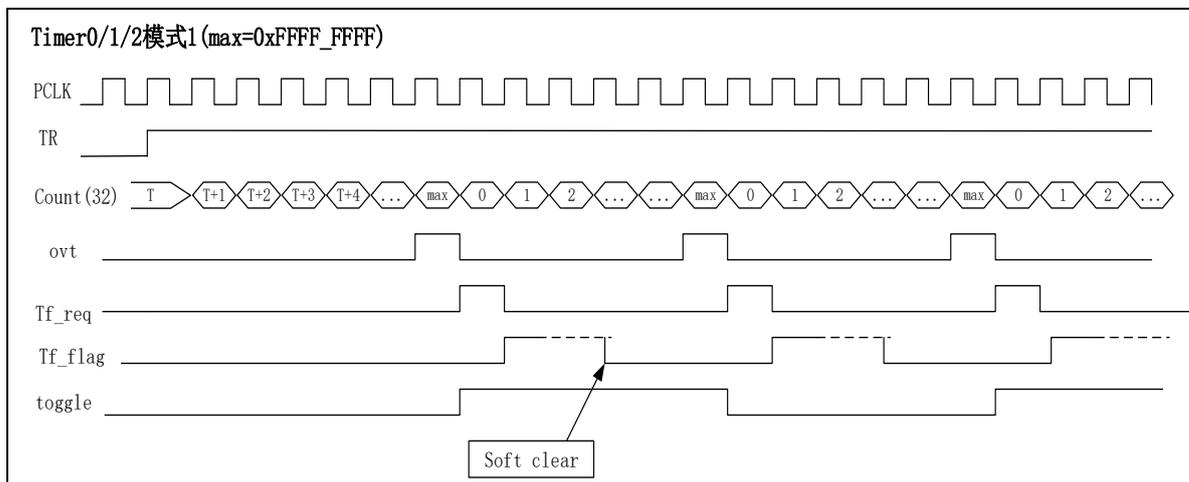


图 9-4 模式 1 时序图
定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

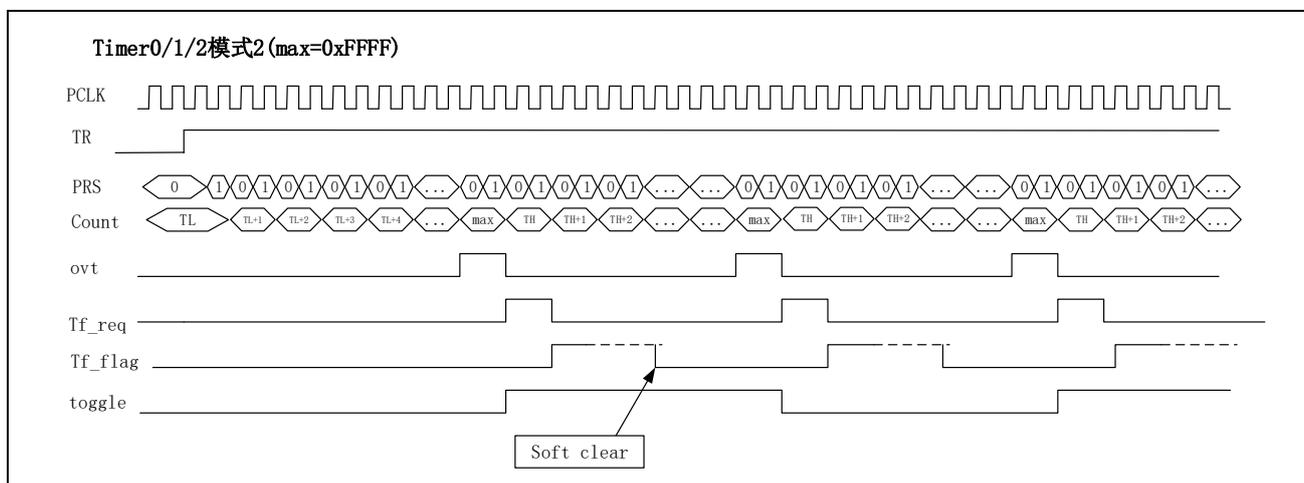


图 9-5 模式 2 时序图（预分频设置为 2）

9.2.2 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。CR.TOG_EN 为 1 时，TOG,TOGN 输出反向。设置 CR.TOG_EN 为 0 可以同时设置端口 TOG, TOGN 输出为 0。在计数时钟为 4M 情况下 Buzzer 输出不同频率的 Timer 重载模式配置如下：

Buzzer 频率	计数器周期	计数器计数值	计数器重载值	CNT 初始值	ARR 重载值
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 互连

9.3.1 GATE 互联

GATE 输入可以从端口直接输入,也可以输入 UART 的 RX 信号;还可以配置为 VC 的输入作为 GATE 信号。Timer0/1/2 的 GATE 都可以配置。

通过内部互联配置,可以实现 UART 波特率的自动识别,可以测量 VC 比较输出的脉冲宽度,可以实现外部控制计数。

配置选择 RX 输入在 GPIO_CTRL 寄存器控制,VC 控制在 VC 控制寄存器控制。端口选择,UART 输入选择与 VC 输入选择作为门控输入只能选择一个有效。

9.3.2 Toggle 输出互联

TIM0 的翻转输出 tog0 到内部模块 UART0,控制 UART0 的波特率;TIM1 的翻转输出 tog1 到内部模块 UART1,控制 UART1 的波特率;TIM0/1/2 的翻转输出还输出到端口上,可以驱动 Buzzer 实现蜂鸣器的控制。

9.4 Base Timer 寄存器描述

x=0,1,2;

Base Timer 基地址 0X4000C00

Timer	偏移地址	描述
TIM0	0x00	TIM0 偏移地址
TIM1	0x20	TIM1 偏移地址
TIM2	0x40	TIM2 偏移地址

寄存器	偏移地址	描述
TIMx_ARR	0X000	TIM0/1/2 重载寄存器
TIMx_CNT	0X004	TIM0/1/2 16 位模式计数寄存器
TIMx_CNT32	0X008	TIM0/1/2 32 位模式计数寄存器
TIMx_CR	0X00C	TIM0/1/2 控制寄存器
TIMx_IFR	0X010	TIM0/1/2 中断标志
TIMx_ICLR	0X014	TIM0/1/2 中断清除寄存器

表 9-1 Base Timer 寄存器列表

9.4.1 16 位模式重载寄存器 (TIMx_ARR)

偏移地址: 0x000

复位值: 0x0000 0000

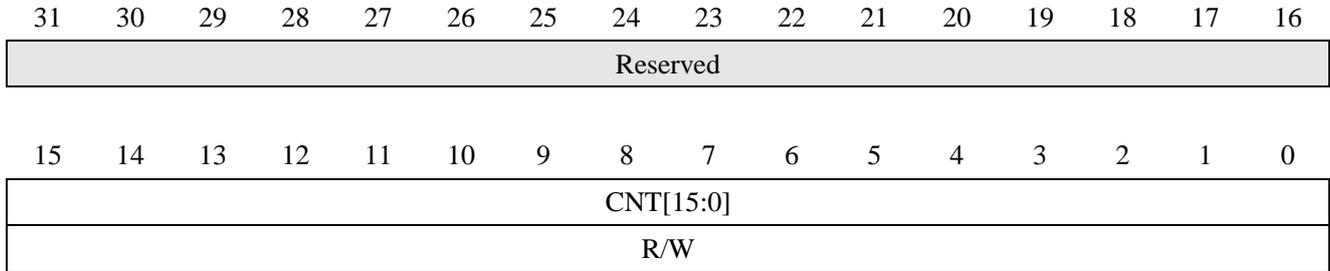
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
R/W															

位	符号	描述
31:16	Reserved	保留位, 读为0
15:0	ARR	定时器模式2 重载寄存器

9.4.2 16 位模式计数寄存器 (TIMx_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

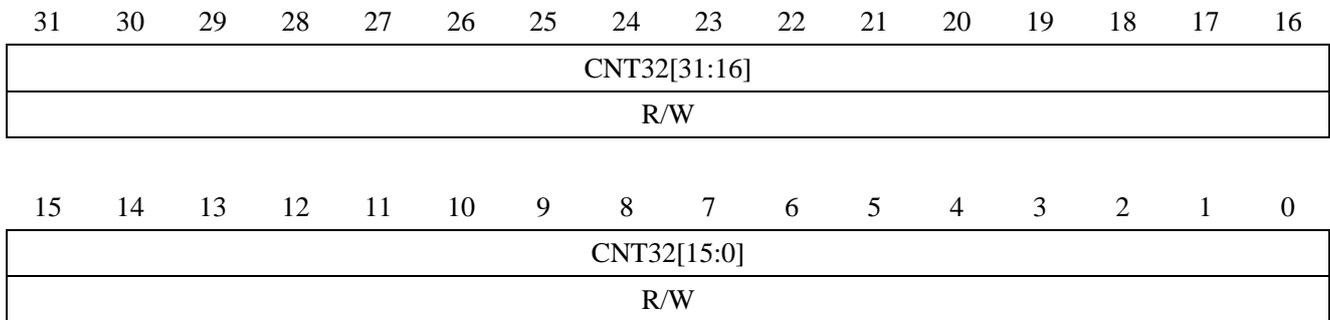


位	符号	描述
31:16	Reserved	保留位, 读为0
15:0	CNT	定时器模式2 计数值寄存器

9.4.3 32 位模式计数寄存器 (TIMx_CNT32)

偏移地址: 0x008

复位值: 0x0000 0000



位	符号	描述
31:0	CNT32	定时器模式1 计数值寄存器

9.4.4 控制寄存器 (TIMx_CR)

偏移地址: 0x00C

复位值: 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

位	符号	描述
31:11	Reserved	保留位, 读为0
10	IE	中断使能控制, 写1后使能中断
9	GATE_P	端口 GATE 极性控制, 默认高电平 gate 有效, 设置为1后低电平有效
8	GATE	定时器门控 0: 无门控, TR=1 时定时器工作; 1: 只有端口 GATE 有效并且TR=1时才工作;
7	Reserved	保留位, 读为0
6:4	PRS	TIM 预除频选择。000:1; 001:2; 010:4; 011:8; 100:16; 101:32; 110:64; 111:256;
3	TOG_EN	TOG 输出使能 0: TOG,TOGN 同时输出0 1: TOG,TOGN 输出相位相反的信号。可供 buzzer 使用。
2	CT	计数器/定时器功能选择 0: 定时器功能, 定时器由 PCLK 来进行计数。 1: 计数器功能, 计数器由外部输入的下降沿进行计数。外部输入由 PCLK 采样, 外部输入时钟频率要低于1/2 采样时钟。
1	MD	定时器工作模式 0: 模式1 32位计数器/定时器 1: 模式2自动重载16位计数器/定时器
0	TR	定时器运行控制 0: 定时器停止 1: 定时器运行

9.4.5 中断标志寄存器 (TIMx_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

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

位	符号	描述
31:1	REV	保留位, 读为0
0	TF	中断标志, 硬件置位。写清除寄存器清零

9.4.6 中断标志清除寄存器 (TIMx_ICLR)

偏移地址: 0x014

复位值: 0x0000 0001

31:8	7	6	5	4	3	2	1	0
Reserved								TFC
								W0

位	符号	描述
31:1	Reserved	保留位, 读为0
0	TFC	中断标志清除, 写0清除, 写1无效

10 低功耗定时器 (LPTIM)

10.1 LPTimer 简介

LPTimer 是异步 16 位定时/计数器，在系统时钟关闭后仍然可以通过内部低速 RC 或者外部低速晶体振荡计时/计数。通过中断在低功耗模式下唤醒系统。

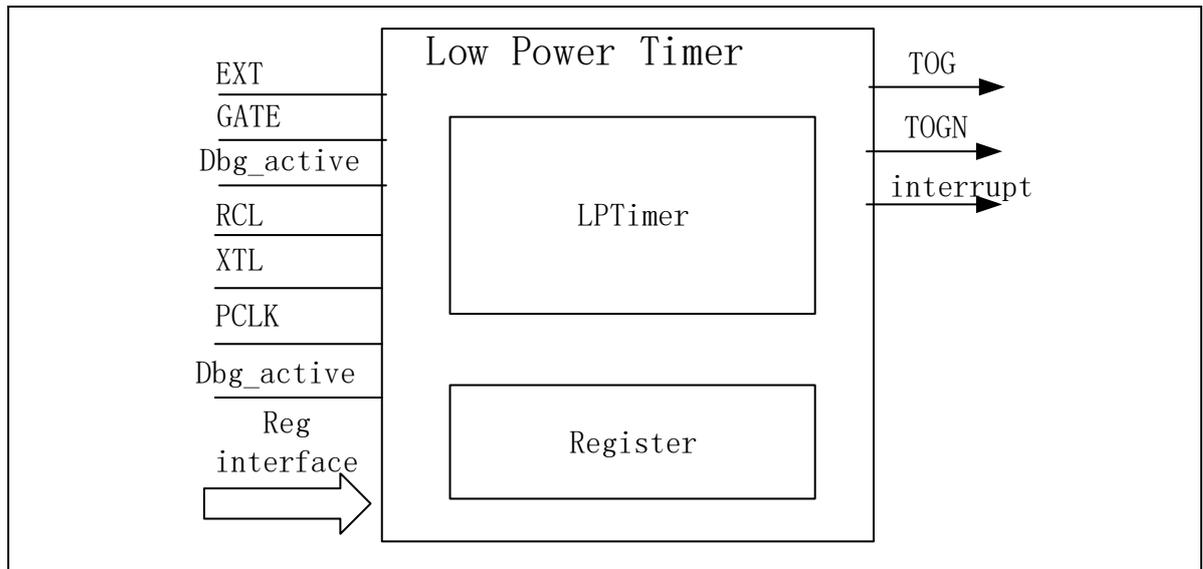


图 10-1 LPTimer 框图

10.2 LPTimer 功能描述

LPTimer 支持 2 种工作模式，每个定时器/计数器都有独立的控制启动信号，以及外部输入时钟，门控信号。

LPTimer 使用 EXT, GATE 来进行计数功能, EXT 用于计数器的外部输入时钟信号, Gate 用于有效电平计数使能信号。

LPTimer 的定时器支持两种工作模式，通过设置定时器控制寄存器（CR）中 MD 选择工作模式。模式 1 为 16 位自由计数模式。模式 2 是 16 位重载模式。

LPTimer 启动时会自动装载重载寄存器 ARR 的值到计数器中。

LPTimer 可选三种时钟作为定时器时钟，通过控制寄存器 CR.TCK_SEL 选择。默认选择 PCLK。时钟选择如表：

TCK_SEL	00	01	10	11
定时器时钟	PCLK	PCLK	XTL	RCL
读定时器计数值	读经过同步	无同步	读经过同步	读经过同步

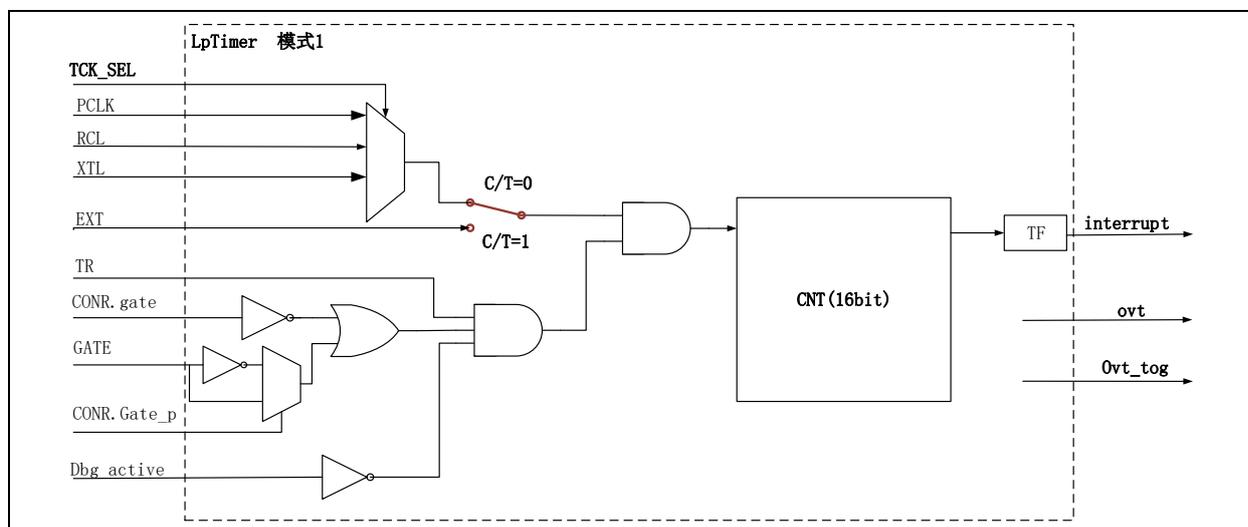


图 10-2 LPTimer 模式 1

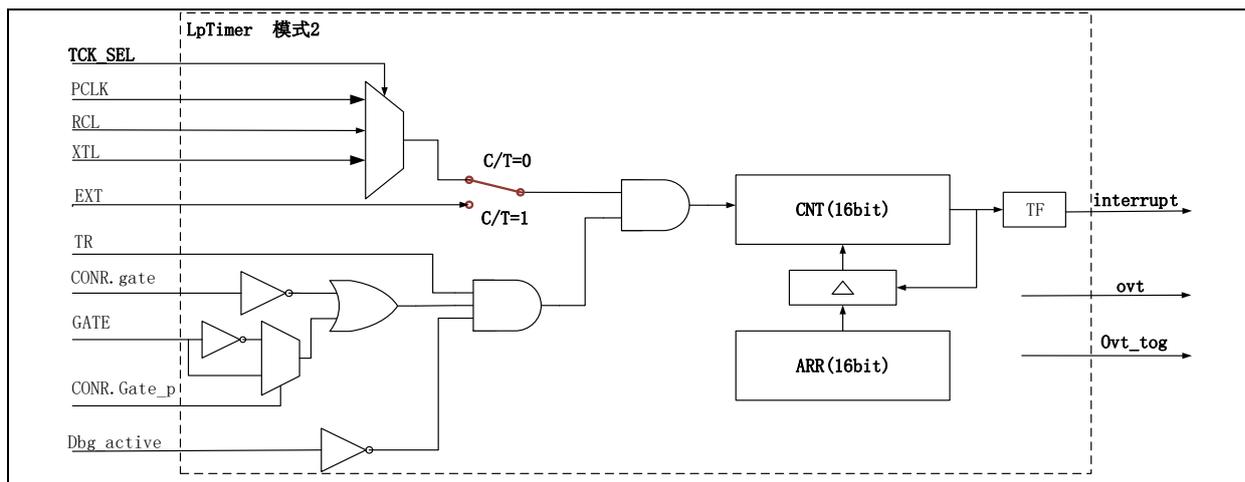


图 10-3 LPTimer 模式 2

当设置对应定时器 TR 为 1 后定时器开始运行。计数器从设定的值开始计数，计数到最大 0XFFFF 后产生溢出中断。模式 1 产生中断后计数器清零继续向上计数，模式 2 产生中断后重载寄存器 ARR 的值到计数器中，向上计数。计数器 CNT 的初值在启动定时器时由 ARR 自动装载。

由于 LPTimer 是异步定时器，定时器中断从 Timer 时钟域同步到 pclk 预，重载模式定时器值设置大于 0Xffff 时,中断会丢失，建议使用中断功能是重载寄存器的值小于 0XFFFC。不使用中断没有这个限制。

10.2.1 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的计数时钟采样，因此外部输入时钟频率不能超过系统的计数时钟。计数到最大值会溢出并且产生中断。

10.2.2 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器一个时钟累加一次，计数到最大值会溢出并且产生中断。

10.3 LPTimer 互连

10.3.1 GATE 互联

GATE 输入可以从端口直接输入，也可以输入 UART 的 RX 信号；还可以配置为 VC 的输入作为 GATE 信号。LPTIM 的 GATE 可以如下配置。

通过内部互联配置，可以实现 UART 波特率的自动识别，可以测量 VC 比较输出的脉冲宽度，可以实现外部控制计数。

UART 选择控制寄存器在端口控制寄存器中 GPIO_CTRL4，VC 输出控制寄存器在 VC 控制模块。

10.3.2 EXT 互联

EXT 输入可以从端口直接输入，也可以配置为 VC 的输入作为 EXT 信号。LPTIM 的 EXT 可以如下配置。

通过内部互联配置，可以测量 VC 脉冲计数。VC 输出控制寄存器在 VC 控制模块。

10.3.3 Toggle 输出互联

LPTimer 的翻转输出到端口上，可以驱动 Buzzer 实现蜂鸣器的控制。

10.4 LPTimer 寄存器描述

基地址 0X40000C00

寄存器	偏移地址	描述
CNT	0X060	LPTimer 计数值只读寄存器
ARR	0X064	LPTimer 重载寄存器
CR	0X06C	LPTimer 控制寄存器
IFR	0X070	LPTimer 中断标志
ICLR	0X074	LPTimer 中断清除寄存器

表 10-1 LPTimer 寄存器列表

10.4.1 计数器计数值寄存器 (LPTIM_CNT)

偏移地址: 0x060

复位值: 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															

位	符号	描述
31:16	Reserved	保留位, 读为0
15:0	CNT	低功耗定时器计数值只读寄存器。启动定时器时, CNT的初值由 ARR 自动装载。

10.4.2 重载寄存器 (LPTIM_ARR)

偏移地址: 0x064

复位值: 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															
位	符号	描述													
31:16	Reserved	保留位, 读为0													
15:0	ARR	低功耗定时器重载寄存器 写 ARR 前需要读取 CR.WT_FLAG, 当且尽当 WT_FLAG 为1时, 写入时才能写入数据。写ARR 寄存器后 WT2_FLAG 会变低。													

10.4.3 控制寄存器 (LPTIM_CR)

偏移地址: 0x06C

复位值: 0x0000 0008

31:11	10	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

位	符号	描述
31:11	Reserved	保留位, 读为0
10	IE	中断使能控制, 写1后使能中断
9	GATE_P	GATE 极性控制, 默认高电平 gate 有效, 设置为1后低电平有效
8	GATE	定时器门控 0: 无门控, TR=1 时定时器工作; 1: 只有门控为1 并且 TR=1 时才工作;
7	WT_FLAG	WT,写同步标志 0: 正在同步, 此时写ARR无效 1: 同步完成, 此时可更改ARR
6	Reserved	保留位, 读为0
5:4	TCK_SEL	LPTimer 时钟选择 00 PCLK; 10:XTL; 11 ,RCL
3	TOG_EN	TOG 输出使能 0: TOG,TOGN 同时输出0 1: TOG,TOGN 输出相位相反的信号。可供 buzzer 使用。
2	CT	计数器/定时器功能选择 0: 定时器功能, 定时器使用 TCK_SEL 选择的时钟进行计数。 1: 计数器功能, 计数器使用外部输入的下降沿进行计数。采样时钟使用 TCK_SEL选择的时钟, 外部输入时钟要低于 1/2 采样时钟。
1	MD	定时器工作模式 0: 模式1无重载模式16位计数器/定时器 1: 模式2自动重装载16位计数器/定时器
0	TR	定时器运行控制位 0: 定时器停止 1: 定时器运行

10.4.4 中断标志寄存器 (LPTIM_IFR)

偏移地址: 0x070

复位值: 0x0000 0000

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

位	符号	描述
31:1	Reserved	保留位, 读为0
0	TF	中断标志,硬件置位。写清除寄存器清零

10.4.5 中断标志清除寄存器 (LPTIM_ICLR)

偏移地址: 0x074

复位值: 0x0000 0001

31:8	7	6	5	4	3	2	1	0
Reserved								TFC
								W0

位	符号	描述
31:1	Reserved	保留位, 读为0
0	TFC	中断标志清除, 写0清除, 写1无效

11 可编程计数阵列 (PCA)

11.1 PCA 简介

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可作为一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个模块都可以进行独立编程，以提供输入捕捉、输出比较或脉冲宽度调制。另外模块 4 有额外的看门狗定时器模式。

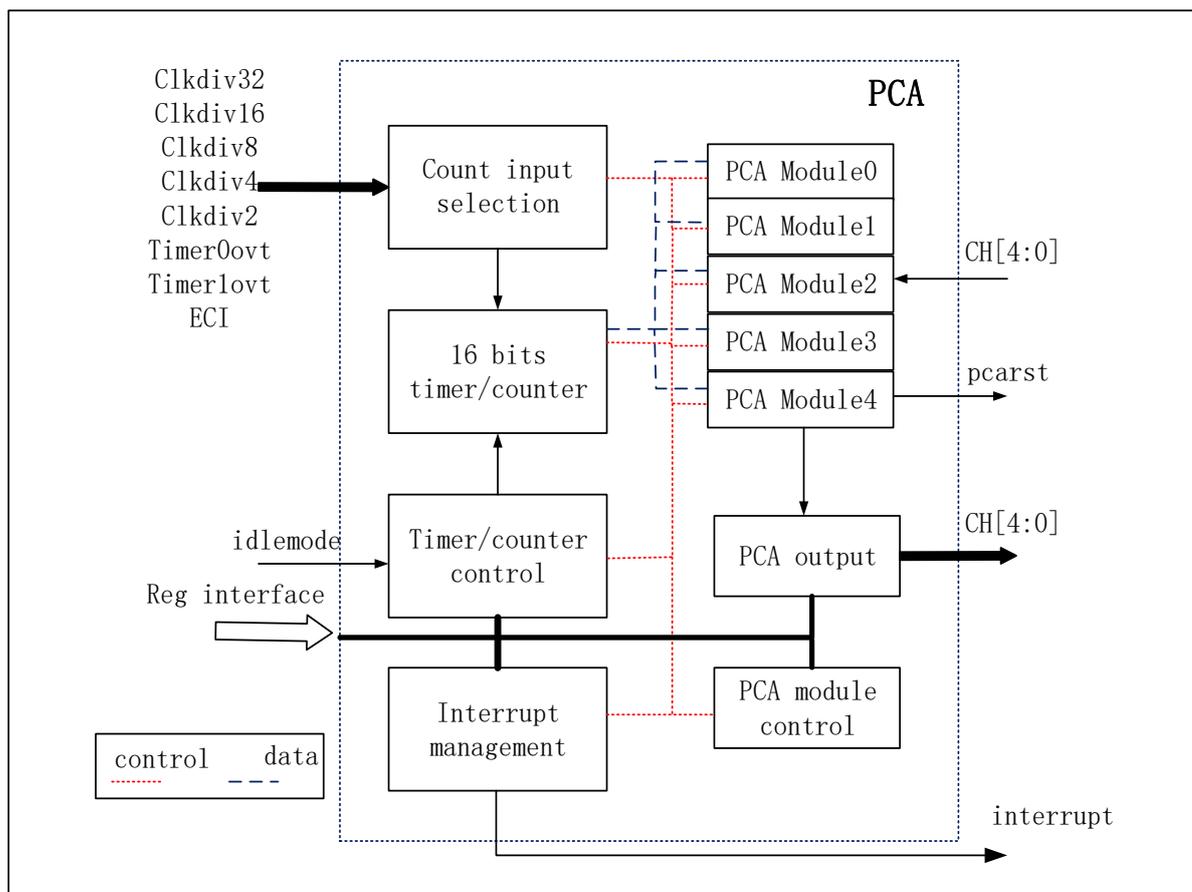


图 11-1 PCA 整体框图

11.2 PCA 功能描述

每个模块都可被配置为独立工作，有三种工作方式：边沿触发捕捉、输出比较、8 位脉宽调制。每个模块在系统控制器中都有属于自己的功能寄存器，这些寄存器用于配置模块的工作方式和与模块交换数据。

每组比较/捕获模块是由一个比较/捕获寄存器组(CCAPx)，以及 1 个 16 位比较器，和各种逻辑门控制组成。寄存器组用来存储时间或次数，针对外部触发捕获条件，或内部触发比较条件。在 PWM 模式下，寄存器(CCAPxL)用来控制输出波形的占空比。

每个模块都可以独立编程的操作在任何以下模式：

- 16 位捕获模式的上升沿，下降沿或任意沿触发。
- 比较模式：16 位软件定时器，16 位高速输出，16 位看门狗定时器（模块 4）或 8 位脉冲宽度调制。
- 未启动。

比较/捕获模块模式寄存器（CCAPMx）确定相应的工作模式。对于比较/捕获模块进行编程时，他们是基于共同的时间计数。定时器/计数器打开和关闭通过 CCON.CR 位即可控制 PCA 定时/计数器的运行。在一个比较/捕获模块捕获，软件定时器，高速输出，设置模块的比较/捕获标志（CCON.CCFx），并产生 PCA 中断请求，如果相应的使能位在 CCAPMx 寄存器设置。CPU 可以在任何时候读写 CCAPx 寄存器。

11.2.1 PCA 定时/计数器

CNT 的这组特殊功能寄存器可用作为一个 16 位定时器/计数器。这是一个 16 位向上计数的计数器。如果 CMOD.CFIE 位被置“1”时，当 CNT 溢出时硬件自动设置 PCA 溢出标志（CCON.CF）并产生 PCA 中断请求。CMOD.CPS[2: 0] 三位选择八个信号输入到定时器/计数器。

- 系统时钟 PCLK 的 32 分频。
- 系统时钟 PCLK 的 16 分频。
- 系统时钟 PCLK 的 8 分频。
- 系统时钟 PCLK 的 4 分频。
- 系统时钟 PCLK 的 2 分频。

- 定时器 0 的溢出。每次定时器 0 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- 定时器 1 的溢出。每次定时器 1 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- ECI。CPU 每过 4 个 PCLK 时钟周期就对 PCA ECI 进行采样，当每次采样结果从高变低时，CL 自动加 1，因此最高的 ECI 输入频率不能高于系统时钟 PCLK 的 1/8，以满足采样需求。

设置运行控制器 (CCON.CR) 启动 PCA 定时/计数器。当 CMOD.CIDL 置“1”后，PCA 定时器/计数器可以继续运行在空闲模式下。CPU 可以随时读取 CNT 的数值，但当计数启动后 (CCON.CR=1) 时，为了防止计数错误，CNT 是禁止写入的。

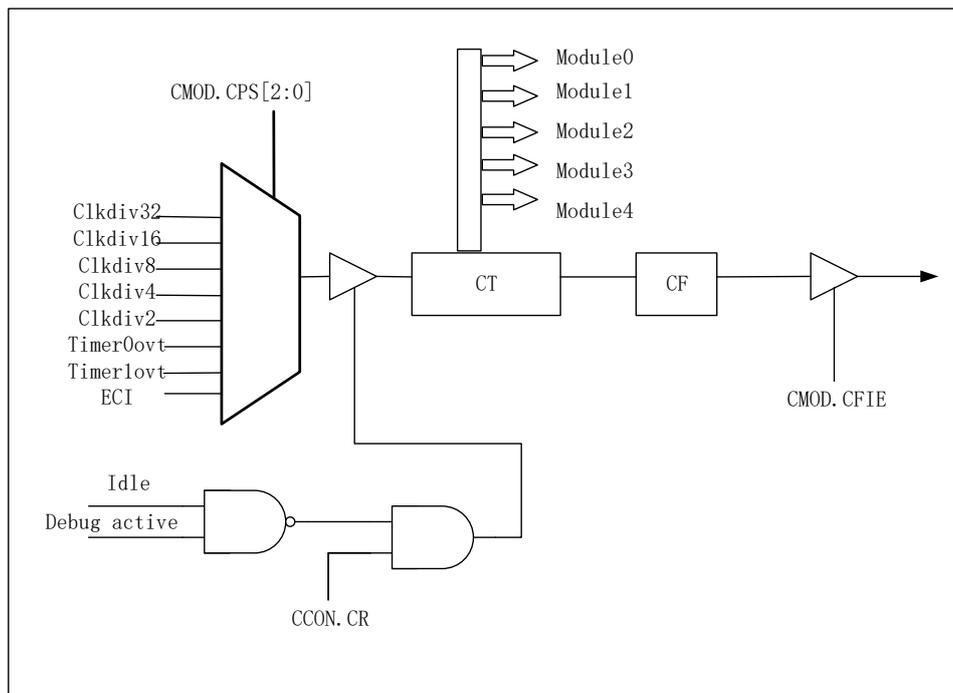


图 11-2 PCA 计数器框图

11.2.2 PCA 捕获功能

PCA 捕获模式提供了 5 路 PCA 测量脉冲周期，脉冲宽度，占空比和相位差的功能。引脚上出现的电平跳变导致 PCA 捕捉 PCA 计数器/定时器的值并将其装入到对应模块的 16 位捕捉/比较寄存器 (CCAPx)。CCAPMx.CAPP 以及 CCAPMx.CAPN 位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何变化（正沿或负沿）。当捕捉发生时，CCON 中的捕捉/比较标志 (CCFn) 被置为逻辑‘1’并产生一个中断请求(如果 CCF 中断被允许)。当 CPU 转向中断服务程序时，CCFn 位不能被硬件自动清除，用户软件写 INTCL 寄存器清除此标志位。需要上下沿同时捕获，建议流程：先使能一种边沿捕获，在发生捕获后在中断服务程序中更使能另外一种捕获，并且关闭前一种捕获中断。依次切换。

分辨率等于定时器/计数器的时钟。输入信号必须在高电平或低电平期间至少保持 2 个时钟周期，以保证输入信号能够被硬件识别。

CPU 可以在任何时候读取或写入 CCAPx 的寄存器。

捕获设置：

- 当需要在外部上升沿进行捕获，CCPMx.CAPP = “1” 以及 CCAPMx.CAPN = “0”
- 当需要在外部下降沿进行捕获，CCPMx.CAPP = “0” 以及 CCAPMx.CAPN = “1”
- 当需要在外部上升、下降沿进行捕获，CCPMx.CAPP = “1” 以及 CCAPMx.CAPN = “1”
- 当需要在外部上升、下降沿同时进行捕获并且需要知道捕获的边沿时，先设置 CCPMx.CAPP = “1”，发生上升沿捕获后，在中断服务子程序中切换到下降沿捕获，设置 CCAPMx.CAPN = “1”，并且清除 CCPMx.CAPP = “0”。下次捕获后再设置为上升沿捕获，依次切换捕获的边沿。

注意：

- 随后由同一模块的捕获值会覆盖现有捕获的值。为了保持捕获的值，在中断服务程序中将它保存在 RAM 里面，这个操作必须在下一次事件出现之前完成，否则就会丢失前面一次捕获采样值。

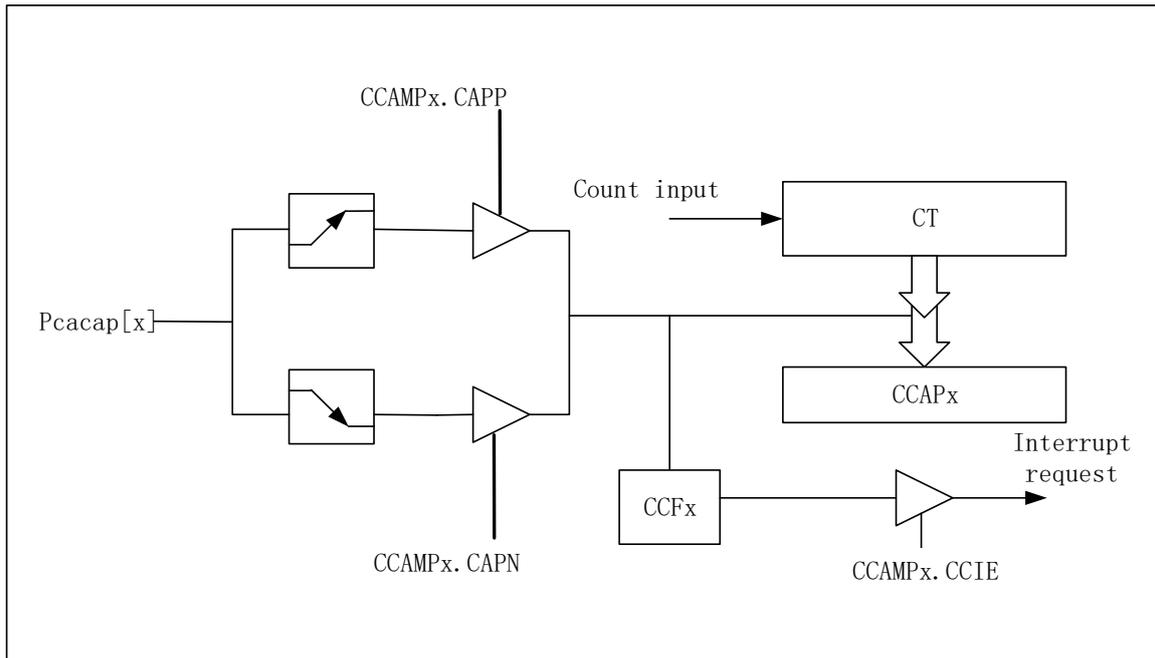


图 11-3 PCA 捕获功能框图

11.2.3 PCA 比较功能

比较功能提供五个模块如下功能，定时器，事件计数器，脉冲宽度调制。四种模式采用比较功能：16 位软件定时器模式，高速输出模式，WDT 模式和 PWM 模式。在前三个功能中，比较/捕获模块比较 16 位 PCA 定时器/计数器的值与预先加载到该模块的 CCAPx 寄存器中的 16 位值。在 PWM 模式下，PCA 模块不断地将 PCA 定时器/计数器低字节寄存器 (CNT) 与一个在 CCAPxL 模块寄存器 8 位的值进行比较。每 4 个时钟周期比较一次，即与最快的 PCA 定时器/计数器的时钟速率相匹配。

设置 CCAPMx.ECOM 位选择该模块的比较功能。

若要正确使用在比较模式下的模块，请遵守以下的一般程序：

- 选择 PCA 模块的操作模式
- 选择 PCA 定时器/计数器的输入信号。
- 比较值加载到模块的比较/捕获寄存器对。
- 设置 PCA 定时器/计数器运行控制位。
- 匹配后产生中断，清除模块的比较/捕获标志。

11.2.3.1 16 位软件计数模式

要设定一个比较/捕获模块的 16 位软件定时器模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.MAT 位。一旦在 PCA 定时器/计数器和比较/捕获的寄存器 (CCAPx) 之间发生了匹配，这将设置模块的比较/捕获标志 (CCON.CCFx)。这将产生一个中断请求，如果相应的中断使能位 (CCAPMx.CCIE) 设置。由于硬件不清除的比较/捕获标志中断处理时，用户必须清除软件标志。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器 (CCAPx)。

注意：在更新这些寄存器时，为了防止无效的匹配发生，用户软件应该先写 CCAPxL，后写 CCAPxH。一旦写如 CCAPxL 就会清除禁用比较功能 ECOMx 位，而写入 CCAPxH 会同时设置的 ECOMx 位，重新启用比较功能。即当向 PCA0 的捕捉/比较寄存器写入一个 16 位数值时，应先写低字节。

11.2.3.2 高速输出模式

在高速输出方式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器 (CCAPx) 发生匹配时，模块 PCA 的 CAP/CMP[x] 引脚上的逻辑电平将发生变化。这可以提供比切换 IO 输出有更高精度，因为这个高速输出不会被中断响应而影响输出频率，靠 CPU 来切换 IO 输出的话，功耗、精度都有所欠缺。

要设定一个比较/捕获模块的高速输出模式，设置 CCAPMx.ECOM, CCAPMx.MAT 和 CCAPMx.TOG 位。PCA 定时器/计数器和比较/捕获的寄存器 (CCAPx) 之间的匹配切换 PCA 的 CAP/CMP[x] 信号，并设置模块的比较/捕获标志 (CCON.CCFx)。通过软件设置或清除 PCA 的 CAP/CMP[x] 信号，用户可以选择匹配切换信号从低到高或高到低。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位 (CCAPMx.CCIE) 当匹配发生时，即可产生中断请求。由于硬件无法清除的比较/捕获标志中断，用户必须在软件中清除这个标志位。如果用户在中断程序中不去改变比较/捕获寄存器，PCA 并重新计数比较值，如相匹配则发生下一次翻转。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器 (CCAPx)。

注意：为了防止无效的匹配，而更新这些寄存器，用户软件应该写 CCAPxL 的首先然后 CCAPxH。写到 CCAPxL 清除禁用比较功能 ECOM 位，而写到 CCAPxH 设置的 ECOM 位，重新启用比较功能。

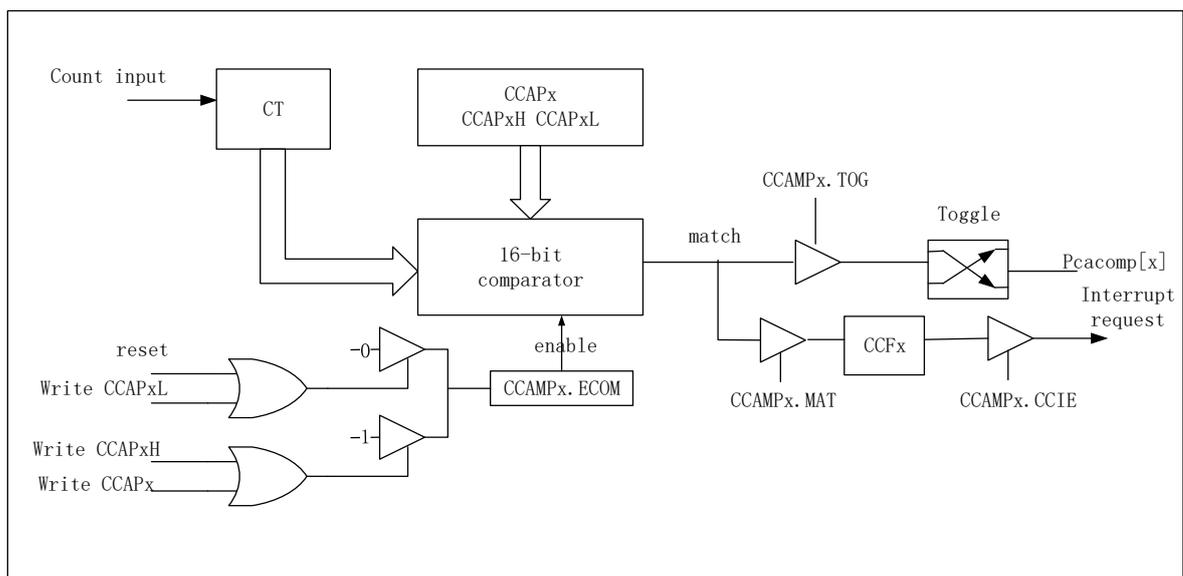


图 11-4 PCA 比较功能框图

11.2.3.3 PCA 模块 4 的 WDT 功能

本产品除了一个 WDT 硬件模块，PCA 的模块 4，还提供一个可编程频率的 16 位 WDT。当 PCA 定时器/计数器的计数值与模块 4 中存储的值比较/捕获寄存器 (CCAP4) 相匹配时，这种模式产生复位信号。PCA 的 WDT 复位信号做为一个独立的复位信号。与外部复位 (RST)，硬件看门狗复位 (WDTRST) 和 LVD 低电压复位，POR 上电下电复位相结合。用户可以自由结合或单独使用它们。模块 4 是具有 WDT 模式唯一 PCA 的模块。当不设置为 WDT 时，它可以在其它模式中独立使用。

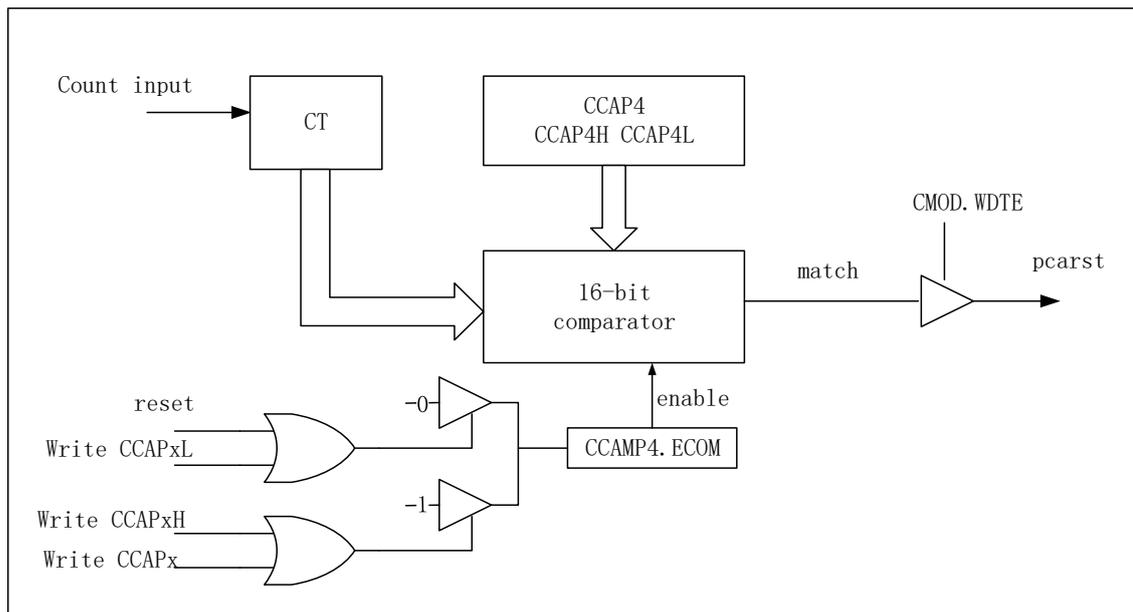


图 11-5 PCA WDT 功能框图

当把 PCA 模块 4 当 WDT 使用时，必须设置 CCAPM4.ECOM4, CCAPM4.MAT4 以及 CMOD.WDTE。另外 PCA 定时器/计数器可以设置 CMOD.CPS 来选择不同的输入计数频率。

在比较/捕获寄存器 (CCAP4) 输入一个 16 位的比较值。在 PCA 定时器/计数器 (CNT)，输入一个 16 位的初始值或使用复位值 (0x0000)。这些值乘以的 PCA 输入脉冲率之间的差额确定的 WDT 匹配运行时间。设置定时器/计数器运行控制位 (CCON.CR) 启动 PCA WDT。每次匹配时，PCA 的 WDT 产生复位信号。要防止一个 PCA WDT 复位，用户有三种选择。

- 定期的比较值 CCAP4 的改变，所以匹配永远不会发生。
- 定期更改 PCA 定时器/计数器值 (CNT) 所以匹配永远不会发生。

- 通过在匹配前清除的 `CMOD.WDTE` 位来禁用模块复位输出信号，后来重新启用它。前两个选项是更可靠的，因为 `WDT` 在第三个选项没有被禁用。第二个选项是不推荐，如果其他 `PCA` 模块都在使用，因为五个模块共享一个共同的时间基。因此，在大多数应用中的第一个选项是最好的。

`PCA WDT` 配置流程：

1. 配置 `WDT` 比较/捕获寄存器 `PCA_CCAP4`
2. 配置 `PCA` 计数寄存器 `PCA_CNT`
3. 配置 `PCA_CCAMP4` 选择比较匹配功能
4. 配置 `PCA_CMOD` 选择输入时钟，使能 `WDT` 功能
5. 启动 `PCA`
6. 选择清除 `PCA WDT` 清除方式在 `PCA WDT` 复位前清除 `PCA WDT`

11.2.3.4 PCA 8 位脉宽调制功能

脉宽调制是一种使用程序来控制波形占空比，周期，相位的技术。5 个 PCA 模块都可以被独立地用于在对应 PCA 的 CAP/CMP[x] 引脚产生脉宽调制 (PWM) 输出，脉冲宽度为 8 位分辨率。PWM 输出的频率取决于 PCA 计数器/定时器的时基。使用模块的捕捉/比较寄存器 CCAPxL 来改变 PWM 输出信号的占空比。当 PCA 计数器/定时器的低字节 (CNTL) 与 CCAPxL 中的值相等时，PCA 的 CAP/CMP[x] 引脚上的输出被置“1”；当 CNTL 中的计数值溢出时，PCA 的 CAP/CMP[x] 输出被复位“0”。当计数器/定时器的低字节 CNTL 溢出时 (从 0xFF 到 0x00)，保存在 CCAPxH 中的值被自动装入到 CCAPxL，不需软件干预。

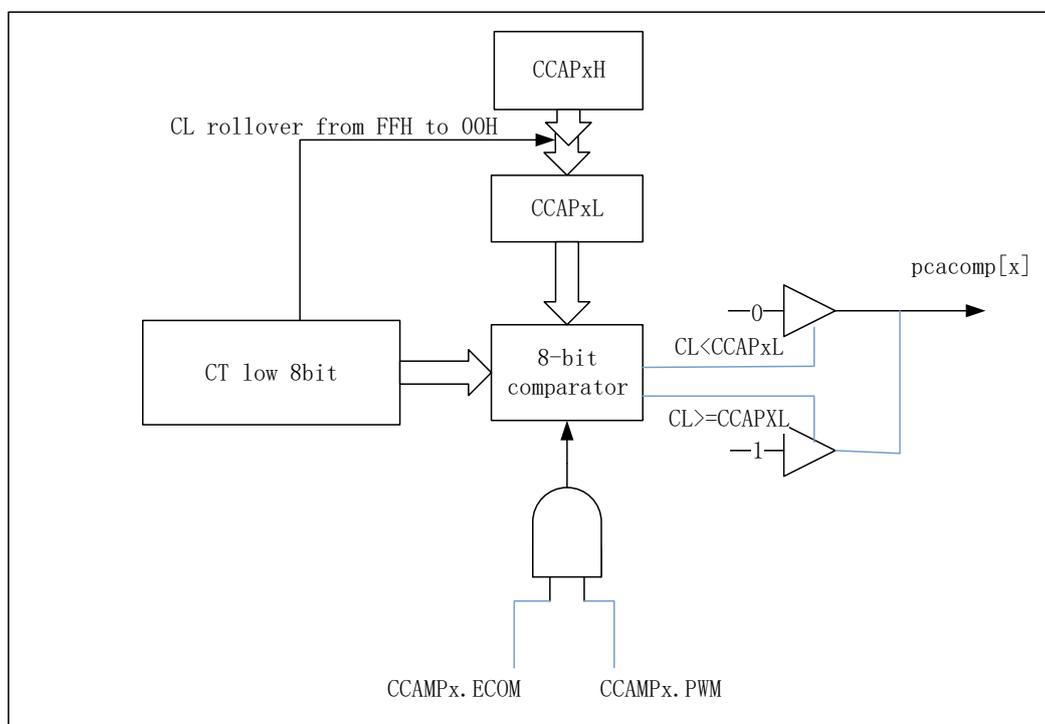


图 11-6 PCA PWM 功能框图

在这种模式下，PCA 定时器/计数器的低字节 (CNTL) 的值不断与比较/捕获寄存器的低字节 (CCAPxL) 的值进行比较。当 $CNTL < CCAPxL$ 时输出波形为低；当 $CNTL \geq CCAPxL$ 时输出波形为高。当 CNTL 溢出时，系统自动将 CCAPxH 的值装载到 CCAPxL 内，开始一个新的计数周期。

CCAPxL 的值决定当前周期的占空比，CCAPxH 的值决定下一个周期的占空比。改变 CCAPxL 中的值即可更改 PWM 的占空比。如下图所示，调整 CCAPxL 的值为 0~255 可以实现占空比 100%~0.4%。为了防止产生毛刺，建议不要直接更改 CCAPxL 的值，

通过更改 CCAPxH 的值在下一周期由硬件自动加载到 CCAPxL。

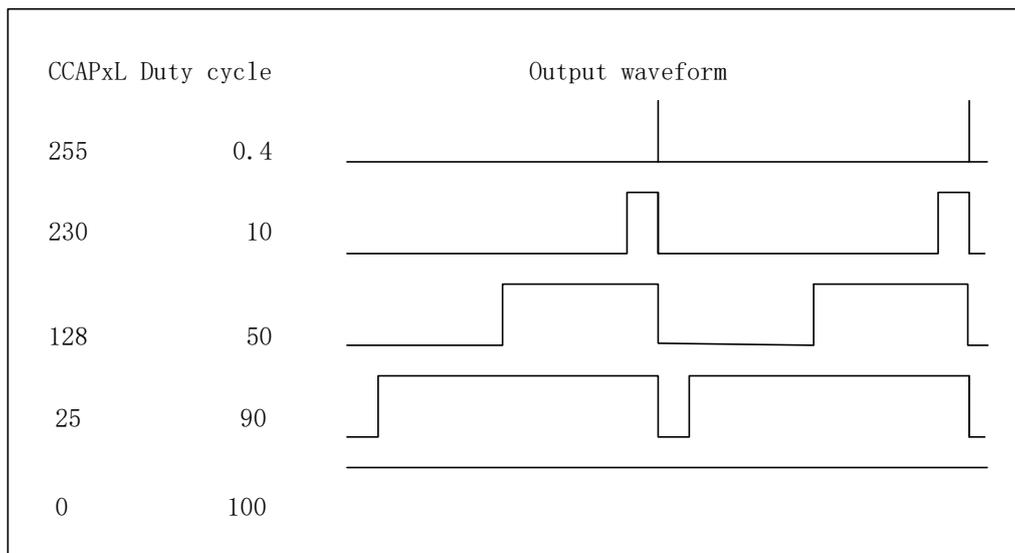


图 11-7 PCA PWM 输出波形

要设定一个比较/捕获模块在 PWM 模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.PWM 位。另外 PCA 定时器/计数器由编程 CMOD.CSP[2:0] 可以选择输入计数信号频率。在 CCAPxL 输入一个 8 位的值指定第一个 PWM 波形的占空比。在 CCAPxH 输入一个 8 位的值会指定第二个 PWM 波形的占空比。设置定时器/计数器运行控制位 (CCON.CR) 启动 PCA 定时器/计数器。

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	工作方式
X	1	0	0	0	0	X	用正沿触发捕获
X	0	1	0	0	0	X	用负沿触发捕获
X	1	1	0	0	0	X	用跳变沿触发捕获
1	0	0	1	0	0	X	软件定时器
1	0	0	1	1	0	X	高速输出
1	0	0	0	0	1	X	8位脉冲宽度调制器

表 11-1 PCA 比较捕获功能模块设置

11.3 PCA 模块与其他模块互连及控制

11.3.1 ECI 互连

ECI 输入可以是外部通过 IO MUX 选择不同的输入端口，也可以是内部 VC 的比较的滤波输出。VC 输出控制寄存器在 VC 控制模块。

11.3.2 PCACAP0

通道 0 的捕获输入可以是：

- 外部的 IO MUX 的输入端口
- 外部 UART 的 RX 的 MUX 输入
- 内部的 VC1 的比较滤波后的输出

UART 选择控制寄存器在端口控制寄存器中 GPIO_CTRL2，VC 输出控制寄存器在 VC 控制模块。

11.3.3 PCACAP1

通道 1 的捕获输入可以是：

- 外部的 IO MUX 的输入端口
- 外部 UART 的 RX 的 MUX 输入
- 内部的 VC2 的比较滤波后的输出

UART 选择控制寄存器在端口控制寄存器中 GPIO_CTRL2，VC 输出控制寄存器在 VC 控制模块。

11.3.4 PCACAP [4:2]

通道 2,3,4 的捕获输入可以是：

- 外部的 IO MUX 的输入端口
- 外部 UART 的 RX 的 MUX 输入

UART 选择控制寄存器在端口控制寄存器中 GPIO_CTRL2。

11.4 PCA 寄存器描述

基地址 0X40001000

寄存器	偏移地址	描述
CCON	0X000	PCA 控制寄存器
CMOD	0X004	PCA 模式寄存器
CNT	0X008	PCA 计数寄存器
ICLR	0X00C	PCA 中断清除寄存器
CCAPM0	0x010	PCA 比较/捕获模块0模式寄存器
CCAPM1	0x014	PCA 比较/捕获模块1模式寄存器
CCAPM2	0x018	PCA 比较/捕获模块2模式寄存器
CCAPM3	0x01C	PCA 比较/捕获模块3模式寄存器
CCAPM4	0x020	PCA 比较/捕获模块4模式寄存器
CCAP0H	0X024	PCA 比较/捕获模块0高8位寄存器
CCAP0L	0X028	PCA 比较/捕获模块0低8位寄存器
CCAP1H	0X02C	PCA 比较/捕获模块1高8位寄存器
CCAP1L	0X030	PCA 比较/捕获模块1低8位寄存器
CCAP2H	0X034	PCA 比较/捕获模块2高8位寄存器
CCAP2L	0X038	PCA 比较/捕获模块2低8位寄存器
CCAP3H	0X03C	PCA 比较/捕获模块3高8位寄存器
CCAP3L	0X040	PCA 比较/捕获模块3低8位寄存器
CCAP4H	0X044	PCA 比较/捕获模块4高8位寄存器
CCAP4L	0X048	PCA 比较/捕获模块4低8位寄存器
CCAPO	0X04C	PCA PWM 与高速输出标志寄存器
CCAP0	0X050	PCA 比较/捕获模块0 的16位寄存器
CCAP1	0X054	PCA 比较/捕获模块1 的16位寄存器
CCAP2	0X058	PCA 比较/捕获模块2 的16位寄存器
CCAP3	0X05C	PCA 比较/捕获模块3 的16位寄存器
CCAP4	0X060	PCA 比较/捕获模块4 的16位寄存器

表 11-2 PCA 寄存器列表

11.4.1 控制寄存器 (PCA_CCON)

偏移地址: 0x000

复位值: 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

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志 (写无效) 当 PCA 计数溢出时, CF 由硬件置位, 如果 CMOD 寄存器的 CFIE 位为1, 则 CF标志可以产生中断 1: 发生计数器溢出; 0: 无溢出;
6	CR	PCA 计数器运行控制位 1: 启动 PCA 计数器计数 0: 关闭 PCA 计数器计数
5	Reserved	保留位
4	CCF4	PCA 计数器模块4 比较/捕获标志位 当出现匹配或捕获时, 该位由硬件置位。(写无效) 当 CCAPM4.CCIE 置位时, 这个标志位会产生一个 PCA 中断
3	CCF3	PCA 计数器模块3 比较/捕获标志位 当出现匹配或捕获时, 该位由硬件置位。(写无效) 当 CCAPM3.CCIE 置位时, 这个标志位会产生一个 PCA 中断
2	CCF2	PCA 计数器模块2 比较/捕获标志位 当出现匹配或捕获时, 该位由硬件置位。(写无效) 当 CCAPM2.CCIE 置位时, 这个标志位会产生一个 PCA 中断
1	CCF1	PCA 计数器模块1 比较/捕获标志位 当出现匹配或捕获时, 该位由硬件置位。(写无效) 当 CCAPM1.CCIE 置位时, 这个标志位会产生一个 PCA 中断
0	CCF0	PCA 计数器模块0 比较/捕获标志位 当出现匹配或捕获时, 该位由硬件置位。(写无效) 当 CCAPM0.CCIE 置位时, 这个标志位会产生一个 PCA 中断

11.4.2 模式寄存器 (PCA_CMOD)

偏移地址: 0x004

复位值: 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	

位	符号	描述
31:8	Reserved	保留位
7	CIDL	空闲模式 IDLE 下, PCA 是否停止工作 1: 休眠模式 (sleep) 下, PCA 停止工作 0: 休眠模式 (sleep) 下, PCA 继续工作
6	WDTE	PCA WDT 功能使能控制位 1: 启动 PCA 模块 4 WDT 功能 0: 关闭 PCA 模块 4 WDT 功能
5:4	Reserved	保留位
3:1	CPS	时钟分频选择及时钟源选择 000: PCLK/32 001: PCLK/16 010: PCLK/8 011: PCLK/4 100: PCLK/2 101: Timer0 overflow 110: Timer1 overflow 111: ECI 外部时钟, 时钟 PCLK 四分频采样
0	CFIE	PCA 计数器中断使能控制信号 1: 使能中断 0: 关闭中断

11.4.3 计数寄存器 (PCA_CNT)

偏移地址: 0x008

复位值: 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															

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	定时器计数器的值 只有在 PCA 停止状态 CNT 才可以写入, 否则写入无效

11.4.4 中断清除寄存器 (PCA_ICLR)

偏移地址: 0x00C

复位值: 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	

位	符号	描述
31:8	Reserved	保留位
7	CF	PCA 计数器溢出标志清除 (软件写0清零, 写1无效), 读出值为1
6:5	Res.	保留位
4	CCF4	PCA 计数器模块4 比较/捕获标志位清除 (软件写0清零, 写1无效), 读出值为1
3	CCF3	PCA 计数器模块3 比较/捕获标志位清除 (软件写0清零, 写1无效), 读出值为1
2	CCF2	PCA 计数器模块2 比较/捕获标志位清除 (软件写0清零, 写1无效), 读出值为1
1	CCF1	PCA 计数器模块1 比较/捕获标志位清除 (软件写0清零, 写1无效), 读出值为1
0	CCF0	PCA 计数器模块0 比较/捕获标志位清除 (软件写0清零, 写1无效), 读出值为1

11.4.5 比较捕获模式寄存器 (PCA_CCAPM0~4)

偏移地址

CCAPM0: 0x010; CCAPM1: 0x014; CCAPM2: 0x018;

CCAPM3: 0x01C; CCAPM4: 0x020;

复位值: 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

位	符号	描述
31:7	Reserved	保留位
6	ECOM	允许比较器功能控制位 1: 允许比较器功能; 0: 禁止比较器功能; 当 PCA 用于软件计数器, 高速输出, PWM 模式, WDT 模式, 要置位 ECOM 写 CCAMPHx 或 CCAMPx 寄存器会自动置位 ECOM; 写 CCAMPLx 寄存器会自动清除 ECOM 位
5	CAPP	正沿捕获控制位 1: 允许上升沿捕获; 0: 禁止上升沿捕获
4	CAPN	负沿捕获控制位 1: 允许下降沿捕获; 0: 禁止下降沿捕获
3	MAT	允许匹配控制位 1: PCA 计数值与模块的比较/捕获寄存器的值一旦匹配, 将置位 CCON 寄存的中断标志 CCFx(x=0-4) 0: 禁止匹配功能
2	TOG	翻转控制位 1: 工作在 PCA 高速输出模式, PCA 计数器的值与模块的比较/捕获寄存器的值一旦匹配, CCPx 引脚翻转 0: 禁止翻转功能
1	PWM	脉宽调制控制位 1: 允许 CCPx 引脚作为 PWM 输出 0: 禁止 PWM 脉宽调制功能 只有 CCAPMx[6:0]=100_0010 时, PWM 功能才有效
0	CCIE	PCA 使能中断 1: 使能比较/捕获中断 0: PCA 比较/捕获功能中断禁止

11.4.6 比较捕获数据寄存器高 8 位 (PCA_CCAP0~4H)

偏移地址

CCAP0H: 0x024; CCAP1H: 0x02C; CCAP2H: 0x034;

CCAP3H: 0x03C; CCAP4H: 0x044;

复位值: 0x0000 0000

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

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[15:8]	比较/捕获模式高8位寄存器 当 PCA 模式用于比较/捕获模式时，用于保存16位捕获计数值的高8位；写 CCAPxH 寄存器会自动置位寄存器 CCAPMx 的 ECOM 位。 当 PCA 模式用于 PWM 模式时，用于控制输出占空比装载寄存器，在计数器低8位溢出时，装载寄存器会自动更新到 PWM 比较寄存器

11.4.7 比较捕获数据寄存器低 8 位 (PCA_CCAP0~4L)

偏移地址

CCAP0L: 0x028; CCAP1L: 0x030; CCAP2L: 0x038;

CCAP3L: 0x040; CCAP4L: 0x048;

复位值: 0x0000 0000

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

位	符号	描述
31:8	Reserved	保留位
7:0	CCAPx[7:0]	比较/捕获模式低8位寄存器 当 PCA 模式用于比较/捕获模式时，用于保存16位捕获计数值的低8位；写 CCAPxH 寄存器会自动清除寄存器 CCAPMx 的 ECOM 位。 当 PCA 模式用于 PWM 模式时，用于控制输出占空比比较寄存器，在 PWM 模式，计数器的低8位的值小于 CCAPx[7:0] 的值 PWM 输出低电平，否则 PWM 输出高电平。

11.4.8 比较捕获 16 位寄存器 (PCA_CCAP0~4)

偏移地址

CCAP0: 0x050; CCAP1: 0x054; CCAP2: 0x058;

CCAP3: 0x05C; CCAP4: 0x060;

复位值: 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															

位	符号	描述
31:16	Reserved	保留位
15:0	CCAPx	比较/捕获模式16位寄存器 当 PCA 模式用于比较/捕获模式时，用于保存16位捕获计数值；写 CCAPx 寄存器会置位寄存器 CCAPMx 的 ECOM 位。 写 CCAPX 寄存器相当于写 CCAPxL 及 CCAPxH 这两个8位寄存器。在比较/捕获模式下可以直接读写这个寄存器，在 PWM 模式下，使用 CCAPxL 及 CCAPxH 寄存器

11.4.9 比较高速输出标志寄存器 (PCA_CCAPO)

偏移地址: 0x04C

复位值: 0x0000 0000

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

位	符号	描述
31:5	Reserved	保留位
4	CCAPO4	比较模块4的输出值
3	CCAPO3	比较模块3的输出值
2	CCAPO2	比较模块2的输出值
1	CCAPO1	比较模块1的输出值
0	CCAPO0	比较模块0的输出值

12 高级定时器 (TIM4/5/6)

12.1 高级定时器简介

高级定时器是一个包含三个定时器 Timer4/5/6。Timer4/5/6 功能相同的高性能计数器，可用于计数产生不同形式的时钟波形，1 个定时器可以产生互补的一对 PWM 或者独立的 2 路 PWM 输出，可以捕获外界输入进行脉冲宽度或周期测量。

高级定时器基本的功能及特性如表所示。

波形模式	锯齿波、三角波
基本功能	● 递加、递减计数方向
	● 软件同步
	● 硬件同步
	● 缓存功能
	● 正交编码计数
	● 通用 PWM 输出
	● 刹车保护
	● AOS 关联动作
中断类型	计数比较匹配中断
	计数周期匹配中断
	死区时间错误中断

表 12-1 Advanced Timer 基本特性

端口名	方向	功能
TIMx_CHA	输入/输出	正交编码计数时钟输入端口或捕获输入端口或比较输出端口 (x=4~6)
TIMx_CHB		
TIMTRIA	输入	硬件计数时钟输入端口或捕获输入端口 硬件启动、停止、清零条件输入端口
TIMTRIB		
TIMTRIC		
TIMTRID		

表 12-2 Advanced Timer 端口列表

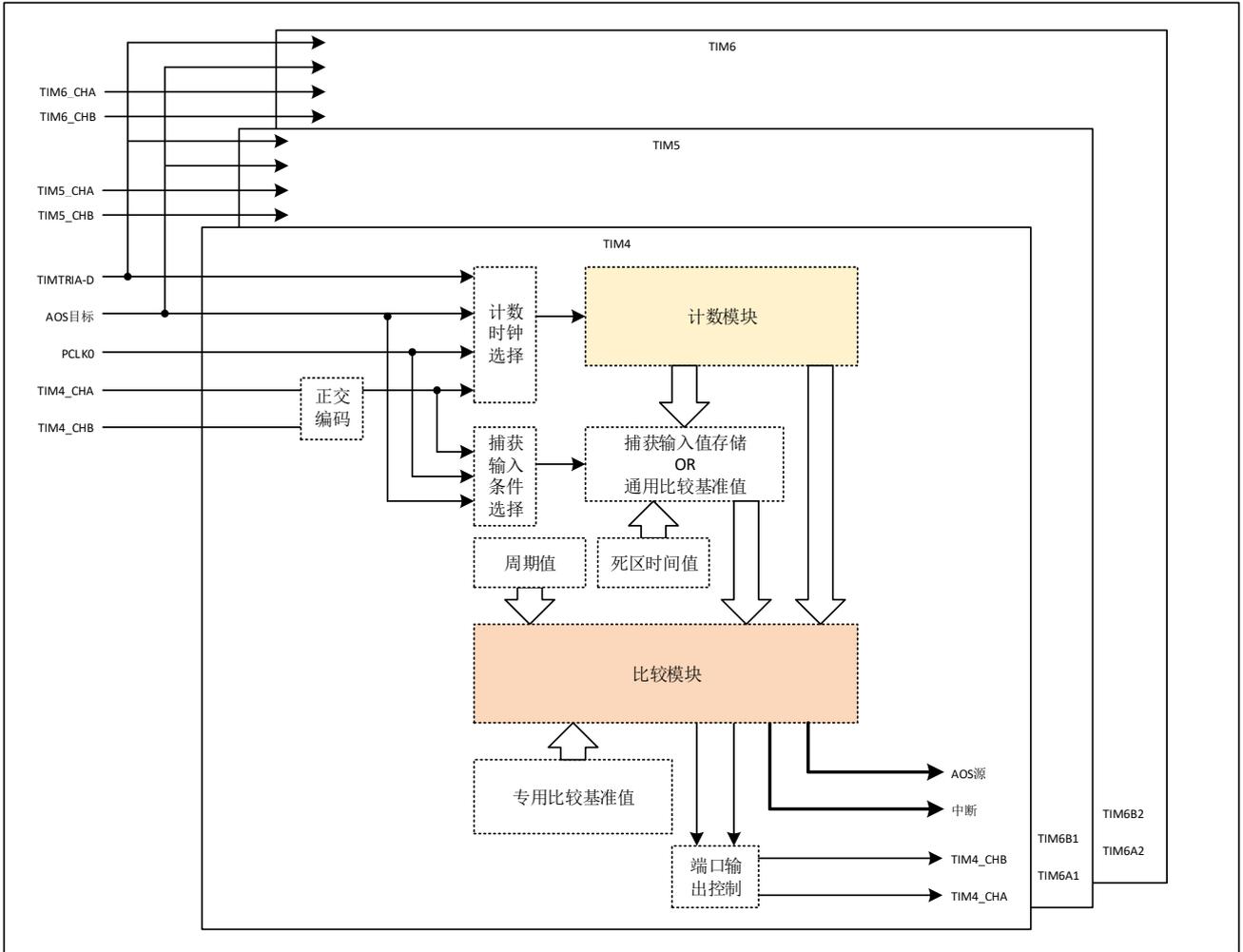


图 12-1 Advanced Timer 框图

12.2 Advanced Timer 功能描述

12.2.1 基本动作

12.2.1.1 基本波形模式

Timer4/5/6 有 2 种基本计数波形模式，锯齿波模式和三角波模式。波形模式又由于不同的内部计数动作有所细分，三角波模式分为三角波 A 模式、三角波 B 模式。锯齿波和三角波的基本波形如图 12-2 图 12-3 所示。三角波 A 模式与三角波 B 模式区别在于缓存传送有差别，三角波 A 模式一个周期只发生一次缓存传送（谷点），而三角波 B 模式一个周期发生两次缓存传送（峰点和谷点）。

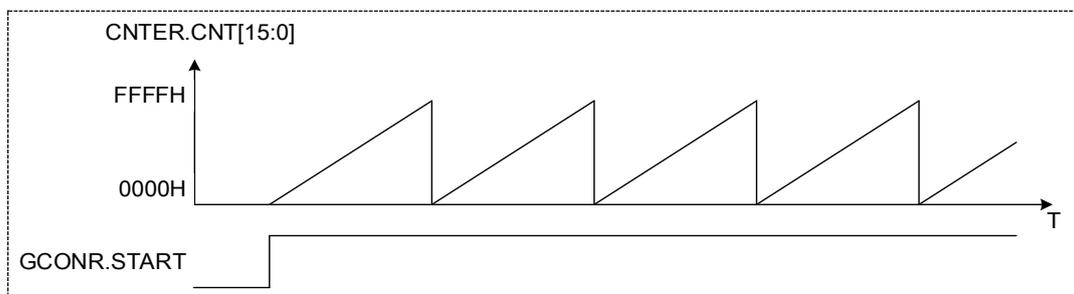


图 12-2 锯齿波波形（递加计数）

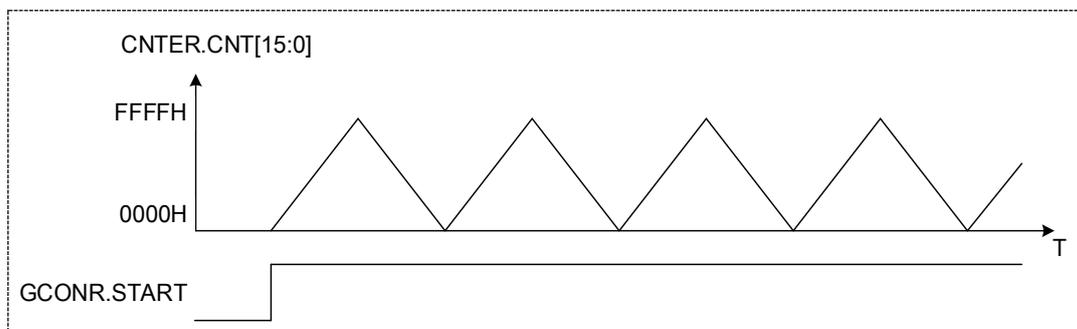


图 12-3 三角波波形

12.2.1.2 比较输出

Timer4/5/6 一个定时器有 2 个比较输出端口 (CHxA、CHxB)，可在计数值与计数基准值比较匹配时输出指定的电平。GCMAR、GCMBR 寄存器分别对应了 CHxA、CHxB 的计数比较基准值。当计数器的计数值和 GCMAR 相等时，CHxA 端口输出指定的电平；当计数器的计数值和 GCMBR 相等时，CHxB 端口输出指定电平。

CHxA、CHxB 端口的计数起始电平、停止电平、计数比较匹配时的电平等，可由端口控制寄存器 (PCONR) 的 PCONR.STACA、PCONR.STPCA、PCONR.STASTPSA、PCONR.CMPCA[1:0]、PCONR.PERCA[1:0] 和 PCONR.STACB、PCONR.STPCB、PCONR.STASTPSB、PCONR.CMPCB[1:0]、PCONR.PERCB[1:0] 位设定。图 12-4 为比较输出的动作例。

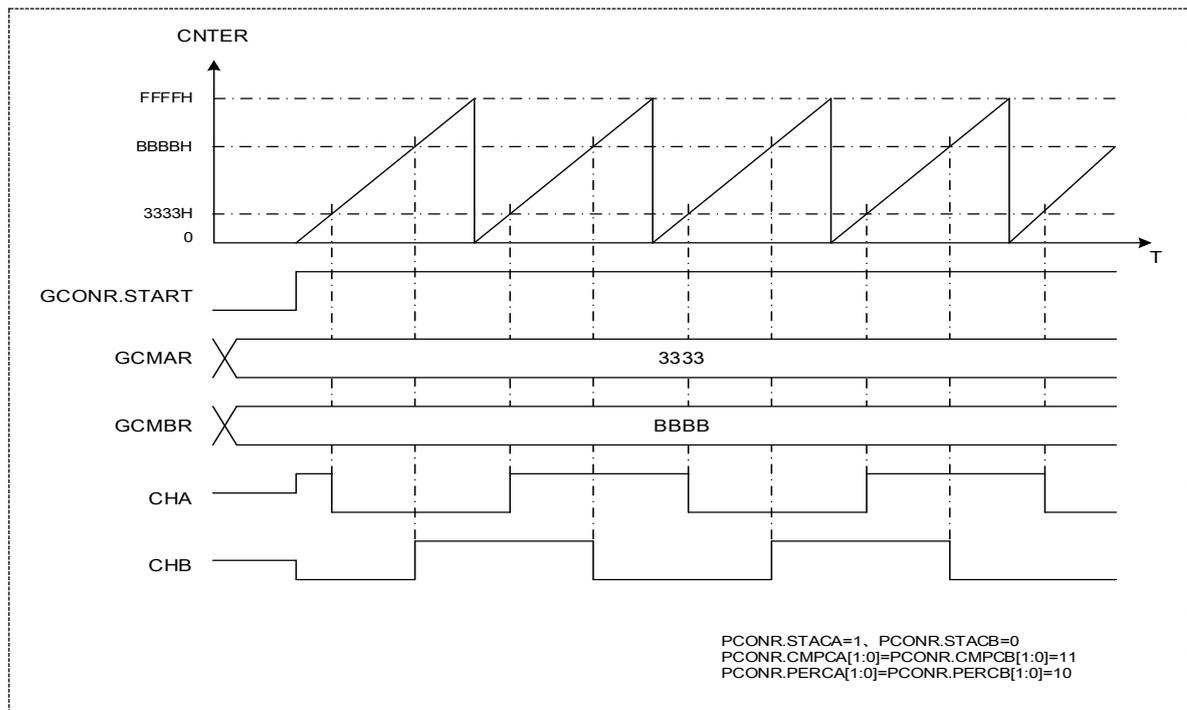


图 12-4 比较输出动作

12.2.1.3 捕获输入

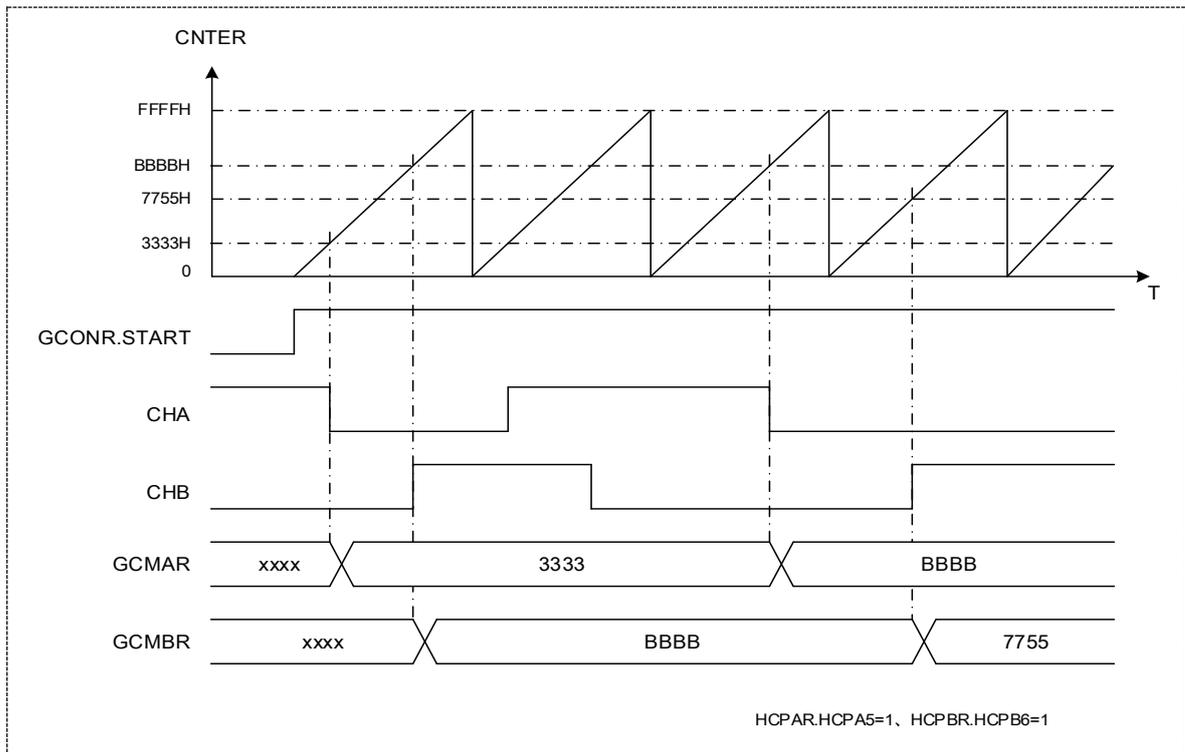


图 12-5 捕获输入动作

Timer4/5/6 都具有捕获输入功能，具备 2 组捕获输入寄存器（GCMAR、GCMBR），用于保存捕获到的计数值。设定端口控制寄存器（PCONR）的 PCONR.CAPCA、PCONR.CAPCB 位为 1，捕获输入功能就有效了。当设定了对应的捕获输入条件且该条件有效时，当前的计数值就被保存到相应的寄存器（GCMAR、GCMBR）中。每组捕获输入的条件可以是 AOS 事件触发、TIMTRIA-TIMTRID 输入、CHxA 或 CHxB 的输入等，具体的条件选择可通过硬件捕获事件选择寄存器（HCPAR、HCPBR）来设定。图 12-5 为捕获输入的动作例。

12.2.2 时钟源选择

Timer4/5/6 的计数时钟可以有以下几种选择：

- a. PCLK 及 PCLK 的 2、4、8、16、64、256、1024 分频 (GCONR.CKDIV[2:0] 设定)
- b. AOS 事件触发输入 (HCUPR.HCUP [19:16] 或 HCDOR.HCDO [19:16] 设定)
- c. CHxA 和 CHxB 的正交编码输入 (HCUPR.HCUP[7:0] 或 HCDOR.HCDO [7:0] 设定)
- d. TIMTRIA-TIMTRID 的端口输入 (HCUPR.HCUP [15:8] 或 HCDOR.HCDO [15:8] 设定)

从上述描述可以看到，b、c、d 时钟互相独立，可分别设定有效或无效，并且当选择 b、c、d 时钟时，a 时钟自动无效。

12.2.3 计数方向

Timer4/5/6 的计数器计数方向可通过软件方式改变。不同波形模式时，改变计数方向的方法略有不同。

12.2.3.1 锯齿波计数方向

锯齿波模式时，计数方向可在计数器计数中或停止时设定。

在递加计数中时，设定 `GCONR.DIR=0`（递减计数），则计数器计数到上溢后变为递减计数模式；在递减计数中时，设定 `GCONR.DIR=1`（递加计数），则计数器计数到下溢后变为递加计数模式。

在计数停止时，设定 `GCONR.DIR` 位。则计数开始后直至上溢或下溢时，`GCONR.DIR` 的设定才会反映到计数中。

12.2.3.2 三角波计数方向

三角波模式时，计数方向只能在计数器停止时设定。在计数中设定计数方向无效。

在计数停止时，设定 `GCONR.DIR` 位。则计数开始后直至上溢或下溢时，`GCONR.DIR` 的设定才会反映到计数中。

12.2.4 数字滤波

Timer4/5/6 的 CHxA、CHxB、TIMTRIA~D 端口输入都有数字滤波功能。可通过设定滤波控制寄存器（FCONR）的相关使能位开启对应端口的滤波功能。滤波用的基准时钟也通过滤波控制寄存器（FCONR）设定。

在滤波采样基准时钟采样到端口上 3 次一致的电平时，该电平被当作有效电平传送到模块内部；小于 3 次一致的电平会被当作外部干扰滤掉，不传送到模块内部。其动作例如图 12-6 所示。

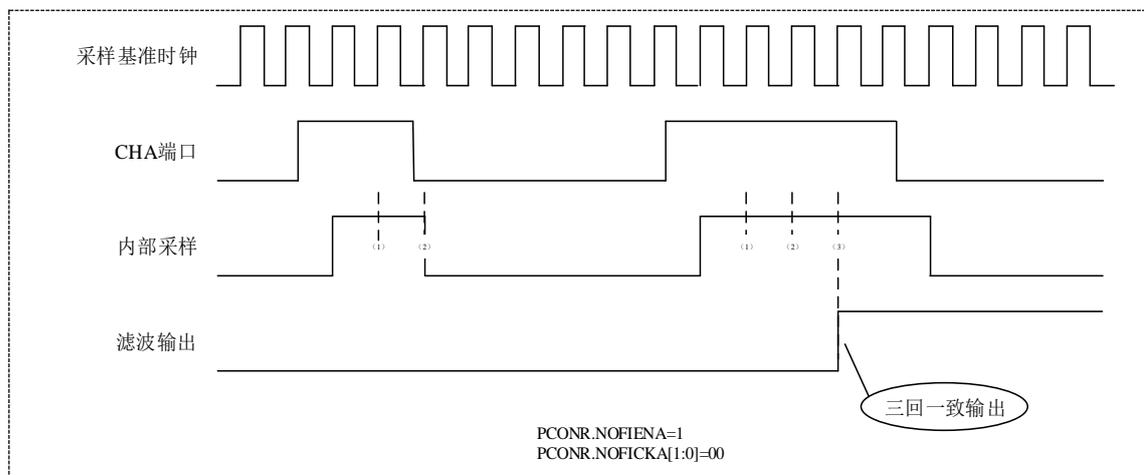


图 12-6 捕获输入端口的滤波功能

TIMTRIA~D 端口是一组 Timer4/5/6 间共用的端口，该组端口的数字滤波功能只在 Timer4 实现，其他定时器 Timer5/6 对该组端口的数字滤波功能设定无效。

12.2.5 软件同步

12.2.5.1 软件同步启动

Timer4/5/6 可通过设定软件同步启动寄存器 (SSTAR) 的相关位, 实现目标 Timer4/5/6 的同步启动。

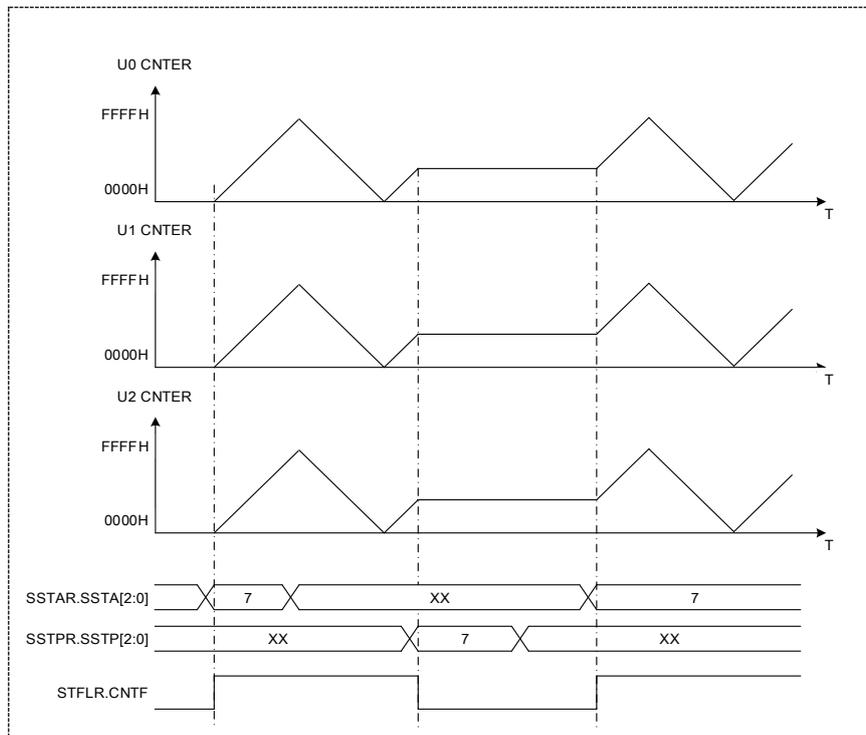


图 12-7 软件同步动作

12.2.5.2 软件同步停止

Timer4/5/6 可通过设定软件同步停止寄存器 (SSTPR) 的相关位, 实现目标 Timer4/5/6 的同步停止。

12.2.5.3 软件同步清零

Timer4/5/6 可通过设定软件同步清零寄存器 (SCLR) 的相关位, 实现目标 Timer4/5/6 的同步清零。

如图 12-7 所示、若设定 Timer4 的 SSTAR.SSTA0=SSTAR.SSTA1=SSTAR.SSTA2,即可实现 Timer4/5/6 的软件同步启动。

软件同步动作相关寄存器 (SSTAR、SSTPR、SCLR) 是一组独立于 Timer4/5/6 外、各个 TIM 间共用的寄存器, 这组寄存器的各个位只在写 1 时有效, 写 0 无效。在读取

SSTAR 寄存器时，会读出各个定时器的计数器状态，在读取 SSTPR 或 SCLRR 时，会读出 0。

12.2.6 硬件同步

每个定时器除独立拥有 2 个通用输入端口（CHxA、CHxB）外，还共同拥有 4 个外部通用输入端口（TIMTRIA、TIMTRIB、TIMTRIC、TIMTRID）及 4 个 AOS 目标，可实现定时器间的硬件同步动作。

12.2.6.1 硬件同步启动

各 Timer4/5/6 均可选择用硬件方式启动计数器，选择相同硬件启动条件的定时器即可在启动条件有效时实现同步启动。具体的硬件启动条件由硬件启动事件选择寄存器（HSTAR）的设定来决定。

12.2.6.2 硬件同步停止

各 Timer4/5/6 均可选择用硬件方式停止计数器，选择相同硬件停止条件的定时器即可在停止条件有效时实现同步停止。具体的硬件停止条件由硬件停止事件选择寄存器（HSTPR）的设定来决定。

12.2.6.3 硬件同步清零

各 Timer4/5/6 均可选择用硬件方式清零计数器，选择相同硬件清零条件的定时器即可在清零条件有效时实现同步清零。具体的硬件清零条件由硬件清零事件选择寄存器（HCLRR）的设定来决定。

12.2.6.4 硬件同步捕获输入

各 Timer4/5/6 均可选择用硬件方式实现捕获输入功能，选择相同捕获输入功能条件的定时器即可在捕获输入功能条件有效时实现同步捕获输入。具体的硬件捕获输入功能条件由硬件捕获事件选择寄存器（HCPAR、HCPBR）的设定来决定。

12.2.6.5 硬件同步计数

Timer4/5/6 均可选择用硬件输入作为 CLOCK 进行计数，选择相同硬件计数条件的定时器即可在硬件计数 CLOCK 有效时实现同步计数。具体的硬件计数条件由硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）的设定来决定。选择硬件同步计数功能时，只是选择了外部输入时钟源，不影响计数器的启动、停止、

清零动作。计数器的启动、停止、清零等还需要单独设定。

图 12-8 所示、Timer4/5/6 的硬件同步动作例。

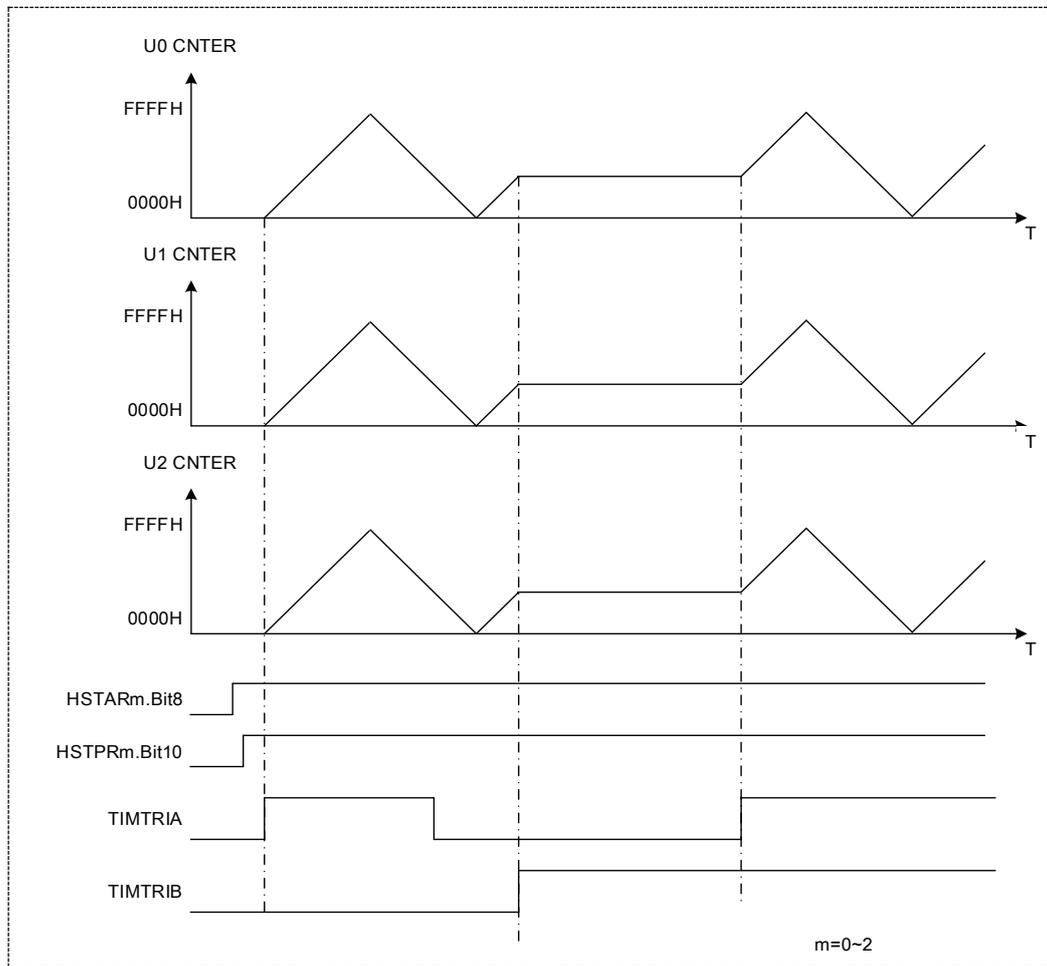


图 12-8 硬件同步动作

12.2.7 缓存功能

缓存动作是指通过设定缓存控制寄存器（BCONR），在缓存传送时间点，选择发生以下事件：

- a. 通用周期基准值缓存寄存器（PERBR）的值自动传送到通用周期基准值寄存器（PERAR）中
- b. 通用比较基准值缓存寄存器（GCMCR、GCMDR）的值自动传送到通用比较基准值寄存器（GCMAR、GCMBR）中（比较输出时）
- c. 通用比较基准值寄存器（GCMAR、GCMBR）的值自动传送到通用比较基准值缓存寄存器（GCMCR、GCMDR）中（捕获输入时）

图 12-9 所示，是比较输出动作时、通用比较基准值寄存器的单缓存方式的时序图。从图中可以看到，在计数期间改变通用比较基准值寄存器（GCMAR）的值可以调整输出占空比，改变通用周期基准值寄存器（PERAR）的值可以调整输出周期。

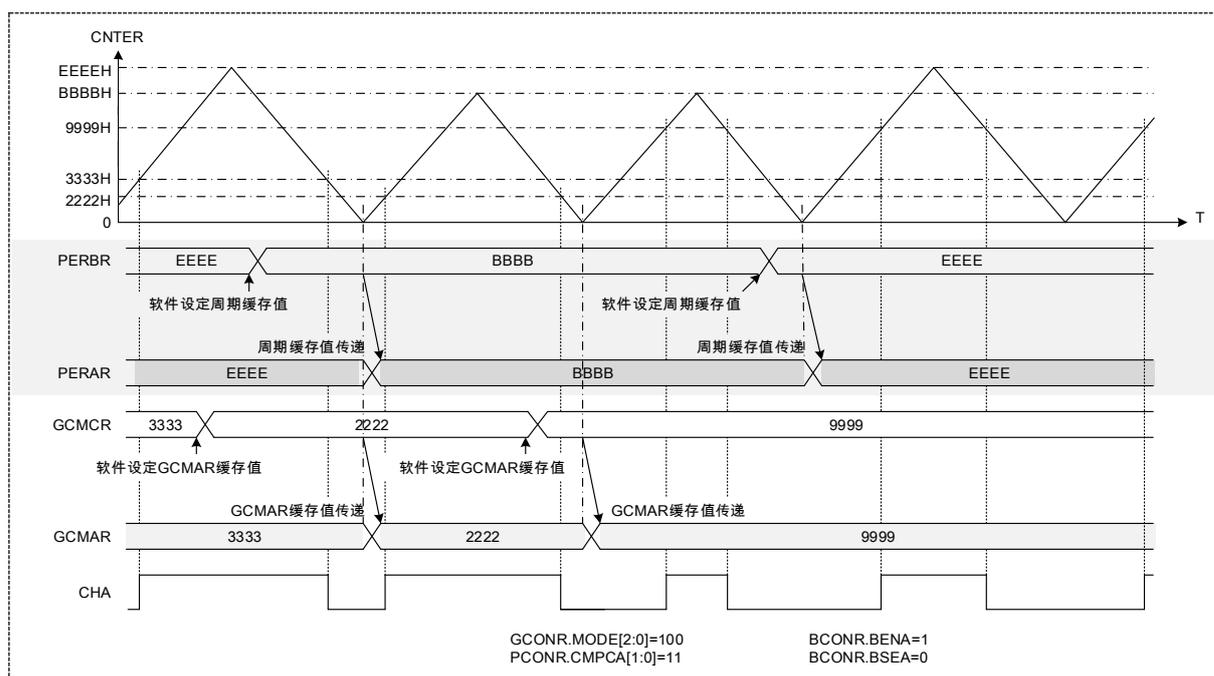


图 12-9 单缓存方式比较输出时序

12.2.7.1 缓存传送时间点

12.2.7.2 通用周期基准值缓存传送时间点

周期基准值缓存传送时间点为锯齿波时递加计数上溢点或递减计数下溢点、三角波时计数谷点。

12.2.7.3 通用比较基准值缓存传送时间点

锯齿波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在上溢点或下溢点。

三角波 A 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在计数谷点。

三角波 B 模式时，设定 BCONR.BENA=1 或 BCONR.BNEB=1，缓存动作有效。缓存传送发生在计数谷点和计数峰点。

12.2.7.4 捕获输入值缓存传送时间点

捕获输入动作缓存传送时间点为捕获输入动作时。

12.2.7.5 清零动作时缓存传送

在锯齿波计数模式或硬件计数模式时，正常的比较输出动作期间若有清零动作产生，通用周期基准值、通用比较基准值、等会根据相应的缓存动作设定状况发生一次缓存传送。

12.2.8 通用 PWM 输出

12.2.8.1 PWM 展频输出

为了降低 PWM 输出对外部的干扰，在 PWM 输出级有展频配置。每个 PWM 输出周期会微调 PWM 输出的相位。

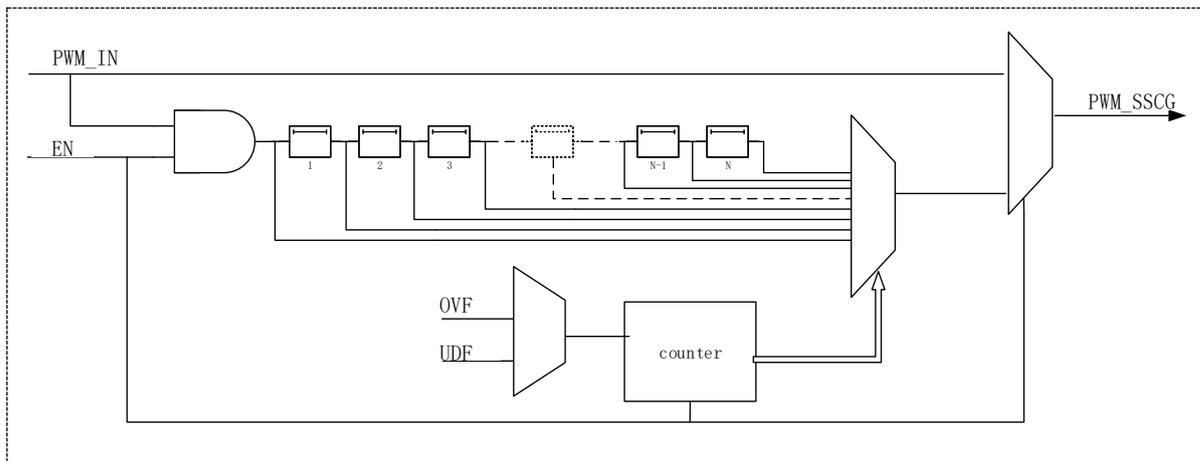


图 12-10 PWM 展频输出示意图

12.2.8.2 独立 PWM 输出

每个定时器的 2 个端口 CHxA、CHxB 能独立的输出 PWM 波。如图 12-11，定时器 Timer6 的 CHA 端口输出 PWM 波。

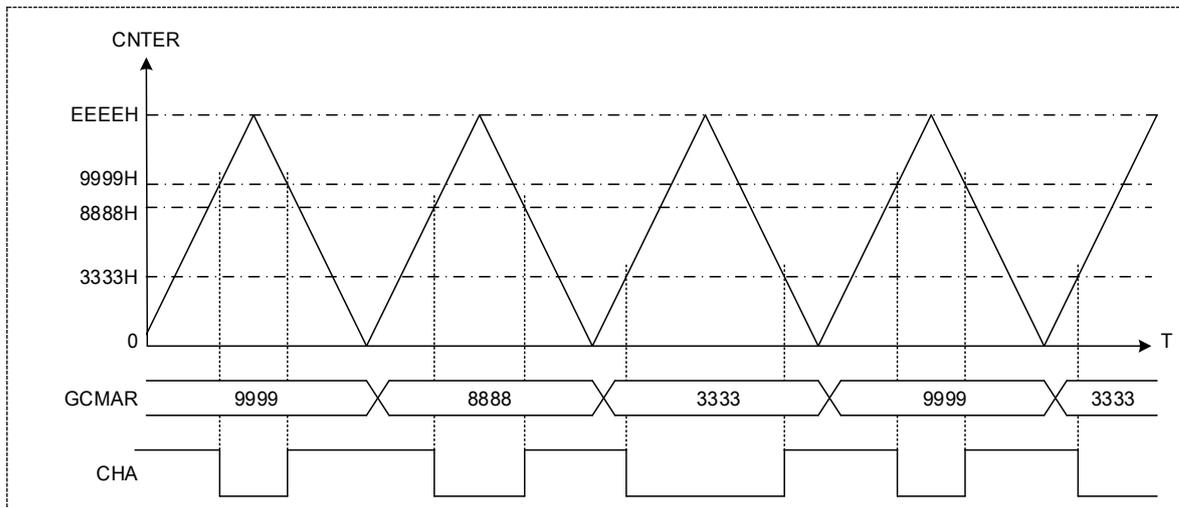


图 12-11 CHA 输出 PWM 波

12.2.8.3 互补 PWM 输出

CHxA 端口和 CHxB 端口，在不同的模式下可组合输出互补 PWM 波形。

12.2.8.3.1 软件设定 GCMBR 互补 PWM 输出

软件设定 GCMBR 互补 PWM 输出是指在锯齿波模式和三角波 A 模式、三角波 B 模式下，用于 CHxB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由寄存器直接设定，与通用比较基准值寄存器（GCMAR）的值没有直接关系。

图 12-12 为软件设定 GCMBR 互补 PWM 波的输出例。

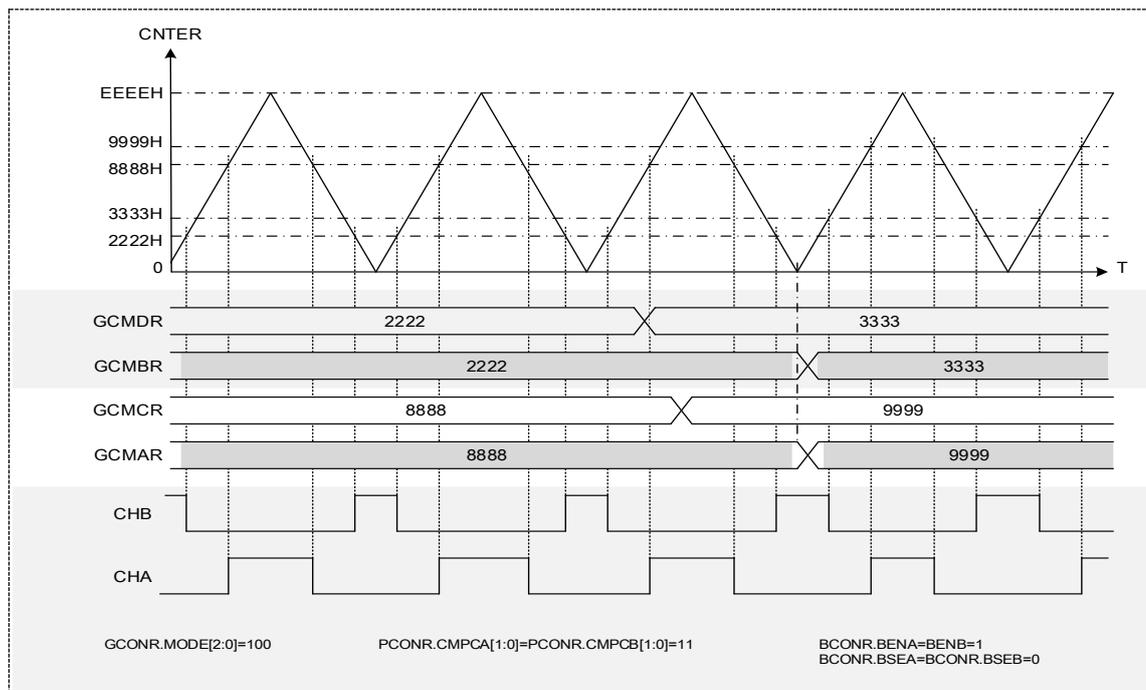


图 12-12 三角波 A 模式时软件设定 GCMBR 互补 PWM 波输出

12.2.8.3.2 硬件设定 GCMBR 互补 PWM 输出

硬件设定 GCMBR 互补 PWM 输出是指在三角波 A 模式、三角波 B 模式下，用于 CHxB 端口波形输出的通用比较基准值寄存器（GCMBR）的值由通用比较基准值寄存器（GCMAR）和死区时间基准值寄存器（DTUAR、DTDAR）的值运算决定。

图 12-13 为硬件设定 GCMBR 互补 PWM 波输出例。

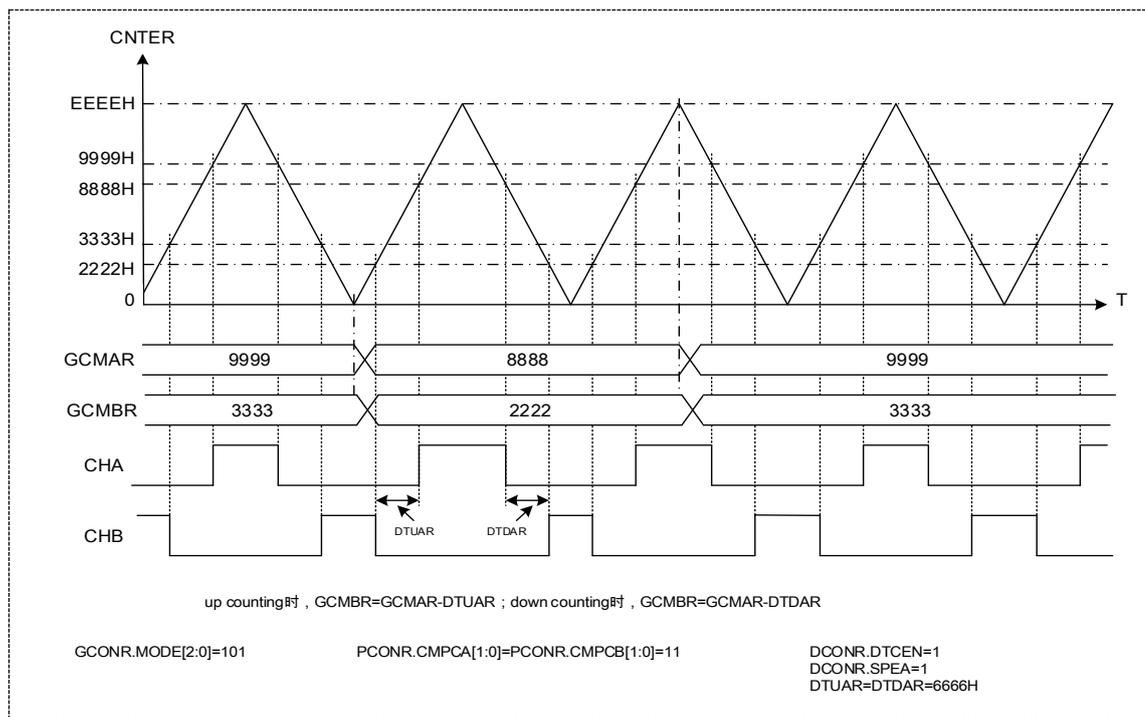


图 12-13 三角波 B 模式时硬件设定 GCMBR 互补 PWM 波输出（对称死区）

12.2.8.4 多相 PWM 输出

每个定时器的 CHxA、CHxB 端口都能输出 2 相独立的 PWM 波或一组互补 PWM 波，多个定时器间组合，同时结合软件、硬件同步动作就可实现多相 PWM 波输出。

如图 12-14, Timer4, Timer5, Timer6 组合输出 6 相 PWM 波；如图 12-15, Timer4, Timer5, Timer6 组合输出 3 相互补 PWM 波。

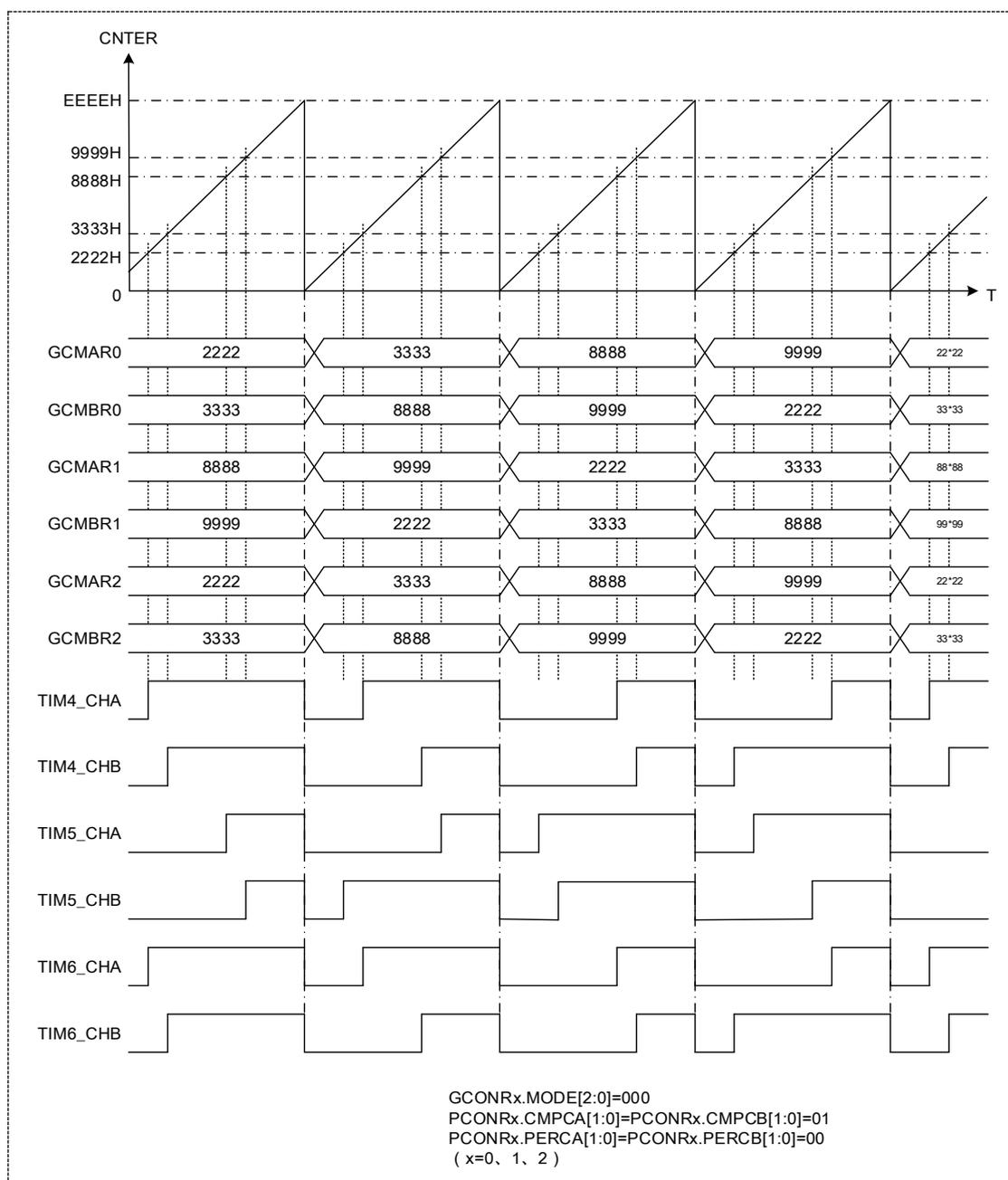


图 12-14 6 相 PWM 波

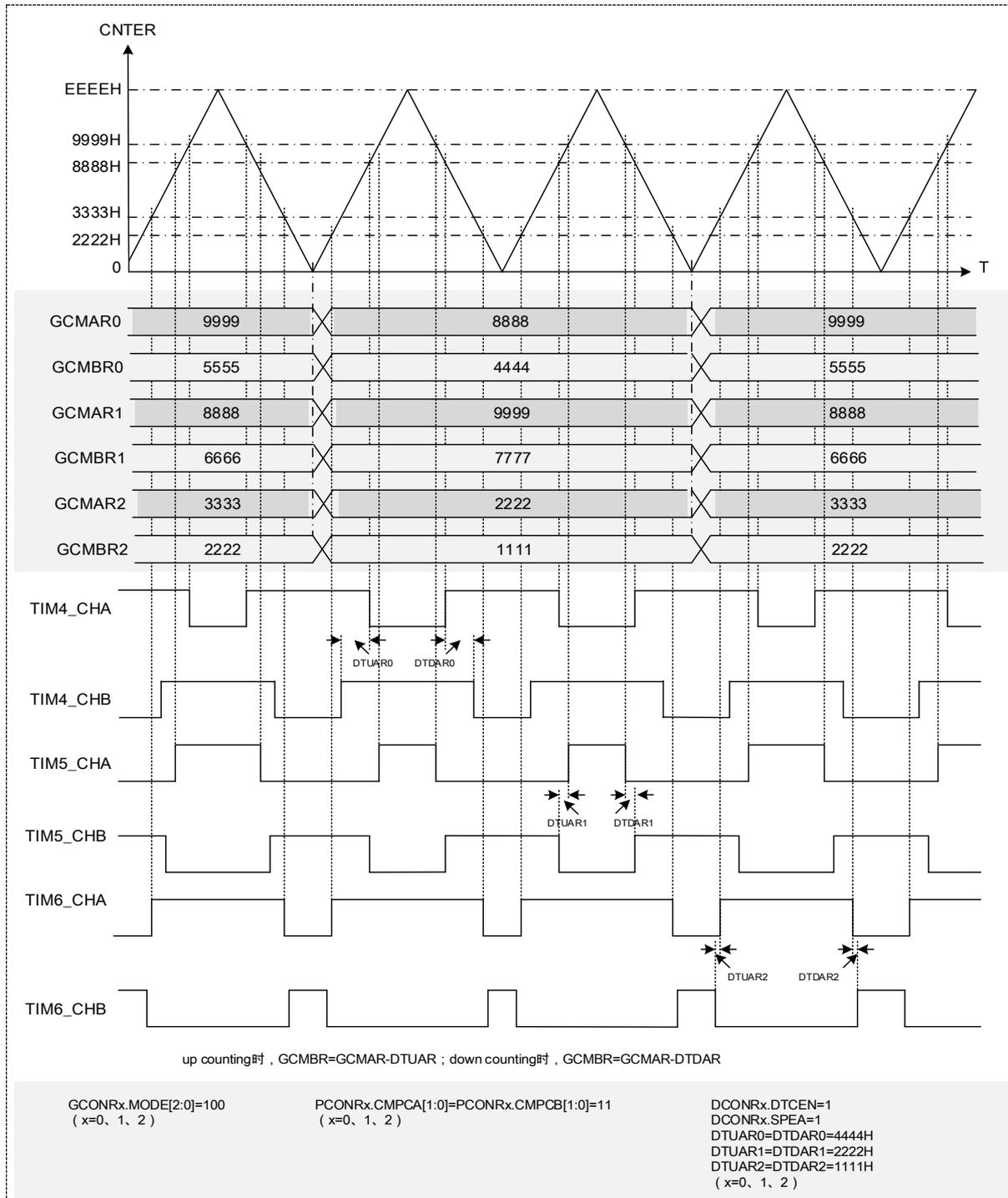


图 12-15 三角波 A 模式时带死区时间三相互补 PWM 波输出

12.2.9 正交编码计数

将 CHxA 输入看作 AIN 输入、CHxB 输入看作 BIN 输入、TIMTRIA-D 中的任意一个输入看作 ZIN 输入，Advanced Timer 就可以实现三路输入的正交编码计数。

一个定时器的 AIN、BIN 单独动作可以实现位置计数模式；两个定时器的 AIN、BIN、ZIN 组合动作可以实现公转计数模式，一个定时器用于位置计数，一个定时器用于公转计数。

公转计数模式时，每两个定时器组合（定时器 4、5 组合，定时器 4 作为位置计数单元，定时器 5 作为公转计数单元）分别实现位置计数和公转计数。

AIN 和 BIN 的计数条件通过设定硬件递加事件选择寄存器（HCUPR）和硬件递减事件选择寄存器（HCDOR）中 CHxA 和 CHxB 的正交关系实现；ZIN 的输入动作通过设定位置单元的硬件清零事件选择寄存器（HCLRR）实现位置计数单元的位置计数器清零、通过设定公转单元的硬件递加事件选择寄存器（HCUPR）实现公转计数单元的公转计数器计数。

12.2.9.1 位置计数模式

正交编码位置模式，是指根据 AIN、BIN 的输入实现基本计数功能、相位差计数功能和方向计数功能。

12.2.9.1.1 基本计数

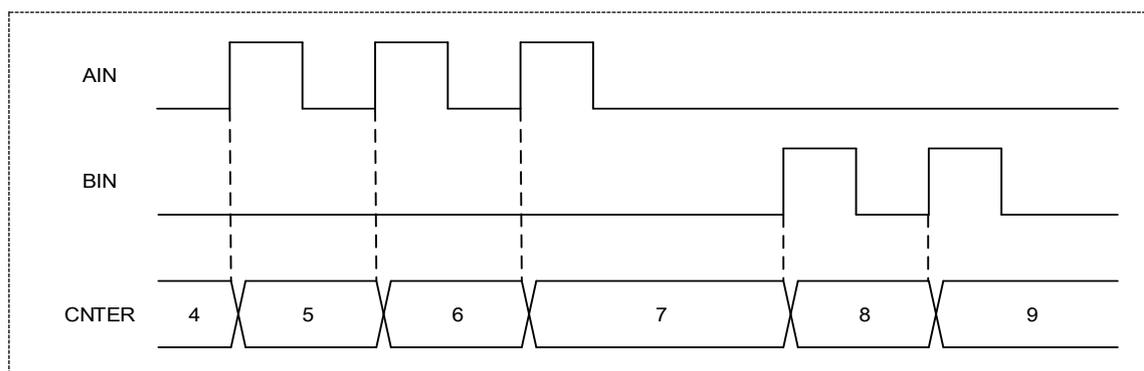


图 12-16 位置模式时基本计数动作

通过设定 HCUPR 与 HCDOR 寄存器，可以灵活的实现各种方式的相位差计数。

12.2.9.1.2 相位差计数

相位差计数是指根据 AIN 和 BIN 的相位关系进行计数。根据设定的不同，可以实现 1 倍计数、2 倍计数、4 倍计数等，如下图图 12-17~图 12-19 所示。

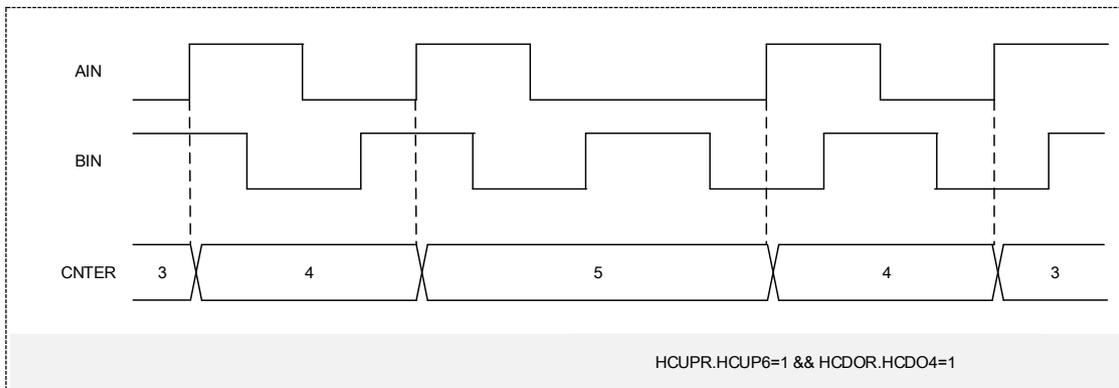


图 12-17 位置模式时相位差计数动作设定(1 倍)

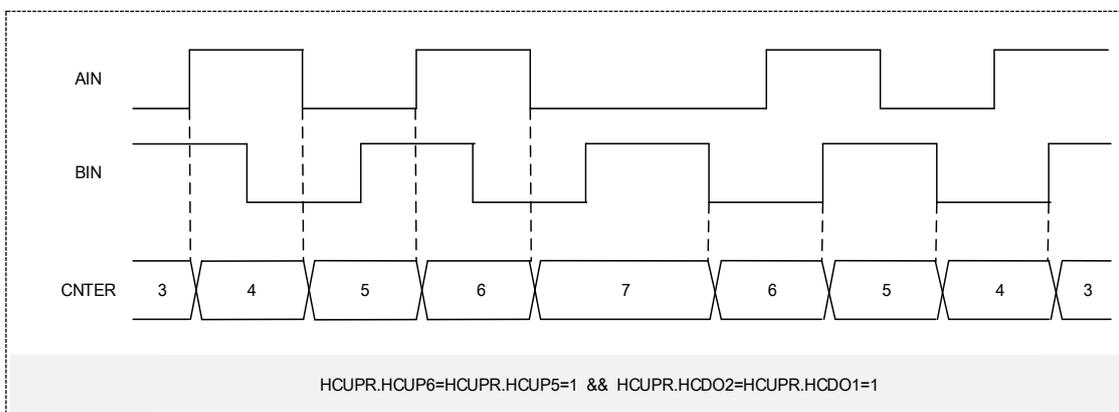


图 12-18 位置模式时相位差计数动作设定(2 倍)

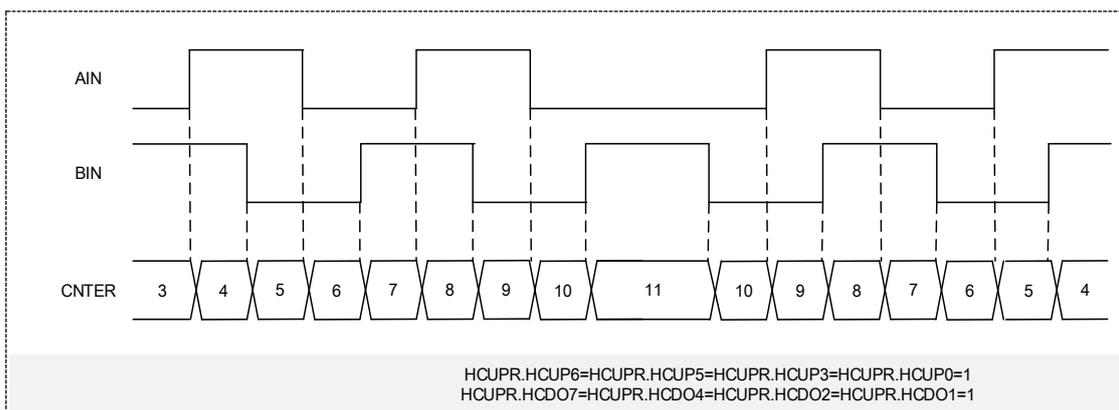


图 12-19 位置模式时相位差计数动作设定(4 倍)

12.2.9.1.3 方向计数

方向计数是指将 AIN 的输入状态设定为方向控制，将 BIN 的输入作为时钟计数，如图 12-20 所示。

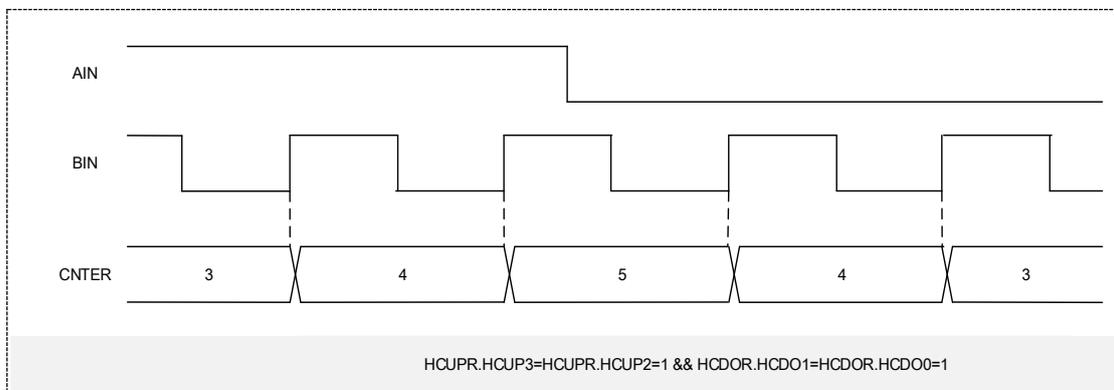


图 12-20 位置模式时方向计数动作

12.2.9.2 公转模式

正交编码公转模式，是指在 AIN、BIN 计数的基础上，加入 ZIN 的输入事件以实现公转圈数等的判断。公转模式时根据公转计数器的计数方式，可实现 Z 相计数功能、位置计数器输出计数功能和 Z 相计数与位置计数器输出混合计数功能。即使用两个 Advanced Timer 实现此功能。

12.2.9.2.1 Z 相计数

Z 相计数是指根据 ZIN 的输入，公转计数单元进行计数，同时将位置计数单元清零的计数动作。

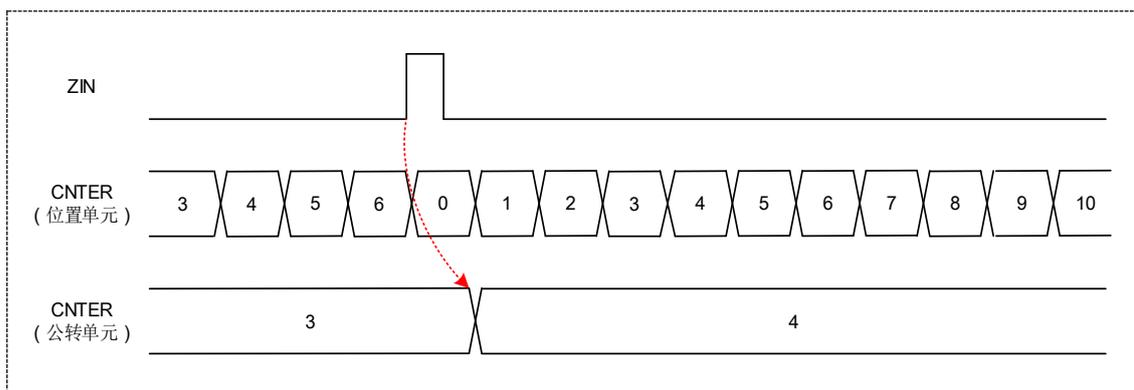


图 12-21 公转模式时 Z 相计数动作

12.2.9.2.2 位置溢出计数

位置溢出计数是指位置计数单元计数发生上溢或下溢时，产生一个溢出事件，从而触发公转计数单元的计数器进行一次计数（在该计数方式时 ZIN 的输入不进行公转计数单元的计数动作和位置计数单元的清零动作）。

位置计数单元的溢出事件通过 AOS 模块的联动选通实现公转计数单元计数，即可实现位置溢出计数。公转计数单元的硬件递加（递减）事件选择寄存器（HCUPR 或 HCDOR）的递加（递减）事件选择 Bit16:it19 中的 1 位，同时 AOS 模块设定对应的递加（递减）事件的事件源为位置计数单元的计数溢出事件，具体请参考 AOS 章节。如图 12-22 所示。

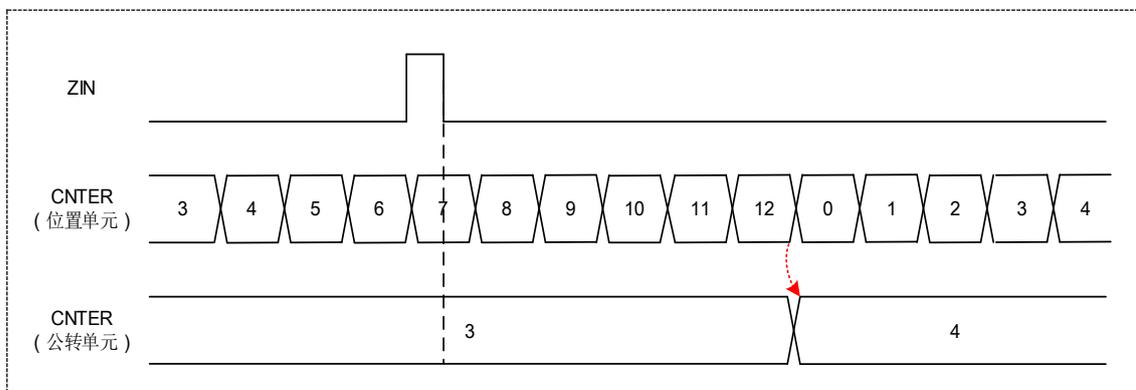


图 12-22 公转模式时位置计数器输出计数动作

12.2.9.2.3 混合计数

混合计数是指上述 Z 相计数和位置溢出计数两种计数方式合并起来的计数动作，其实现方式也是上述两种计数方式的组合。如图 12-23 所示。

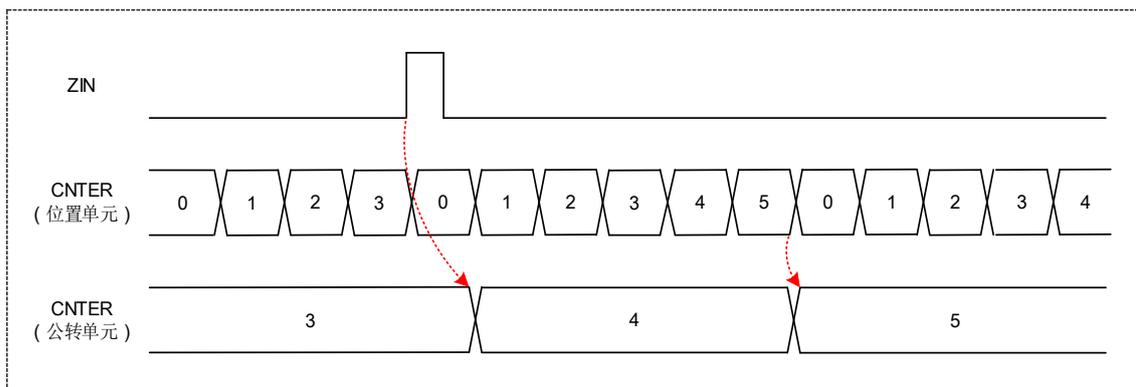


图 12-23 公转模式时 Z 相计数和位置计数器输出混合计数动作

12.2.9.2.4 Z 相动作屏蔽

在公转计数模式的 Z 相计数功能或混合计数功能时，可以设定在位置计数器的上溢点或下溢点后的几个周期内（GCONR.ZMSK[0:1]设定），将 ZIN 的有效输入屏蔽，不进行公转计数单元的计数和位置计数单元的清零。

位置计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKPOS 为 1 时，位置计数单元的 Z 相屏蔽功能使能，Z 相屏蔽的周期数由 GCONR.ZMSK 设定；公转计数单元的通用控制寄存器（GCONR）的 GCONR.ZMSKREV 为 1 时，公转计数单元的 Z 相屏蔽功能使能。

图 12-24 是公转计数模式混合计数时，在位置计数单元计数上溢后的 4 个计数周期内有 ZIN 相输入时，ZIN 相输入的动作无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

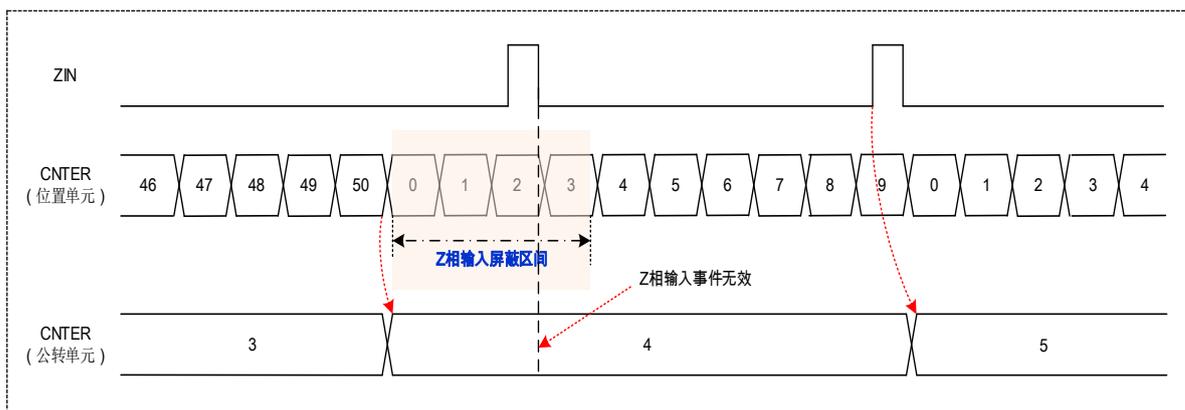


图 12-24 公转计数模式-混合计数 Z 相屏蔽动作例 1

图 12-25 是公转计数模式混合计数时，在位置计数单元计数上溢后的第 3 个周期，计数方向发生变化，此时设定的 4 个周期的屏蔽周期变为无效（实际 ZIN 相屏蔽功能维持了 3 个周期），开始向下计数。在位置计数单元发生计数下溢后，ZIN 相屏蔽功能重新开启，维持 4 个周期后变为无效。在 ZIN 相屏蔽期间，ZIN 相的输入功能无效，即公转计数单元不计数、位置计数单元不清零；之后再来的 ZIN 相输入正常动作。

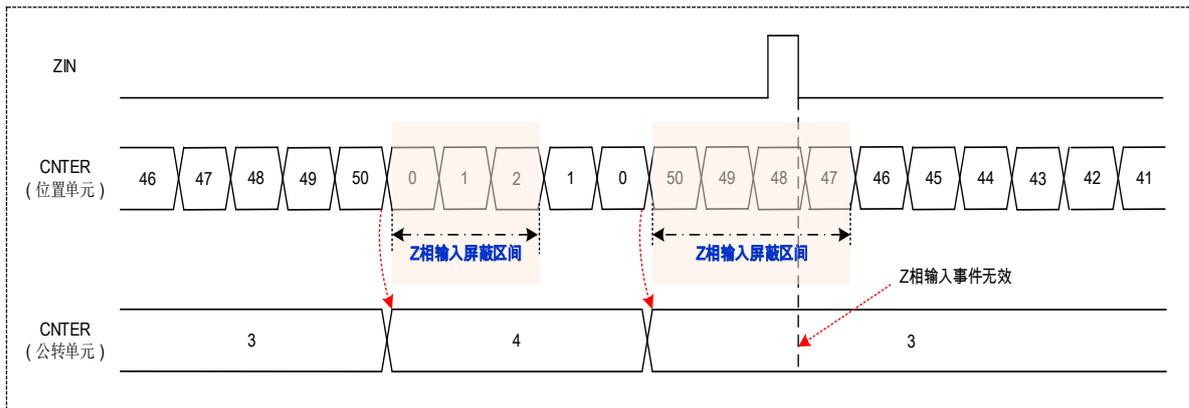


图 12-25 公转计数模式-混合计数 Z 相屏蔽动作例 2

12.2.10 周期间隔响应

Timer4/5/6 的通用比较基准值寄存器 (GCMAR~GCMDR)，在计数比较匹配时可分别产生专用有效请求信号，送到 AOS 模块中用于和其它模块关联动作。

该请求信号可以每间隔几个周期后产生一次有效的请求信号。通过设定有效周期寄存器 (VPERR) 的 VPERR.PCNTS 位来指定每隔多少个周期请求信号有效一次，其它周期内即使计数值和比较基准值寄存器 GCMAR 或 GCMBR 的值相等，也不会输出有效的请求信号。图 12-26 所示是周期间隔有效请求信号的动作例。

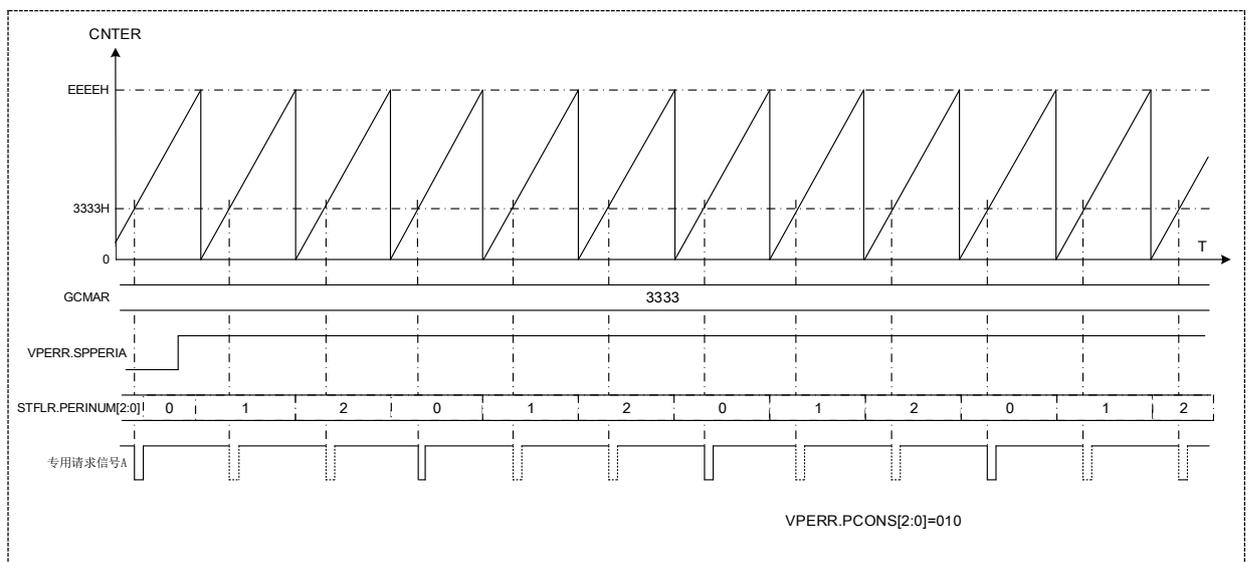


图 12-26 周期间隔有效请求信号动作

12.2.11 保护机制

Advanced Timer 可以对端口的输出状态进行保护控制。

Advanced Timer 有 4 个共用的端口输出无效事件接口，这 4 个接口连接刹车控制模块输出的 4 组刹车事件。每个接口上选通的异常状况事件可从刹车控制设定，当这些接口上监测到异常状况时，可以实现对通用 PWM 输出的控制。

端口在正常输出期间，若监测到从刹车控制过来的刹车事件，则端口的输出状态可变为预先设定好的状态。通用 PWM 输出端口在刹车控制异常事件发生时，端口状态可以变为输出高阻态、输出低电平或输出高电平（PCONR.DISVALA、PCONR.DISVALB 的设定决定）。

例如，若 PCONR.DISSELA[1:0]=01&PCONR.DISVALA=01 设定时，则在 CHxA 端口正常输出期间，若输出无效条件 1 上产生刹车事件，则 CHxA 端口上输出变为高阻态。

12.2.12 中断说明

Timer4/5/6 各含有 3 类共计 9 个中断。分别是 4 个通用计数比较匹配中断（含 2 个捕获输入中断）、2 个计数周期匹配中断、1 个死区时间错误中断。

12.2.12.1 计数比较匹配中断

通用比较基准值寄存器（GCMAR-GCMDR）共计 4 个，可分别与计数值比较产生比较匹配有效信号。计数比较匹配时，状态标志寄存器（STFLR）中的 STFLR.CMAF~STFLR.CMDF 位分别会被置为 1。此时若设定中断控制寄存器（ICONR）的 ICONR.INTENA~ICONR.INTEND 中相应位为 1 使能中断，则对应的中断请求也会被触发。

在硬件捕获事件选择寄存器（HCPAR、HCPBR）选择的捕获输入有效条件产生时，捕获输入动作发生。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENA 或 ICONR.INTENB 位为 1 使能中断，则对应的中断请求被触发。

12.2.12.2 计数周期匹配中断

锯齿波递加计数至上溢点、锯齿波递减计数至下溢点、三角波计数至谷点或三角波计数至峰点时，状态标志寄存器（STFLR）的 STFLR.OVFF 或 STFLR.UDFF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENOVF 位与 ICONR.INTENUDF 位使能中断，则在对应的时间点可触发计数周期匹配中断。

12.2.12.3 死区时间错误中断

将死区时间基准值寄存器（DTUAR、DTDAR）的值加载到通用比较基准值寄存器（GCMBR）中时，若超过周期限制，则会产生死区时间错误，状态标志寄存器（STFLR）的 STFLR.DTEF 位会被置为 1。此时若设置中断控制寄存器（ICONR）的 ICONR.INTENDE 位使能中断，则会在该时刻触发死区时间错误中断。

12.2.13 刹车保护

当可以设定无效条件 0~3，配置 PCONR.DISVALA, PCONR.DISVALB。无效条件有效时硬件自动将端口状态改变为预设状态(高电平, 低电平, 高阻态, 保持正常输出)。

12.2.13.1 端口刹车与软件刹车

端口经过极性选择控制，有效使能后，经过数字滤波，同步，产生端口刹车标志；端口刹车标志做为 Advanced Timer 的无效条件 3。端口刹车标志需要软件清除。

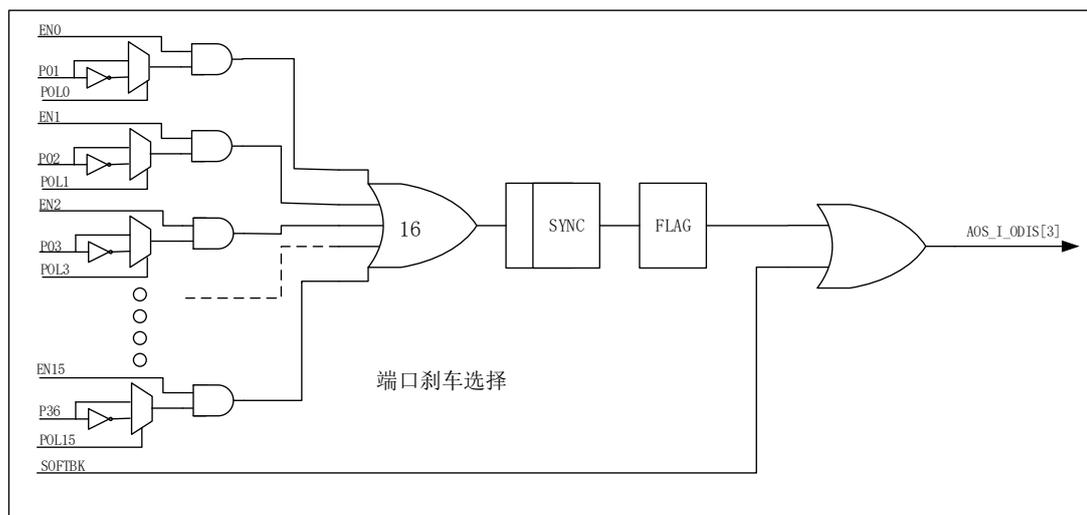


图 12-27 端口刹车与软件刹车示意图

12.2.13.2 低功耗模式自动刹车

系统进入低功耗模式，时钟停止后 PWM 将不能正常工作。低功耗模式作为 Advanced Timer 的无效条件 2 控制 PWM 刹车。

12.2.13.3 输出电平同高同低刹车

输出电平经过电平监测，有效使能后，经过同步，产生同高同低刹车标志；端口刹车标志做为 Advanced Timer 的无效条件 1。同高同低刹车标志需要软件清除。

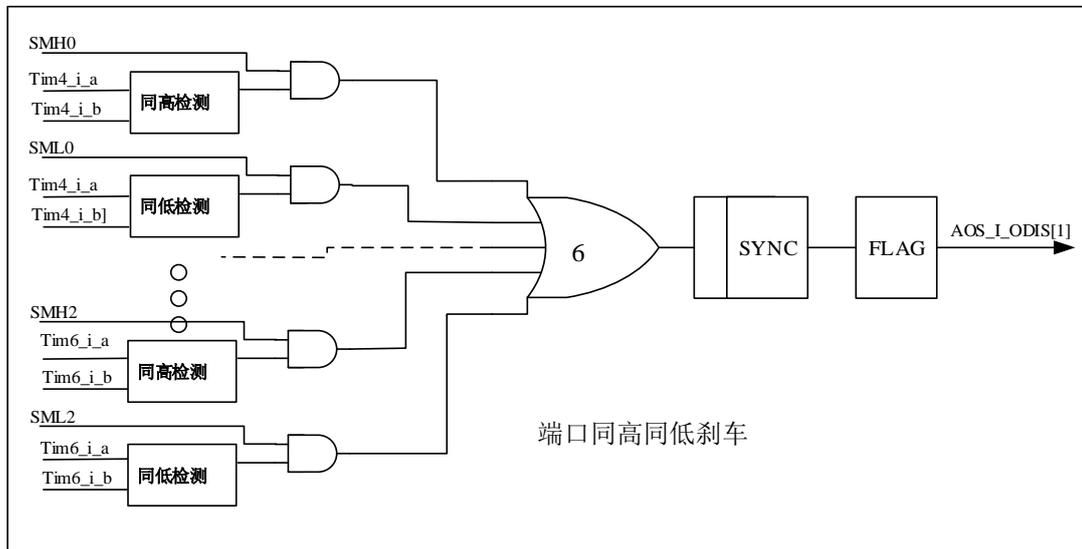


图 12-28 输出同高同低刹车示意图

12.2.13.4 VC 刹车

VC1, VC2 中断标志经过使能后作为 Advanced Timer 的无效条件 0。

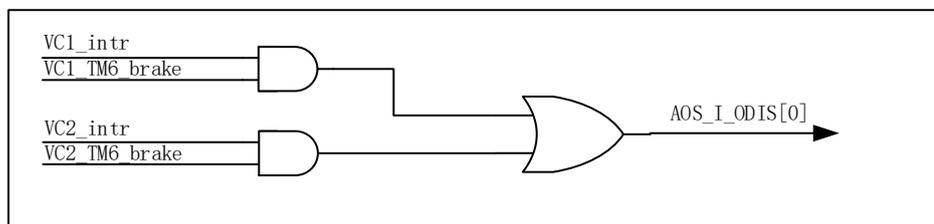


图 12-29 VC 刹车控制示意图

12.2.14 内部互连

12.2.14.1 中断触发输出

由于 Timer4/5/6 的一个中断包含多个中断源。控制触发 ADC 与控制 AOS 的中断信号有单独控制可以选择不同的源，可以选择上溢出，下溢出，4 个比较匹配共 6 个 TIMx 的中断源的任意中断源作为触发条件。

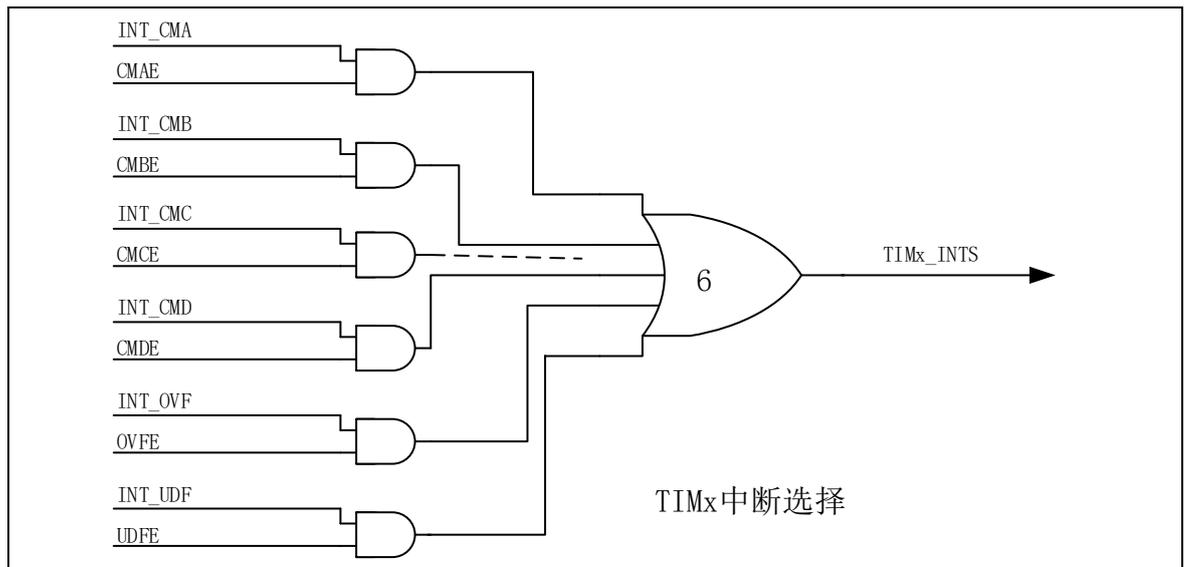


图 12-30 Timer4/5/6 中断选择

12.2.14.2 AOS 触发

AOS 是系统的内部信号，通过选择控制后可以触发 Advanced Timer 的计数器的开始，停止，清零，加 1，减 1 等功能。Advanced Timer 有 4 路 AOS 触发，每路触发可以选择不同模块的中断源。选择后的信号产生单脉冲触发输入到 Advanced Timer，控制 Advanced Timer 的计数器的开始、停止、清零。

Timer4/5/6 内部使用寄存器选择不同的 AOS_I_TRIG 作为自己的触发信号。如可使用 HSTAR 寄存器可以使用中断触发相应定时器的硬件启动。

	AOS_i_trig0	AOS_i_trig1	AOS_i_trig2	AOS_i_trig3
选择控制信号	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
0010	TIM2_INT	TIM2_INT	TIM2_INT	TIM2_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

	AOS_i_trig0	AOS_i_trig1	AOS_i_trig2	AOS_i_trig3
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

表 12-3 AOS 源选择

12.2.14.3 端口触发 TRIGA-TRIGD

端口触发可以控制 Advanced Timer 的硬件启动、停止、清零、捕获、计数器加减计数等功能，有数字滤波功能可选，端口可以配置为芯片的任意一个端口上。

选择控制信号分别独立控制	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

表 12-4 端口触发选择

12.2.14.4 比较输出 VC 与 Advanced Timer 互连

VC 内部可以互连到 Advanced Timer 的捕获输入端,可以对 VC 输出的边沿进行捕获;

12.2.14.5 UART 与 Advanced Timer 互连

UARTx_RX / LPUART_RX 可以通过内部互连到 BaseTimer, LPTimer,PCA 与 Advanced Timer。通过软件可以实现波特率的自动识别。

UART 选择控制寄存器在端口控制寄存器中 GPIO_CTRL3, VC 输出控制寄存器在 VC 控制模块。

12.3 寄存器描述

CH0 基地址 0x40003000

CH1 基地址 0x40003400

CH2 基地址 0x40003800

寄存器	偏移地址	描述
TIMx_CNTER	0x000	通用计数基准值寄存器
TIMx_PERAR	0x004	通用周期基准值寄存器
TIMx_PERBR	0x008	通用周期基准值缓存寄存器
TIMx_GCMAR	0x010	通用比较A基准值寄存器
TIMx_GCMBR	0x014	通用比较B基准值寄存器
TIMx_GCMCR	0x018	通用比较C基准值寄存器
TIMx_GCMDR	0x01C	通用比较D基准值寄存器
TIMx_DTUAR	0x040	死区时间基准值寄存器
TIMx_DTDAR	0x044	死区时间基准值寄存器
TIMx_GCONR	0x050	通用控制寄存器
TIMx_ICONR	0x054	中断控制寄存器
TIMx_PCONR	0x058	端口控制寄存器
TIMx_BCONR	0x05C	缓存控制寄存器
TIMx_DCONR	0x060	死区控制寄存器
TIMx_FCONR	0x068	滤波控制寄存器
TIMx_VPERR	0x06C	有效周期寄存器
TIMx_STFLR	0x070	状态标志寄存器
TIMx_HSTAR	0x074	硬件启动事件选择寄存器
TIMx_HSTPR	0x078	硬件停止事件选择寄存器
TIMx_HCELR	0x07C	硬件清零事件选择寄存器
TIMx_HCPAR	0x080	硬件捕获事件选择寄存器
TIMx_HCPBR	0x084	硬件捕获事件选择寄存器
TIMx_HCUPR	0x088	硬件递减事件选择寄存器
TIMx_HCDOR	0x08C	硬件递减事件选择寄存器
TIMx_IFR	0x100	中断标志寄存器
TIMx_ICLR	0x104	中断清除寄存器
TIMx_CR	0x108	展频及中断触发选择寄存器
TIMx_AOSSR	0x110	AOS 选择寄存器, 三个通道共用
TIMx_AOSCL	0x114	AOS 刹车标志清除寄存器, 三个通道共用
TIMx_PTBKS	0x118	端口刹车控制寄存器, 三个通道共用
TIMx_TTRIG	0x11C	端口触发控制寄存器, 三个通道共用
TIMx_ITRIG	0x120	AOS 触发控制寄存器, 三个通道共用

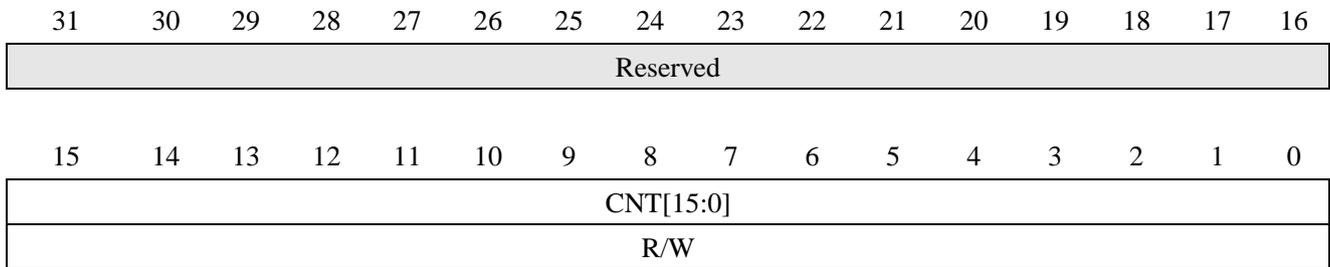
寄存器	偏移地址	描述
TIMx_PTBKP	0x124	端口刹车极性控制寄存器, 三个通道共用
TIMx_SSTAR	0x3F4	软件同步启动寄存器
TIMx_SSTPR	0x3F8	软件同步停止寄存器
TIMx_SCLRR	0x3FC	软件同步清零寄存器

表 12-5 Advanced Timer 寄存器列表

12.3.1 通用计数基准值寄存器 (TIMx_CNTER)

地址偏移量: 0x000

复位值: 0x0000 0000

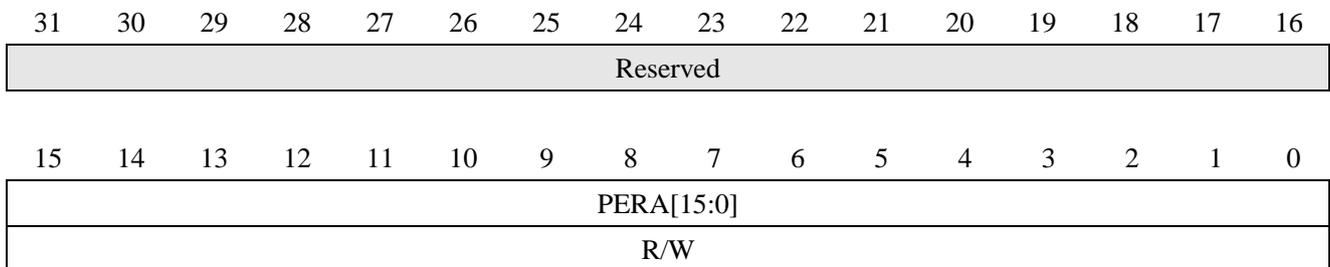


位	符号	功能描述
31:16	Reserved	-
15:0	CNT[15:0]	当前计数器的计数值

12.3.2 通用周期基准值寄存器 (TIMx_PERAR)

地址偏移量: 0x004

复位值: 0x0000 FFFF

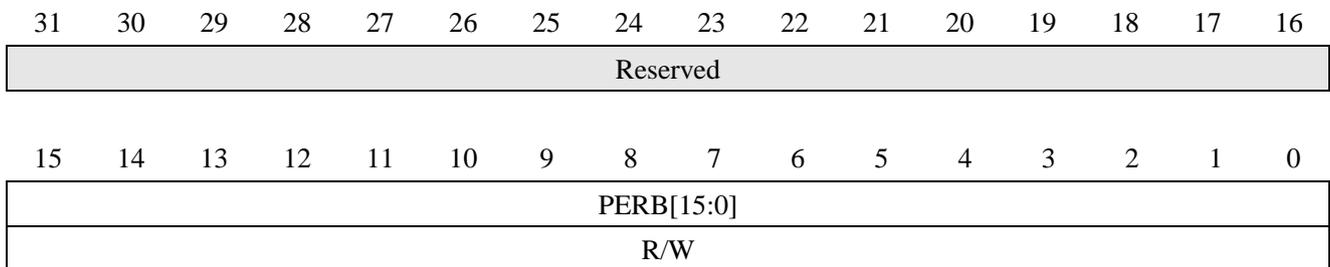


位	符号	功能描述
31:16	Reserved	-
15:0	PERA[15:0]	计数周期值, 设定每轮计数的计数周期值

12.3.3 通用周期缓存寄存器 (TIMx_PERBR)

地址偏移量: 0x008

复位值: 0x0000 FFFF

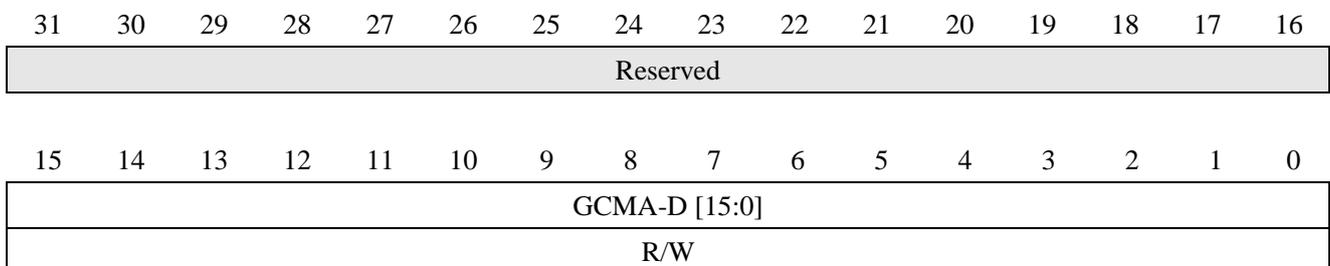


位	符号	功能描述
31:16	Reserved	-
15:0	PERB[15:0]	缓存计数周期值, 计数周期的缓存值

12.3.4 通用比较基准值寄存器 (TIMx_GCMAR-GCMDR)

地址偏移量: 0x0010, 0x0014, 0x0018, 0x001C

复位值: 0x0000 FFFF

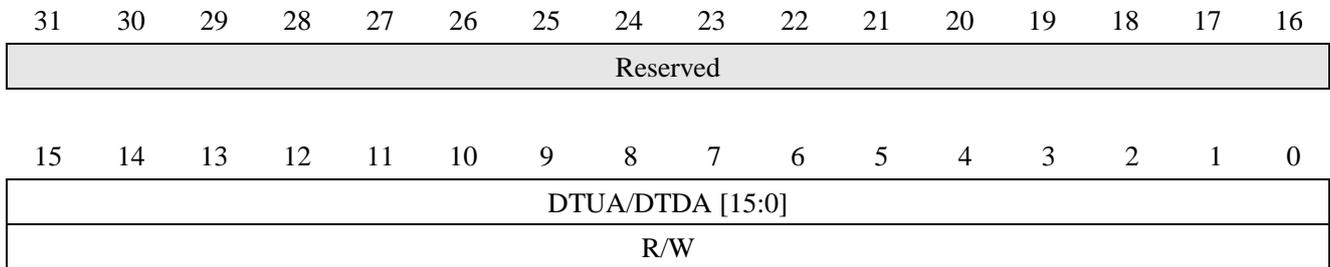


位	符号	功能描述
31:16	Reserved	-
15:0	GCMA-D [15:0]	计数比较基准值, 比较基准值设定, 与计数值相等时匹配信号有效

12.3.5 死区时间基准值寄存器 (TIMx_DTUAR- DTDAR)

地址偏移量: 0x040, 0x044

复位值: 0x0000 FFFF



位	符号	功能描述
31:16	Reserved	-
15:0	DTUA/DA [15:0]	死区时间值, 死区时间设定值

12.3.6 通用控制寄存器 (TIMx_GCONR)

地址偏移量: 0x050

复位值: 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		
Reserved												DIR	Res.	CKDIV	MODE	START
												R/W		R/W	R/W	R/W

位	标记	功能描述
31:20	Reserved	-
19:18	ZMSK	Z相输入屏蔽周期数 正交编码Z相输入被屏蔽的计数周期值 00: Z相输入屏蔽功能无效 01: 位置计数上溢后或下溢后的4个计数周期内的Z相输入被屏蔽 10: 位置计数上溢后或下溢后的8个计数周期内的Z相输入被屏蔽 11: 位置计数上溢后或下溢后的16个计数周期内的Z相输入被屏蔽
17	ZMSKPOS	Z相输入位置计数器选择 0: Z相输入时该定时器作为位置计数器, 在屏蔽周期期间内位置计数器清零功能正常动作 1: Z相输入时该定时器作为位置计数器, 在屏蔽周期期间内位置计数器清零功能被屏蔽
16	ZMSKREV	Z相输入公转计数器选择 0: Z相输入时该定时器作为公转计数器, 在屏蔽周期期间内公转计数器计数功能正常动作 1: Z相输入时该定时器作为公转计数器, 在屏蔽周期期间内公转计数器计数功能被屏蔽
15:9	Reserved	-
8	DIR	计数方向 0: 递减计数; 1: 递加计数
7	Reserved	-
6:4	CKDIV	计数时钟选择 000: PCLK0 001: PCLK0/2 010: PCLK0/4 011: PCLK0/8 100: 101: PCLK0/64 110: PCLK0/256 111: PCLK0/1024 PCLK0/16
3:1	MODE	计数模式 000: 锯齿波A模式 100: 三角波A模式 101: 三角波B模式

		请不要设定其它值
0	START	计数器启动 0: 计数器关闭; 1: 计数器启动 <i>注: 该位在软件停止条件或硬件停止条件有效时, 会自动变为0</i>

12.3.7 中断控制寄存器 (TIMx_ICONR)

地址偏移量: 0x054

复位值: 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			
Reserved				R/W	R/W	R/W	Reserved		R/W	R/W	R/W	R/W			

位	标记	功能
31:9	Reserved	-
8	INTENDE	死区时间错误中断使能 0: 死区时间错误时, 该中断无效 1: 死区时间错误时, 该中断使能
7	INTENUDF	下溢中断使能 0: 锯齿波时下溢发生或三角波时计数到谷点, 该中断无效 1: 锯齿波时下溢发生或三角波时计数到谷点, 该中断使能
6	INTENOVF	上溢中断使能 0: 锯齿波时上溢发生或三角波时计数到峰点, 该中断无效 1: 锯齿波时上溢发生或三角波时计数到峰点, 该中断使能
5:4	Reserved	-
3	INTEND	计数匹配中断使能D 0: GCMDR 寄存器与计数值相等时, 该中断无效 1: GCMDR 寄存器与计数值相等时, 该中断使能
2	INTENC	计数匹配中断使能C 0: GCMCR 寄存器与计数值相等时, 该中断无效 1: GCMCR 寄存器与计数值相等时, 该中断使能
1	INTENB	计数匹配中断使能B 0: GCMBR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: GCMBR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能
0	INTENA	计数匹配中断使能A 0: GCMAR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断无效 1: GCMAR 寄存器与计数值相等时, 或者发生捕获输入事件时, 该中断使能

12.3.8 端口控制寄存器 (TIMx_PCONR)

地址偏移量: 0x058

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DISVALB	DISSELB	OUTENB	PERCB	CMPCB	STASTPS	STPC	STAC	CAP					
		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
Reserved		DISVALA	DISSELA	OUTENA	PERCA	CMPCA	STASTPS	STPC	STAC	CAP					
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W					

位	标记	功能
31:29	Reserved	-
28:27	DISVALB	CHxB 输出状态控制 00: 强制输出无效条件0~3中被选择的条件成立时, CHxB 端口正常输出 01: 强制输出无效条件0~3中被选择的条件成立时, CHxB 端口输出高阻态 10: 强制输出无效条件0~3中被选择的条件成立时, CHxB 端口输出低电平 11: 强制输出无效条件0~3中被选择的条件成立时, CHxB 端口输出高电平
26:25	DISSELB	强制输出无效条件选择B 00: 选择强制输出无效条件0; 01: 选择强制输出无效条件1 10: 选择强制输出无效条件2; 11: 选择强制输出无效条件3
24	OUTENB	输出使能B 0: Advanced Timer 功能时的 CHxB 端口输出无效 1: Advanced Timer 功能时的 CHxB 端口输出有效
23:22	PERCB	周期值匹配时端口状态设定B 00: 计数器计数值与周期值相等时, CHxB 端口输出保持为低电平 01: 计数器计数值与周期值相等时, CHxB 端口输出设定为高电平 10: 计数器计数值与周期值相等时, CHxB 端口输出设定为先前状态 11: 计数器计数值与周期值相等时, CHxB 端口输出设定为反转电平
21:20	CMPCB	比较值匹配时端口状态设定B 00: 计数器计数值与 GCMBR 相等时, CHxB 端口输出保持为低电平 01: 计数器计数值与 GCMBR 相等时, CHxB 端口输出设定为高电平 10: 计数器计数值与 GCMBR 相等时, CHxB 端口输出设定为先前状态 11: 计数器计数值与 GCMBR 相等时, CHxB 端口输出设定为反转电平
19	STASTPSB	计数开始停止端口状态选择B 0: 计数开始或停止时, CHxB 端口输出由 STACB、STPCB 决定 1: 计数开始或停止时, CHxB 端口输出设定为先前状态 <i>注: 此处的计数开始是指初始计数开始或停止再开始; 计数停止是指初始时停止或计数开始后再停止</i>

18	STPCB	计数停止端口状态设定B 0: 计数停止时, CHxB 端口输出设定为低电平 1: 计数停止时, CHxB 端口输出设定为高电平
17	STACB	计数开始端口状态设定B 0: 计数开始时, CHxB 端口输出设定为低电平 1: 计数开始时, CHxB 端口输出设定为高电平
16	CAPCB	功能模式选择B 0: 比较输出功能; 1: 捕获输入功能
15:13	Reserved	-
12:11	DISVALA	CHxA 输出状态控制 00: 强制输出无效条件0~3中被选择的条件成立时, CHxA 端口正常输出 01: 强制输出无效条件0~3中被选择的条件成立时, CHxA 端口输出高阻态 10: 强制输出无效条件0~3中被选择的条件成立时, CHxA 端口输出低电平 11: 强制输出无效条件0~3中被选择的条件成立时, CHxA 端口输出高电平
10:9	DISSELA	强制输出无效条件选择A 00: 选择强制输出无效条件0; 01: 选择强制输出无效条件1 10: 选择强制输出无效条件2; 11: 选择强制输出无效条件3
8	OUTENA	输出使能A 0: Advanced Timer 功能时的 CHxA 端口输出无效 1: Advanced Timer 功能时的 CHxA 端口输出有效
7:6	PERCA	周期值匹配时端口状态设定A 00: 计数器计数值与周期值相等时, CHxA 端口输出保持为低电平 01: 计数器计数值与周期值相等时, CHxA 端口输出设定为高电平 10: 计数器计数值与周期值相等时, CHxA 端口输出设定为先前状态 11: 计数器计数值与周期值相等时, CHxA 端口输出设定为反转电平
5:4	CMPCA	比较值匹配时端口状态设定A 00: 计数器计数值与 GCMAR 相等时, CHxA 端口输出保持为低电平 01: 计数器计数值与 GCMAR 相等时, CHxA 端口输出设定为高电平 10: 计数器计数值与 GCMAR 相等时, CHxA 端口输出设定为先前状态 11: 计数器计数值与 GCMAR 相等时, CHxA 端口输出设定为反转电平
3	STASTPSA	计数开始停止端口状态选择A 0: 计数开始或停止时, CHxA 端口输出由 STACA、STPCA 决定 1: 计数开始或停止时, CHxA 端口输出设定为先前状态 <i>注: 此处的计数开始是指初始计数开始或停止再开始; 计数停止是指初始时停止或计数开始后再停止</i>
2	STPCA	计数停止端口状态设定A 0: 计数停止时, CHxA 端口输出设定为低电平; 1: 计数停止时, CHxA 端口输出设定为高电平
1	STACA	计数开始端口状态设定A 0: 计数开始时, CHxA 端口输出设定为低电平 1: 计数开始时, CHxA 端口输出设定为高电平
0	CAPCA	功能模式选择A

		0: 比较输出功能; 1: 捕获输入功能
--	--	----------------------

12.3.9 缓存控制寄存器 (TIMx_BCONR)

地址偏移量: 0x05C

复位值: 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							BENP	Reserved					BENB	Res.	BENA
							R/W						R/W		R/W

位	标记	功能
31:9	Reserved	-
8	BENP	周期值缓存传送 0: 缓存传送无效 1: 缓存传送使能 (PERBR->PERAR)
7:3	Reserved	-
2	BENB	通用比较值缓存传送B 0: 缓存传送无效 1: 缓存传送使能 比较输出功能时: (GCMDB->GCMBR); 捕获输入功能时: (GCMBR->GCMDB)
1	Reserved	-
0	BENA	通用比较值缓存传送A 0: 缓存传送无效 1: 缓存传送使能 比较输出功能时: (GCMCR->GCMAR); 捕获输入功能时: (GCMAR->GCMCR)

12.3.10 死区控制寄存器 (TIMx_DCONR)

地址偏移量: 0x060

复位值: 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						DTCEN	
							R/W							R/W	

位	标记	功能
31:9	Reserved	-
8	SEPA	分离设定 0: DTUAR 和 DTDAR 分别设定 1: DTDAR 的值和 DTUAR 的值自动相等
7:1	Reserved	-
0	DTCEN	死区功能 0: 死区功能无效 1: 死区功能有效

12.3.11 滤波控制寄存器 (TIMx_FCONR)

地址偏移量: 0x068

复位值: 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		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

位	标记	功能
31	Reserved	-
30:29	NOFICKTD[1:0]	TRID 端口滤波采样基准时钟选择 00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64
28	NOFIENDT	TRID 端口捕获输入滤波使能, 0无效; 1使能
27	Reserved	-
26:25	NOFICKTC[1:0]	TRIC 端口滤波采样基准时钟选择 00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64
24	NOFIENTC	TRIC 端口捕获输入滤波使能, 0无效; 1使能
23	Reserved	-
22:21	NOFICKTB[1:0]	TRIB 端口滤波采样基准时钟选择 00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64
20	NOFIENBTB	TRIB 端口捕获输入滤波使能, 0无效; 1使能
19	Reserved	-
18:17	NOFICKTA[1:0]	TRIA 端口滤波采样基准时钟选择 00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64
16	NOFIENTA	TRIA 端口捕获输入滤波使能, 0无效; 1使能
15:7	Reserved	-
6:5	NOFICKGB[1:0]	CHxIB 端口滤波采样基准时钟选择 00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64
4	NOFIENGB	CHxIB 端口捕获输入滤波使能, 0无效; 1使能

3	Reserved	-
2:1	NOFICKGA[1:0]	CHxIA 端口滤波采样基准时钟选择 00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64
0	NOFIENGA	CHxIA 端口捕获输入滤波使能, 0无效; 1使能

注意:

- TRIGA-D 滤波设置只有在 TIM4 中设置有效, 在 Timer5/6 设置无效。

12.3.12 有效周期寄存器 (TIMx_VPERR)

地址偏移量: 0x06C

复位值: 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	GE	GE	GE	
											PERI	PERIC	PERI	PERI	
											R/W	R/W	R/W	R/W	

位	标记	功能
31:21	Reserved	-
20:18	PCNTS	有效周期选择 000: 有效周期选择功能无效 001: 每隔1个周期有效一次 010: 每隔2个周期有效一次 011: 每隔3个周期有效一次 100: 每隔4个周期有效一次 101: 每隔5个周期有效一次 110: 每隔6个周期有效一次 111: 每隔7个周期有效一次
17:16	PCNTE	有效周期计数条件选择 00: 有效周期选择功能无效 01: 锯齿波计数上、下溢点或三角波波谷做为计数条件 10: 锯齿波计数上、下溢点或三角波波峰做为计数条件 11: 锯齿波计数上、下溢点或三角波波谷、波峰做为计数条件
15:4	Reserved	-
3	GEPERID	通用信号有效周期选择D 0: 有效周期选择功能无效; 1: 有效周期选择功能使能
2	GEPERIC	通用信号有效周期选择C 0: 有效周期选择功能无效; 1: 有效周期选择功能使能
1	GEPERIB	通用信号有效周期选择B 0: 有效周期选择功能无效; 1: 有效周期选择功能使能
0	GEPERIA	通用信号有效周期选择A 0: 有效周期选择功能无效; 1: 有效周期选择功能使能

12.3.13 状态标志寄存器 (TIMx_STFLR)

地址偏移量: 0x070

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	
DIR	Reserved						VPERNUM			Reserved					
R							R								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DTEF	UDFF	OVFF	Reserved			CMD	CMC	CMB	CMA
						R/W	R/W	R/W				R/W	R/W	R/W	R/W

位	标记	功能
31	DIRF	计数方向 0: 递减计数 1: 递加计数
30:24	Reserved	-
23:21	VPERNUM	周期次数 有效周期选择功能使能时, 计数后的周期次数
20:9	Reserved	-
8	DTEF	死区时间错误 0: 未发生死区时间错误; 1: 发生死区时间错误
7	UDFF	下溢匹配 0: 未发生锯齿波下溢或三角波计数到谷点 1: 发生锯齿波下溢或三角波计数到谷点
6	OVFF	上溢匹配 0: 未发生锯齿波上溢或三角波计数到峰点 1: 发生锯齿波上溢或三角波计数到峰点
5:4	Reserved	-
3	CMDF	计数匹配D 0: GCMDF 寄存器的值与计数值不相等; 1: GCMDF 寄存器的值与计数值相等
2	CMCF	计数匹配C 0: GCMCF寄存器的值与计数值不相等; 1: GCMCF寄存器的值与计数值相等
1	CMBF	计数匹配B 0: GCMBR寄存器的值与计数值不相等, 且未发生 CHxB 捕获完成动作 1: GCMBR寄存器的值与计数值相等, 或发生 CHxB 捕获完成动作
0	CMAF	计数匹配A 0: GCMAF寄存器的值与计数值不相等, 且未发生 CHxA 捕获完成动作

		1: GCMAR 寄存器的值与计数值相等, 或发生 CHxA 捕获完成动作
--	--	---------------------------------------

12.3.14 硬件启动事件选择寄存器 (TIMx_HSTAR)

地址偏移量: 0x074

复位值: 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	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W															

位	标记	功能
31	STARTS	硬件启动使能 0: 硬件启动无效 1: 硬件启动有效 <i>注: 硬件启动有效时, SSTAR 的设定无效</i>
30:16	Reserved	-
15	HSTA15	硬件启动条件15: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
14	HSTA14	硬件启动条件14: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
13	HSTA13	硬件启动条件13: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
12	HSTA12	硬件启动条件12: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
11	HSTA11	硬件启动条件11: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
10	HSTA10	硬件启动条件10: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
9	HSTA9	硬件启动条件9: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效

8	HSTA8	硬件启动条件8: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
7	HSTA7	硬件启动条件7: CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
6	HSTA6	硬件启动条件6: CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
5	HSTA5	硬件启动条件5: CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
4	HSTA4	硬件启动条件4: CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
3	HSTA3	硬件启动条件3: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
2	HSTA2	硬件启动条件2: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
1	HSTA1	硬件启动条件1: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效
0	HSTA0	硬件启动条件0: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件启动无效 1: 条件匹配时, 硬件启动有效

12.3.15 硬件停止事件选择寄存器 (TIMx_HSTPR)

地址偏移量: 0x078

复位值: 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
HSTP15	HSTA14	HSTP13	HSTP12	HSTP11	HSTP10	HSTP9	HSTP8	HSTP7	HSTP6	HSTP5	HSTP4	HSTP3	HSTP2	HSTP1	HSTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能
31	STOPS	硬件停止使能 0: 硬件停止无效 1: 硬件停止有效 <i>注: 硬件停止有效时, 软件停止的设定无效</i>
30:16	Reserved	-
15	HSTP15	硬件停止条件15: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
14	HSTP14	硬件停止条件14: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
13	HSTP13	硬件停止条件13: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
12	HSTP12	硬件停止条件12: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
11	HSTP11	硬件停止条件11: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
10	HSTP10	硬件停止条件10: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
9	HSTP9	硬件停止条件9: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效

8	HSTP8	硬件停止条件8: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
7	HSTP7	硬件停止条件7: CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
6	HSTP6	硬件停止条件6: CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
5	HSTP5	硬件停止条件5: CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
4	HSTP4	硬件停止条件4: CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
3	HSTP3	硬件停止条件3: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
2	HSTP2	硬件停止条件2: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
1	HSTP1	硬件停止条件1: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效
0	HSTP0	硬件停止条件0: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件停止无效 1: 条件匹配时, 硬件停止有效

12.3.16 硬件清零事件选择寄存器 (TIMx_HCELR)

地址偏移量: 0x07C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLEARS		Reserved													
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCEL															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W															

位	标记	功能
31	STARTS	硬件清零使能 0: 硬件清零无效 1: 硬件清零有效 <i>注: 硬件清零有效时, 软件清零的设定无效</i>
30:16	Reserved	-
15	HCEL15	硬件清零条件15: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
14	HCEL14	硬件清零条件14: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
13	HCEL13	硬件清零条件13: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
12	HCEL12	硬件清零条件12: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
11	HCEL11	硬件清零条件11: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
10	HCEL10	硬件清零条件10: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
9	HCEL9	硬件清零条件9: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效

8	HCEL8	硬件清零条件8: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
7	HCEL7	硬件清零条件7: CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
6	HCEL6	硬件清零条件6: CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
5	HCEL5	硬件清零条件5: CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
4	HCEL4	硬件清零条件4: CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
3	HCEL3	硬件清零条件3: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
2	HCEL2	硬件清零条件2: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
1	HCEL1	硬件清零条件1: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效
0	HCEL0	硬件清零条件0: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件清零无效 1: 条件匹配时, 硬件清零有效

12.3.17 硬件捕获 A 事件选择寄存器 (TIMx_HCPAR)

地址偏移量: 0x080

复位值: 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
HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA	HCPA
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能
31:16	Reserved	-
15	HCPA15	硬件捕获A条件15: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
14	HCPA14	硬件捕获A条件14: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
13	HCPA13	硬件捕获A条件13: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
12	HCPA12	硬件捕获A条件12: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
11	HCPA11	硬件捕获A条件11: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
10	HCPA10	硬件捕获A条件10: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
9	HCPA9	硬件捕获A条件9: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
8	HCPA8	硬件捕获A条件8: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
7	HCPA7	硬件捕获A条件7: CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效

		1: 条件匹配时, 硬件捕获A有效
6	HCPA6	硬件捕获A条件6: CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
5	HCPA5	硬件捕获A条件5: CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
4	HCPA4	硬件捕获A条件4: CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
3	HCPA3	硬件捕获A条件3: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
2	HCPA2	硬件捕获A条件2: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
1	HCPA1	硬件捕获A条件1: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效
0	HCPA0	硬件捕获A条件0: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件捕获A无效 1: 条件匹配时, 硬件捕获A有效

12.3.18 硬件捕获 B 事件选择寄存器 (TIMx_HCPBR)

地址偏移量: 0x084

复位值: 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
HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB	HCPB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能
31:16	Reserved	-
15	HCPB15	硬件捕获B条件15: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
14	HCPB14	硬件捕获B条件14: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
13	HCPB13	硬件捕获B条件13: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
12	HCPB12	硬件捕获B条件12: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
11	HCPB11	硬件捕获B条件11: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
10	HCPB10	硬件捕获B条件10: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
9	HCPB9	硬件捕获B条件9: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
8	HCPB8	硬件捕获B条件8: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
7	HCPB7	硬件捕获B条件7: CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效

		1: 条件匹配时, 硬件捕获B有效
6	HCPB6	硬件捕获B条件6: CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
5	HCPB5	硬件捕获B条件5: CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
4	HCPB4	硬件捕获B条件4: CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
3	HCPB3	硬件捕获B条件3: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
2	HCPB2	硬件捕获B条件2: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
1	HCPB1	硬件捕获B条件1: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效
0	HCPB0	硬件捕获B条件0: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件捕获B无效 1: 条件匹配时, 硬件捕获B有效

12.3.19 硬件递加事件选择寄存器 (TIMx_HCUPR)

地址偏移量: 0x088

复位值: 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

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

位	标记	功能
31:20	Reserved	-
19	HCUP19	硬件递加条件: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
18	HCUP18	硬件递加条件: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
17	HCUP17	硬件递加条件: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
16	HCUP16	硬件递加条件: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
15	HCUP15	硬件递加条件: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
14	HCUP14	硬件递加条件: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
13	HCUP13	硬件递加条件: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
12	HCUP12	硬件递加条件: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效

11	HCUP11	硬件递加条件: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
10	HCUP10	硬件递加条件: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
9	HCUP9	硬件递加条件: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
8	HCUP8	硬件递加条件: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
7	HCUP7	硬件递加条件: CHxB 端口为高电平时, CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
6	HCUP6	硬件递加条件: CHxB 端口为高电平时, CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
5	HCUP5	硬件递加条件: CHxB 端口为低电平时, CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
4	HCUP4	硬件递加条件: CHxB 端口为低电平时, CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
3	HCUP3	硬件递加条件: CHxA 端口为高电平时, CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
2	HCUP2	硬件递加条件: CHxA 端口为高电平时, CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
1	HCUP1	硬件递加条件: CHxA 端口为低电平时, CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效
0	HCUP0	硬件递加条件: CHxA 端口为低电平时, CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件递加无效 1: 条件匹配时, 硬件递加有效

12.3.20 硬件递减事件选择寄存器 (TIMx_HCDOR)

地址偏移量: 0x08C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												HCD	HCD	HCD	HCD
												0	0	0	0
												R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCD															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W															

位	标记	功能
31:20	Reserved	-
19	HCDO19	硬件递减条件: 从 AOS 来的事件触发3有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
18	HCDO18	硬件递减条件: 从 AOS 来的事件触发2有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
17	HCDO17	硬件递减条件: 从 AOS 来的事件触发1有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
16	HCDO16	硬件递减条件: 从 AOS 来的事件触发0有效 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
15	HCDO15	硬件递减条件: TIMTRID 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
14	HCDO14	硬件递减条件: TIMTRID 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
13	HCDO13	硬件递减条件: TIMTRIC 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
12	HCDO12	硬件递减条件: TIMTRIC 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效

11	HCDO11	硬件递减条件: TIMTRIB 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
10	HCDO10	硬件递减条件: TIMTRIB 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
9	HCDO9	硬件递减条件: TIMTRIA 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
8	HCDO8	硬件递减条件: TIMTRIA 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
7	HCDO7	硬件递减条件: CHxB 端口为高电平时, CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
6	HCDO6	硬件递减条件: CHxB 端口为高电平时, CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
5	HCDO5	硬件递减条件: CHxB 端口为低电平时, CHxA 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
4	HCDO4	硬件递减条件: CHxB 端口为低电平时, CHxA 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
3	HCDO3	硬件递减条件: CHxA 端口为高电平时, CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
2	HCDO2	硬件递减条件: CHxA 端口为高电平时, CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
1	HCDO1	硬件递减条件: CHxA 端口为低电平时, CHxB 端口上采样到下降沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效
0	HCDO0	硬件递减条件: CHxA 端口为低电平时, CHxB 端口上采样到上升沿 0: 条件匹配时, 硬件递减无效 1: 条件匹配时, 硬件递减有效

12.3.21 软件同步启动寄存器 (TIMx_SSTAR)

地址偏移量: 0x3F4

复位值: 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

位	标记	功能
31:3	Reserved	
2	SSTA2	Timer6 软件启动 0: 软件启动无效 1: 软件启动使能
1	SSTA1	Timer5 软件启动 0: 软件启动无效 1: 软件启动使能
0	SSTA0	Timer4 软件启动 0: 软件启动无效 1: 软件启动使能

12.3.22 软件同步停止寄存器 (TIMx_SSTPR)

地址偏移量: 0x3F8

复位值: 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

位	标记	功能
31:3	Reserved	
2	SSTP2	Timer6 软件停止 0: 软件停止无效 1: 软件停止使能
1	SSTP1	Timer5 软件停止 0: 软件停止无效 1: 软件停止使能
0	SSTP0	Timer4 软件停止 0: 软件停止无效 1: 软件停止使能

12.3.23 软件同步清零寄存器 (TIMx_SCLRR)

地址偏移量: 0x3FC

复位值: 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													SCLR2	SCLR1	SCLR0
													R/W	R/W	R/W

位	标记	功能
31:3	Reserved	
2	SCLR2	Timer6 软件清零 0: 软件清零无效 1: 软件清零使能
1	SCLR1	Timer5 软件清零 0: 软件清零无效 1: 软件清零使能
0	SCLR0	Timer4 软件清零 0: 软件清零无效 1: 软件清零使能

12.3.24 中断标志寄存器 (TIMx_IFR)

地址偏移量: 0x100

复位值: 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	
SAMHF	SAMLF	Reserved					DTEF	UDFF	OVFF	Reserved			CMDF	CMCF	CMBF	CMAF
RO	RO						RO	RO	RO				RO	RO	RO	RO

位	标记	位名	功能
31:16	Reserved	-	
15	SAMHF	CHxA/B 端口高状态中断标志	0: CHxA 和 CHxB 端口上没有同时出现高电平 1: CHxA 和 CHxB 端口上同时出现高电平
14	SAMLF	CHxA/B 端口低状态中断标志	0: CHxA 和 CHxB 端口上没有同时出现低电平 1: CHxA 和 CHxB 端口上同时出现低电平
13:9	Reserved	-	
8	DTEF	死区时间错误中断标志	0: 未发生死区时间错误; 1: 发生死区时间错误
7	UDFF	下溢匹配中断标志	0: 未发生锯齿波下溢或三角波计数到谷点 1: 发生锯齿波下溢或三角波计数到谷点
6	OVFF	上溢匹配中断标志	0: 未发生锯齿波上溢或三角波计数到峰点 1: 发生锯齿波上溢或三角波计数到峰点
5:4	Reserved	-	
3	CMDF	计数匹配D中断标志	0: GCMDR 寄存器的值与计数值不相等; 1: GCMDR 寄存器的值与计数值相等
2	CMCF	计数匹配C中断标志	0: GCMCR 寄存器的值与计数值不相等; 1: GCMCR 寄存器的值与计数值相等
1	CMBF	计数匹配B中断标志	0: GCMBR 寄存器的值与计数值不相等, 且未发生 CHxB 捕获完成动作 1: GCMBR 寄存器的值与计数值相等, 或发生 CHxB 捕获完成动作
0	CMAF	计数匹配A中断标志	0: GCMAA 寄存器的值与计数值不相等, 且未发生 CHxA 捕获完成动作

		1: GCMAR 寄存器的值与计数值相等, 或发生 CHxA 捕获完成动作
--	--	---------------------------------------

12.3.25 中断标志清除寄存器 (TIMx_ICLR)

地址偏移量: 0x104

复位值: 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	
SAMHC	SAMLC	Reserved -					DTEC	UDFC	OVFC	Reserved			CMDC	CMCC	CMBC	CMAC
W0	W0						W0	W0	W0				W0	W0	W0	W0

位	标记	功能
31:16	Reserved	-
15	SAMHC	CHxA/B 端口高状态中断标志清除, 写1无效, 写0清除对应中断
14	SAMLC	CHxA/B 端口低状态中断标志清除, 写1无效, 写0清除对应中断
13:9	Reserved	-
8	DTEC	死区时间错误中断标志清除, 写1无效, 写0清除对应中断
7	UDFC	下溢匹配中断标志清除, 写1无效, 写0清除对应中断
6	OVFC	上溢匹配中断标志清除, 写1无效, 写0清除对应中断
5:4	Reserved	-
3	CMDC	计数匹配D中断标志清除, 写1无效, 写0清除对应中断
2	CMCC	计数匹配C中断标志清除, 写1无效, 写0清除对应中断
1	CMBC	计数匹配B中断标志清除, 写1无效, 写0清除对应中断
0	CMAC	计数匹配A中断标志清除, 写1无效, 写0清除对应中断

12.3.26 展频及中断触发选择 (TIMx_CR)

地址偏移量: 0x108

复位值: 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

位	标记	功能
31:11	Reserved	-
10	DITENS	PWM 展频计数选择 0: 选择下溢出, 1: 选择上溢出
9	DITENB	PWM 通道B展频使能 0: 使能无效, 1: 使能有效, 每个周期改变 PWM 的输出延时
9	DITENA	PWM 通道A展频使能 0: 使能无效, 1: 使能有效, 每个周期改变 PWM 的输出延时
7	UDFE	下溢匹配使能触发 ADC 0: 使能无效, 1: 使能有效, 这个中断可以控制 ADC/AOS_i_tirg
6	OVFE	上溢匹配使能触发 ADC 0: 使能无效, 1: 使能有效, 这个中断可以控制 ADC/AOS_i_tirg
5:4	Reserved	-
3	CMDE	计数匹配D使能触发 ADC 0: 使能无效, 1: 使能有效, 这个中断可以控制 ADC/AOS_i_tirg
2	CMCE	计数匹配C使能触发 ADC 0: 使能无效, 1: 使能有效, 这个中断可以控制 ADC/AOS_i_tirg
1	CMBE	计数匹配B使能触发 ADC 0: 使能无效, 1: 使能有效, 这个中断可以控制 ADC/AOS_i_tirg
0	CMAE	计数匹配A使能触发 ADC 0: 使能无效, 1: 使能有效, 这个中断可以控制 ADC/AOS_i_tirg

12.3.27 AOS 选择控制寄存器 (TIMx_AOSSR)

地址偏移量: 0x110

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器, 任意一个定时器其更改后, 在另外两个定时器的值会同时更改。

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	FBRAKE
		R/W	R/W	R/W	R/W	R/W	R	R							

位	标记	功能
31:14	Reserved	-
13	SMH2	通道2同高选择 0: 选择无效, 1: 选择有效, 出现同高时 AOS_i_odis[1]
12	SMH1	通道1同高选择 0: 选择无效, 1: 选择有效, 出现同高时 AOS_i_odis[1]
11	SMH0	通道0同高选择 0: 选择无效, 1: 选择有效, 出现同高时 AOS_i_odis[1]
10	SML2	通道2同低选择 0: 选择无效, 1: 选择有效, 出现同低时 AOS_i_odis[1]
9	SML1	通道1同低选择 0: 选择无效, 1: 选择有效, 出现同低时 AOS_i_odis[1]
8	SML0	通道0同低选择 0: 选择无效, 1: 选择有效, 出现同低时 AOS_i_odis[1]
7	SOFTBK	软件刹车: 写1实现软件刹车
13	Reserved	-
4	BFILTEN	端口刹车滤波使能
3:2	BFILTS	端口刹车滤波时钟选择
1	FSAME	同高同低刹车标志, 只读
0	FBRAKE	端口刹车标志, 只读

12.3.28 AOS 选择控制寄存器标志清除 (TIMx_AOSCL)

地址偏移量: 0x114

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器, 任意一个定时器其更改后, 在另外两个定时器的值会同时更改。

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

位	标记	功能
31:2	Reserved	-
1	FSAME	同高同低刹车标志清除, 写0清除, 写1无效, 读恒为1
0	FBRAKE	端口刹车标志清除, 写0清除, 写1无效, 读恒为1

12.3.29 端口刹车控制寄存器 (TIMx_PTBKS)

地址偏移量: 0x118

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器, 任意一个定时器其更改后, 在另外两个定时器的值会同时更改。

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

位	标记	功能
31:16	Reserved	-
15	EN15	P36 刹车端口使能: 1 选择, 0无效
14	EN14	P35 刹车端口使能: 1 选择, 0无效
13	EN13	P34 刹车端口使能: 1 选择, 0无效
12	EN12	P33 刹车端口使能: 1 选择, 0无效
11	EN11	P32 刹车端口使能: 1 选择, 0无效
10	EN10	P31 刹车端口使能: 1 选择, 0无效
9	EN9	P27 刹车端口使能: 1 选择, 0无效
8	EN8	P26 刹车端口使能: 1 选择, 0无效
7	EN7	P26 刹车端口使能: 1 选择, 0无效
6	EN6	P24 刹车端口使能: 1 选择, 0无效
5	EN5	P23 刹车端口使能: 1 选择, 0无效
4	EN4	P15 刹车端口使能: 1 选择, 0无效
3	EN3	P14 刹车端口使能: 1 选择, 0无效
2	EN2	P03 刹车端口使能: 1 选择, 0无效
1	EN1	P02 刹车端口使能: 1 选择, 0无效
0	EN0	P01 刹车端口使能: 1 选择, 0无效

12.3.30 端口触发控制寄存器 (TIMx_TTRIG)

地址偏移量: 0x11C

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器, 任意一个定时器其更改后, 在另外两个定时器的值会同时更改。

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			

位	标记	功能
31:16	Reserved	-
15:12	TRIGDS	TIMx 触发D端口选择
11:8	TRIGCS	TIMx 触发C端口选择
7:4	TRIGBS	TIMx 触发B端口选择
3:0	TRIGAS	TIMx 触发A端口选择

控制信号与端口选择如下

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 触发控制寄存器 (TIMx_ITRIG)

地址偏移量: 0x120

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器, 任意一个定时器其更改后, 在另外两个定时器的值会同时更改。

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			

位	标记	功能
31:16	Reserved	-
15:12	IAOS3S	TIMx AOS3 触发源选择
11:8	IAOS2S	TIMx AOS2 触发源选择
7:4	IAOS1S	TIMx AOS1 触发源选择
3:0	IAOS0S	TIMx AOS0 触发源选择

控制信号 (IAOSxS) 与中断源选择如下 (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 端口刹车极性控制寄存器 (TIMx_PTBKP)

地址偏移量: 0x124

复位值: 0x0000 0000

Timer4/5/6 使用同一个实体寄存器, 任意一个定时器其更改后, 在另外两个定时器的值会同时更改。

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

位	标记	功能
31:16	Reserved	-
15	POL15	P36 刹车端口极性选择: 1 低电平有效, 0 高电平有效
14	POL14	P35 刹车端口极性选择: 1 低电平有效, 0 高电平有效
13	POL13	P34 刹车端口极性选择: 1 低电平有效, 0 高电平有效
12	POL12	P33 刹车端口极性选择: 1 低电平有效, 0 高电平有效
11	POL11	P32 刹车端口极性选择: 1 低电平有效, 0 高电平有效
10	POL10	P31 刹车端口极性选择: 1 低电平有效, 0 高电平有效
9	POL9	P27 刹车端口极性选择: 1 低电平有效, 0 高电平有效
8	POL8	P26 刹车端口极性选择: 1 低电平有效, 0 高电平有效
7	POL7	P26 刹车端口极性选择: 1 低电平有效, 0 高电平有效
6	POL6	P24 刹车端口极性选择: 1 低电平有效, 0 高电平有效
5	POL5	P23 刹车端口极性选择: 1 低电平有效, 0 高电平有效
4	POL4	P15 刹车端口极性选择: 1 低电平有效, 0 高电平有效
3	POL3	P14 刹车端口极性选择: 1 低电平有效, 0 高电平有效
2	POL2	P03 刹车端口极性选择: 1 低电平有效, 0 高电平有效
1	POL1	P02 刹车端口极性选择: 1 低电平有效, 0 高电平有效
0	POL0	P01 刹车端口极性选择: 1 低电平有效, 0 高电平有效

13 实时时钟 (RTC)

13.1 实时时钟简介

实时时钟/日历提供秒、分、时、日、周、月、年的信息，每月的天数和闰年的天数可自动调整。时钟操作可通过 AM/PM 寄存器位，决定采用 24 或 12 小时格式。表 14-1 所示是其基本特性。

时钟源	片外低速晶振 XTL (32.768kHz) 片内低速振荡器 RCL(32kHz, 1%精度) 片外高速晶振 XTH
基本功能	可计算 00~99 年之间的秒、分、时、日、周、月、年
	可自动进行闰年调整
	可配置为24或12小时格式
	可程序控制启动或停止
	具有闹钟功能
	具有高精度 1Hz 方波信号输出
中断	具有周期中断
	具有闹钟中断

表 13-1 RTC 的基本特性

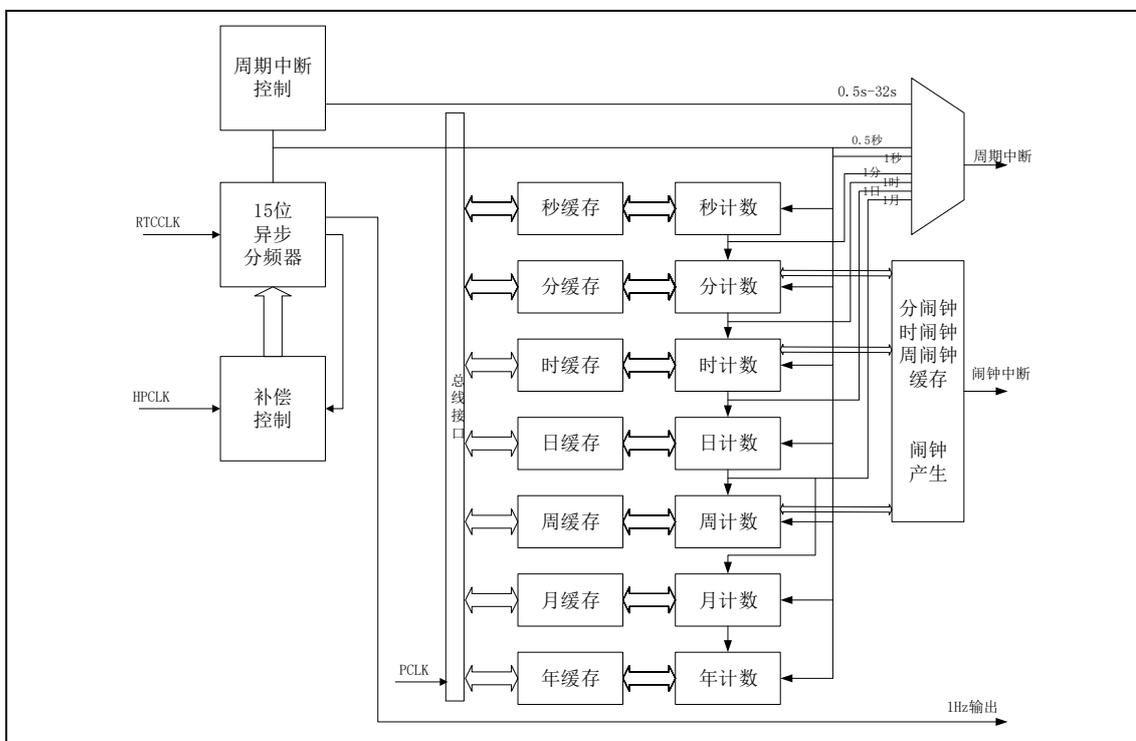


图 13-1 RTC 框图

13.2 实时时钟功能描述

实时时钟的时钟源可配置为外部低速晶振，外部高速晶振，内部低速 RC；默认使用外部低速晶振。控制寄存器 CR0、CR1 与 COMPEN 只受上电复位控制，其他复位源不能复位这三个控制寄存器。其他数据寄存器上电状态不定，上电后需要初始化，不受任何复位影响。

所有软件写入和读取的日期时间值都为 BCD 码，无须十六进制转换为十进制。

任何无效的日期时间将无法写入，比如 32 日，25 时，70 秒，B 月等。

13.2.1 上电设定

RTC 在上电之后复位一次，在系统不掉电的情况下，外部各种复位请求都不能复位 RTC，RTC 会一直处于计数状态。在上电之后，设定日历初始值、闹钟设置、误差补偿、中断等之后，启动 RTC。

13.2.2 RTC 计数开始设定

1. 设定 CR0.START=0，计数停止；
2. 设定 CR0.AMPM 和 CR0.PRDS，CR0.PRDX 设定时制和中断周期；
3. 设定 CR1.CKSEL 选择 RTC 的计时时钟；
4. 设定秒，分，时，周，日，月，年的日历计数寄存器；
5. 需要进行时钟误差补偿时，设定计数时钟误差补偿寄存器 COMPEN；
6. 清除中断标志位 CR1.ALMF，CR1.PRDF，并使能中断；
7. 设定 CR0.START=1，计数开始。

13.2.3 系统低功耗模式切换

在 RTC 计数开始后，系统如果立即切换为低功耗模式时，请执行下列任意一种确认后再进行模式切换。

控制寄存器在系统控制寄存器 SYSCTRL1.RTC_LPW

1. 在 CR0.START=1 设定后，经过 2 个以上的 RTC 计数时钟后再进行模式切换。
2. 在 CR0.START=1 设定后，设定 CR1.WAIT=1，查询 CR1.WAITF=1。再设定

CR1.WAIT=0，查询 CR1.WAITF=0 后再进行模式切换。

在 RTC 低功耗模式下，RTC 寄存器不能读写。在低功耗模式下，RTC 消耗更少的电流。

RTC 运行时切换低功耗模式不需要等待。

13.2.4 读出计数寄存器

有三种读取计数寄存器的方式：

- 方式 1：任意时刻读取方式 1

1. 设定 CR1.WAIT=1，停止日历寄存器计数，进入读写模式；
2. 查询直到 CR1.WAITF=1；
3. 读出秒，分，时，周，日，月，年计数寄存器值；
4. 设定 CR1.WAIT=0，计数器计数；
5. 查询直到 CR1.WAITF=0。

- 方式 2：任意时刻读取方式 2

1. 读出分，时，周，日，月，年计数寄存器值；
2. 读出秒计数寄存器值；
3. 再次读出秒计数寄存器值；
4. 判断两次秒的读出值是否相同，不同重新从第一步开始，相同读取结束。

- 方式 3：中断读取方式

在 RTC 周期中断服务中读取秒，分，时，周，日，月，年计数寄存器值。因为中断发生后到下次数据改变至少 0.5s 的时间。

13.2.5 写入计数寄存器

1. 设定 CR1.WAIT=1，停止日历寄存器计数，进入读写模式；
2. 查询直到 CR1.WAITF=1；
3. 写入秒，分，时，周，日，月，年计数寄存器值；
4. 设定 CR1.WAIT=0，计数器重新开始计数。注意，须在 1 秒内完成所有写操作；
5. 查询直到 CR1.WAITF=0。

在 RTC 未启动模式下写秒，分，时，周，日，月，年计数寄存器不需要等 WAIT。

注：

- 在计数模式下更改秒寄存器会复位秒计数，写分，时，周，日，月，年计数寄存器值不会影响 RTC 计数。

13.2.6 闹钟设定

1. 设定 CR1.ALMEN=0，闹钟禁止；
2. 设定 CR1.ALMIE=1，闹钟中断许可；
3. 分闹钟 ALMMIN，时闹钟 ALMHOUR，周闹钟 ALMWEEL 设定；
4. 设定 CR1.ALMEN=1，闹钟许可；
5. 等待发生中断；
6. 由于闹钟中断和定周期中断共用中断请求信号，则当 CR1.ALMF=1 时，进入闹钟中断处理；否则进入定周期中断处理。

13.2.7 1Hz 输出

RTC 可选择输出一般精度，较高精度和高精度 3 种 1Hz 时钟。当时钟误差补偿功能有效时输出较高精度的 1Hz 时钟；当使用不同频率的 PCLK 时输出高精度的 1Hz 时钟。需要根据 PCLK 频率配置系统控制寄存器，其中，

- 一般精度的 1Hz 输出设定如下：（无时钟补偿）
 1. 设定 CR0.START=0，计数停止；
 2. RTC 输出引脚设定；
 3. CR0.1HZOE=1，时钟输出许可；
 4. 设定 CR0.START=1，计数开始；
 5. 等待 2 个计数周期以上；
 6. 1Hz 输出开始。
- 较高精度的 1Hz 输出设定如下：（低速补偿）
 1. 设定 CR0.START=0，计数停止；
 2. RTC 输出引脚设定；
 3. CR0.1HZOE=1，时钟输出许可；
 4. 时钟误差补偿寄存器 COMPEN.CR 补偿数设定；

5. 时钟误差补偿寄存器 COMPEN.EN=1，误差补偿有效；
 6. 设定 CR0.START=1，计数开始；
 7. 等待 2 个计数周期以上；
 8. 1Hz 输出开始。
- 当高精度的 1Hz 输出时，需要在较高精度输出的基础上，为 RTC 提供 4M,6M,8M,12M,16M,20M,24M,32MHz 的高速 PCLK 时钟，输出设定如下：
 1. 设定 CR0.START=0，计数停止；
 2. RTC 输出引脚设定；
 3. CR0.1HZOE=1，时钟输出许可；
 4. CR0.1HZSEL=1，选择输出高精度 1Hz 时钟；
 5. 配置高速时钟补偿时钟 SYSCTRL1.RTC_FREQ_ADJUST
 6. 时钟误差补偿寄存器 COMPEN.CR[8:0] 补偿数设定；
 7. 时钟误差补偿寄存器 COMPEN.EN=1，精度补偿有效；
 8. 设定 CR0.START=1，计数开始；
 9. 等待 2 个计数周期以上；
 10. 1Hz 输出开始。

13.2.8 时钟误差补偿

由于外部晶振存在误差，在需要得到高精度的计数结果时，需要对该误差进行补偿。补偿方法分为两种：第一种，基于自身时钟的误差补偿；第二种，基于高速时钟的误差补偿。

基于自身时钟的误差补偿原理与计算：

由于计数器采用 32.768KHz 的时钟计数，如果需要对每秒精度进行补偿时，只能按照 32.768KHz 的整数周期补偿，则每秒补偿的最小单位为 $(1/32768) * 10^6 = 30.5\text{ppm}$ ，无法满足高精度的要求。

那么要在 32.768KHz 的计数时钟下实现精度较高的时钟补偿时，需要在算法上做调整，将最大补偿周期扩大 32 倍。则在只能补偿的最小单位为 30.5ppm 的情况下，平均到每秒的补偿单位变为 $30.5\text{ppm}/32 = 0.96\text{ppm}$ 。满足了精度较高的时钟补偿要求。而

且补偿发生在每 32 秒内比较均匀的范围。所以，该寄存器中引入了 5 位小数的设定。

例 1:

当默认状态下直接输出 1Hz 时钟，通过测定该时钟的精度，计算补偿目标值。

假设实际测定值为 0.9999888Hz，则：

实际发振频率 = 32768 X 0.9999888 ≈ 32767.63

$$\begin{aligned} \text{补偿目标值} &= (\text{实际发振频率} - \text{目标频率}) / \text{目标频率} \times 10^6 \\ &= (32767.96 - 32768) / 32768 \times 10^6 \\ &= -11.29\text{ppm} \end{aligned}$$

根据

$$\text{CR}[8 : 0] = \left(\frac{\text{补偿目标值}[\text{ppm}] \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.00000\text{B}$$

如果补偿目标值为 -11.29ppm，计算相应的寄存器值如下：

$$\begin{aligned} \text{CR}[8:0] &= (-11.29 \times 2^{15} / 10^6)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= (-0.37)_{\text{取2的补码}} + 0001.00000\text{B} \\ &= 1111.10101\text{B} + 0001.00000\text{B} \\ &= 0000.10101\text{B} \end{aligned}$$

基于高速 24MHz 时钟的误差补偿原理与计算：

该方式的计算方法与基于自身时钟的误差补偿相同。由于引入了 4M-32MHz 高速时钟，本来需要在最多 32 秒内累计的 1/32768 秒误差可分散到每 1 秒，针对每 1 秒进行最小 0.96ppm（23 个 24MHz 时钟周期）的补偿，实现平均的每秒高精度 1Hz 时钟输出。

13.3 RTC 中断

RTC 支持两种中断类型。闹钟中断、定周期中断。闹钟中断与定周期中断共用一个中断信号。

13.3.1 RTC 闹钟中断

当 CR1.ALMIE=1 时，若当前日历时间与分闹钟寄存器（ALMMIN）、时闹钟寄存器（ALMHOUR）、周闹钟寄存器（ALMWEED）相等时，触发闹钟中断。

13.3.2 RTC 周期中断

控制寄存器 1（CR1）的 ALMIE=1 时，选择的周期发生后，触发定周期唤醒中断，由于闹钟和定周期共用中断，通过标志寄存器位来区分。

13.4 RTC 寄存器描述

基地址 0X40001400

寄存器	偏移地址	描述
RTC_CR0	0X000	控制寄存器0
RTC_CR1	0X004	控制寄存器1
RTC_SEC	0X008	秒计数寄存器
RTC_MIN	0X00C	分计数寄存器
RTC_HOUR	0X010	时计数寄存器
RTC_WEEK	0X014	周计数寄存器
RTC_DAY	0X018	日计数寄存器
RTC_MON	0X01C	月计数寄存器
RTC_YEAR	0X020	年计数寄存器
RTC_ALMMIN	0X024	分闹钟寄存器
RTC_ALMHOUR	0X028	时闹钟寄存器
RTC_ALMWEEK	0X02C	周闹钟寄存器
RTC_COMPEN	0X030	时钟误差补偿寄存器

表 13-2 RTC 寄存器列表

13.4.1 控制寄存器 0 (RTC_CR0)

*只有上电对该寄存器复位有效

地址偏移量: 0x000

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13:8	7	6	5	4	3	2:0
Res.	PRDSEL	PRDX	START	HZ1SEL	HZ1OE	Res.	AMPM	PRDS
	R/W	R/W	R/W	R/W	R/W		R/W	R/W

位	符号	功能描述
31:15	Reserved	-
14	PRDSEL	0: 使用 PRDS 所设定的周期中断时间间隔 1: 使用 PRDX 所设定的周期中断时间间隔
13:8	PRDX	设置产生周期中断的时间间隔, 可设定的范围为0.5秒到32秒, 步进为0.5秒。 000000: 0.5秒 000001: 1秒 111110: 31.5秒 111111: 32秒
7	START	0: 停止 RTC 计数器 1: 使能 RTC 计数器
6	1HZSEL	0: 普通精度 1Hz 输出 1: 高精度 1Hz 输出
5	1HZOE	0: 禁止 1Hz 输出 1: 使能 1Hz 输出
4	Reserved	-
3	AMPM	0: 12小时制 1: 24小时制
2:0	PRDS	设置产生中断的时间间隔: 000: 不产生周期中断 001: 0.5秒 010: 1秒 011: 1分钟 100: 1小时 101: 1天 11x: 1月 注意: 如需要在 START=1 时写入更改周期中断的时间间隔操作步骤如下:

		step1, 在 NVIC 中关闭 RTC 中断; step2, 更改周期中断的时间间隔; step3, 清除 RTC 中断标志; step4, 使能 RTC 中断。
--	--	--

13.4.2 控制寄存器 1 (RTC_CR1)

*只有上电对该寄存器复位有效

地址偏移量: 0x004

复位值 0X00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15:11			10:8			7	6	5	4	3	2	1	0		
Reserved			CKSEL	ALMEN	ALMIE	Res.	ALMF	PRDF	Res.	WAITF	WAIT				
			R/W	R/W	R/W		RO	RO		R/W	R/W				

位	符号	功能描述
31:11	Reserved	-
10:8	CKSEL	RTC 时钟选择 00x: XTL 32.768k 01x: RCL 32k 100: XTH/128 (晶振为 4M 时选择此项) 101: XTH/256 (晶振为 8M 时选择此项) 110: XTH/512 (晶振为 16M 时选择此项) 111: XTH/1024 (晶振为 32M 时选择此项)
7	ALMEN	0: 禁止闹钟 1: 使能闹钟 注意: 在 START=1 日历计数过程中并且 ALMIE=1 中断许可的情况下使能 ALMEN 时, 为防止误动作请将系统中断关闭。使能后请将 ALMF 标志位清除。
6	ALMIE	0: 禁止闹钟中断 1: 使能闹钟中断
5	Reserved	-
4	ALMF	0: 未发生闹钟中断 1: 已发生闹钟中断 注意: 该位仅在 ALMEN=1 时有效。闹钟匹配时, 32.768KHz 一个时钟后置1。写0时清除标志, 写1无效。
3	PRDF	0: 未发生周期中断 1: 已发生周期中断 注意: 发生周期中断后, 该位置1。写0时清除该标志, 写1无效。
2	Reserved	-
1	WAITF	0: 非写入/读出状态 1: 写入/读出状态 注意: WAIT 位设定是否有效标志。在写入/读出前请确认该位是否为“1”。计

		数过程中，在 WAIT 位清” 0 “后等待写入完成后该位才清” 0 “。
0	WAIT	<p>0: 正常计数模式</p> <p>1: 写入/读出模式</p> <p>注意：在写入/读出时请将该位置 “1”，由于计数器在连续计数，请在1秒的时间内完成写入/读出操作并将该位清 “0 “。</p>

13.4.3 秒计数寄存器 (RTC_SEC)

地址偏移量: 0x008

复位值: 不定

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										SECH		SECL			
Reserved										R/W		R/W			

位	符号	功能描述
31:7	Reserved	-
6:4	SECH	秒计数十位值
3:0	SECL	秒计数个位值

表示 0-59 秒, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

13.4.4 分计数寄存器 (RTC_MIN)

地址偏移量: 0x00C

复位值: 不定

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										MINH		MINL			
Reserved										R/W		R/W			

位	符号	功能描述
31:7	Reserved	-
6:4	MINH	分计数十位值
3:0	MINL	分计数个位值

表示 0-59 分, 采用十进制计数。请写入十进制 0-59 的 BCD 码, 写入错误值时, 写入值将被忽略。

13.4.5 时计数寄存器 (RTC_HOUR)

地址偏移量: 0x010

复位值: 不定

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										HOURL		HOURL			
										R/W		R/W			

位	符号	功能描述
31:6	Reserved	-
5:4	HOURH	时计数十位值
3:0	HOURL	时计数个位值

24 小时时制时，表示 0-23 小时。12 小时时制时，b5=0 表示 AM，则 01:12 表示上午；b5=1 表示 PM，则 21:32 表示下午。

请根据控制为 AMPM 的值，设定正确十进制的 0:23 或者 01:12,21:32 的 BCD 码。写入超出范围的值将被忽略。

具体时间表示参考下表：

24小时时制	AMPM=1	12小时时制	AMPM=0
时间	寄存器表示	时间	寄存器表示
00时	00H	AM 12时	12H
01时	01H	AM 01时	01H
02时	02H	AM 02时	02H
03时	03H	AM 03时	03H
04时	04H	AM 04时	04H
05时	05H	AM 05时	05H
06时	06H	AM 06时	06H
07时	07H	AM 07时	07H
08时	08H	AM 08时	08H
09时	09H	AM 09时	09H
10时	10H	AM 10时	10H
11时	11H	AM 11时	11H
12时	12H	PM 12时	32H
13时	13H	PM 01时	21H
14时	14H	PM 02时	22H
15时	15H	PM 03时	23H
16时	16H	PM 04时	24H
17时	17H	PM 05时	25H
18时	18H	PM 06时	26H
19时	19H	PM 07时	27H
20时	20H	PM 08时	28H
21时	21H	PM 09时	29H
22时	22H	PM 10时	30H
23时	23H	PM 11时	31H

13.4.6 日计数寄存器 (RTC_DAY)

地址偏移量: 0x018

复位值: 不定

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										DAYH		DAYL			
Reserved										R/W		R/W			

位	符号	功能描述
31:6	Reserved	-
5:4	DAYH	日计数十位值
3:0	DAYL	日计数个位值

十进制表示 1:31 日，自动计算闰年和月份。具体表示如下：

月份	日计数表示
2月（普通年）	01:28
2月（闰年）	01:29
4、6、9、11月	01:30
1、3、5、7、8、10、12月	01:31

13.4.7 周计数寄存器 (RTC_WEEK)

地址偏移量: 0x014

复位值: 不定

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		

位	符号	功能描述
31:3	Reserved	-
2:0	WEEK	周计数值

十进制 0:6 表示周日:周六。请写入正确的十进制 0:6 的 BCD 码，写入其他值，将被忽略。周计数值对应关系如下：

周	周计数表示
周日	00H
周一	01H
周二	02H
周三	03H
周四	04H
周五	05H
周六	06H

13.4.8 月计数寄存器 (RTC_MON)

地址偏移量: 0x01C

复位值: 不定

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												MON			
Reserved												R/W			

位	符号	功能描述
31:5	Reserved	-
4:0	MON	月计数值

十进制 1:12 表示 1:12 月。请写入正确的十进制 1:12 的 BCD 码，写入其他值，将被忽略。

13.4.9 年计数寄存器 (RTC_YEAR)

地址偏移量: 0x020

复位值: 不定

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								YEARH				YEARL			
Reserved								R/W				R/W			

位	符号	功能描述
31:8	Reserved	-
7:4	YEARH	年计数十位值
3:0	YEARL	年个位计数值

十进制 0:99 表示 0:99 年。根据月进位计数。自动计算闰年如: 00、04、08、...、92、96 等。请写入正确的十进制年计数值，写入错误值将被忽略。

13.4.10 分闹钟寄存器 (RTC_ALMMIN)

地址偏移量: 0x024

复位值: 不定

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										ALMMINH		ALMMINL			
										R/W		R/W			

位	符号	功能描述
31:6	Reserved	-
5:4	ALMMINH	分闹钟匹配值十位
3:0	ALMMINL	分闹钟匹配值个位

请设定十进制 0:59 的 BCD 码。写入其他值，不会发生闹钟匹配。

13.4.11 时闹钟寄存器 (RTC_ALM HOUR)

地址偏移量: 0x028

复位值: 不定

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										ALM HOURH		ALM HOURL			
										R/W		R/W			

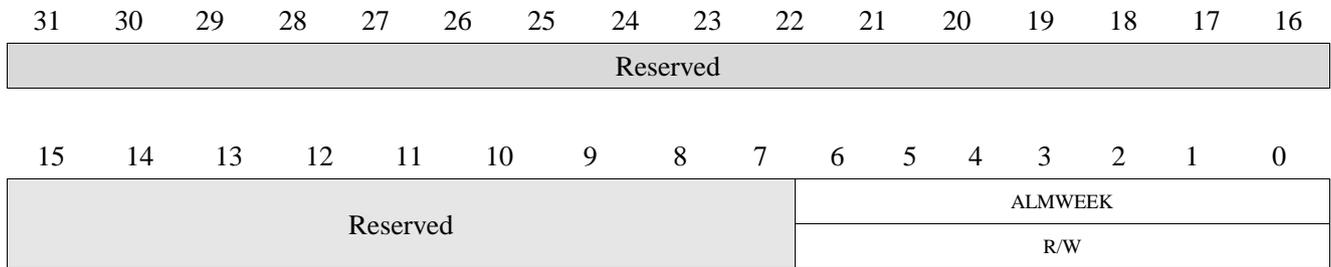
位	符号	功能描述
31:6	Reserved	-
5:4	ALM HOURH	时闹钟十位匹配值
3:0	ALM HOURL	时闹钟个位匹配值

请根据时制设定正确的闹钟匹配值，否则不会发生时闹钟匹配。

13.4.12 周闹钟寄存器 (RTC_ALMWEAK)

地址偏移量: 0x02C

复位值: 不定



位	符号	功能描述
31:7	Reserved	-
6:0	ALMWEAK	周闹钟匹配值。 b0:b6 分别对应周日:周六, 对应为置“1”时, 代表每周该日闹钟有效。 如, b0=1, b5=1 代表周日和周五闹钟设定有效。

请根据时制设定正确的闹钟匹配值, 否则不会发生时闹钟匹配。

13.4.13 时钟误差补偿寄存器 (RTC_COMPEN)

地址偏移量: 0x030

*只有上电对该寄存器复位有效, 复位值: 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	Reserved						R/W								

位	符号	功能描述																																																																																																																				
31:16	Reserved	-																																																																																																																				
15	EN	补偿使能 0: 禁止时钟误差补偿 1: 使能时钟误差补偿																																																																																																																				
14:b	Reserved	-																																																																																																																				
8:0	CR	补偿值 通过补偿值设定, 可针对每秒进行 +/-0.96ppm 的精度补偿。补偿值为9位带小数点的2的补码, 后5位为小数部分。可补偿范围 274.6ppm:212.6ppm。最小微分误差 +/-0.48ppm。最小分辨率 0.96ppm。具体补偿精度请参考下表: <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th colspan="10">补偿值设定</th><th>补偿数</th></tr> <tr> <th>EN</th><th colspan="9">CR[8:0]</th><th></th></tr> </thead> <tbody> <tr> <td rowspan="8" style="text-align: center; vertical-align: middle;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">-274.6ppm</td></tr> <tr> <td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">-273.7ppm</td></tr> <tr> <td style="text-align: center;">:</td><td style="text-align: center;">:</td></tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">-0.95ppm</td></tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0ppm</td></tr> <tr> <td style="text-align: center;">:</td><td style="text-align: center;">:</td></tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">+211.7ppm</td></tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">+212.6ppm</td></tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">X</td><td style="text-align: center;">无补偿</td></tr> </tbody> </table>	补偿值设定										补偿数	EN	CR[8:0]										1	1	0	0	0	0	0	0	0	0	-274.6ppm	1	0	0	0	0	0	0	0	1	-273.7ppm	:	:	:	:	:	:	:	:	:	:	0	0	0	0	1	1	1	1	1	-0.95ppm	0	0	0	1	0	0	0	0	0	0ppm	:	:	:	:	:	:	:	:	:	:	0	1	1	1	1	1	1	1	1	0	+211.7ppm	0	1	1	1	1	1	1	1	1	1	+212.6ppm	0	X	X	X	X	X	X	X	X	X	无补偿
补偿值设定										补偿数																																																																																																												
EN	CR[8:0]																																																																																																																					
1	1	0	0	0	0	0	0	0	0	-274.6ppm																																																																																																												
	1	0	0	0	0	0	0	0	1	-273.7ppm																																																																																																												
	:	:	:	:	:	:	:	:	:	:																																																																																																												
	0	0	0	0	1	1	1	1	1	-0.95ppm																																																																																																												
	0	0	0	1	0	0	0	0	0	0ppm																																																																																																												
	:	:	:	:	:	:	:	:	:	:																																																																																																												
	0	1	1	1	1	1	1	1	1	0	+211.7ppm																																																																																																											
	0	1	1	1	1	1	1	1	1	1	+212.6ppm																																																																																																											
0	X	X	X	X	X	X	X	X	X	无补偿																																																																																																												

补偿原理说明与计算:

由于计数器采用 32.768KHz 的时钟计数, 如果需要对每秒精度进行补偿时, 只能按照 32.768KHz 的整数周期补偿, 则每秒补偿的最小单位为 $(1/32768) * 10^6 = 30.5\text{ppm}$, 无法满足高精度的要求。

那么要在 32.768KHz 的计数时钟下实现精度较高的时钟补偿时, 需要在算法上做调

整，将最大补偿周期扩大 32 倍。则在只能补偿的最小单位为 30.5ppm 的情况下，平均每秒的补偿单位变为为 30.5ppm/32=0.96ppm。满足了精度较高的时钟补偿要求。而且补偿发生在每 32 秒内比较均匀的范围。所以，该寄存器中引入了 5 位小数的设定。

设定值计算如下：

$$CR[8 : 0] = \left(\frac{\text{补偿目标值 [ppm]} \times 2^{15}}{10^6} \right)_{\text{取2的补码}} + 0001.00000B$$

如果补偿目标值为 +20.6ppm，计算相应的寄存器值如下：

$$\begin{aligned} CR[8:0] &= (20.3 \times 2^{15}/10^6)_{\text{取 2 的补码}} + 0001.00000B \\ &= (0.6651904)_{\text{取 2 的补码}} + 0001.00000B \\ &= 0000.10101B + 0001.00000B \\ &= 0001.10101B \end{aligned}$$

如果补偿目标值为-20.6ppm，计算相应的寄存器值如下：

$$\begin{aligned} CR[8:0] &= (-20.3 \times 2^{15}/10^6)_{\text{取 2 的补码}} + 0001.00000B \\ &= (-0.6651904)_{\text{取 2 的补码}} + 0001.00000B \\ &= 1111.01011B + 0001.00000B \\ &= 0000.01011B \end{aligned}$$

14 看门狗定时器 (WDT)

14.1 WDT 简介

WDT 用来检测和解决由软件错误引起的故障。当 WDT 计数器达到设定的溢出时间后，会触发中断或产生系统复位。WDT 由专用的 10KHz 片内振荡器驱动。

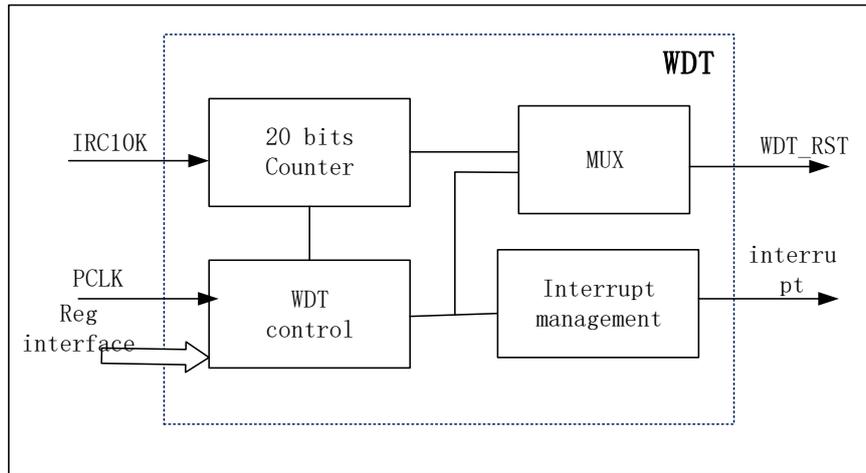


图 14-1 WDT 整体框图

14.2 WDT 功能描述

- 20Bit 自由运行的递增计数器，溢出时间可配置为 1.6ms – 50s。
- 溢出后的动作可配置为中断或复位。
- WDT 时钟由独立的 RC 振荡器提供，可在 Sleep 和 DeepSleep 模式下工作。
- WDTCON 寄存器只有在 WDT 未被启动时才能修改，以防止启动后无意之间修改 WDT 的配置。

14.2.1 WDT 溢出后产生中断

在本模式下，WDT 将按所设定的时间周期性地产生中断。在中断服务程序中需要清除 WDT 中断标志。

配置方法如下所示：

Step1: 配置 WDT_CON.WOV，选择 WDT 计时溢出时间。

Step2: 设置 WDT_CON.WINT_EN 为 1，选择 WDT 溢出后产生中断。

Step3: 使能 NVIC 中断向量表中的 WDT 中断。

Step4: 向 WDT_RST 寄存器依次写入 0x1E、0xE1，启动 WDT 定时器。

Step5: 在中断服务程序中向 WDT_RST 寄存器依次写入 0x1E、0xE1 以清除中断标志。

14.2.2 WDT 溢出后产生复位

在本模式下，WDT 计数器溢出后会产生 Reset 信号，该信号会复位 MCU。用户程序需要在 WDT 溢出前清零 WDT 计数器，从而避免产生 WDT 复位。

配置方法如下所示：

Step1: 配置 WDT_CON.WOV，选择 WDT 计数器溢出时间。

Step2: 设置 WDT_CON.WINT_EN 为 0，选择 WDT 溢出后产生复位。

Step3: 向 WDT_RST 寄存器依次写入 0x1E、0xE1，启动 WDT 定时器。

Step4: 在 WDT 溢出前向 WDT_RST 寄存器依次写入 0x1E、0xE1 以清零 WDT 计数器。

注：由于 WDT 振荡器是低精度的 RC 振荡器，强烈建议在 WDT 计数器计数值到达溢出值的一半之前对 WDT 进行清零。

14.3 WDT 寄存器描述

基地址 0X40000C00

寄存器	偏移地址	描述
WDT_RST	0X080	WDT清除控制寄存器
WDT_CON	0X084	WDT控制寄存器

表 14-1 WDT 寄存器列表

14.3.1 WDT 清除控制寄存器 (WDT_RST)

偏移地址: 0x080

复位值: 0x0000 0000

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

位	符号	描述
31:8	Reserved	保留位, 读为0
7:0	WDTRST	看门狗启动/清零控制 当看门狗未启动时, 向该寄存器依次写入0x1E、0xE1, 启动WDT定时器。 当看门狗已启动时, 向该寄存器依次写入0x1E、0xE1, 清零WDT定时器及中断标志。

14.3.2 WDT_CON 寄存器

偏移地址: 0x084

复位值: 0x0000 000F

注:

- 该寄存器只有在 WDT 未运行时才可以写入。

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			

位	符号	描述
31:16	Reserved	保留位, 读为0
15:8	WCNTL	WDT计数器低8位
7	WDTINT	WDT中断标志 1: 已发生WDT中断, 向WDT_RST寄存器依次写入0x1E、0xE1以清除该中断标志。 0: 未发生WDT中断。
5	WINT_EN	WDT溢出后的动作配置 1: WDT溢出后产生中断。 0: WDT溢出后产生复位。
4	WDTR	WDT运行标志 1: WDT正在运行 0: WDT停止
3:0	WOV[3:0]	WDT计时溢出时间配置 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 通用同步异步收发器 (UART)

15.1 概述

本产品带有 2 个通用 UART 模块 (UART0/1)，通用同步异步收发器 (UART) 能够灵活地与外部设备进行全双工数据交换，它支持同步单向通信以及多处理器通信。常用于短距离、低速的串行通信中。UART 通过可编程波特率发生器提供了多种波特率。UART0 的波特率由 TIMER0 产生，UART1 的波特率由 TIMER1 产生。UART 支持多种工作模式。

15.2 结构框图

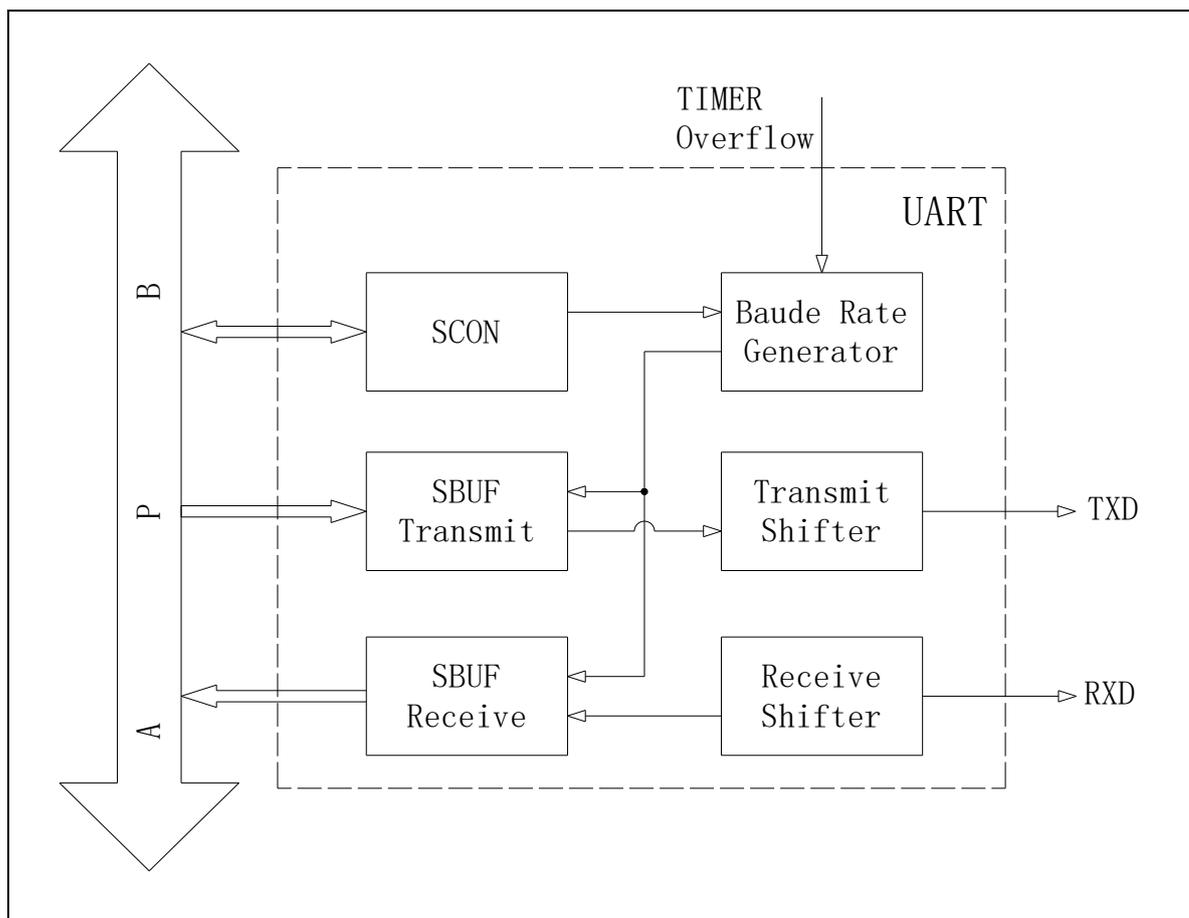


图 15-1 UART 结构框图

15.3 主要特性

通用 UART 模块支持以下基本功能：

- 全双工传输、半双工传输
- 可编程串行通信功能
 - 两种字符长度：8 比特、9 比特
 - Mode0/1/2/3 四种传输模式
- 16 比特波特率发生器
- 多机通讯
- 自动地址识别

15.4 功能描述

15.4.1 工作模式

UART 支持多种工作模式: 同步半双工模式、异步全双工模式。通过 UARTx_SCON.SM0 和 UARTx_SCON.SM1 搭配, 即可配置出所需要的各种工作模式。

15.4.1.1 Mode0~Mode3 功能对比

配置 UARTx_SCON.SM 可选择不同的传输模式: Mode0~Mode3。这四种工作模式的主要功能对如下表所示:

工作模式		传输位宽	数据组成	波特率
Mode0	同步模式 半双工	8bit	Data(8bit)	$BaudRate = \frac{f_{PCLK}}{12}$
Mode1	异步模式 全双工	10bit	Start (1bit) + Data(8bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{PCLK}}{32 * (65536 - TM)}$
Mode2	异步模式 全双工	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{PCLK}}{64}$
Mode3	异步模式 全双工	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{PCLK}}{32 * (65536 - TM)}$

表 15-1 Mode0/1/2/3 数据结构

注:

- Mode0 只能作为主机发送 UART 同步移位时钟, 不可以作为从机接受外部输入的 UART 同步移位时钟。
- f_{PCLK} 代表当前 PCLK 的频率。
- DBAUD 的定义详见 UARTx_SCON。
- TM 为 TIMER 计数值。注意, TIMER 必须配置为 16 位自动重载入模式, 计数寄存器和重载寄存器都得写入 TM 值。

15.4.1.2 Mode0 (同步模式, 半双工)

当工作在 Mode0 时, UART 工作在同步模式, 其波特率为固定的 PCLK 时钟的 1/12。

UART 接收数据由 RXD 输入、UART 发送数据有 RXD 输出，RXD 此时为输入输出端口。UART 同步移位时钟由 TXD 输出，TXD 此时为输出端口。注意，本模式只能作为主机发送同步移位时钟，不可以作为从机从外部接收该时钟。该模式下，传输的数据位宽只能是 8 位的，没有起始位和结束位。

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。

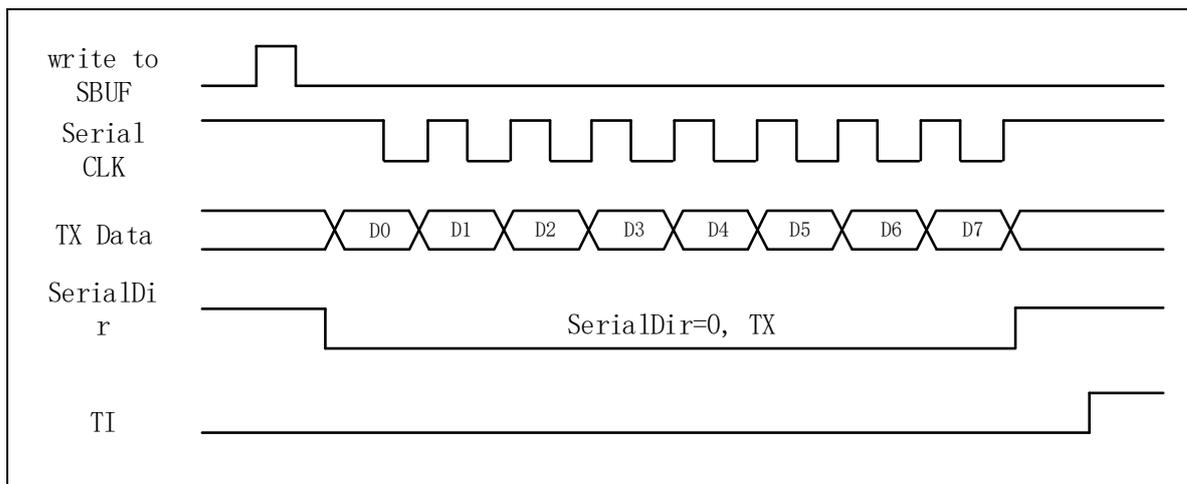


图 15-2 Mode0 发送数据

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。

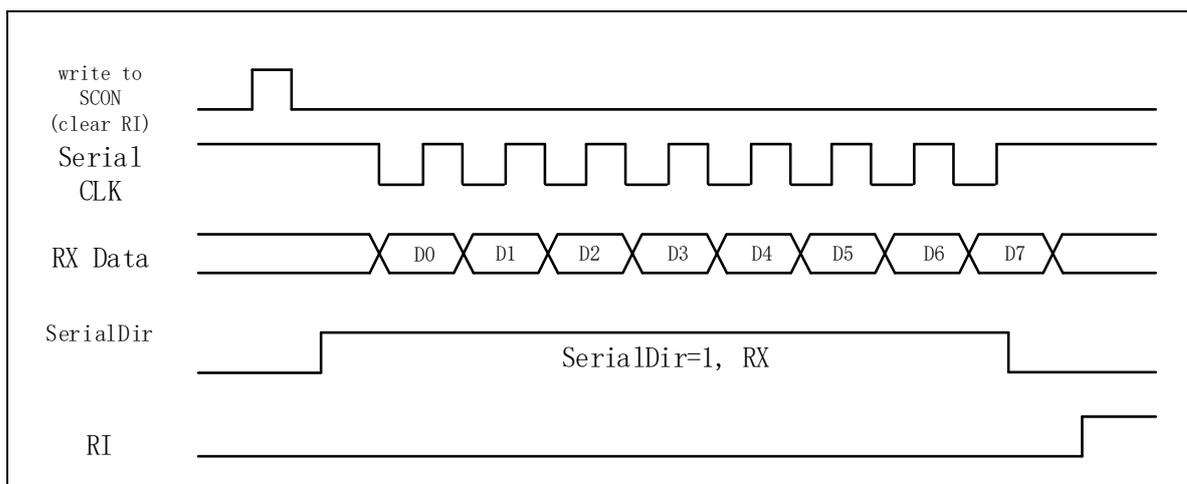


图 15-3 Mode0 接收数据

15.4.1.3 Mode1（异步模式，全双工）

当工作在 Mode1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 10 位组成：起始位“0”开始，紧接着 8 位数据位（低位在先，高位在后），最后是结束位“1”。

该模式下，波特率由定时器模块产生，并且是可编程的。UART0 的波特率由 TIMER0 产生，UART1 的波特率由 TIMER1 产生。

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）。

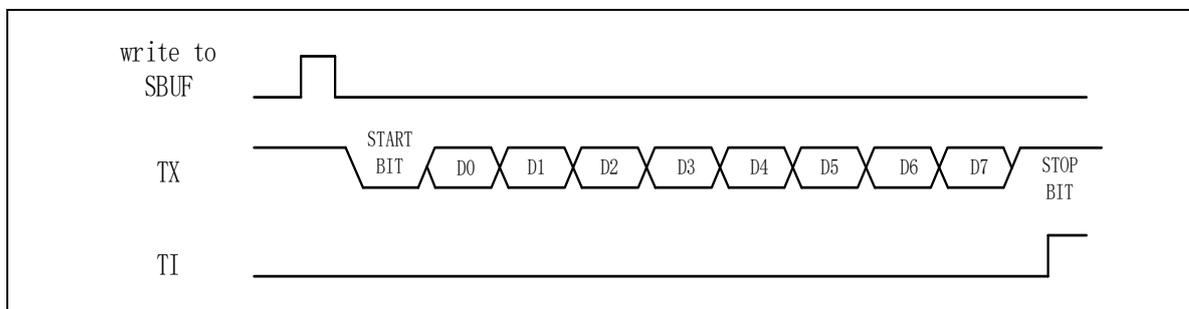


图 15-4 Mode1 发送数据

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。

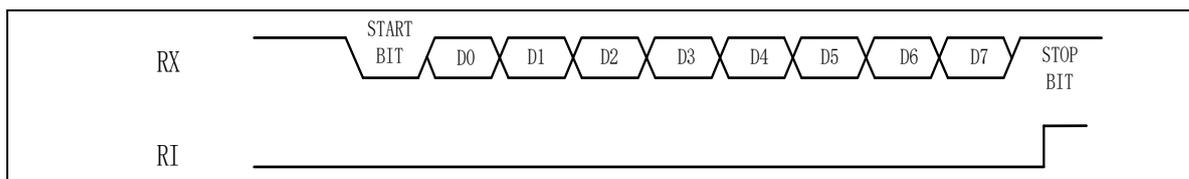


图 15-5 Mode1 接收数据

15.4.1.4 Mode2（异步模式，全双工）

当工作在 Mode2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 11 位组成：起始位“0”开始，接着是 8 个数据位，1 个 TB8 位和结束位。额外的 TB8 位是用来在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位来使用。该模式下，波特率可以独立产生，不需要外部 TIMER 产生。

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）。

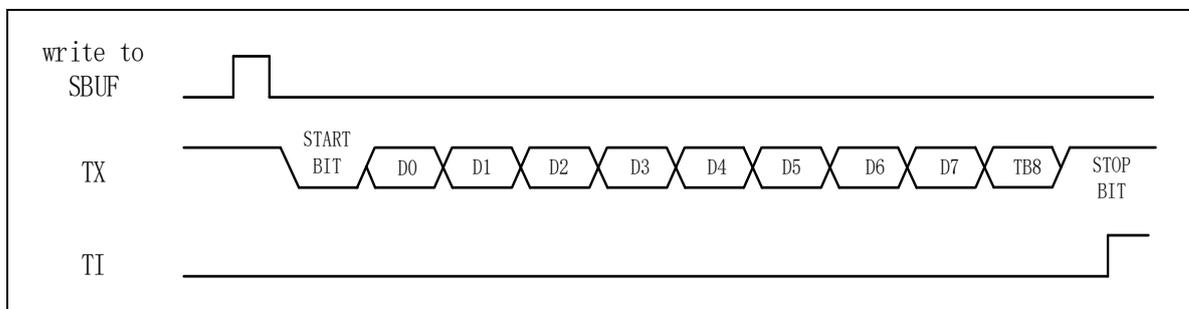


图 15-6 Mode2 发送数据

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。

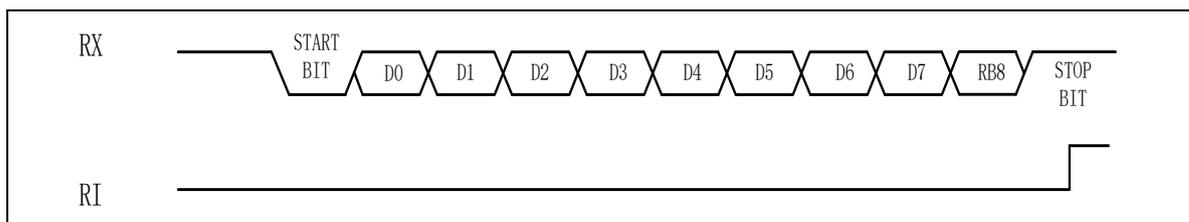


图 15-7 Mode2 接收数据

15.4.1.5 Mode3（异步模式，全双工）

Mode3 的数据格式，传输时序以及操作方式都与 Mode2 相同，唯一的区别是 Mode3 的波特率由 TIMER 产生，而不是像 Mode2 由设备自己独立产生。Mode3 的波特率是可编程的，波特率生成方式与 Mode1 相同。本产品中，UART0 的波特率由 TIMER0 产生，UART1 的波特率由 TIMER1 产生。

将 UARTx_SCON.SM0 置 1，UARTx_SCON.SM1 置 1，可进入 Mode3 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）。

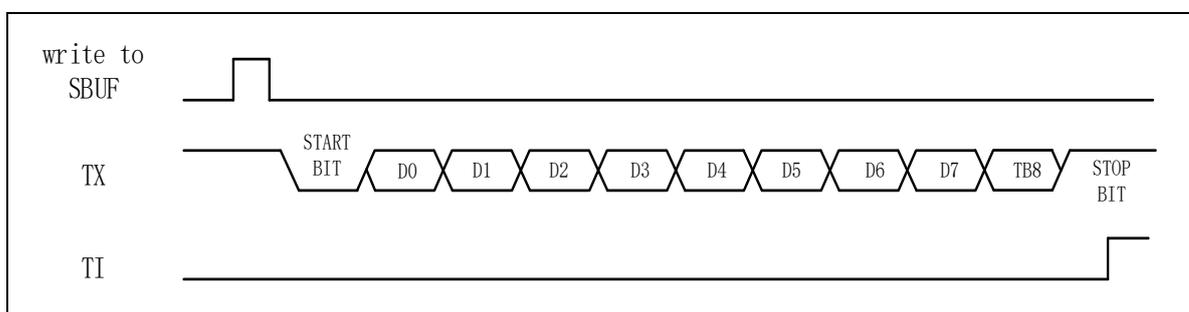


图 15-8 Mode3 发送数据

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。

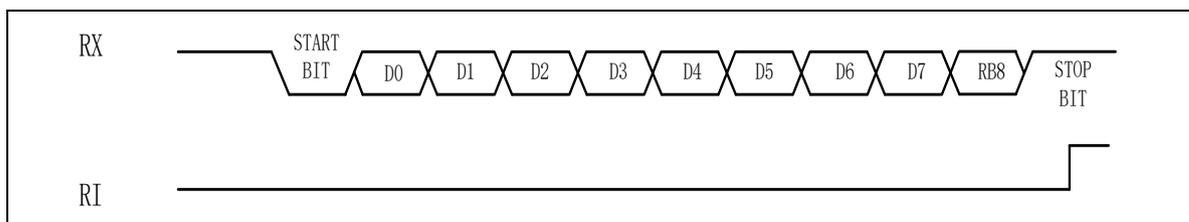


图 15-9 Mode3 接收数据

15.4.2 波特率生成

Mode0~Mode3 生成波特率的公式不尽相同，详见下方所示：

$$\text{Mode0 波特率生成公式: } \mathit{BaudRate} = \frac{f_{PCLK}}{12}$$

$$\text{Mode1 波特率生成公式: } \mathit{BaudRate} = \frac{(DBAUD+1)f_{PCLK}}{32*(65536-TM)}$$

$$\text{Mode2 波特率生成公式: } \mathit{BaudRate} = \frac{(DBAUD+1)f_{PCLK}}{64}$$

$$\text{Mode3 波特率生成公式: } \mathit{BaudRate} = \frac{(DBAUD+1)f_{PCLK}}{32*(65536-TM)}$$

注：

- f_{PCLK} 代表当前 PCLK 的频率。
- DBAUD 的定义详见 UARTx_SCON。
- TM 为 TIMER 计数值。注意，TIMER 必须配置为 16 位自动重载入模式，计数寄存器和重载寄存器都得写入 TM 值。

15.4.2.1 Mode1/Mode3 波特率设置示例

波特率	PCLK = 1 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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!

波特率	PCLK = 4 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 10 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 14 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 20 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 24 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 2 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 8 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 11.0592 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 16 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 32 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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%

波特率	PCLK = 22.12 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
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 帧错误检测

当工作在 Mode1/2/3 时，UART 具有帧错误检测功能，硬件会自动检测接收到的帧数

据是否带有效的 Stop 位。如果接受数据是硬件未在预期内收到有效 Stop 位，从而出现同步失效或过度的噪声，则 UARTx_ISR.FE 置 1。UARTx_ISR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有效 Stop 位，也不会把 UARTx_ISR.FE 标志清 0。

15.6 多机通讯

Mode2/3 具有多机通讯功能，为此在其帧格式中增加了 1 位 TB8/RB8。将 UARTx_SCON.SM2 置“1”，可开启多机通讯位。

当开启多机通讯位后，发送数据时，主机可以通过 UARTx_SCON.TB8 来区分当前帧是地址帧 (UARTx_SCON.TB8=1) 还是数据帧 (UARTx_SCON.TB8=0)。接收数据时，从机会忽略 RB8 位 (第 9 位) 为“0”的当前接收帧。

- 当为数据帧时，该帧数据不会存入到从机的 UARTx_SBUF 寄存器中，从机也不会产生接收中断。
- 当为地址帧时，由于多机通讯中自动地址识别功能已开启，使得从机可以检测接收到的地址与其自身地址是否相符合。
 - 如果地址相符合，从机会对 UARTx_SCON.RB8 置“1”，UARTx_ISR.RI 置“1”。从机软件看到 UARTx_SCON.RB8=1 并且 UARTx_ISR.RI=1 后，将 UARTx_SCON.SM2 位清“0”，接受数据帧。
 - 如果地址不符合，表明主机并不是寻址该从机，从机硬件保持 UARTx_SCON.RB8 和 UARTx_ISR.RI 为“0”，软件保持 UARTx_SCON.SM2 位为“1”，继续处于地址监听状态。

注：如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 stop 位代替。当从机接收到匹配的地址帧和有效的 stop 位时，UARTx_ISR.RI 会被置“1”。

15.7 自动地址识别

当开启多机通讯位后（UARTx_SCON.SM2 置“1”），自动地址识别功能也将开启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收端会给出 UARTx_ISR.RI 接收标志。如果地址不匹配，则接收端不会给出任何接收标志。

15.7.1 给定地址

UART 设备的 UARTx_SADDR 寄存器用来表示自己的设备给定地址，UARTx_SADEN 寄存器是地址掩码，可以用来定义地址中的无关位。当 UARTx_SADEN 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，UARTx_SADEN 寄存器必须设为 8'hFF。

$$\text{GivenAddr} = \text{SADDR} \& \text{SADEN}$$

15.7.1.1 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 8'hFF。

$$\text{BroadCastAddr} = \text{SADDR} | \text{SADEN}$$

15.7.1.2 举例

假设某从机的 UARTx_SADDR 和 UARTx_SADEN 配置如下：

SADDR: 8'b01101001

SADEN: 8'b11111011

那么其给定地址和广播地址如下：

Given: 8'b01101x01

Broadcast: 8'b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

8'b01101001 和 8'b01101101 (given address)

8'b11111011 和 8'b11111111 (broadcast address)。

15.8 收发端缓存

15.8.1 接收缓存

通用 UART (UART0/1) 接收端有一个帧长度 (8/9bits) 的接收缓存, 也就是说当一帧数据接收完毕后, 接收缓存中的数据会被一直保持, 直到下一帧数据的 Stop 位接收完毕后, 接收缓存才会更新为新一帧数据。

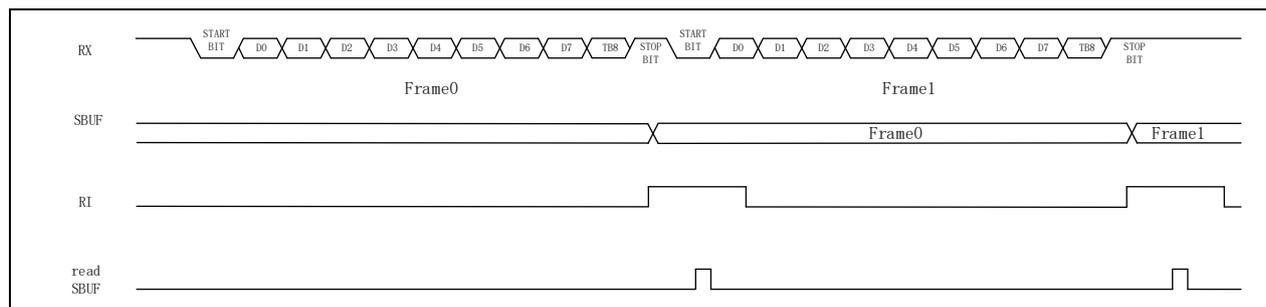


图 15-10 接收缓存

15.8.2 发送缓存

通用 UART (UART0/1) 发送端不支持发送缓存。如果在发送数据过程中, 填写 UARTx_SBUF 寄存器, 将会破坏当前正在发送数据。软件应该避免这种操作。

15.9 寄存器

UART0 基地址: 0x4000 0000

UART1 基地址: 0x4000 0100

寄存器	偏移地址	描述
UARTx_SBUF	0x00	数据寄存器
UARTx_SCON	0x04	控制寄存器
UARTx_SADDR	0x08	地址寄存器
UARTx_SADEN	0x0C	地址掩码寄存器
UARTx_ISR	0x10	中断标志位寄存器
UARTx_ICR	0x14	中断标志位清除寄存器

15.9.1 数据寄存器 (UARTx_SBUF)

偏移地址: 0x00

复位值: 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							

位	标记	功能描述
31:8	Reserved	
7:0	SBUF	发送数据时, 待发送数据写入该寄存器; 接收数据时, 从该寄存器中读出接收到的数据; 注意, 对该寄存器读的值实际是 RxBuffer 中的值, 对该寄存器写的值实际是写到了 TXShifter 中。

15.9.2 控制寄存器 (UARTx_SCON)

偏移地址: 0x04

复位值: 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

位	标记	功能描述
31:10	Reserved	
9	DBAUD	波特率倍率设置 0: 单倍波特率; 1: 双倍波特率
8	Reserved	
7:6	SM01	工作模式配置 00: mode0; 01: mode1; 10: mode2; 11: mode3
5	SM2	多机通讯使能控制 0: 关闭多机通信功能 1: 使能多机通信功能
4	REN	接收使能控制 Mode0: 0: 发送, 1: 接收 其他: 0: 发送, 1: 接收/发送
3	TB8	发送数据时待发送的TB8位
2	RB8	接收数据时收到的RB8位
1	TIEN	发送完成中断使能 0: 禁止发送完成中断 1: 使能发送完成中断
0	RIEN	接收完成中断使能 0: 禁止接收完成中断 1: 使能接收完成中断

15.9.3 地址寄存器 (UARTx_SADDR)

偏移地址: 0x08

复位值: 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							

位	标记	功能描述
31:8	Reserved	
7:0	SADDR	从机设备地址

15.9.4 地址掩码寄存器 (UARTx_SADEN)

偏移地址: 0x0C

复位值: 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							

位	标记	功能描述
31:8	Reserved	
7:0	SADEN	从机设备地址掩码

15.9.5 标志位寄存器 (UARTx_ISR)

偏移地址: 0x10

复位值: 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	TI	RI
R													R	R	R

位	符号	描述
31:3	Reserved	
2	FE	接收帧错误标志位; 硬件置位, 软件清零 1: FE中断标志有效 0: FE中断标志无效
1	TI	发送完成中断标志位; 硬件置位, 软件清零 1: TI中断标志有效 0: TI中断标志无效
0	RI	接收完成中断标志位; 硬件置位, 软件清零 1: RI中断标志有效 0: RI中断标志无效

15.9.6 标志位清除寄存器 (UARTx_ICR)

偏移地址: 0x14

复位值: 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

位	标记	功能描述
31:3	RESERVED	
2	FECLR	接收帧错误标志清除 写0清零, 写1无效
1	TICLR	发送完成中断标志清除 写0清零, 写1无效
0	RICLR	接收完成中断标志清除 写0清零, 写1无效

16 低功耗同步异步收发器（LPUART）

16.1 概述

本产品带有 1 个 LPUART 模块，LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信；波特率既可由外部 TIMER2 产生，也可以由模块内部逻辑产生。

LPUART 为支持低功耗应用，除了原本的 PCLK 时钟外，增加了一路 SCLK 时钟。LPUART 模块内部寄存器配置逻辑工作于 PCLK 时钟域，数据收发逻辑工作于 SCLK 时钟域。当系统进入低功耗模式后，关闭高频 PCLK 时钟，打开低频 SCLK 时钟，LPUART 仍旧可以进行正常的收发。

SCLK 时钟来源可以选择：PCLK、外部低速时钟（XTL），内部低速时钟（RCL）。在 LPMODE=1 时，SCLK 时钟还支持 1/2/4/8/16/32/64/128 倍的预分频。

注意，在 LPMODE=0 时，LPUART 接收的是 TIMER2 时钟的 Toggle 输出信号而非 Overflow 信号，因此必须使能 TIMER2 的 Toggle 输出。

16.2 结构框图

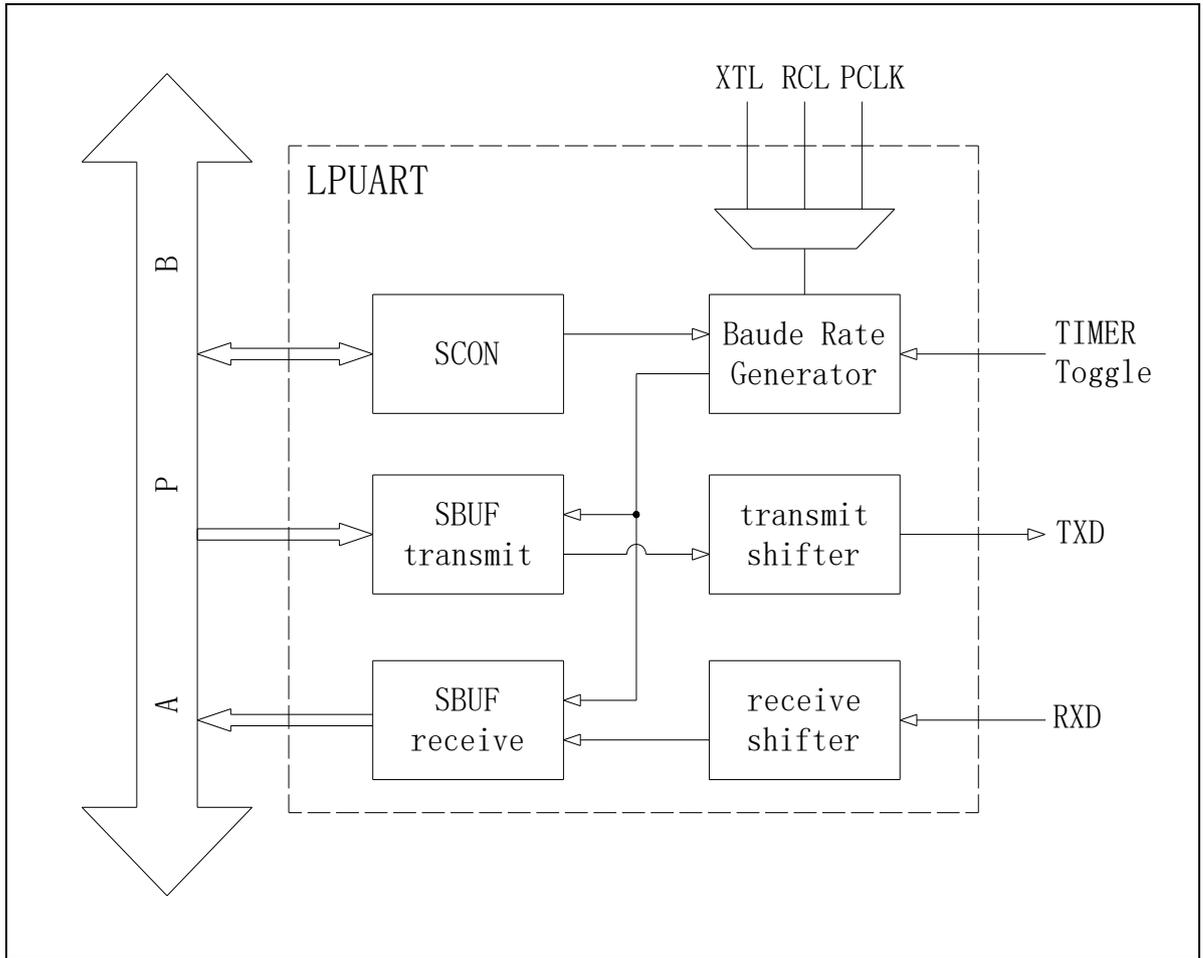


图 16-1 LPUART 结构框图

16.3 主要特性

LPUART 模块支持以下基本功能：

- 配置时钟 PCLK
- 传输时钟 SCLK（SCLK 可选择 XTL、RCL 以及 PCLK）
- 系统低功耗模式下收发数据
- 全双工传输、半双工传输
- 可编程串行通信功能
 - 两种字符长度：8 比特、9 比特
 - Mode0/1/2/3 四种传输模式
- 16 比特波特率计数器
- 多机通讯
- 自动地址识别

16.4 功能描述

16.4.1 配置时钟和传输时钟

LPUART 模块有两个时钟：配置时钟 PCLK 和传输时钟 SCLK。

- **配置时钟**

配置时钟用于系统 APB 总线对 LPUART 模块进行寄存器配置，固定为 PCLK。

- **传输时钟**

传输时钟用于 LPUART 数据收发逻辑工作，可选择 XTL、RCL 及 PCLK。当 SCLK 为 XTL 或 RCL，系统进入 DeepSleep 时，LPUART 仍可正常收发数据。

16.4.2 工作模式

LPUART 支持多种工作模式：同步半双工模式、异步全双工模式。通过配置 LPUARTx_SCON.SM 的值，即可获取所需要的工作模式。

LPUART 比 UART 增加了一个 LPMODE 控制位。当该位置“1”时，只支持 Mode1/3 工作模式，并且波特率生成方式也会发生改变。

16.4.2.1 Mode0~Mode3 功能对比

当 LPMODE=0 时:

配置 LPUARTx_SCON.SM 可选择 4 种传输模式, 各模式功能对比如下所示:

工作模式		传输位宽	数据组成	波特率
Mode0	同步模式 半双工	8bit	Data(8bit)	$BaudRate = \frac{f_{SCLK}}{12}$
Mode1	异步模式 全双工	10bit	Start (1bit) + Data(8bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{SCLK}}{32 * (65536 - TM)}$
Mode2	异步模式 全双工	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{SCLK}}{64}$
Mode3	异步模式 全双工	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{(DBAUD + 1)f_{SCLK}}{32 * (65536 - TM)}$

表 16-1 Mode0/1/2/3 数据结构

当 LPMODE=1 时:

Mode2	异步模式 全双工	11bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop(1bit)	$BaudRate = \frac{f_{SCLK}}{PreScale * 4}$
-------	-------------	-------	--	--

注:

- Mode0 只能作为主机发送 LPUART 同步移位时钟, 不可以作为从机接受外部输入的 LPUART 同步移位时钟。
- f_{SCLK} 代表当前 SCLK 的频率。
- DBAUD 的定义详见 LPUARTx_SCON。
- TM 为 TIMER 计数值。注意, TIMER 必须配置为 16 位自动重载入模式, 计数寄存器和重载寄存器都得写入 TM 值。
- PreScale 为预分频系数。

16.4.2.2 Mode0（同步模式，半双工）数据收发说明

设置 LPUART_SCON.SM 为 0, LpUart 工作于 Mode0, 时钟与数据同步输入输出, 波特率固定为 SCLK/12。传输的数据宽度固定为 8 比特, 没有起始位和结束位。数据的接收及发送均通过 RXD 引脚实现; 同步移位时钟由 TXD 引脚输出。注意, TXD 引脚不接受输入时钟。

当 LPMODE=1 时, 不支持 Mode0 工作模式。

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。

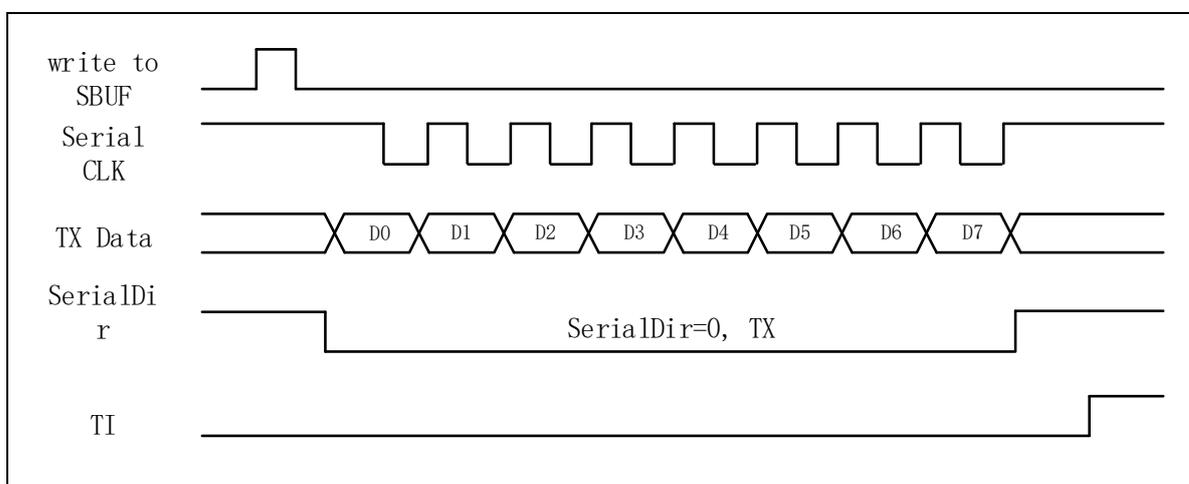


图 16-2 Mode0 发送数据

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。

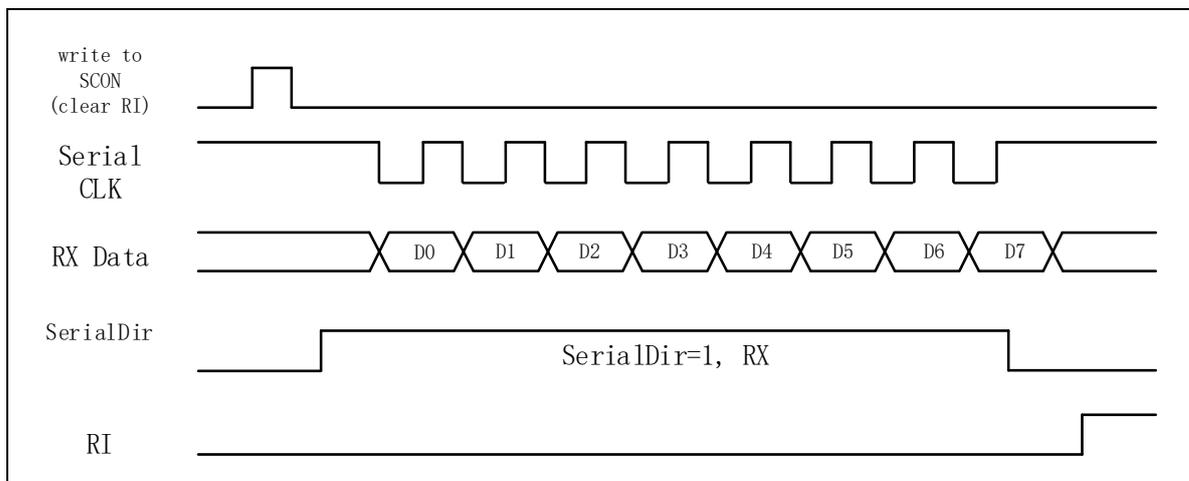


图 16-3 Mode0 接收数据

16.4.2.3 Mode1（异步模式，全双工）数据收发说明

当工作在 Mode1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 10 位组成：起始位“0”开始，紧接着 8 位数据位（低位在先，高位在后），最后是结束位“1”。

该模式下，LPUART 的波特率由定时器 TIMER2 模块产生，并且是可编程的。

将 LPUART_SCON.SM0 清 0, LPUART_SCON.SM1 置 1, 可进入 Mode1 工作模式。

当 LPMODE=1 时，支持 Mode1 工作模式，但是波特率计算方式发生改变，具体参考波特率生成章节。

发送数据时，与 LPUART_SCON.REN 的值无关，将所发送数据写入 LPUART_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

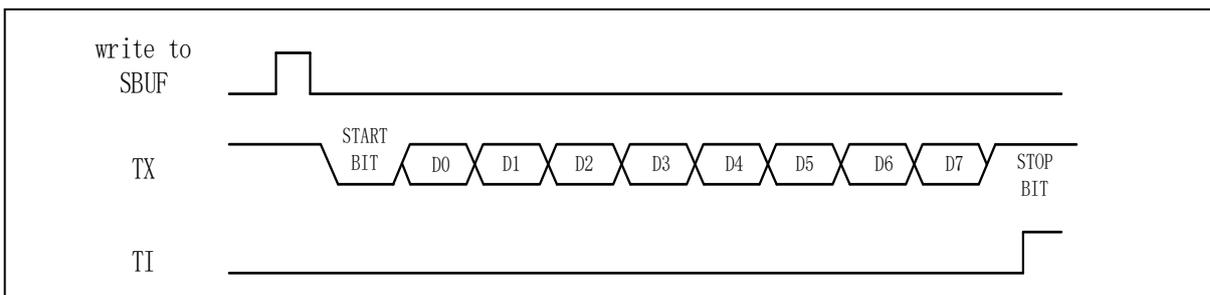


图 16-4 Mode1 发送数据

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。

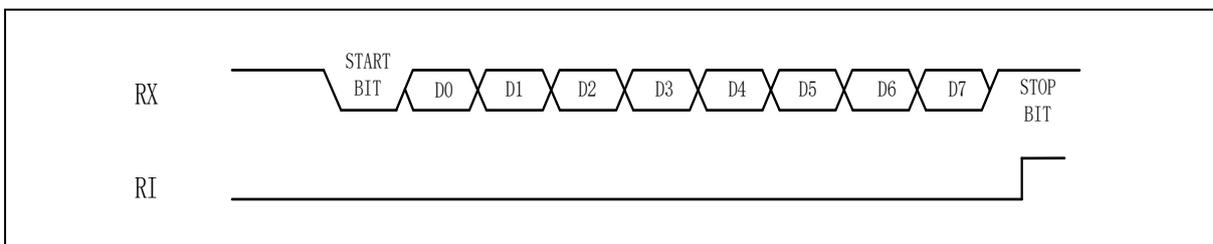


图 16-5 Mode1 接收数据

16.4.2.4 Mode2（异步模式，全双工）数据收发说明

当工作在 Mode2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 11 位组成：起始位“0”开始，接着是 8 个数据位，1 个 TB8 位和结束位。额外的 TB8 位是用来在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位来使用。该模式下，波特率可以独立产生，不需要外部 TIMER 产生。

将 LPUART_SCON.SM0 置 1, LPUART_SCON.SM1 清 0, 可进入 Mode2 工作模式。当 LPMODE=1 时，不支持 Mode2 工作模式。

发送数据时，与 LPUART_SCON.REN 的值无关，并将所发送数据写入 LPUART_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

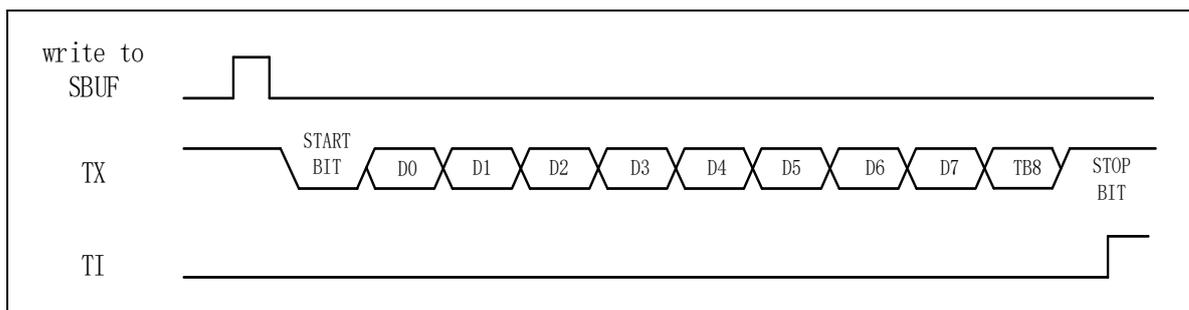


图 16-6 Mode2 发送数据

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。

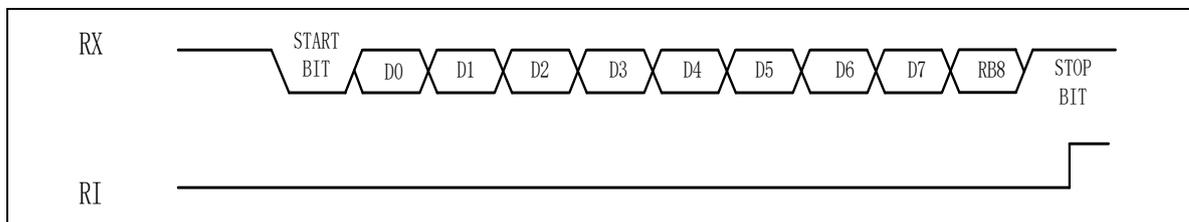


图 16-7 Mode2 接收数据

16.4.2.5 Mode3（异步模式，全双工）数据收发说明

Mode3 的数据格式，传输时序以及操作方式都与 Mode2 相同，唯一的区别是 Mode3 的波特率由 TIMER 产生，而不是像 Mode2 由设备自己独立产生。Mode3 的波特率是可编程的，波特率生成方式与 Mode1 相同。

将 LPUART_SCON.SM0 置 1, LPUART_SCON.SM1 置 1, 可进入 Mode3 工作模式。当 LPMODE=1 时，支持 Mode3 工作模式。但是波特率计算方式发生改变，具体参考波特率生成章节。

发送数据时，与 LPUART_SCON.REN 的值无关，并将所发送数据写入 LPUART_SBUF 寄存器中，数据就会从 TXD 移出（低位在先，高位在后）。

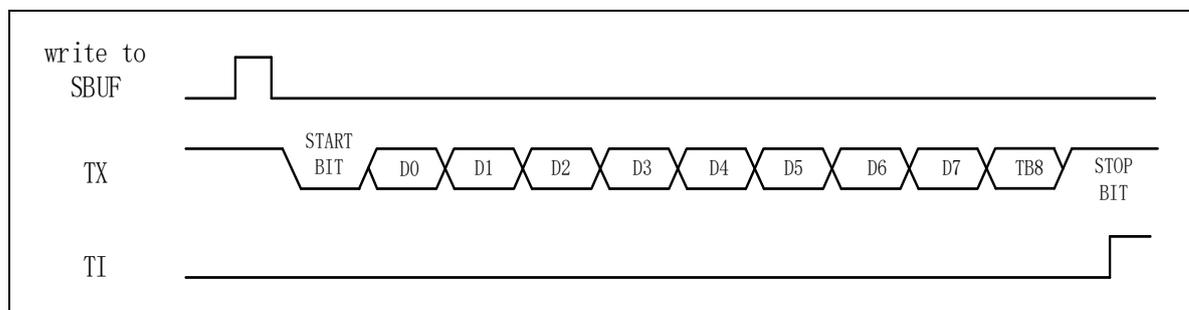


图 16-8 Mode3 发送数据

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。

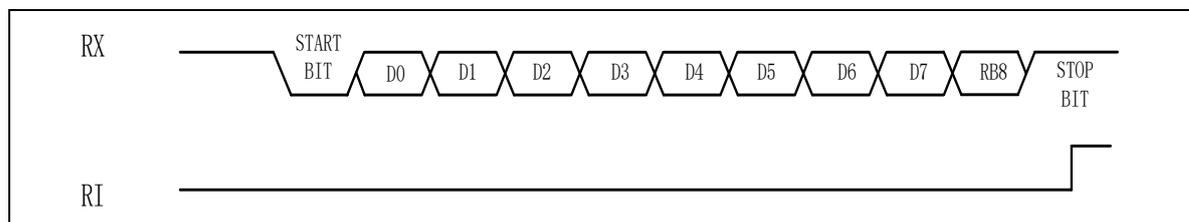


图 16-9 Mode3 接收数据

16.4.3 波特率生成

LPMODE=0

Mode0~Mode3 生成波特率的公式如下所示：

$$\text{Mode0 波特率生成公式: } \text{BaudRate} = \frac{f_{SCLK}}{12}$$

$$\text{Mode1 波特率生成公式: } \text{BaudRate} = \frac{(DBAUD+1)f_{SCLK}}{32*(65536-TM)}$$

$$\text{Mode2 波特率生成公式: } \text{BaudRate} = \frac{(DBAUD+1)f_{SCLK}}{64}$$

$$\text{Mode3 波特率生成公式: } \text{BaudRate} = \frac{(DBAUD+1)f_{SCLK}}{32*(65536-TM)}$$

LPMODE=1

当 LPMODE=1 时，不支持 Mode0、Mode2。

$$\text{Mode1、Mode3 波特率生成公式: } \text{BaudRate} = \frac{f_{SCLK}}{\text{PreScale} * 4}$$

注：

- f_{SCLK} 代表当前 SCLK 的频率。
- DBAUD 的定义详见 LPUARTx_SCON。
- TM 为 TIMER 计数值。注意，TIMER 必须配置为 16 位自动重载入模式，计数寄存器和重载寄存器都得写入 TM 值。
- PreScale 为预分频系数。

16.5 帧错误检测

当工作在 Mode1/2/3 时，LPUART 具有帧错误检测功能，硬件会自动检测接收到的帧数据是否带有有效的 Stop 位。如果接受数据时未在预期时间内收到有效 Stop 位，从而出现同步失效或过度的噪声，则 LPUART_ISR.FE 位置 1。LPUART_ISR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有有效 Stop 位，也不会把 LPUART_ISR.FE 标志清 0。

16.6 多机通讯

Mode2/3 具有多机通讯功能,为此在其帧格式中增加了 1 位 TB8/RB8。将 SCON.SM2 置“1”,可开启多机通讯位。

当开启多机通讯位后,发送数据时,主机可以通过 LPUART_SCON.TB8 来区分当前帧是地址帧 (LPUART_SCON.TB8=1) 还是数据帧 (LPUART_SCON.TB8=0)。

- 当为数据帧时,该帧数据不会存入到从机的 LPUARTx_SBUF 寄存器中,从机也不会产生接收中断。
- 当为地址帧时,由于多机通讯中自动地址识别功能已开启,使得从机可以检测接收到的地址与其自身地址是否相符合。
 - 如果地址相符合,从机会对 LPUARTx_ISR.RI 置“1”, LPUARTx_SCON.RB 置“1”。从机软件看到 LPUARTx_SCON.RB=1 并且 LPUARTx_ISR.RI=1 后,将 LPUARTx_SCON.SM2 位清“0”,接受数据帧。
 - 如果地址不符合,表明主机并不是寻址该从机,从机硬件保持 LPUARTx_RB8 和 LPUARTx_ISR.RI 为“0”,软件保持 LPUARTx_SCON.SM2 位为“1”,从机继续处于地址监听状态。

注:

- 如果有需要,也可以在 Mode1 下开启多机通讯位,此时 TB8 位由 stop 位代替。当从机接收到匹配的地址帧和有效的 stop 位时, LPUARTx_ISR.RC 会被置“1”。

16.7 自动地址识别

当开启多机通讯位后（LPUART_SCON.SM2 置“1”），自动地址识别功能也将开启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收端会给出 LPUART_ISR.RI 接收标志。如果地址不匹配，则接收端不会给出任何接收标志。

16.7.1 给定地址

LPUART 设备的 LPUART_SADDR 寄存器用来表示自己的设备给定地址，LPUART_SADEN 寄存器是地址掩码，可以用来定义地址中的无关位。当 LPUART_SADEN 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，LPUART_SADEN 寄存器必须设为 8'hFF。

$$\text{GivenAddr} = \text{SADDR} \& \text{SADEN}$$

16.7.2 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 8'hFF。

$$\text{BroadCastAddr} = \text{SADDR} | \text{SADEN}$$

16.7.3 举例

假设某从机的 LPUART_SADDR 和 LPUART_SADEN 配置如下：

SADDR: 8'b01101001

SADEN: 8'b11111011

那么其给定地址和广播地址如下：

Given: 8'b01101x01

Broadcast: 8'b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

8'b01101001 和 8'b01101101 (given address)

8'b11111011 和 8'b11111111 (broadcast address)。

16.8 收发端缓存

16.8.1 接收缓存

LPUART 接收端有一个帧长度（8/9bits）的接收缓存，也就是说当一帧数据接收完毕后，接收缓存中的数据会被一直保持，直到下一帧数据的 Stop 位接收完毕后，接收缓存才会更新为新一帧数据。

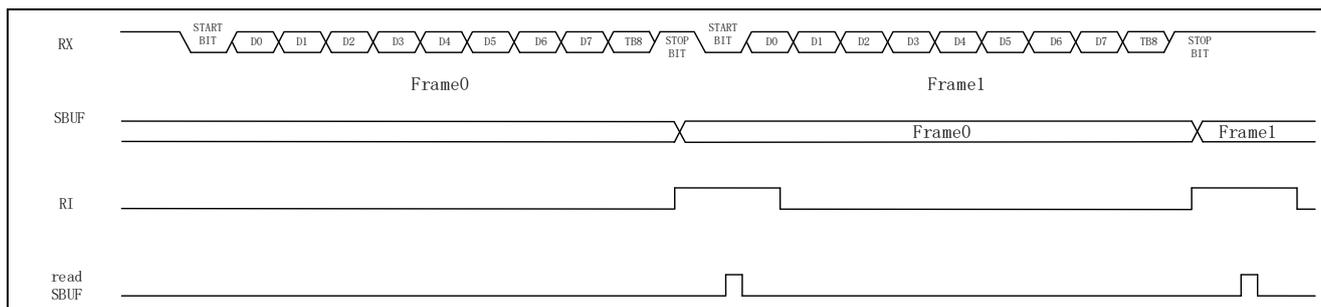


图 16-10 接收缓存

16.8.2 发送缓存

LPUART 发送端有一个帧长度（8/9bits）的发送缓存，当发送移位寄存器在发送当前帧数据时，CPU 可以向发送缓存写入下一帧需要发送的数据。

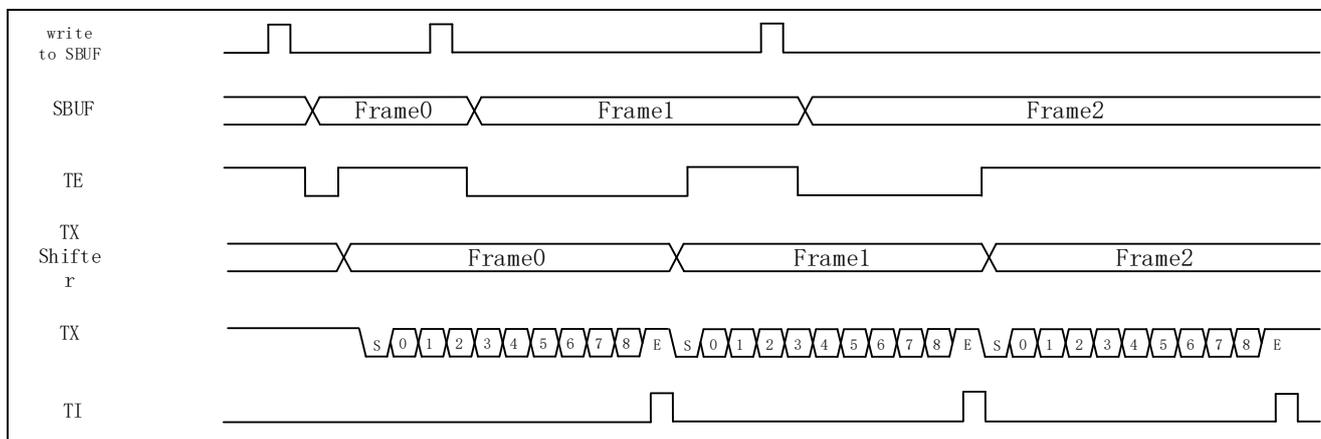


图 16-11 发送缓存

其中，寄存器位 LPUART_ISR.TE 是发送缓存空满标志位。LPUART 模块只包含一帧（8/9bits）发送缓存，因此当 LPUART_ISR.TE 位为“1”期间，硬件会自动屏蔽软件对 LPUART_SBUF 寄存器的写操作，直到 LPUART_ISR.TE 位变“0”为止。软件在将发送数据填入 LPUART_SBUF 寄存器前，必须判断 LPUART_ISR.TE 位“0”“1”状态，否则发送数据将会发生丢失。

16.9 寄存器

LPUART 基地址： 0x4000 0200

寄存器名	偏移地址	描述
LPUART_SBUF	0x00	数据寄存器
LPUART_SCON	0x04	控制寄存器
LPUART_SADDR	0x08	地址寄存器
LPUART_SADEN	0x0C	地址掩码寄存器
LPUART_ISR	0x10	中断标志位寄存器
LPUART_ICR	0x14	中断标志位清除寄存器

16.9.1 数据寄存器（LPUART_SBUF）

偏移地址： 0x00

复位值： 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							

位	标记	功能描述
31:8	Reserved	
7:0	SBUF	发送数据时，当发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读出。 注意，对该寄存器读的值实际是 RXBuffer 中的值，对该寄存器写的值实际是写到了 TXShifter 中。

16.9.2 控制寄存器 (LPUART_SCON)

偏移地址: 0x04

复位值: 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	DBA	TEEN	SM01		SM2	REN	TB8	RB8	TIEN	RIEN	
				ODE	UD										
RW		RW					RW		RW	RW	RW	RW	RW	RW	

位	标记	功能描述
31:16	Reserved	
15:13	PRS	传输时钟 SCLK 预分频选择; 000: div128; 001: div64; 010: div32; . . . ; 110: div2; 111: div1 PRS[2:0] 只有当 LPMODE=1 时有效; 当 LPMODE=0 时, PRS[2:0] 不会对 SCLK 预分频。
12:11	SCLKSEL	传输时钟 SCLK 选择; 00: PCLK; 01: PCLK; 10: XTL; 11: RCL
10	LPMODE	低功耗模式; 0: 正常工作模式; 1: 低功耗工作模式
9	DBAUD	双倍波特率; 0: 单倍波特率; 1: 双倍波特率
8	TEEN	发送缓存空中断使能; 0: disable; 1: enable
7:6	SM01	工作模式; 00: mode0; 01: mode1; 10: mode2; 11: mode3
5	SM2	多主机通讯; 0: disable, 1: enable

4	REN	接收使能; mode0: 0: 发送, 1: 接收 其他: 0: 发送, 1: 接收/发送
3	TB8	发送 TB8 位
2	RB8	接收 RB8 位
1	TIEN	发送完成中断使能; 0: disable; 1: enable
0	RIEN	接收完成中断使能; 0: disable; 1: enable

16.9.3 地址寄存器 (LPUART_SADDR)

偏移地址: 0x08

复位值: 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							

位	标记	功能描述
31:8	Reserved	
7:0	SADDR	从机设备地址寄存器

16.9.4 地址掩码寄存器 (LPUART_SADEN)

偏移地址: 0x0C

复位值: 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							

位	符号	功能描述
31:8	Reserved	
7:0	SADEN	从机设备地址掩码寄存器

16.9.5 中断标志位寄存器 (LPUART_ISR)

偏移地址: 0x10

复位值: 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

位	标记	功能描述
31:4	Reserved	
3	TE	发送缓存空中断标志位，硬件置位，硬件清零。 注意：当该位值为“0”时，硬件自动屏蔽软件写SBUF操作。 1:TE中断有效 0:TE中断无效
2	FE	接收帧错误标志位，硬件置位，软件清零 1:FE中断有效 0:FE中断无效
1	TI	发送完成中断标志位，硬件置位，软件清零 1:TI中断有效 0:TI中断无效
0	RI	接收完成中断标志位，硬件置位，软件清零 1:RI中断有效 0:RI中断无效

16.9.6 中断标志位清除寄存器 (LPUART_ICR)

偏移地址: 0x14

复位值: 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

位	符号	描述
31:3	Reserved	
2	FECLR	清除接收帧错误标志位; 写0清零, 写1无效
1	TICLR	清除发送完成中断标志位; 写0清零, 写1无效
0	RICLR	清除接收完成中断标志位; 写0清零, 写1无效

17 I2C 总线 (I2C)

17.1 简介

I2C 是双线双向的同步串行总线，它利用一根时钟线和一根数据线在连接总线的两个器件之间进行信息的传递，为设备之间数据交换提供了一种简单高效的方法。每个连接到总线上的器件都有唯一的地址，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器 (I2C_STAT)能反映 I2C 总线控制器和 I2C 总线的状态。

17.2 主要特性

I2C 控制器支持以下特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps) / 快速(400Kbps) / 高速(1Mbps) 三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

17.3 协议描述

I2C 总线使用连接设备的"SCL"(串行时钟总线)和"SDA"(串行数据总线)来传送信息。主机在 SCL 线上输出串行时钟信号，数据在 SDA 线上进行传输，每传输一个字节（最高位 MSB 开始传输），后面跟随一个应答位。一个 SCL 时钟脉冲传输一个数据位。

17.3.1 I2C 总线上数据传输

通常标准 I2C 传输协议包含四个部分：起始(S)或重复起始信号(Sr)，从机地址及读写位，传输数据，停止信号(P)。

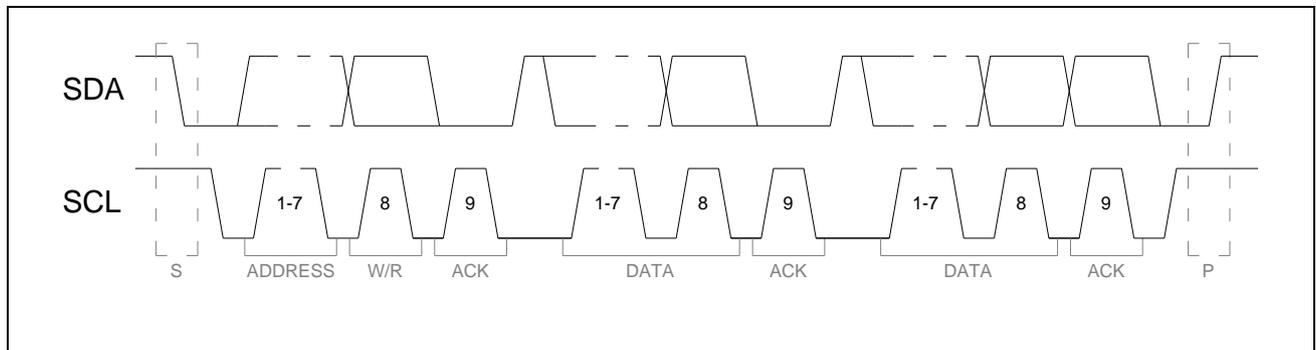


图 17-1 I2C 传输协议

- 起始信号、重复起始信号、停止信号

当总线处于空闲状态下（SCL 和 SDA 线同时为高），SDA 线上出现由高到低的信号，表明总线上产生了起始信号。

当两个起始信号之间没有停止信号时，即产生了重复起始信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信（例如：从写入设备到从设备读出）而不释放总线。

当 SCL 线为高时，SDA 线上出现由低到高的信号，被定义为停止信号。主机向总线发出停止信号结束数据传送。

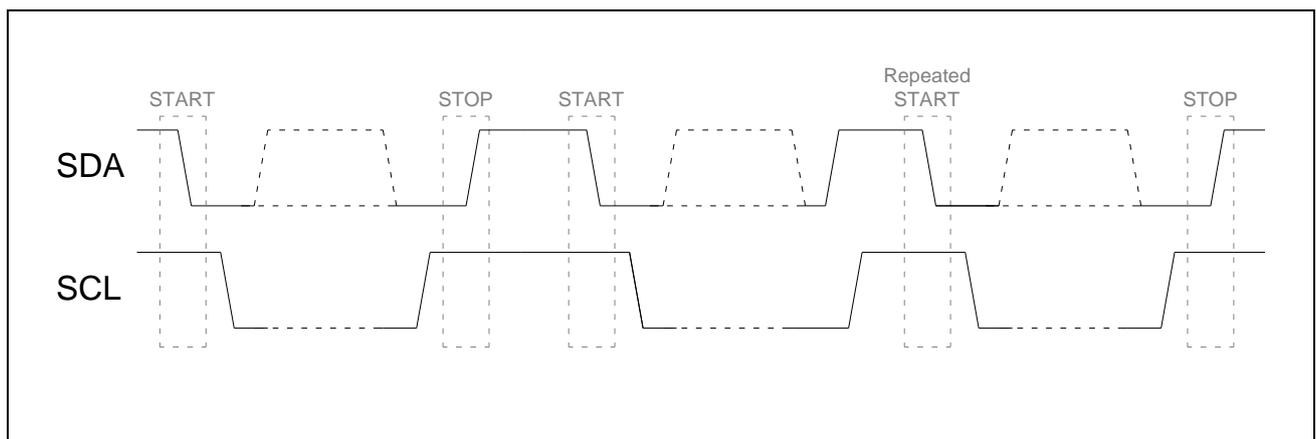


图 17-2 START 和 STOP 条件

- 从机地址及读写位

当起始信号产生后，主机立即传输数据的第一字节：7 位从机地址 + 读写位，读写位控制从机的数据传输方向（0：写；1：读）。被主机寻址的从机会通过在第 9 个

SCL 时钟周期将 SDA 置为低电平作为应答。

- 传输数据

数据传输过程中，一个 SCL 时钟脉冲传输一个数据位，且 SDA 线只有在 SCL 为低时才可以改变。

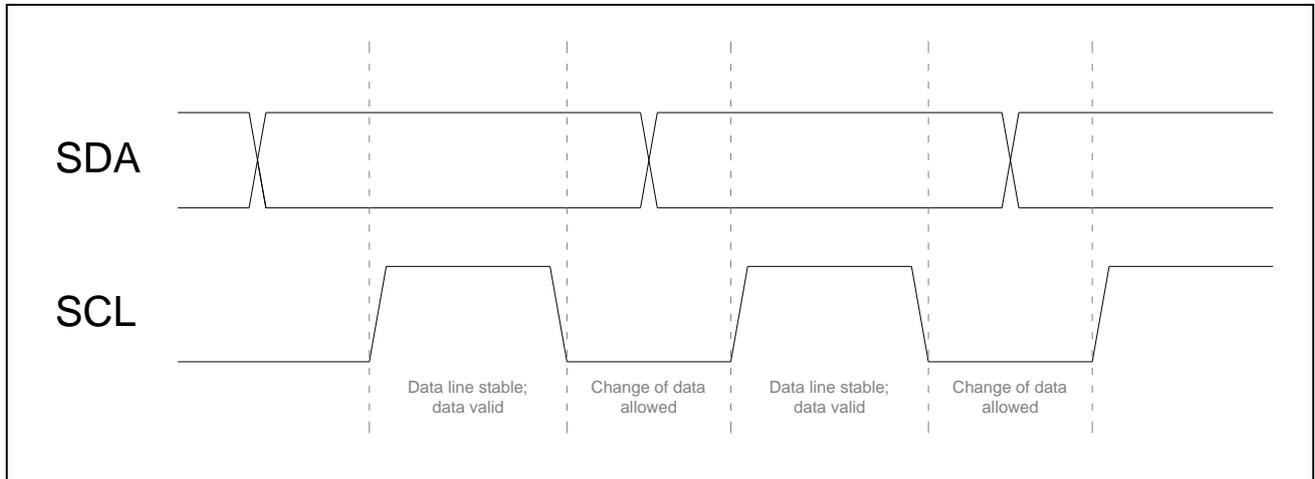


图 17-3 I2C 总线上位传输

17.3.2 I2C 总线上的应答

每传输一个字节，后面跟随一个应答位。通过将 SDA 线拉低，来允许接收端回应发送端。ACK 为一个低电平信号，当时钟信号为高时，SDA 保持低电平则表明接收端已成功接收到发送端的数据。

当主机作为发送器件时，如果从机上产生无响应信号(NACK)，主机可以产生停止信号来退出数据传输，或者产生重复起始信号开始新一轮的数据传输。当主机作为接收器件时，发生无响应信号(NACK)，从机释放 SDA 线，使主机产生停止信号或重复起始信号。

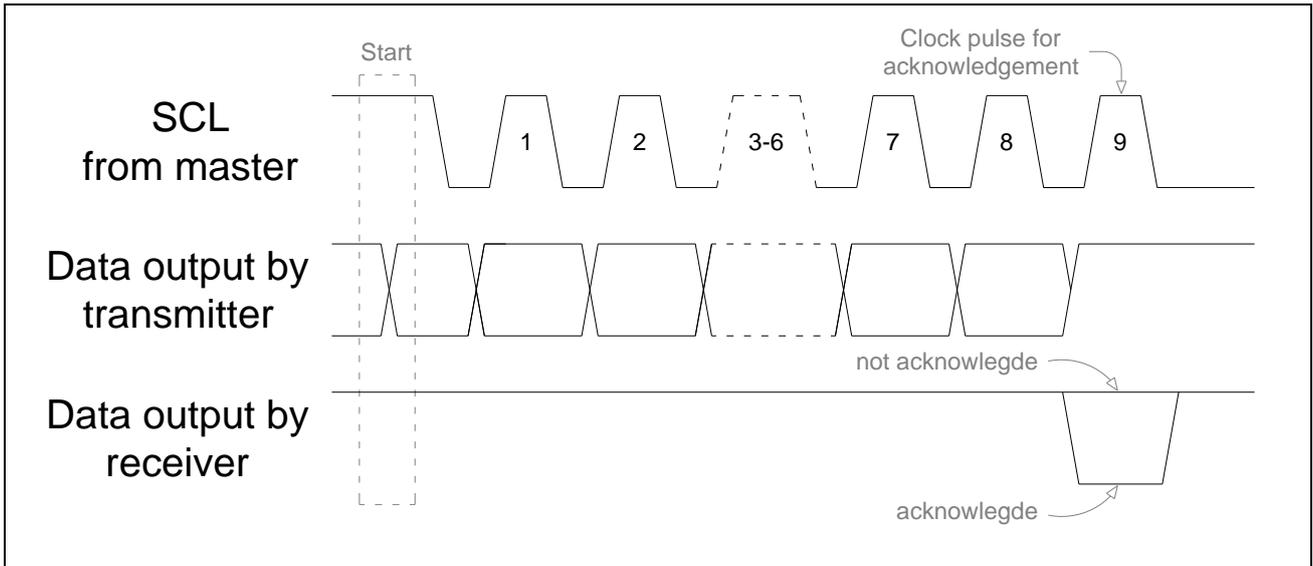


图 17-4 I2C 总线上应答信号

17.3.3 I2C 总线上的仲裁

I2C 总线上的仲裁分为两个部分：SCL 线上的同步和 SDA 线上的仲裁

- SCL 线上的同步（时钟同步）

由于 I2C 总线具有线“与”的逻辑功能，SCL 线上只要有一个节点发送低电平，总线上就表现低电平。当所有的节点都发送高电平时，总线才能表现为高电平。所以，时钟低电平的时间由时钟电平期最长的器件决定，而时钟的高电平时间由时钟高电平期最短的器件决定。由于 I2C 这种特性，当多个主机同时发送时钟信号时，在总线上表示的是统一的时钟信号。如果从机希望主机降低传送速度可以通过将 SCL 主动拉低延长其低电平时间来通知主机，当主机在准备下一次传送时发现 SCL 的电平被拉低时进行等待，直到从机完成操作并释放 SCL 线的控制权。

- SDA 线上的仲裁

SDA 线上的仲裁也是由于 I2C 总线具有线“与”的逻辑功能。主机在发送数据后，通过比较总线上的数据来决定是否退出竞争。丢失仲裁的主机立即切换到未被寻址的从机状态，以确保自身能被仲裁胜利的主机寻址到。仲裁失败的主机继续输出时钟脉冲（在 SCL 上），直到发送完当前的串行字节。通过这种原理可以保证 I2C 总线在多个主机企图控制总线时保证数据的不丢失。

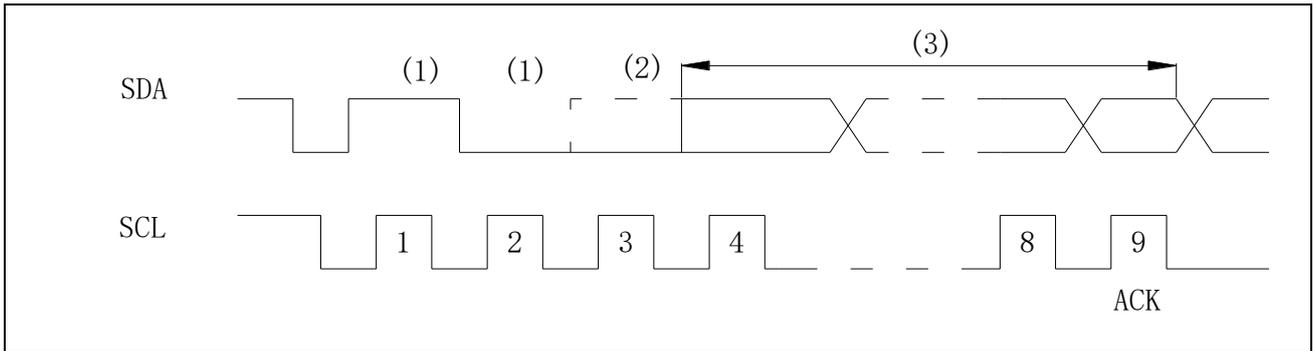


图 17-5 I2C 总线上的仲裁

- a) 另一器件发送串行数据；
- b) 另一器件通过拉低 SDA 先撤消了该 I2C 主机发送的一个逻辑 1（虚线）。仲裁丢失，I2C 进入从接收模式；
- c) 此时 I2C 处于从接收模式，但仍产生时钟脉冲，直至发送完当前字节。I2C 将不为下个字节的传输产生时钟脉冲。一旦赢得仲裁，SDA 上的数据传输由新的主机来启动。

17.4 功能描述

I2C 总线使用双线在连接到总线"SCL"（串行时钟线）和"SDA"（串行数据线）的设备间传送信息。滤波逻辑可以过滤数据总线上的毛刺来保护数据的完整性。由于只有无方向端口，I2C 组件需要使用到引脚的漏端开路缓冲器。每个连接到总线的设备都能使用软件通过特定地址寻址。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时开始传输数据时发生数据冲突。I2C 总线状态能在状态寄存器中进行查询。

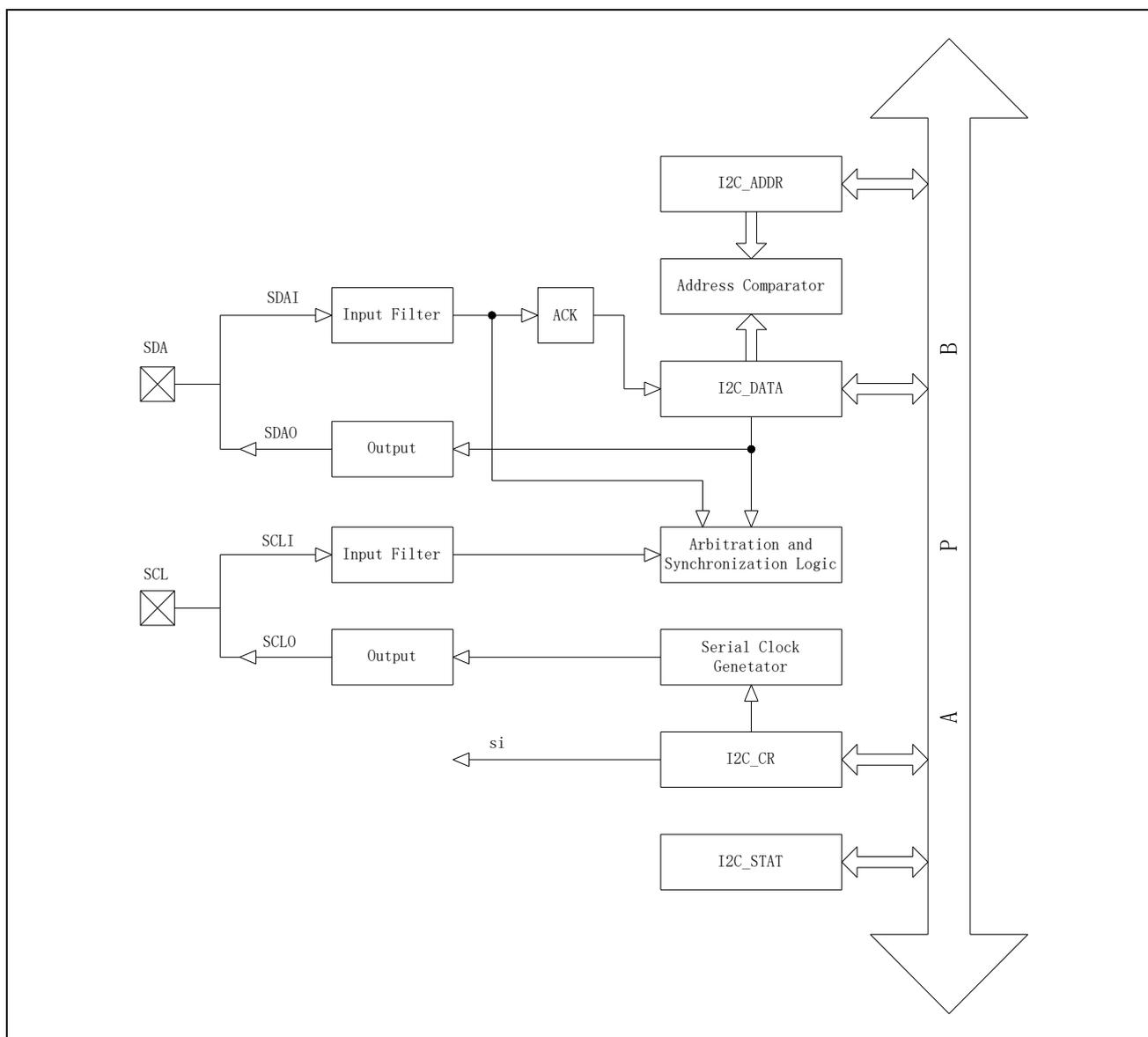


图 17-6 I2C 功能模块图

17.4.1 串行时钟发生器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 PCLK 信号的频率关系为 $F_{scl} = F_{pclk} / 8 / (I2C_TM.tn+1)$ ，其中 I2C_TM.tn 应大于 0。

下表列出了 PCLK 频率与 I2C_TM.tn 组合时，SCL 信号的输出频率值。

PCLK (KHz)	I2C_TM.tn						
	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

表 17-1 I2C 时钟信号波特率

17.4.2 输入滤波器

输入信号与 PCLK 同步，低于 PCLK 周期的尖峰脉冲信号会被滤除。

当本模块作为主机时，如果 I2C_TM 的值小于等于 9，应设置 I2C_CR.H1M 为 1；如果 I2C_TM 的值大于 9，应设置 I2C_CR.H1M 为 0。

当本模块作为从机时，如果 PCLK 与 SCL 频率比值小于等于 30 时，应设置 I2C_CR.H1M 为 1；如果 PCLK 与 SCL 频率比值大于 30 时，应设置 I2C_CR.H1M 为 0。

17.4.3 地址比较器

I2C 比较器将自己的从机地址与接收到的 7 位从机地址做比较。它可使用 "I2C_ADDR" 寄存器对自己的从机地址进行编程。并且会根据 "I2C_ADDR" 寄存器的 "i2cadr" 位与首次接收到的 8 位字节或广播地址 (0x00) 相比较。如果任何一者相同，"I2C_CR" 寄存

器的"si"位会被置 1 并产生一个中断请求。

17.4.4 应答标志位

"I2C_CR"寄存器的"aa"标志位为应答标志位。当"aa"位为 1 时，I2C 模块收到数据后回应答位，当"aa"位为 0 时，I2C 模块收到数据后回非应答位。

17.4.5 中断产生器

"I2C_CR"寄存器的"si"标志位为中断标志位。每当状态寄存器（I2C_STAT）的值产生变化时（除变为 0xF8 外），"si"标志位都会被置 1。当产生中断时，通过查询状态寄存器（I2C_STAT）可获知 I2C 总线的状态，以确定中断的实际来源。为了进行下一步的操作，"si"标志位必须通过软件清零。

17.4.6 工作模式

I2C 组件可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100Kbps 而在高速模式下可达 400Kbps，在超高速模式下可达 1Mbps，并且可以在四种模式下工作：主机发送模式、主机接收模式、从机接收模式、从机发送模式。还有一种特殊模式广播呼叫模式，其操作方式与从机接收模式类似。

- 主机发送模式

主机发送多个字节到从机，主机产生时钟，所以需要在 I2C_TM 中填入设定值。主机发送模式中需要将 I2C_CR.sta 置 1。当总线空闲时，主机发起一个起始位 START。若成功 I2C_CR.si 被置 1。接下来把从机地址和写位 (SLA+W) 写入 I2C_DATA 中，清零"si"位后，总线上发出 SLA+W。

主机发出 SLA+W 收到从机应答位 ACK 后，"si"被置 1。接下来根据用户定义格式发送数据。所有数据发送完后，将 I2C_CR.sto 置 1，清零"si"位后发出 STOP 信号，也可以发送重复起始信号进行新一轮数据传输。

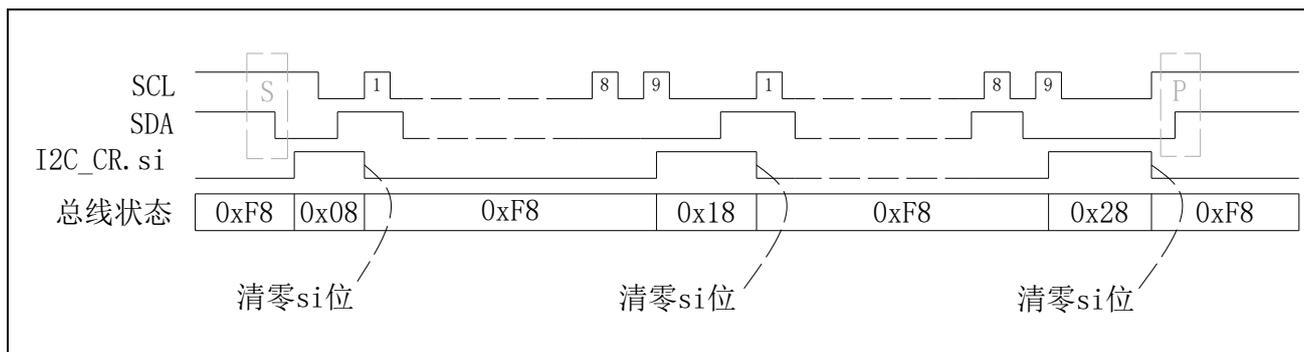


图 17-7 主发送模式数据同步图

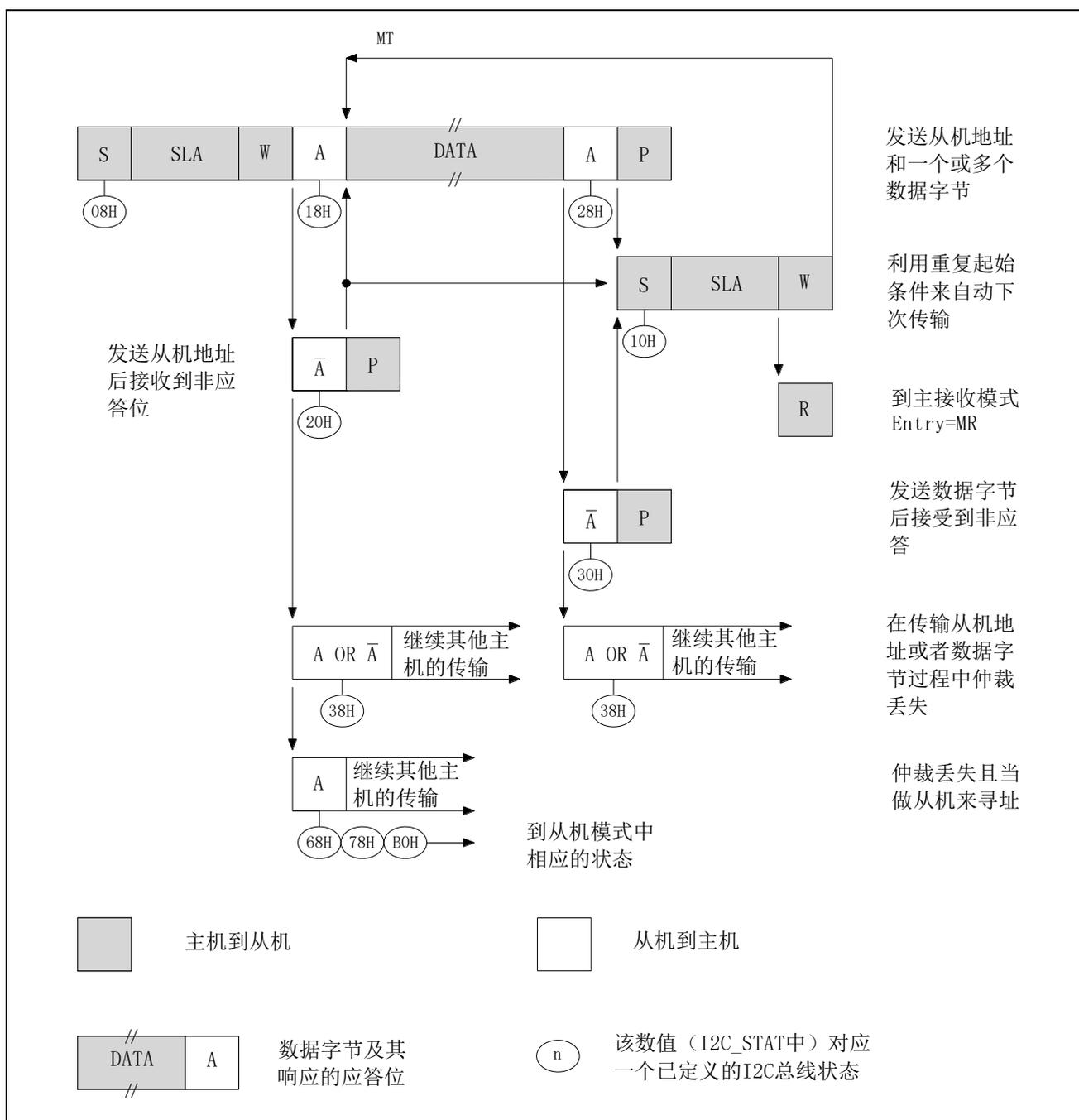


图 17-8 I2C 主机发送状态图

- 主机接收模式

主机接收模式，由从机传输数据。初始化设置与主机发送模式相同，主机发送起始位以后， I2C_DATA 应写入从机地址和“读位” (SLA+R)。收到从机应答位 ACK 后 I2C_CR.si 被置 1。"si"清 0 后开始接收从机数据，若 I2C_CR.aa 为 1，主机收到数据后回应答位；若为 0 主机收到数据后不回应答 NACK。然后主机可以发停止信号或重复起始信号开始下一轮的数据传输。

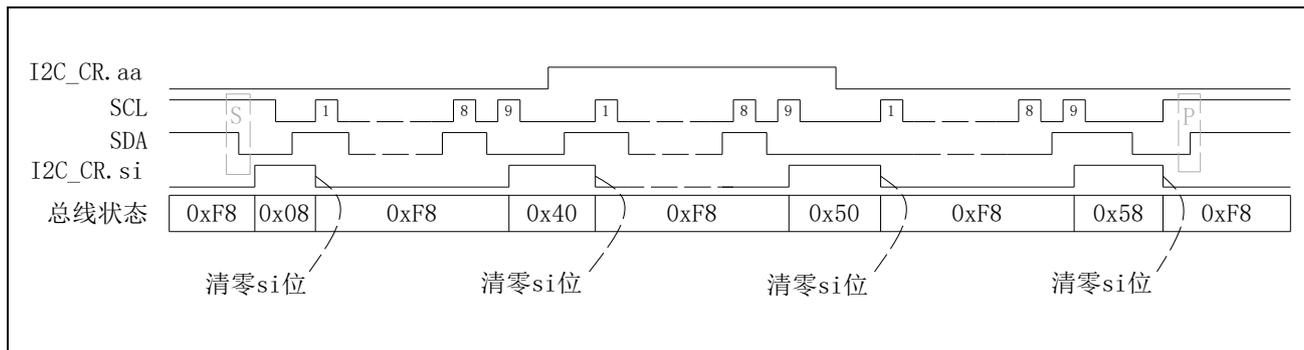


图 17-9 主接收模式数据同步图

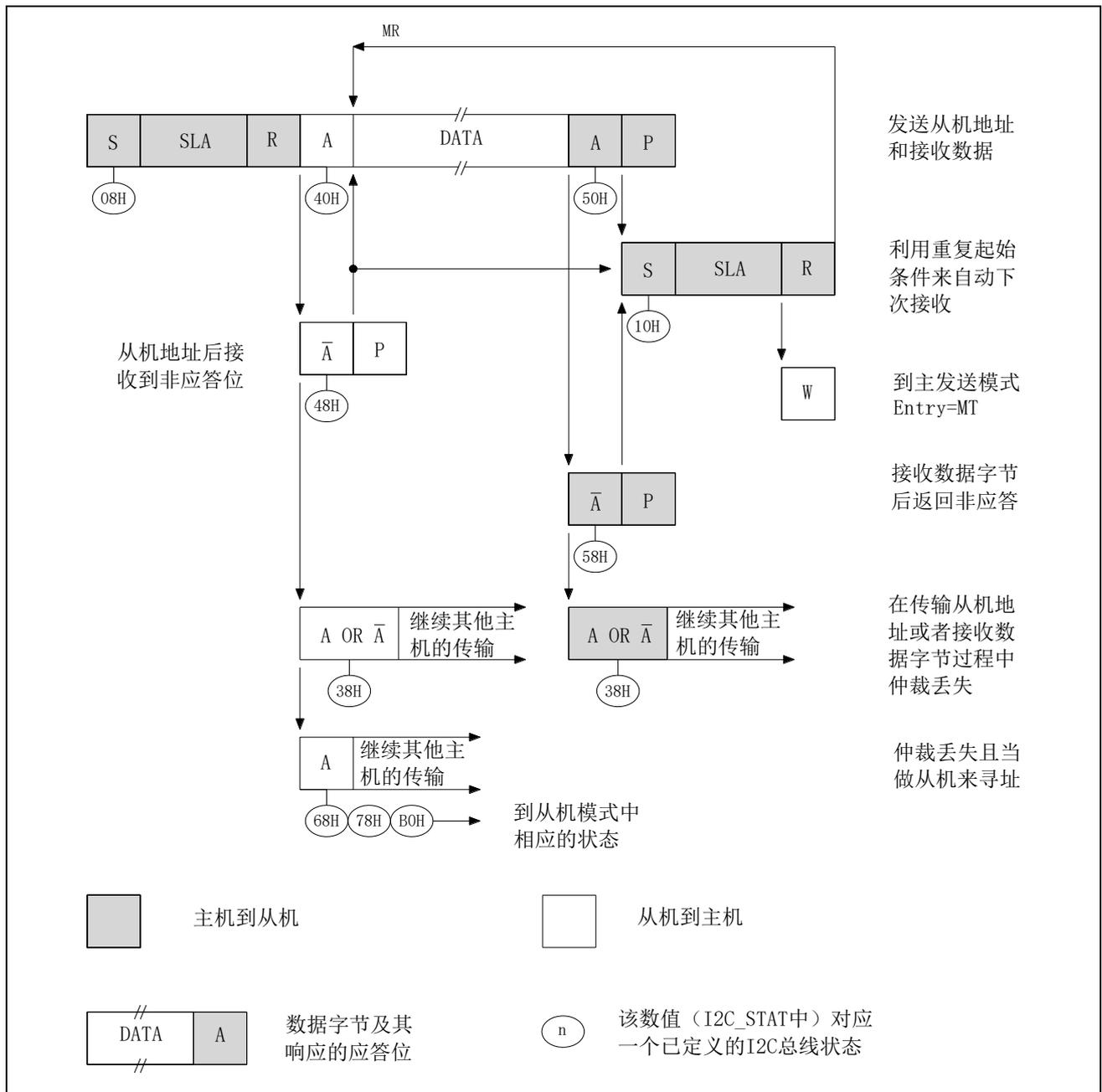


图 17-10 I2C 主机接收状态图

• 从机接收模式

在从机接收模式下，从机接收主机发来的数据。在传输开始前，I2C_ADDR 应写入从机地址，I2C_CR.aa 置 1 用以响应主机的寻址。上述初始化后，从机进入空闲模式，等待“写”信号（SLA+W）。若主机仲裁失败，也会直接进入从机接收模式。当从机被“写”信号 SLA+W 寻址到后，需要清零"si"位，以便从主机接收数据。如果在传输过程中 I2C_CR.aa=0，从机将在下一字节返回无应答位 NACK，从机也将转为未寻址从机，与主机联系终止，不再接收数据，且 I2C_DATA 保持之前接收到的数据。从机地址识别可通过置位"aa"来恢复，这意味着"aa"位可临时将 I2C 模块从 I2C 总线上分离出来。

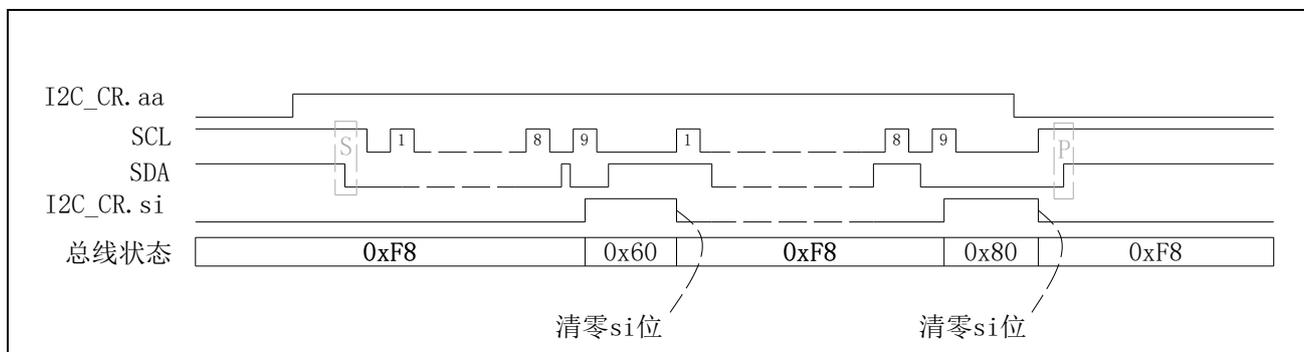


图 17-11 从接收模式数据同步图

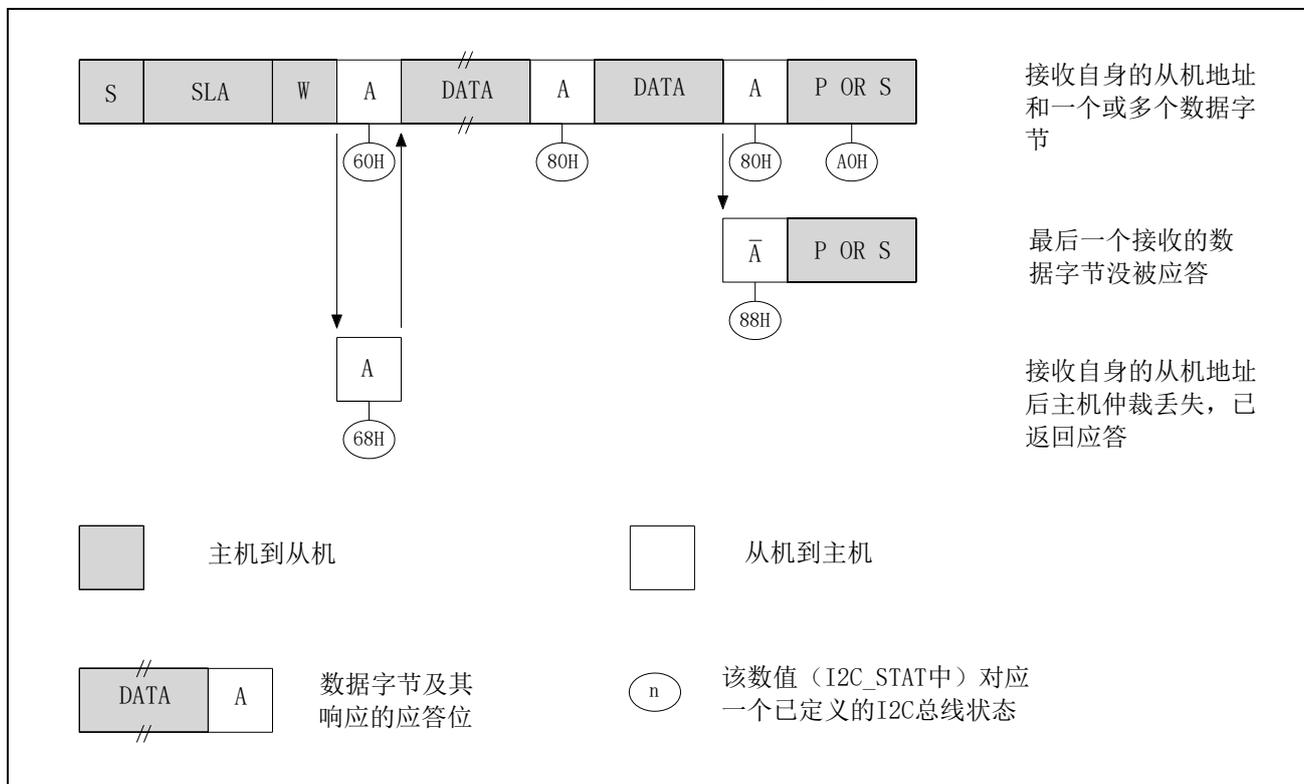


图 17-12 从机接收状态图

• 从机发送模式

从机发送模式，数据由从机发送给主机。当初始化 I2C_ADDR 及 I2C_CR.aa 值后，器件等待直到自身地址被“读”信号(SLA+R)寻址。若主机仲裁失败，也可进入从机发送模式。

当从机被“读”信号 SLA+R 寻址，需要将"si"清零用以向主机发送数据。通常主机接收每字节数据后会返回应答位。

如果传输过程中 I2C_CR.aa 清零，从机将发送最后一个字节数据，并在接下去的传输中发送全 1 数据，并将自身变为未寻址从机。

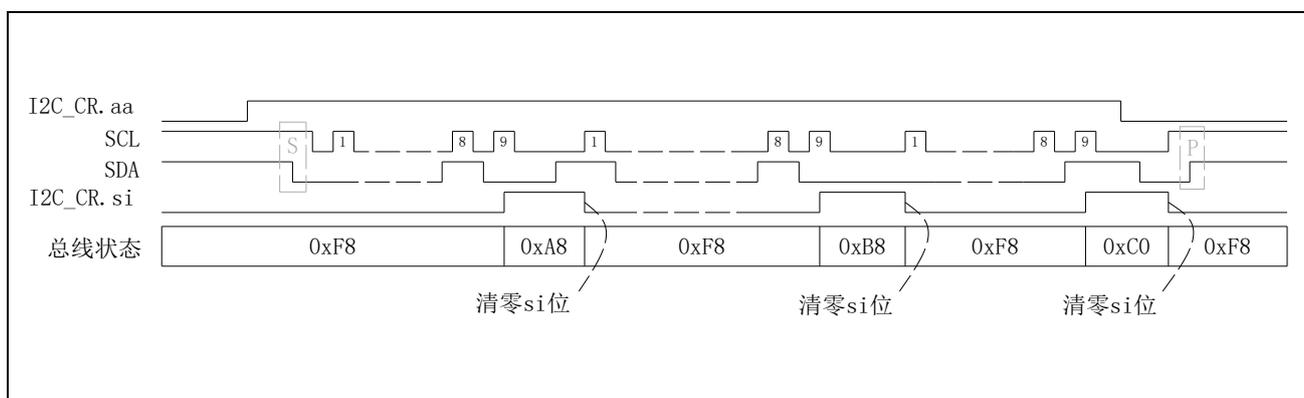


图 17-13 从发送模式数据同步图

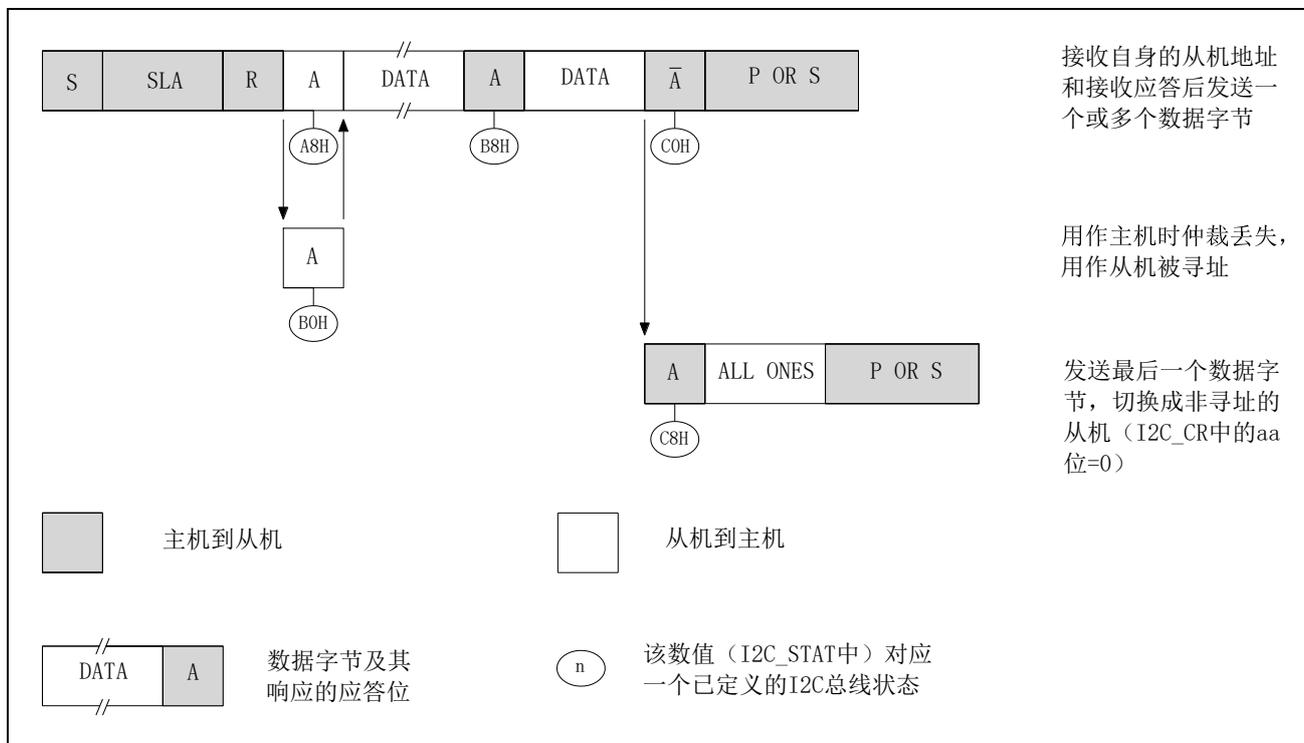


图 17-14 I2C 从机发送状态图

• 广播呼叫模式

广播呼叫模式是一种特殊的从机接收模式，寻址方式为 0x00，从机地址和读写都为 0。当 I2C_ADDR.GC 及 I2C_CR.aa 都为置 1，使能接收广播呼叫模式。在该模式下 I2C_STAT 值与普通从机接收模式 I2C_STAT 值不同。仲裁失败也可能进入广播呼叫模式。

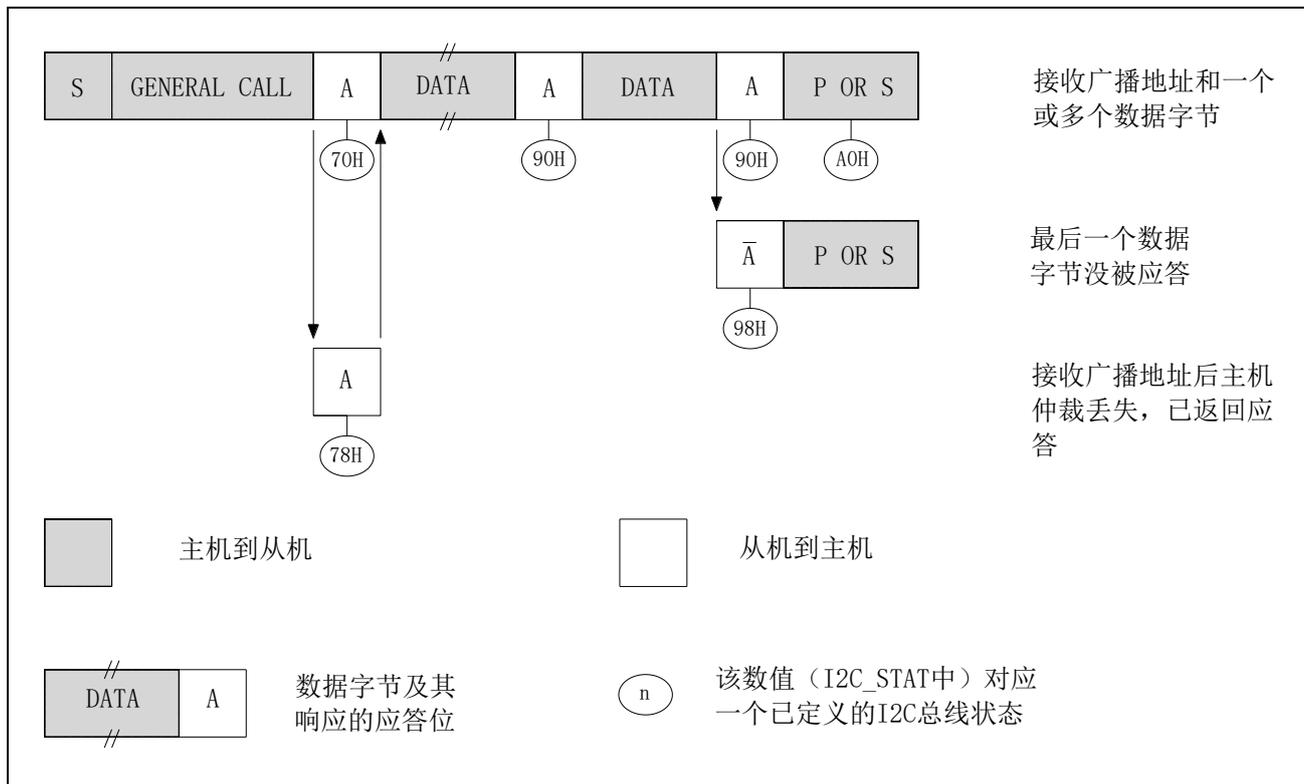


图 17-15 I2C 广播呼叫状态图

17.4.7 状态码表述

I2C 状态寄存器中有两种特殊的状态：F8H 和 00H。

F8H: 这个状态码表示没有任何可用的相关信息，因为串行中断标志"si"还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

00H: 该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时"si"置位。

状态代码	描述
主发送模式	
08H	已发送起始条件
10H	已发送重复起始条件
18H	已发送 SLA+W，已接收 ACK
20H	已发送 SLA+W，已接收非 ACK
28H	已发送 I2C_DATA 中的数据，已接收 ACK
30H	已发送 I2C_DATA 中的数据，已接收非 ACK
38H	在 SLA+读写或写数据字节时丢失仲裁
主接收模式	
08H	已发送起始条件
10H	已发送重复起始条件
38H	在非 ACK 中丢失仲裁
40H	已发送 SLA+R，已接收 ACK
48H	已发送 SLA+R，已接收非 ACK
50H	已接收数据字节，ACK 已返回
58H	已接收数据字节，非 ACK 已返回
从接收模式	
60H	已接收自身的 SLA+W，已返回 ACK

68H	主控时在 SLA+读写丢失仲裁, 已接收自身的 SLA+W, 已返回 ACK
80H	前一次寻址使用自身从地址, 已接收数据字节, 已返回 ACK
88H	前一次寻址使用自身从地址, 已接收数据字节, 已返回非 ACK
A0H	静态寻址时, 接收到停止条件或重复起始条件
从发送模式	
A8H	已接收自身的 SLA+R, 已返回 ACK
B0H	当主机时丢失仲裁, 已接收自身 SLA+R, 已返回 ACK
B8H	已发送数据, 已接收 ACK
C0H	已发送数据字节, 已接收非 ACK
C8H	装入的数据字节已被发送, 已接收 ACK
广播呼叫模式	
70H	已接收广播地址 (0x00), 已返回 ACK
78H	主控时在 SLA+读写丢失仲裁, 已接收广播地址, 已返回 ACK
90H	前一次寻址使用广播地址, 已接收数据字节, 已返回 ACK
98H	前一次寻址使用广播地址, 已接收数据字节, 已返回非 ACK
A0H	静态寻址时, 接收到停止条件或重复起始条件
其余杂项状态	
F8H	无可用的相关状态信息, si=0
00H	传输过程出现总线错误, 或外部干扰使 I2C 进入未定义的状态

表 17-2 I2C 状态码表述

17.5 编程示例

17.5.1 主机发送示例

Step1: 设置 PERI_CLKEN.I2C 为 1, 使能 I2C 模块时钟。

Step2: 向 PERI_RESET.I2C 依次写入 0、1, 复位 I2C 模块。

Step3: 将 I2C 的 SCL、SDA 使用的端口配置为开漏模式, 根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出, 在 DIR 相应位清零之前, 需要先初始化相应的输出寄存器为 1), 最后将 SCL、SDA 映射到相应的管脚。

Step4: 配置 I2C_TM, 使 SCL 的时钟速率符合应用需求。

Step5: 设置 I2C_TMRUN 为 1, 使能 SCL 时钟发生器。

Step6: 设置 I2C_CR.ens 为 1, 使能 I2C 模块。

Step7: 设置 I2C_CR.sta 为 1, 总线尝试发送 Start 信号。

Step8: 等待 I2C_CR.si 变为 1, Start 信号已发送到总线上。

Step9: 查询 I2C_STAT, 如果该寄存器值为 0x08 或 0x10, 继续执行下一步骤, 否则进行出错处理。

Step10: 向 I2C_DATA 中写入 SLA+W, 设置 I2C_CR.sta 为 0, 设置 I2C_CR.si 为 0, 发送 SLA+W。

Step11: 等待 I2C_CR.si 变为 1, SLA+W 已发送到总线上。

Step12: 查询 I2C_STAT, 如果该寄存器值为 0x18, 继续执行下一步骤。否则进行出错处理。

Step13: 向 I2C_DATA 写入待发送的数据, 设置 I2C_CR.si 为 0, 发送数据。

Step14: 等待 I2C_CR.si 变为 1, 数据已发送到总线上。

Step15: 查询 I2C_STAT, 如果该寄存器值为 0x28, 继续执行下一步骤。否则进行出错处理。

Step16: 如待发送的数据未完成, 则跳转到 Step13 继续执行。

Step17: 设置 I2C_CR.sto 为 1, 设置 I2C_CR.si 为 0, 总线尝试发送 Stop 信号。

Step18: 等待 I2C_CR.si 变为 1, Stop 信号已发送到总线上。

17.5.2 主机接收示例

- Step1: 设置 PERI_CLKEN.I2C 为 1, 使能 I2C 模块时钟。
- Step2: 向 PERI_RESET.I2C 依次写入 0、1, 复位 I2C 模块。
- Step3: 将 I2C 的 SCL、SDA 使用的端口配置为开漏模式, 根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出, 在 DIR 相应位清零之前, 需要先初始化相应的输出寄存器为 1), 最后将 SCL、SDA 映射到相应的管脚。
- Step4: 配置 I2C_TM, 使 SCL 的时钟速率符合应用需求。
- Step5: 设置 I2C_TMRUN 为 1, 使能 SCL 时钟发生器。
- Step6: 设置 I2C_CR.ens 为 1, 使能 I2C 模块。
- Step7: 设置 I2C_CR.sta 为 1, 总线尝试发送 Start 信号。
- Step8: 等待 I2C_CR.si 变为 1, Start 信号已发送到总线上。
- Step9: 查询 I2C_STAT, 如果寄存器值为 0x08 或 0x10, 继续执行下一步骤, 否则进行出错处理。
- Step10: 向 I2C_DATA 写入 SLA+R, 设置 I2C_CR.sta 为 0, 设置 I2C_CR.si 为 0, 发送 SLA+R。
- Step11: 等待 I2C_CR.si 变为 1, SLA+R 已发送到总线上。
- Step12: 查询 I2C_STAT, 如果寄存器值为 0x40, 继续执行下一步骤, 否则进行出错处理。
- Step13: 设置 I2C_CR.aa 为 1, 使能应答标志。
- Step14: 设置 I2C_CR.si 为 0, 从机发送数据, 主机根据 I2C_CR.aa 发送 ACK 或 NACK。
- Step15: 等待 I2C_CR.si 变为 1, 从 I2C_DATA 读取已接收到的数据。
- Step16: 查询到 I2C_STAT, 如果该寄存器值为 0x50 或 0x58, 继续执行下一步骤, 否则进行出错处理。
- Step17: 如果待接收的数据只差最后一个字节, 设置 I2C_CR.aa 为 0, 使能非应答标志。
- Step18: 如待接收的数据未完成, 则跳转到 Step14 继续执行。
- Step19: 设置 I2C_CR.sto 为 1, 设置 I2C_CR.si 为 0, 总线尝试发送 Stop 信号。
- Step20: 等待 I2C_CR.si 变为 1, Stop 信号已发送到总线上。

17.5.3 从机接收示例

Step1: 设置 PERI_CLKEN.I2C 为 1, 使能 I2C 模块时钟。

Step2: 向 PERI_RESET.I2C 依次写入 0、1, 复位 I2C 模块。

Step3: 将 I2C 的 SCL、SDA 使用的端口配置为开漏模式, 根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出, 在 DIR 相应位清零之前, 需要先初始化相应的输出寄存器为 1), 最后将 SCL、SDA 映射到相应的管脚。

Step4: 设置 I2C_CR.ens 为 1, 使能 I2C 模块。

Step5: 配置 I2C_ADDR 为从机地址。

Step6: 设置 I2C_CR.aa 为 1, 使能应答标志。

Step7: 等待 I2C_CR.si 变为 1, 被 SLA+W 寻址。

Step8: 查询 I2C_STAT, 如果该寄存器值为 0x60, 继续执行下一步骤, 否则进行出错处理。

Step9: 设置 I2C_CR.si 为 0, 主机发送数据, 从机根据 I2C_CR.aa 返回 ACK 或 NACK。

Step10: 等待 I2C_CR.si 变为 1, 从 I2C_DATA 中读取已接收到的数据。

Step11: 查询 I2C_STAT, 如果该寄存器值为 0x80, 继续执行下一步骤, 否则进行出错处理

Step12: 如待接收的数据未完成, 则跳转到 Step9 继续执行。

Step13: 设置 I2C_CR.aa 为 0, 设置 I2C_CR.si 为 0。

17.5.4 从机发送示例

Step1: 设置 PERI_CLKEN.I2C 为 1, 使能 I2C 模块时钟。

Step2: 向 PERI_RESET.I2C 依次写入 0、1, 复位 I2C 模块。

Step3: 将 I2C 的 SCL、SDA 使用的端口配置为开漏模式, 根据 I2C 的模式需要配置合适的端口方向(如配置端口方向为输出, 在 DIR 相应位清零之前, 需要先初始化相应的输出寄存器为 1), 最后将 SCL、SDA 映射到相应的管脚。

Step4: 设置 I2C_CR.ens 为 1, 使能 I2C 模块。

Step5: 配置 I2C_ADDR 为从机地址。

Step6: 设置 I2C_CR.aa 为 1, 使能应答标志。

Step7: 等待 I2C_CR.si 变为 1, 被 SLA+R 寻址。

Step8: 查询 I2C_STAT, 如果该寄存器的值为 0xA8, 继续执行下一步骤, 否则进行出错处理。

Step9: 向 I2C_DATA 写入待发送的数据, 设置 I2C_CR.si 为 0, 发送数据。

Step10: 等待 I2C_CR.si 变为 1, 数据已发送到总线上。

Step11: 查询 I2C_STAT, 如果该寄存器的值为 0xB8 或 0xC0 时, 继续执行下一步骤, 否则进行出错处理。

Step12: 如待发送的数据未完成, 则跳转到 Step9 继续执行。

Step13: 设置 I2C_CR.aa 为 0, 设置 I2C_CR.si 为 0。

17.6 寄存器描述

寄存器列表

I2C 基地址: 0x40000400

偏移量	寄存器名称	访问	寄存器描述
0x00	I2C_TMRUN	RW	I2C 波特率计数器使能寄存器.
0x04	I2C_TM	RW	I2C 波特率计数器配置寄存器.
0x08	I2C_CR	RW	I2C 配置寄存器.
0x0c	I2C_DATA	RW	I2C 数据寄存器.
0x10	I2C_ADDR	RW	I2C 地址寄存器.
0x14	I2C_STAT	RO	I2C 状态寄存器.

表 17-3 寄存器列表

17.6.1 I2C 波特率计数器使能寄存器(I2C_TMRUN)

地址偏移量: 0x00

复位值: 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														tme	
Reserved														RW	

位	标记	功能描述
31: 1	Reserved	
0	tme	波特率计数器使能. 0 – 禁止 1 – 使能

17.6.2 I2C 波特率计数器配置寄存器(I2C_TM)

地址偏移量: 0x04

复位值: 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

位	标记	功能描述
31:8	Reserved	
7:0	tm	tm:波特率计数器配置值. $F_{scl} = F_{pclk} / 8 / (tm+1)$, 其中 $tm > 0$

17.6.3 I2C 配置寄存器(I2C_CR)

地址偏移量：0x08

复位值：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
Reserved										RW	RW	RW	RW	RW		RW

位	标记	功能描述
31:7	Reserved	
6	ens	I2C 模块使能控制 0 – 禁止 1 – 使能
5	sta	I2C 总线控制 0 – 无功能 1 – 向总线发送 START
4	sto	I2C 总线控制 0 – 无功能 1 – 向总线发送 STOP
3	si	I2C 中断标志 读出 1，已发生 I2C 中断 写入 0，I2C 进行下一步操作
2	aa	应答控制位 0 – 发送 NAK 1 – 发送 ACK
1	Reserved	
0	h1m	I2C 滤波参数配置

		0 – 高级滤波，更高的抗干扰性能 1 – 简单滤波，更快的通信速率 注：详见【输入滤波器】章节。
--	--	---

17.6.4 I2C 数据寄存器(I2C_DATA)

地址偏移量: 0x0c

复位值: 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								i2cdat							
								RW							

位	标记	功能描述
31:8	Reserved	
7:0	i2cdat	I2C 数据寄存器 在 I2C 发送模式下, 写入待发送的数据 在 I2C 接收模式下, 读出收到的数据

17.6.5 I2C 地址寄存器(I2C_ADDR)

地址偏移量: 0x10

复位值: 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

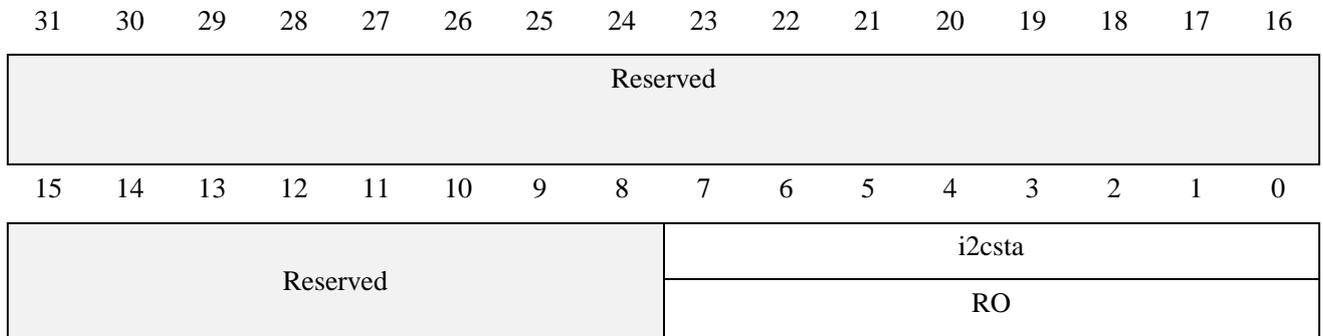
Reserved	i2cadr	GC
	RW	RW

位	标记	功能描述
31:8	Reserved	
7:1	i2cadr	I2C 从机模式地址.
0	GC	广播地址应答使能 0 – 禁止 1 – 使能

17.6.6 I2C 状态寄存器(I2C_STAT)

地址偏移量: 0x14

复位值: 0x00000000



位	标记	功能描述
31:8	Reserved	
7:0	i2csta	I2C 状态寄存器 状态值的具体定义详见【状态码表述】章节

18 串行外设接口（SPI）

18.1 SPI 简介

SPI 接口是工作于全双工模式下的同步串行数据通信接口，使用 4 个引脚进行通信：MISO、MOSI、SCK、CS/SSN。当 SPI 作为主机时，输出 CS 和 SCK 信号以控制通信过程。当 SPI 作为从机时，在 SSN 和 SCK 信号的控制下进行通信。

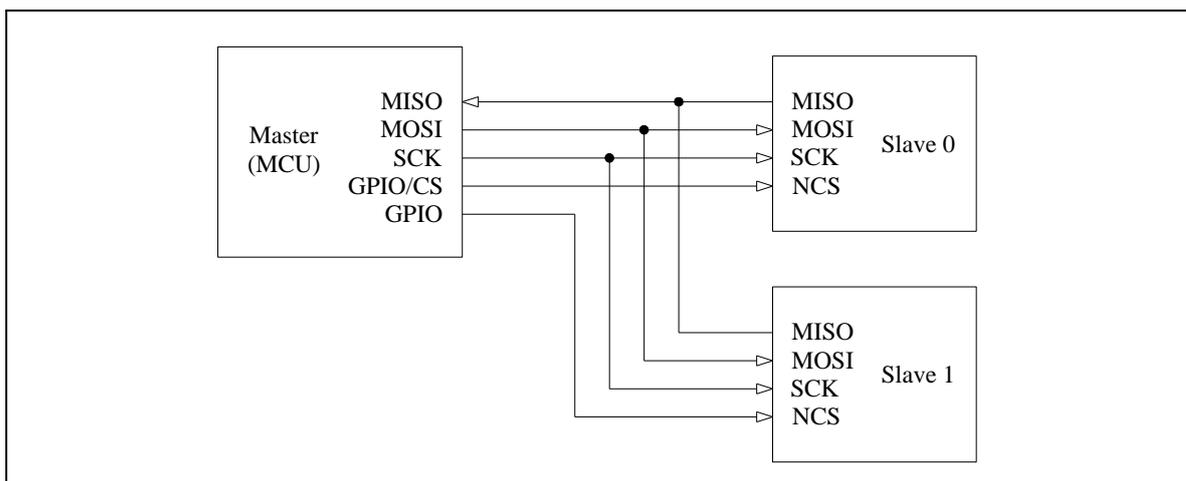
18.2 SPI 主要特性

- 支持 SPI 主机模式、SPI 从机模式
- 支持标准四线全双工通信
- 支持配置串行时钟极性和相位
- 主机模式支持 7 种通信速率
- 主机模式最大分频系数为 PCLK/2，最高通信速率为 16M bps
- 从机模式最大分频系数为 PCLK/4，最高通信速率为 12M bps
- 帧长度固定为 8 比特，优先传送 MSB

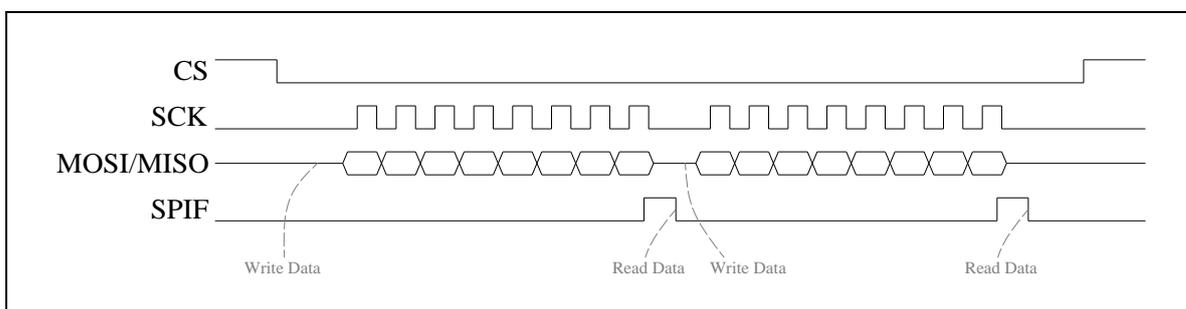
18.3 SPI 功能描述

18.3.1 SPI 主机模式

每个数据帧的长度固定为 8 比特，发送数据的第一位固定为 MSB。设置 SPI_CR.mstr 为 1，则 SPI 接口工作于主机模式。将数据写入 SPI_Data 寄存器，即可启动 SPI 传输。SCK 引脚会自动产生串行时钟；在串行时钟边沿，移位寄存器中的数据被发送到 MOSI 引脚，MISO 引脚的数据被接收到移位寄存器中。每完成一个数据帧的传送时，SPIF 会被硬件置 1，读取 SPI_DATA 可清除 SPIF 标志。SCK 引脚输出时钟的频率由 SPI_CR[spr2:spr0] 进行控制，其输出频率的范围为 $PCLK / 2 \sim PCLK / 128$ ；CS 引脚的输出电平由 SPI_SSN.ssn 进行控制，GPIO 引脚的输出电平由 GPIO 相关寄存器控制。主机模式的典型应用框图如下所示。

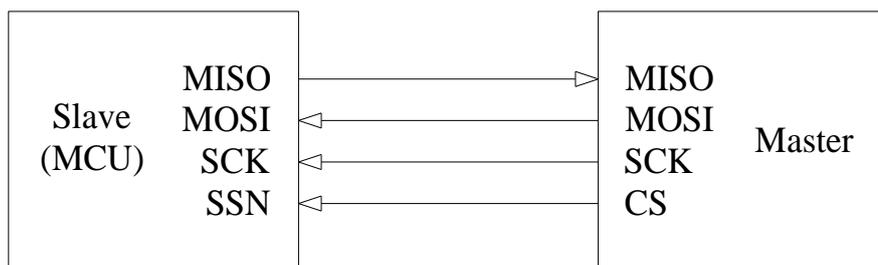


主机模式的通信示意如下图所示，其中 CPOL=0，CPHA=0。



18.3.2 SPI 从机模式

每个数据帧的长度固定为 8 比特，接收数据的第一位固定为 MSB。设置 SPI_CR.mstr 为 0，则 SPI 接口工作于从机模式。该模式下，SCK 引脚作为输入脚且串行时钟来自于外部主机；SSN 引脚作为输入脚且片选信号来自外部主机或固定为低电平。SSN 引脚详见 GPIO 章节辅助寄存器。从机模式的典型应用框图如下所示。



当 SPI 从机从 SPI 主机接收到一个数据帧时，SPIF 位会被置高；用户程序应尽快读取收到的数据。从机模式接收数据的通信时序如下所示，其中 CPOL=0，CPHA=0。

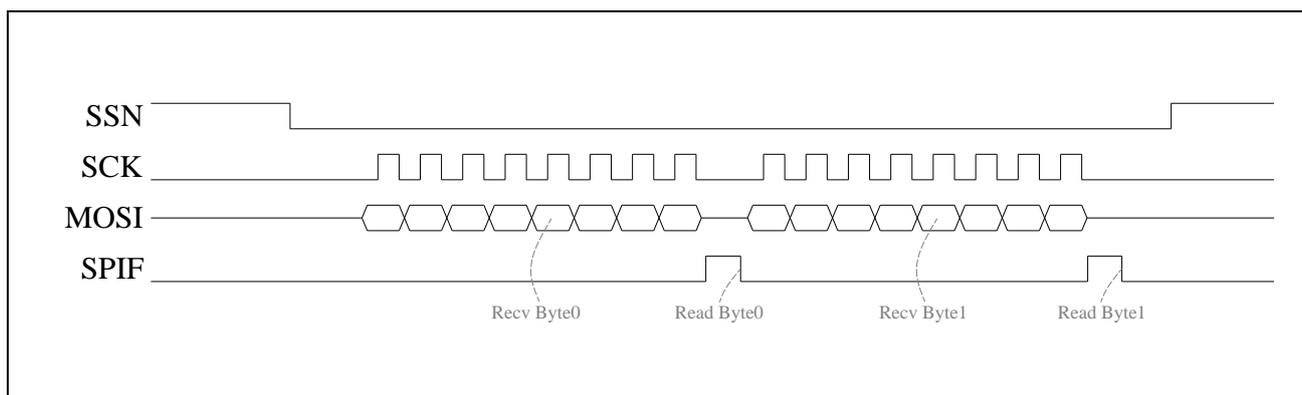


图 18-1 从机接收示意图

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。

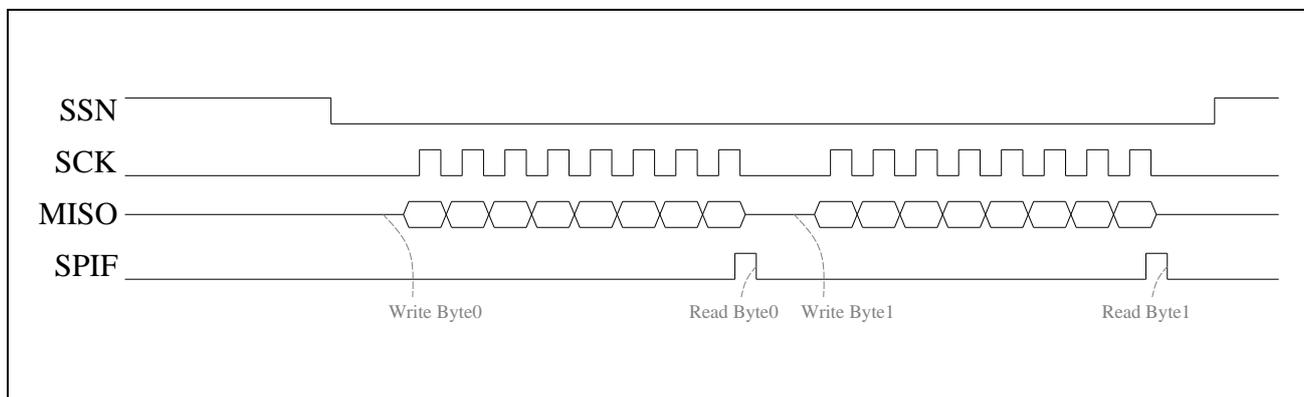


图 18-2 从机发送示意图

18.3.3 SPI 数据帧格式

SPI 接口帧格式取决于时钟极性位 CPOL 和时钟相位位 CPHA 的配置。

当 CPOL 为 0 时，SCK 线空闲状态为低电平。当 CPOL 为 1 时，SCK 线空闲状态为高电平。当 CPHA 为 0 时，数据会在第一个 SCK 时钟转换信号跳变时被采样。当 CPHA 为 1 时，数据会在第二个 SCK 时钟信号跳变时被采样。

SPI 接口主机帧格式如下图所示。

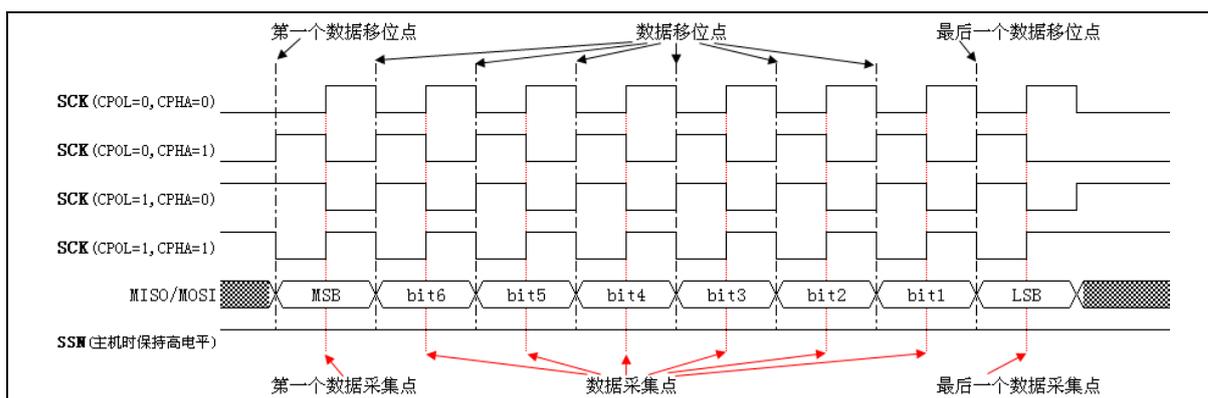


图 18-3 主机模式帧格式

SPI 接口从机帧格式如下图所示。

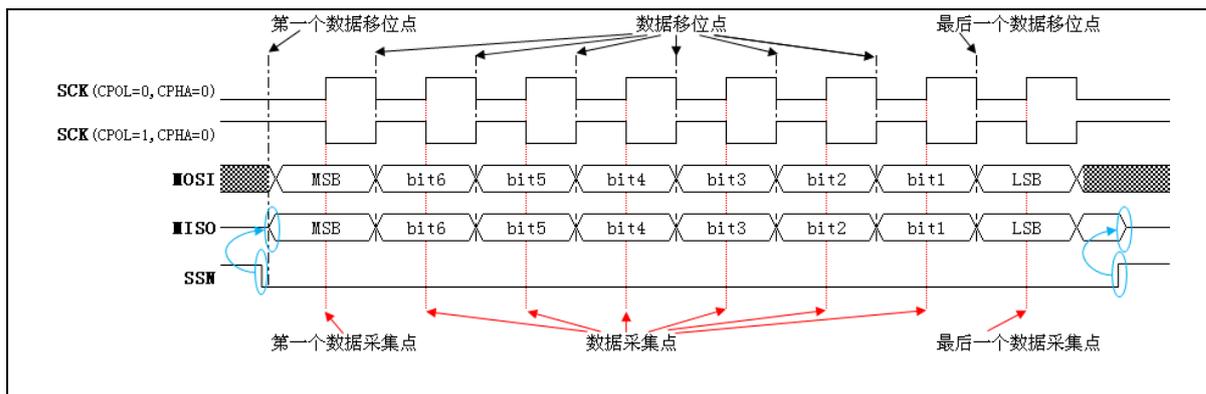


图 18-4 从机 CPHA 为 0 时数据帧格式

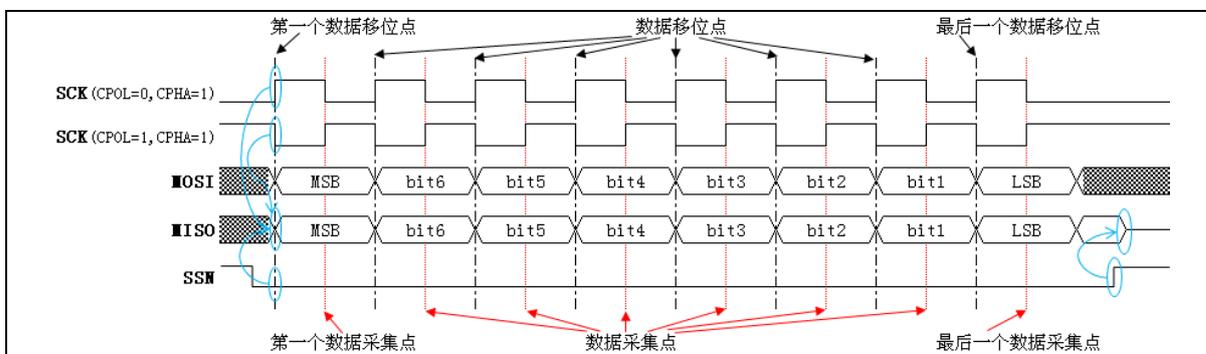


图 18-5 从机 CPHA 为 1 时数据帧格式

18.3.4 SPI 状态标志及中断

SPI 在工作中会产生如下三种状态标志，其产生条件及清除方法如下所示。

- 当完成一个数据帧的传送时，SPIF 会被硬件置位。对 SPI_DATA 寄存器进行读操作即可清除该标志位。
- 当 SPI 工作于主机模式且外部 SSN 输入为低电平，SPI_STAT.mdf 会被硬件置位，说明有其它 SPI 主机在占用总线。当 SSN 输入为高电平时，SPI_STAT.mdf 会被硬件自动清除。
- 当 SPI 工作于从机模式，如果正在进行数据传输时 SSN 管脚被拉高，则 SPI_STAT.sserr 会被置位。设置 SPI_CR.spen 为 0 可以清除该标志。

如果允许 SPI 中断向量，以下两种情况均可产生中断：

- SPI 传送完成，即 SPI_STAT.SPIF 为 1
- SPI 的主机模式错误，即 SPI_STAT.mdf 为 1

18.3.5 SPI 多机系统配置说明

- 当 SPI 模块作为主机且工作于单主机系统时，可通过 SPI_CS 管脚或 GPIO 管脚控制从机。当选择 SPI_CS 管脚作为从机的片选信号时，设置 SPI_SSN.ssn 为 0 即可选中从机，设置 SPI_SSN.ssn 为 1 即可释放从机。当选择 GPIO 管脚作为从机的片选信号时，设置 GPIOx_OUT 寄存器相应的比特为 0 即可选中从机，设置 GPIOx_OUT 寄存器相应的比特为 1 即可释放从机。
- 当 SPI 模块为从机时，根据需要配置 SPI_SSN 的来源（详见 GPIO 端口辅助控制器）。当 SSN 为低时，即可选中本从机以进行通信；当 SSN 为高时，则本机处于未选中态。
- 当 SPI 模式工作于多主机多从机时，所有的从机片选信号都通过 GPIO 管脚连接，主机还必须通过 GPIO 管脚和其他主机的 SSN 信号相连，来监测总线是否被占用。下图所示的 Master0 需要进行通信的操作方法为：等待 Master0.SSN 变为高；从 GPIO0 输出低以通知 Master1 释放 SPI 总线；从 GPIO2 输出低以选中 Slave1；与 Slave1 进行通信；从 GPIO2 输出高以释放 Slave1；从 GPIO0 输出高以释放 Master1。

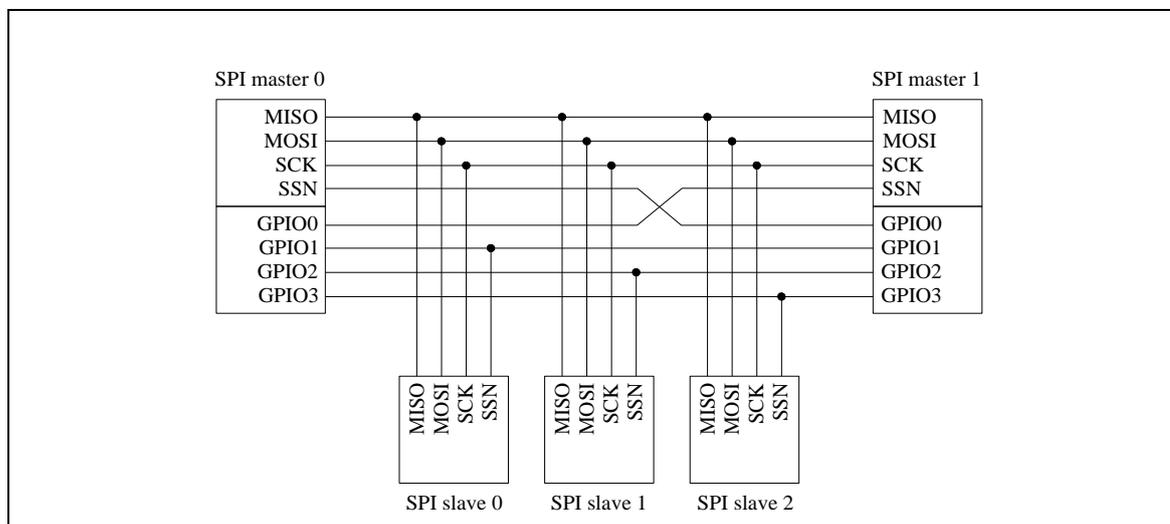


图 18-6 SPI 多主机/多从机系统的示意图

18.3.6 SPI 管脚配置说明

SPI 可以在一些特殊的管脚配置下保持部分或全部的功能。

具体情况如下表（“√”代表该管脚被配置并使用，空白代表该管脚未被配置）：

	SPI_CS(主)/ SPI_SSN(从)	SCK	MOSI	MISO	功能说明
主机模式	√	√	√	√	一般配置 全部主机功能正常
		√	√	√	全部主机功能正常
	√	√	√		主机发送功能正常
	√	√		√	主机接收功能正常
		√	√		主机发送功能正常
		√		√	主机接收功能正常
从机模式	√	√	√	√	一般配置 全部从机功能正常
	√	√	√		从机接收功能正常
	√	√		√	从机发送功能正常
	√ 固定低电平	√	√	√	全部从机功能正常
	√ 固定低电平	√	√		从机接收功能正常
	√ 固定低电平	√		√	从机发送功能正常

表 18-1 SPI 管脚配置说明表

注意：

- 表中未列举的情况暂不支持。
- 主机模式下，即使不使用 SPI_CS 片选输出，发送数据前需要设置 SPI_SSN 为 1，发送数据后需要设置 SPI_SSN 为 0。
- 从机模式且片选输入固定为低电平时，为维持正常功能，必须满足 SPI_CR.cpha=1。

18.4 SPI 编程示例

18.4.1 SPI 主机发送示例

Step1: 按 GPIO 章节管脚数字复用功能的相关描述, 将 CS/SCK/MISO/MOSI 映射到需要的管脚; 并配置 CS/SCK/MOSI 管脚为输出模式, 配置 MISO 管脚为输入模式。

Step2: 设置 SPI_CR.mstr 为 1, 使 SPI 工作于主机模式。

Step3: 配置 SPI_CR[spr2:spr0], 使 SCK 输出的时钟速率符合应用需求。

Step4: 配置 SPI_CR.cpol 及 SPI_CR.cpha, 使数据帧格式符合应用需求。

Step5: 设置 SPI_CR.spen 为 1, 使能 SPI 接口。

Step6: 设置 SPI_SSN.ssn 为 0, 使 CS 管脚输出低电平以选中从机。

Step7: 将待发送的数据写入 SPI_DATA, 等待 SPIF 变为 1。

Step8: 读取 SPI_DATA 以清除 SPIF 标志。

Step9: 如待发送的数据未完成, 则跳转到 Step7 继续执行。

Step10: 设置 SPI_SSN.ssn 为 1, 使 CS 管脚输出高电平以释放从机。

注意:

- 可使用 GPIO 代替 CS 实现片选输出, 多用于多机通信系统。
- 在传输过程必需设置 SPI_SSN.ssn 为 0, 传输完成后必需设置 SPI_SSN.ssn 为 1。

18.4.2 SPI 主机接收示例

Step1: 按 GPIO 章节管脚数字复用功能的相关描述, 将 CS/SCK/MISO/MOSI 映射到需要的管脚; 并配置 CS/SCK/MOSI 管脚为输出模式, 配置 MISO 管脚为输入模式。

Step2: 设置 SPI_CR.mstr 为 1, 使 SPI 工作于主机模式。

Step3: 配置 SPI_CR[spr2:spr0], 使 SCK 输出的时钟速率符合应用需求。

Step4: 配置 SPI_CR.cpol 及 SPI_CR.cpha, 使数据帧格式符合应用需求。

Step5: 设置 SPI_CR.spen 为 1, 使能 SPI 接口。

Step6: 设置 SPI_SSN.ssn 为 0, 使 CS 管脚输出低电平以选中从机。

Step7: 向 SPI_DATA 写入任意数据以触发主机发送 SCK。

Step8: 查询等待 SPI_STAT.SPIF 变为 1, 已收到从机发送的数据。

Step9: 从 SPI_DATA 读出收到的数据。

Step10: 如果待接收的数据未完成, 则跳转到 Step7 继续执行。

Step11: 设置 SPI_SSN.ssn 为 1, 使 CS 管脚输出高电平以释放从机。

注意:

- 可使用 GPIO 代替 CS 实现片选输出, 多用于多机通信系统。
- 在传输过程必需设置 SPI_SSN.ssn 为 0, 传输完成后必需设置 SPI_SSN.ssn 为 1。

18.4.3 SPI 从机发送示例

Step1: 按 GPIO 章节管脚数字复用功能的相关描述, 将 SSN/SCK/MISO/MOSI 映射到需要的管脚; 并配置 SSN/SCK/MOSI 管脚为输入模式, 配置 MISO 管脚为输出模式。SSN 管脚来源详见 GPIO 端口辅助控制器。

Step2: 设置 SPI_CR.mstr 为 0, 使 SPI 工作于从机模式。

Step3: 配置 SPI_CR.cpol 及 SPI_CR.cpha, 使数据帧格式符合应用需求。

Step4: 设置 SPI_CR.spen 为 1, 使能 SPI 接口。

Step5: 查询等待 SSN 管脚拉低, 主机选中 SPI 从机。

Step6: 将待发送的数据写入 SPI_DATA, 等待 SPIF 变为 1。

Step7: 读取 SPI_DATA 以清除 SPIF 标志。

Step8: 当查询到 SSN 管脚为低且待发送的数据尚未完成, 跳转到 Step6。

Step9: 查询等待 SSN 管脚拉高, 主机释放 SPI 从机。

18.4.4 SPI 从机接收示例

Step1: 按 GPIO 章节管脚数字复用功能的相关描述, 将 SSN/SCK/MISO/MOSI 映射到需要的管脚; 并配置 SSN/SCK/MOSI 管脚为输入模式, 配置 MISO 管脚为输出模式。SSN 管脚来源详见 GPIO 端口辅助控制器。

Step2: 设置 SPI_CR.mstr 为 0, 使 SPI 工作于从机模式。

Step3: 配置 SPI_CR.cpol 及 SPI_CR.cpha, 使数据帧格式符合应用需求。

Step4: 设置 SPI_CR.spen 为 1, 使能 SPI 接口。

Step5: 查询等待 SSN 管脚拉低, 主机选中 SPI 从机。

Step6: 查询等待 SPI_STAT.SPIF 变为 1, 已收到主机发送的数据。

Step7: 从 SPI_DATA 读出收到的数据。

Step8: 如果待接收的数据未完成, 则跳转到 Step6 继续执行。

Step9: 查询等待 SSN 管脚拉高, 主机释放 SPI 从机。

18.6 SPI 寄存器描述

寄存器列表

SPI 基地址: 0x40000800

偏移量	寄存器名称	访问	寄存器描述
0x00	SPI_CR	RW	SPI 配置寄存器
0x04	SPI_SSN	RW	SPI 片选配置寄存器
0x08	SPI_STAT	RO	SPI 状态寄存器
0x0c	SPI_DATA	RW	SPI 数据寄存器

表 18-2 SPI 寄存器列表

18.6.1 SPI 配置寄存器(SPI_CR)

地址偏移量：0x00

复位值：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

位	标记	功能描述
31:8	Reserved	
7	spr2	波特率选择位 2 参考 spr0.
6	spen	SPI 模块使能控制 0 – 禁止 1 – 使能
5	Reserved	
4	mstr	SPI 工作模式配置 0 – 从机模式 1 – 主机模式
3	cpol	SCK 线空闲状态配置 0 – 低电平 1 – 高电平
2	cpha	时钟相位配置 0 – 第一边沿 1 – 第二边沿
1	spr1	波特率选择位 1 参考 spr0

0	spr0	波特率选择位 0			
		spr2	spr1	spr0	SCK Rate
		0	0	0	PCLK /2
		0	0	1	PCLK /4
		0	1	0	PCLK /8
		0	1	1	PCLK /16
		1	0	0	PCLK /32
		1	0	1	PCLK /64
		1	1	0	PCLK /128
		1	1	1	Reserved
表 18-3 主机模式波特率选择					

18.6.2 SPI 片选配置寄存器(SPI_SSN)

地址偏移量: 0x04

复位值: 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	
Reserved														RW	

位	标记	功能描述
31:1	Reserved	
0	ssn	主机模式时 SPI_CS 输出电平配置 0: SPI_CS 端口输出低电平 1: SPI_CS 端口输出高电平

18.6.3 SPI 状态寄存器(SPI_STAT)

地址偏移量: 0x08

复位值: 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								spif	Res.	sserr	mdf	Reserved			
								RO		RO	RO				

位	标记	功能描述
31:8	Reserved	
7	spif	传输完成标志 1: SPI 总线已完成一个字节的传输 0: SPI 总线正在传输中
6	Reserved	
5	sserr	从机模式 SSN 错误标志
4	mdf	主机模式时, 冲突标志 1: SSN 管脚电平为低 0: SSN 管脚电平为高
3:0	Reserved	

18.6.4 SPI 数据寄存器(SPI_DATA)

地址偏移量: 0x0c

复位值: 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							
Reserved								RW							

位	标记	功能描述
31:8	Reserved	
7:0	spdat	数据寄存器 在发送模式, 向这个寄存器写入待发送的字节; 在接收模式, 从这个寄存器读取接收到的字节;

19 时钟校准模块 (CLKTRIM)

19.1 CLK_TRIM 简介

CLK_TRIM(Clock Trimming)模块是一个专门用来校准/监测时钟的电路。在校准模式下选择精准的时钟源来校准不精准的时钟源，反复校准，调节不精准时钟源的参数，直到被校准时钟源的频率达到精度要求。校准模式下计数值会有一些的误差，但是在允许的精度误差范围内。在监测模式下选择稳定的时钟源来监测系统工作时钟，在设定的监测周期下，监测系统工作时钟是否有失效的情况发生并产生中断。在校准模式和监控模式下，所需的时钟源都必须初始化和使能，具体的配置过程请参考第4章系统控制器。

19.2 CLK_TRIM 主要特性

CLK_TRIM 支持以下特性：

- 校准模式
- 监测模式
- 32 位参考时钟计数器可加载初值
- 32 位待校准时钟计数器可配置溢出值
- 6 种参考时钟源
- 4 种待校准时钟源
- 支持中断方式

19.3 CLK_TRIM 功能描述

19.3.1 CLK_TRIM 校准模式

校准模式主要用于选择一个精准的时钟源作为参考时钟来校准一个不精准的待校准时钟源。

由软件按照下面的操作流程反复校准，调节待校准时钟源参数，直到待校准时钟源满足频率精度要求。

19.3.1.1 操作流程

1. 设置 CLKTRIM_CR.refclk_sel 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR.calclk_sel 寄存器选择被校准时钟。
3. 设置 CLKTRIM_REFCON.rcntval 寄存器为校准时间。
4. 设置 CLKTRIM_CR.IE 寄存器使能中断。
5. 设置 CLKTRIM_CR.trim_start 寄存器开始校准。
6. 参考时钟计数器和待校准时钟计数器开始计数。
7. 当参考时钟计数器从初始值减计数到 0 时，CLKTRIM_IFR.stop 置 1，触发中断。
8. 中断服务子程序判断 CLKTRIM_IFR.stop 为 1，读取寄存器 CLKTRIM_REFCNT 和 CLKTRIM_CALCNT 的值，
9. 清零 CLKTRIM_CR.trim_start 寄存器结束校准。

注意：

- 校准模式在校准过程中有可能因为校准时间设置过长，发生待校准时钟计数器在 CLKTRIM_IFR.stop 置 1 之前溢出的情况，CLKTRIM_IFR.calcnt_of 置 1，触发中断。中断服务子程序发现 CLKTRIM_IFR.calcnt_of 置 1 时，清零 CLKTRIM_CR.trim_start 寄存器结束校准。

这种情况下校准是无法正确进行的，必须调整校准时间，重新校准。

具体步骤是：

设置 CLKTRIM_REFCON.rcntval 寄存器调整校准时间。

设置 CLKTRIM_CR.trim_start 寄存器重新开始校准。

19.3.2 CLK_TRIM 监测模式

监测模式主要用于选择一个稳定的时钟源作为参考时钟，在设定的时间周期下监测系统工作时钟的异常状态。在监测模式下只能选择外部 XTH 时钟或者外部 XTL 时钟作为被监测时钟。

19.3.2.1 操作流程

1. 设置 CLKTRIM_CR .refclk_sel 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR .calclk_sel 寄存器选择被监控时钟。
3. 设置 CLKTRIM_REFCON .rentval 寄存器为监控间隔时间。
4. 设置 CLKTRIM_CALCON.ccntval 寄存器为被监控时钟计数器溢出时间。
5. 设置 CLKTRIM_CR .mon_en 寄存器使能监控功能。
6. 设置 CLKTRIM_CR .IE 寄存器使能中断。
7. 设置 CLKTRIM_CR .trim_start 寄存器开始监控。
8. 参考时钟计数器和被监控时钟计数器开始计数。
9. 当参考时钟计数器计数到达监控间隔时间时，判断被监控时钟计数器是否溢出。
如果溢出表示被监控时钟工作正常。如果没有溢出表示被监控时钟失效，
CLKTRIM_IFR .xtal32k_fault/xtal32m_fault 置 1，触发中断。
10. 处理中断服务子程序，清除中断标志位
CLKTRIM_IFR .xtal32k_fault/xtal32m_fault，清零 CLKTRIM_CR .trim_start 寄存器
结束监测。

19.4 CLK_TRIM 寄存器描述

寄存器列表

基地址：0x40001800

偏移量	寄存器名称	访问	寄存器描述
0x00	CLKTRIM_CR	RW	配置寄存器.
0x04	CLKTRIM_REFCON	RW	参考计数器初值配置寄存器.
0x08	CLKTRIM_REFCNT	RO	参考计数器值寄存器.
0x0c	CLKTRIM_CALCNT	RO	校准计数器值寄存器.
0x10	CLKTRIM_IFR	RO	中断标志位寄存器.
0x14	CLKTRIM_ICLR	RW	中断标志位清除寄存器
0x18	CLKTRIM_CALCON	RW	校准计数器溢出值配置寄存器

表 19-1 寄存器列表

19.4.1 配置寄存器(CLKTRIM_CR)

偏移地址：0x00

复位值：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			

位	标记	功能描述
31:8	Reserved	
7	IE	中断使能寄存器 0 – 禁止 1 – 使能
6	mon_en	监测模式使能寄存器 0 – 禁止 1 – 使能
5:4	calclk_sel	待校准/监测时钟选择寄存器 00 ---- RCH 01 ---- XTH 10 ---- RCL 11 ---- XTL
3:1	refclk_sel	参考时钟选择寄存器 000 ---- RCH 001 ---- XTH 010 ---- RCL 011 ---- XTL 100 ---- IRC10K 101 ---- EXT_CLK_IN
0	trim_start	校准/监测开始寄存器 0 – 停止 1 – 开始

19.4.2 参考计数器初值配置寄存器(CLKTRIM_REFCON)

偏移地址：0x04

复位值：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															

位	标记	功能描述
31:0	rcntval	参考计数器初始值

19.4.3 参考计数器值寄存器(CLKTRIM_REFCNT)

偏移地址：0x08

复位值：0x00000000

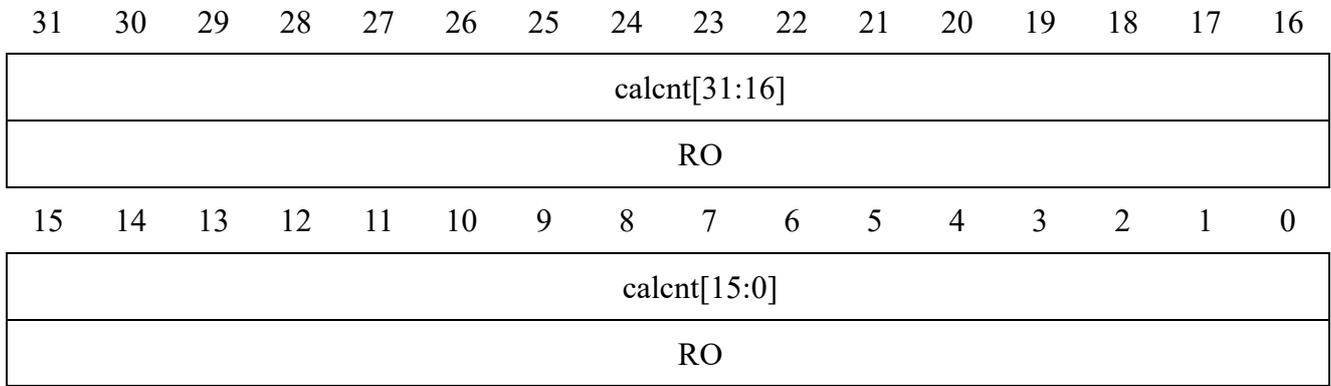
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
refcnt[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
refcnt[15:0]															
RO															

位	标记	功能描述
31:0	refcnt	参考计数器值

19.4.4 校准计数器值寄存器(CLKTRIM_CALCNT)

偏移地址: 0x0c

复位值: 0x00000000



位	标记	功能描述
31:0	calcnt	校准计数器值

19.4.5 中断标志位寄存器(CLKTRIM_IFR)

偏移地址：0x10

复位值：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	xtl_f	calcn	stop
Reserved												ault	ault	t_of	
Reserved												RO	RO	RO	RO

位	标记	功能描述
31:4	Reserved	
3	xth_fault	XTH 失效标志. CLKTRIM_ICLR.xth_fault_clr 写零清除此标志位
2	xtl_fault	XTL 失效标志. CLKTRIM_ICLR.xtl_fault_clr 写零清除此标志位
1	calcnt_of	校准计数器溢出标志. CLKTRIM_CR.start 写零清除此标志位
0	stop	参考计数器停止标志. CLKTRIM_CR.start 写零清除此标志位

19.4.6 中断标志位清除寄存器(CLKTRIM_ICLR)

偏移地址：0x14

复位值：0xf

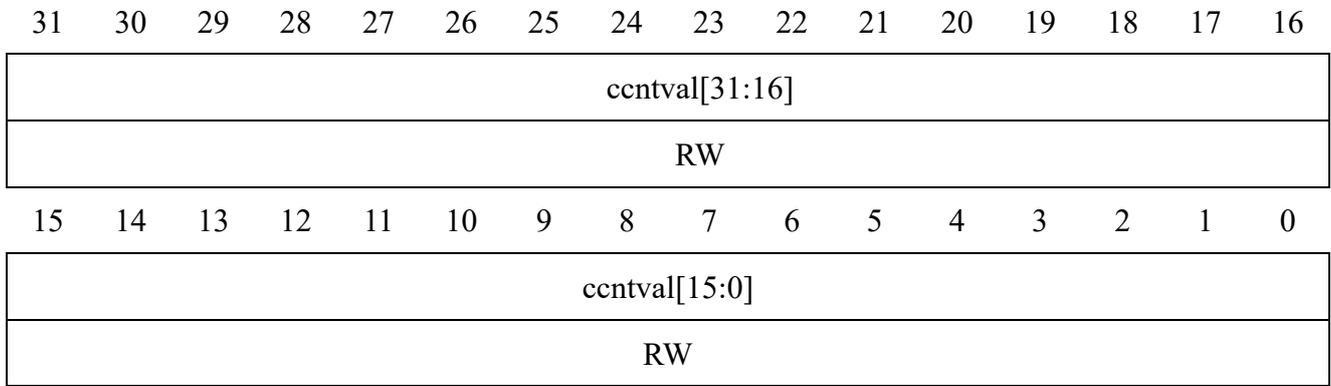
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												xth_f	xtl_f	Reserved	
												ault_	ault_		
												clr	clr		
												RW	RW		

位	标记	功能描述
31:4	Reserved	
3	xth_fault_clr	清除 XTH 失效标志，写零清除。
2	xtl_fault_clr	清除 XTL 失效标志，写零清除。
1:0	Reserved	

19.4.7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON)

偏移地址：0x18

复位值：0xffff ffff



位	标记	功能描述
31:0	ccntval	校准计数器溢出值

20 循环冗余校验（CRC）

20.1 概述

循环冗余校验 (CRC) 计算单元将数据流或数据块作为输入，在生成多项式的控制下生成一个输出数。该输出数常用于验证数据传输或存储的正确性和完整性。本模块支持计算 CRC 值和检验 CRC 值。

20.2 主要特性

- 一种执行标准：ISO/IEC13239
- 一种编码方式：CRC-16， $x^{16} + x^{12} + x^5 + 1$
- 三种写入位宽：8bit，16bit，32bit
- 两种工作模式：CRC 编码模式、CRC 校验模式

20.3 功能描述

20.3.1 工作模式

本模块支持两种工作模式：CRC 编码模式、CRC 校验模式。

CRC 编码模式是指向 CRC 模块输入一定数量的原始数据，获取 CRC 模块生成的输出值（CRC_RESULT.RESULT）。CRC 校验模式是指向 CRC 模块输入一定数量的原始数据+CRC 校验值，验证原始数据与 CRC 校验值是否匹配（CRC_RESULT.FLAG）。

20.3.2 编码方式

本模块支持 CRC-16 编码，其计算结果为 16 比特，生成多项式为 $x^{16} + x^{12} + x^5 + 1$ 。

20.3.3 写入位宽

本模块支持三种写入位宽：8bit，16bit，32bit。不同位宽的写入需要符合“位宽一致、先低后高”的原则，即“每次写入数据，都必须写入到与本次有效数据位宽相等的寄存器中，并且较低位数据的写入先于较高位数据”。

下方展示了同一序列数据采用三种位宽进行写入的方法，其输出结果相同。

- 8bit 位宽写入：0x00, 0x11, 0x22 , 0x33, 0x44, 0x55, 0x66, 0x77
- 16bit 位宽写入：0x1100, 0x3322, 0x5544, 0x7766
- 32bit 位宽写入：0x33221100, 0x77665544

20.4 编程示例

20.4.1 CRC-16 编码模式

Step 1: 向 CRC_RESULT 写入 0xFFFF, 初始化 CRC 计算。

Step 2: 将待编码的原始数据依次写入 CRC_DATA 寄存器, 写入位宽可选择 8bit、16bit、32bit。

Step 3: 读取 CRC_RESULT.RESULT 以获取 CRC 值。

20.4.2 CRC-16 检验模式

Step 1: 向 CRC_RESULT 写入 0xFFFF, 初始化 CRC 计算。

Step 2: 将已编码的数据序列依次写入 CRC_DATA 寄存器, 写入位宽可选择 8bit、16bit、32bit。

Step 3: CRC_RESULT.FLAG 的值判定已编码的数据序列是否被篡改。

20.5 寄存器描述

20.5.1 寄存器列表

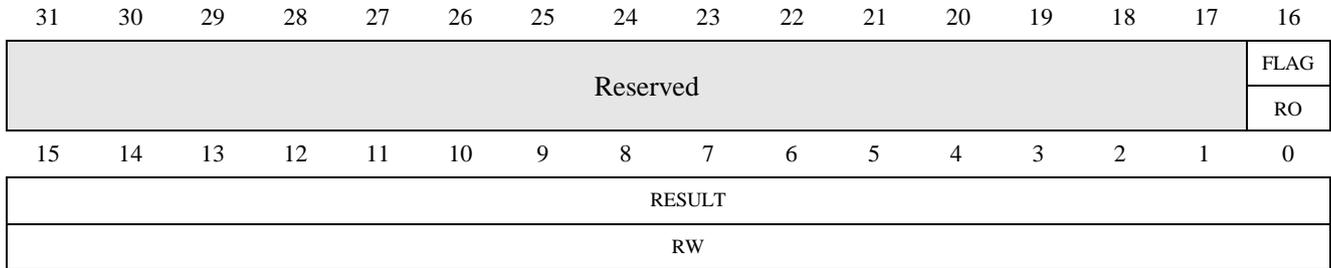
基地址：0x4002 0900

寄存器	偏移地址	描述
CRC_RESULT	0x04	CRC 结果寄存器
CRC_DATA	0x80	CRC 数据寄存器

20.5.2 结果寄存器 (CRC_RESULT)

偏移地址: 0x04

复位值: 0x0000 0000

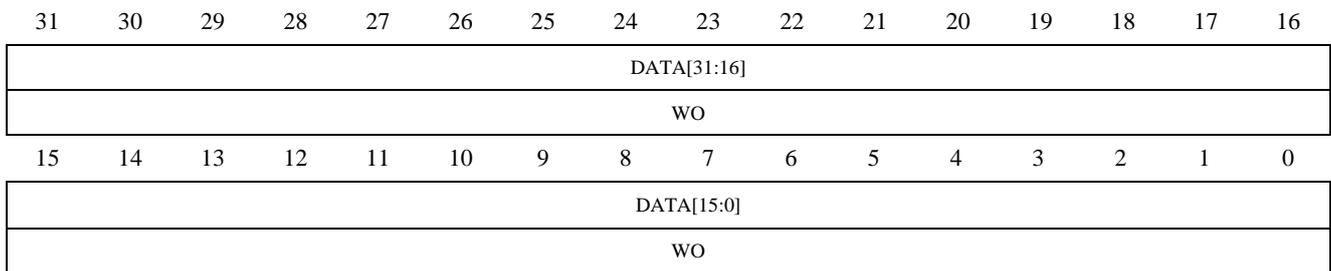


位	符号	描述
16	FLAG	CRC校验结果 0: 当前CRC校验错误 1: 当前CRC校验正确
15:0	RESULT	CRC计算结果 读取该寄存器以获取CRC计算结果 向该寄存器写入0xFFFF可初始化CRC计算

20.5.3 数据寄存器 (CRC_DATA)

偏移地址: 0x80

复位值: 0x0000 0000



位	符号	功能描述
31:0	DATA	本寄存器用于写入需要运算的数据, 支持3种写入位宽 8bit写入方式: $((\text{uint8_t} *)0x40020980) = 0xXX$ 16bit写入方式: $((\text{uint16_t} *)0x40020980) = 0xFFFF$ 32bit写入方式: $((\text{uint32_t} *)0x40020980) = 0xFFFFFFFF$

21 模数转换器 (ADC)

21.1 模块简介

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。本系列内部集成了一个 12 位高精度、高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 12 位转换精度；
- 1Msps 转换速度($V_{CC} > 2.7V$)；
- 12 路转换通道：9 个引脚通道、内置温度传感器、内置 1.2v 基准电压、1/3 电源电压；
- 4 种参考源：电源电压、ExRef 引脚、内置 1.5v 参考电压、内置 2.5v 参考电压；
- ADC 的电压输入范围：0~Vref；
- 3 种转换模式：单次转换、连续转换、累加转换；
- 软件可配置 ADC 的转换速率；
- 内置信号放大器，可转换高阻信号；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

21.2 ADC 框图

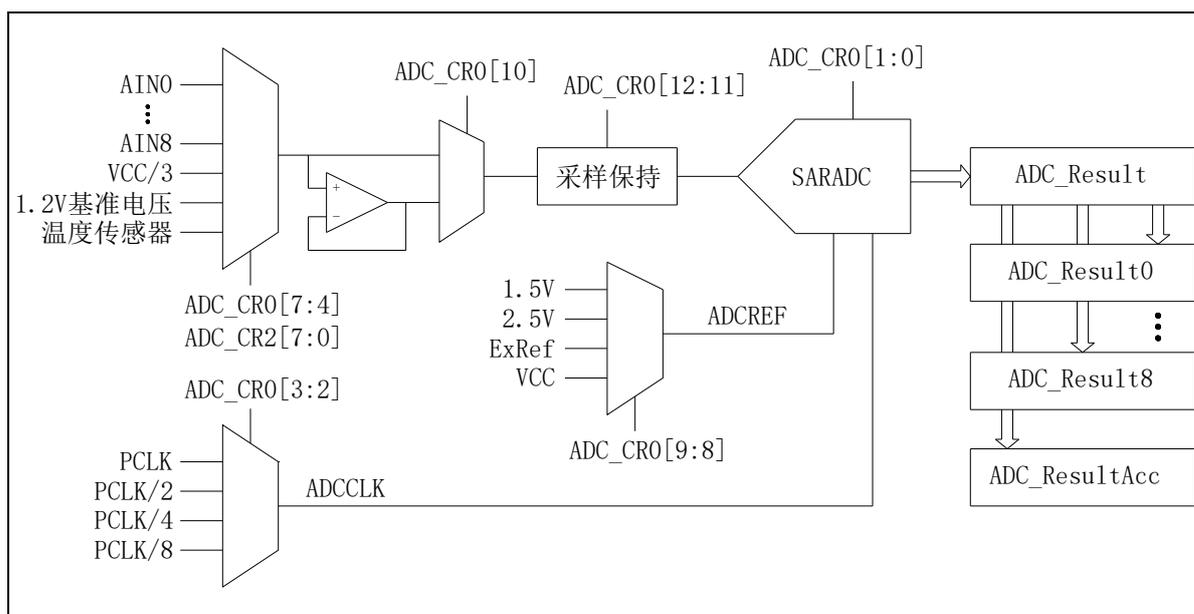


图 21-1 ADC 示意框图

21.3 转换时序及转换速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由采样过程及逐次比较过程组成。其中采样过程需要 4~12 个 AdcClk，由 ADC_CR0.SAM 配置；逐次比较过程需要 16 个 AdcClk。所以，一次 ADC 转换共需要 20~28 个 AdcClk。

ADC 转换速度的单位为 sps，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：AdcClk 的频率 / 一次 ADC 转换所需要的 AdcClk 的个数。

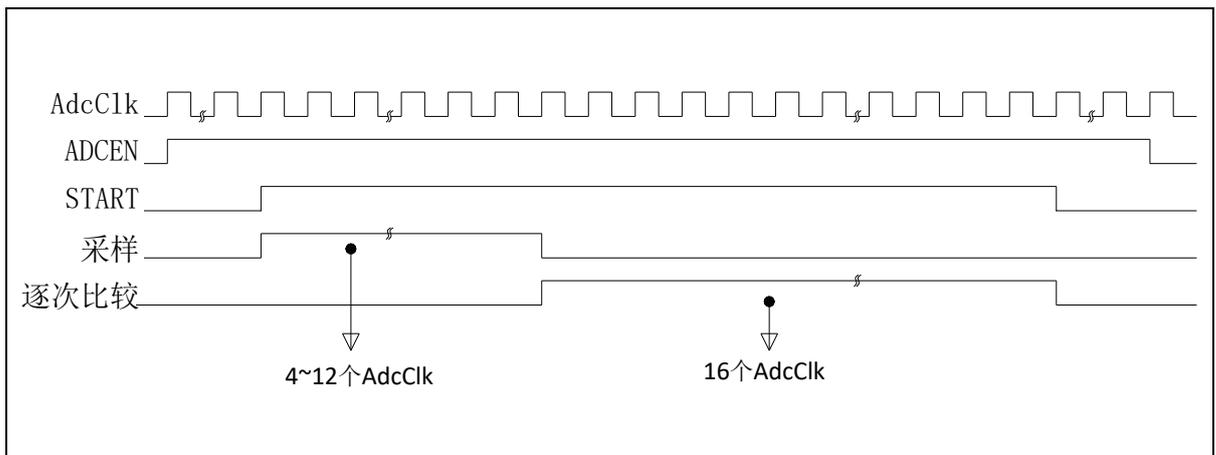


图 21-2 ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 VCC 电压相关，最高转换速度如下表所示：

ADC 参考电压	VCC 电压	最高转换速度	最大 AdcClk 频率
内部 1.5V	1.8~5.5V	200Ksps	4MHz
内部 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 单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 12 路 ADC 通道进行转换。该模式既可通过设置 ADC_CR0.START 位启动也可通过设置 ADC_CR1[9:0]的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_CR0.START 位自动清零，转换结果保存在 ADC_result 寄存器中。

通过 START 位启动 ADC 单次转换操作流程：

Step1: 配置 P0ADS~P3ADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 P3ADS.6 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.ADCEN 为 1，使能 ADC 模块。

Step5: 延时 20uS，等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.CT 为 0，选择单次转换模式。

Step7: 配置 ADC_CR0.SREF，选择 ADC 的参考电压。

Step8: 配置 ADC_CR0.SAM 及 ADC_CR0.CLKSEL，设置 ADC 的转换速度。

Step9: 配置 ADC_CR0.SEL，选择待转换的通道。

Step10: 设置 ADC_CR0.START 为 1，启动 ADC 单次转换。

Step11: 等待 ADC_CR0.START 变为 0，读取 ADC_result 寄存器以获取 ADC 转换结果。

Step12: 如需对其它通道进行转换，重复执行 Step9~Step11。

Step13: 设置 ADC_CR0.ADCEN 及 BGR_CR.BGR_EN 为 0，关闭 ADC 模块、BGR 模块。

通过外部触发启动 ADC 单次转换操作流程：

Step1: 配置 P0ADS~P3ADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 P3ADS.6 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

- Step3: 设置 BGR_CR.BGR_EN 为 1, 使能 BGR 模块。
- Step4: 设置 ADC_CR0.ADCEN 为 1, 使能 ADC 模块。
- Step5: 延时 20uS, 等待 ADC 及 BGR 模块启动完成。
- Step6: 设置 ADC_CR1.CT 为 0, 选择单次转换模式。
- Step7: 设置 ADC_HT 为 0x00。
- Step8: 设置 ADC_CR1.HtCmp 为 1, 使能 ADC 高阈值比较功能。
- Step9: 设置 ADC_CR0.IE 为 1, 使能 ADC 中断。
- Step10: 使能 NVIC 中断向量表中的 ADC 中断。
- Step11: 配置 ADC_CR0.SREF, 选择 ADC 的参考电压。
- Step12: 配置 ADC_CR0.SAM 及 ADC_CR0.CLKSEL, 设置 ADC 的转换速度。
- Step13: 配置 ADC_CR0.SEL, 选择待转换的通道。
- Step14: 设置 ADC_IFR 为 0x00, 清除 ADC 中断标志。
- Step15: 配置 ADC_CR1.TRIGS1 及 ADC_CR1.TRIGS0, 选择外部触发条件。
- Step16: 当外部触发条件触发 ADC 完成转换时, ADC 模块会产生中断。用户可在 ADC 中断服务程序中读取 ADC_result 寄存器以获取 ADC 转换结果。
- Step17: 如需对其它通道进行转换, 重复执行 Step13~Step16。
- Step18: 设置 ADC_CR0.ADCEN 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

21.5 连续转换模式

在连续转换模式下，启动一次 ADC 可对多个通道依次进行多次转换；可转换的 ADC 通道为 AIN0~AIN7。ADC 转换的总次数由 ADCCR2.ADCCNT 进行配置；待进行转换的通道由 ADC_CR2[7:0]进行配置。该模式既可通过设置 ADC_CR0.START 位启动也可通过设置 ADC_CR2[9:0]的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.CONT_INTF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_result0~ADC_result7 寄存器中。如果总转换次数大于待转换的 ADC 通道的数量，则 ADC_result0~ADC_result7 寄存器中只保存最后一次的转换结果。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换的过程。通过寄存器将 START 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。

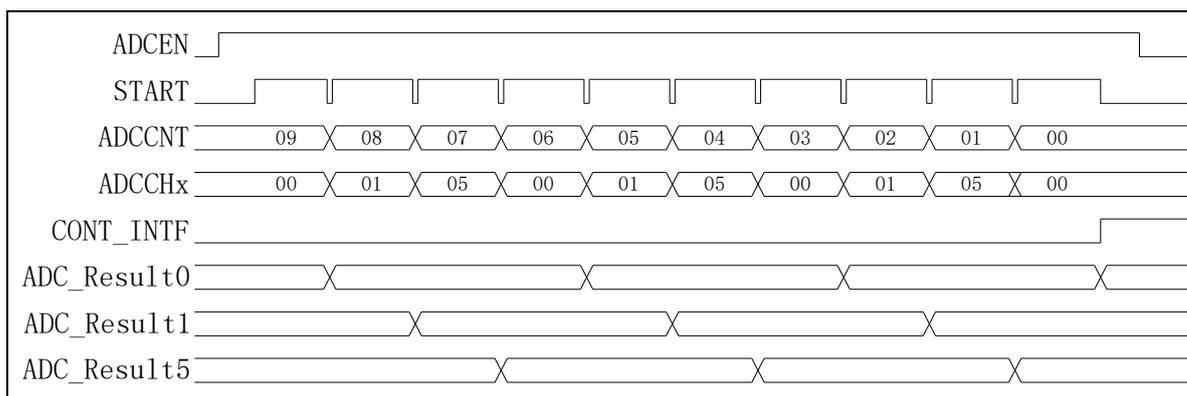


图 21-3 ADC 连续转换过程示例

通过 START 位启动 ADC 连续转换操作流程：

Step1: 配置 P0ADS~P3ADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 P3ADS.6 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.ADCEN 为 1，使能 ADC 模块。

Step5: 延时 20uS，等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.CT 为 1，选择连续转换模式。

- Step7: 设置 ADC_CR1[14:12]为 0, 关闭转换结果比较功能。
- Step8: 配置 ADC_CR2.ADCNT, 选择连续转换的总转换次数。
- Step9: 配置 ADC_CR0.SREF, 选择 ADC 的参考电压。
- Step10: 配置 ADC_CR0.SAM 及 ADC_CR0.CLKSEL, 设置 ADC 的转换速度。
- Step11: 配置 ADC_CR2[7:0], 选择待转换的通道。
- Step12: 设置 ADC_ICLR.CONT_INTC 为 0, 清除 ADC_IFR.CONT_INTF 标志。
- Step13: 设置 ADC_CR0.StateRst 为 1, 复位连续转换状态。
- Step14: 设置 ADC_CR0.START 为 1, 启动 ADC 连续转换。
- Step15: 等待 ADC_IFR.CONT_INTF 变为 1, 读取 ADC_result0~ ADC_result7 寄存器以获取相应通道的转换结果。
- Step16: 如需对其它通道进行转换, 重复执行 Step11~Step15。
- Step17: 设置 ADC_CR0.ADCEN 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

21.6 连续转换累加模式

在连续转换累加模式下，启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加；可转换的 ADC 通道为 AIN0~AIN7。ADC 转换的总次数由 ADCCR2.ADCCNT 进行配置；待进行转换的通道由 ADC_CR2[7:0]进行配置。该模式既可通过设置 ADC_CR0.START 位启动也可通过设置 ADC_CR2[9:0]的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_IFR.CONT_INTF 位会自动置 1，转换结果的累加值保存在 ADC_result_acc 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。通过寄存器将 START 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。每次转换完成时，ADC_result_acc 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

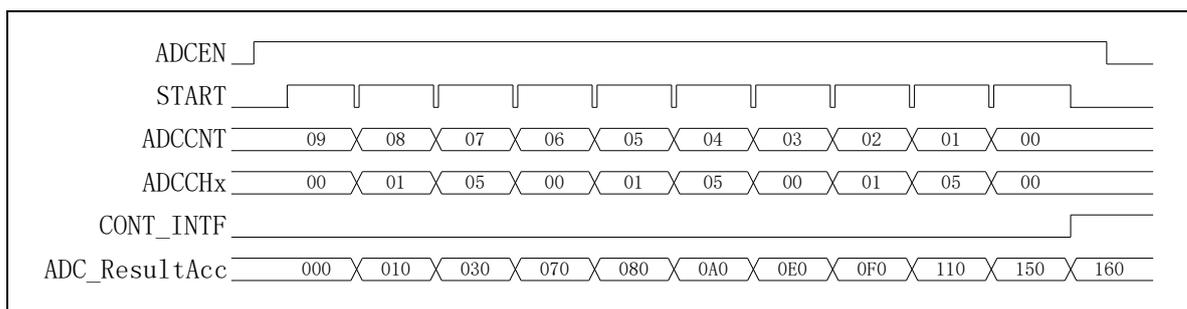


图 21-4 ADC 连续转换累加过程示例

通过 START 位启动 ADC 连续转换累加操作流程：

Step1: 配置 P0ADS~P3ADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2: 设置 P3ADS.6 为 1，将 ADC 外部参考电压引脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压引脚，则可以略过本步骤。

Step3: 设置 BGR_CR.BGR_EN 为 1，使能 BGR 模块。

Step4: 设置 ADC_CR0.ADCEN 为 1，使能 ADC 模块。

Step5: 延时 20uS，等待 ADC 及 BGR 模块启动完成。

Step6: 设置 ADC_CR1.CT 为 1，选择连续转换模式。

Step7: 设置 ADC_CR1[14:12]为 0，关闭转换结果比较功能。

- Step8: 设置 ADC_CR1.RACC_EN 为 1, 使能 ADC 转换自动累加功能。
- Step9: 配置 ADC_CR2.ADCNT, 选择连续转换的总转换次数。
- Step10: 配置 ADC_CR0.SREF, 选择 ADC 的参考电压。
- Step11: 配置 ADC_CR0.SAM 及 ADC_CR0.CLKSEL, 设置 ADC 的转换速度。
- Step12: 配置 ADC_CR2[7:0], 选择待转换的通道。
- Step13: 设置 ADC_ICLR.CONT_INTC 为 0, 清除 ADC_IFR.CONT_INTF 标志。
- Step14: 设置 ADC_CR1.RACC_CLR 为 0, 清零 ADC_result_acc 寄存器。
- Step15: 设置 ADC_CR0.StateRst 为 1, 复位连续转换状态。
- Step16: 设置 ADC_CR0.START 为 1, 启动 ADC 连续转换。
- Step17: 等待 ADC_IFR.CONT_INTF 变为 1, 读取 ADC_result_acc 寄存器以获取转换结果累加值。
- Step18: 如需对其它通道进行转换, 重复执行 Step12~Step17。
- Step19: 设置 ADC_CR0.ADCEN 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。

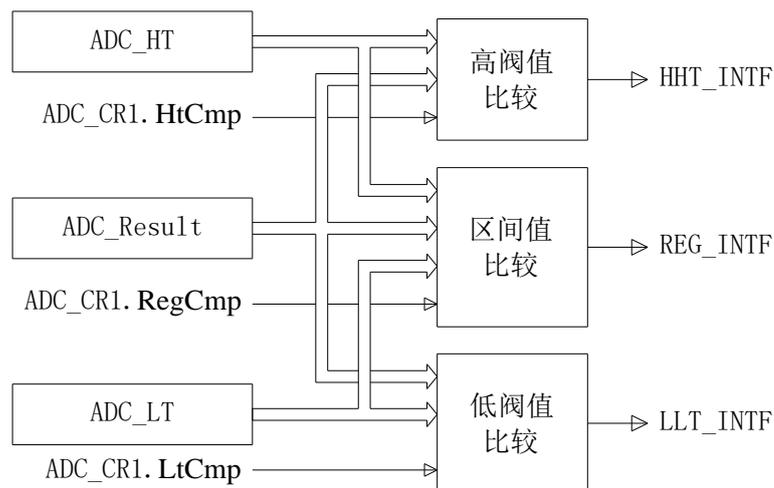
21.7 ADC 转换结果比较

ADC 转换完成时，ADC 转换结果可以与用户设定的阈值进行比较，支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 HtCmp、LtCmp、RegCmp 置 1。该功能可实现对模拟量的自动监测，直到 ADC 转换结果符合用户预期时才产生中断申请用户程序介入。

上阈值比较：当 ADC 转换结果位于[ADC_HT, 4095]区间内则 ADC_IFR. HHT_INTF 置 1；向 ADC_ICLR. HHT_INTC 写入 0 则清零 ADC_IFR. HHT_INTF。

下阈值比较：当 ADC 转换结果位于[0,ADC_LT)区间内则 ADC_IFR. LLT_INTF 置 1；向 ADC_ICLR. LLT_INTC 写入 0 则清零 ADC_IFR. LLT_INTF。

区间值比较：当 ADC 转换结果位于[ADC_LT, ADC_HT)区间内则 ADC_IFR. REG_INTF 置 1；向 ADC_ICLR. REG_INTC 写入 0 则清零 ADC_IFR. REG_INTF。



21.8 ADC 中断

ADC 中断请求如下表所示：

中断源	中断标志	中断使能
ADC 连续转换完成	ADC_IFR.CONT_INTF	ADC_CR0.IE
ADC 转换结果位于区间值区域	ADC_IFR.REG_INTF	
ADC 转换结果位于上阈值区域	ADC_IFR.HHT_INTF	
ADC 转换结果比较下阈值区域	ADC_IFR.LLT_INTF	

21.9 使用温度传感器测量环境温度

温度传感器的输出电压会随环境温度的改变而变化，故根据温度传感器的输出电压即可计算出相应环境温度。当 ADC 模块的测量通道选择温度传感器的输出电压时，即可测量环境温度。

计算公式如下：

$$\text{环境温度} = 25 + 0.0839 \times V_{\text{ref}} \times (\text{AdcValue} - \text{Trim})$$

其中：Vref 为当前 ADC 模块的参考电压，取值为 1.5 或 2.5。

AdcValue 为 ADC 模块测量温度传感器输出电压的结果，取值为 0~4095。

Trim 为 16bit 的校准值，计算时需要从 Flash 存储器中读出，其存放地址详见下表。

ADC 参考电压	校准值存放地址	校准值精度
内部 1.5V	0x00100C34	±5℃
内部 2.5V	0x00100C36	±5℃

计算示例如下：

条件 1：Vref=2.5、AdcValue=0x7E5、Trim=0x76C：

温度 1： $25 + 0.0839 \times 2.5 \times (0x7E5 - 0x76C) = 50^{\circ}\text{C}$ 。

条件 2：Vref=1.5、AdcValue=0x72D、Trim=0x76C：

温度 3： $25 + 0.0839 \times 1.5 \times (0x72D - 0x76C) = 17^{\circ}\text{C}$ 。

通过 ADC 测量环境温度操作流程:

- Step1: 设置 BGR_CR.BGR_EN 为 3, 使能 BGR 模块和温度传感器模块。
- Step2: 设置 ADC_CR0.ADCEN 为 1, 使能 ADC 模块。
- Step3: 延时 20uS, 等待 ADC 及 BGR 模块启动完成。
- Step4: 设置 ADC_CR1.CT 为 0, 选择单次转换模式。
- Step5: 配置 ADC_CR0.SREF, 选择 ADC 的参考电压为内部 1.5V 或内部 2.5V。
- Step6: 配置 ADC_CR0.SAM 及 ADC_CR0.CLKSEL, 设置 ADC 的转换速度。
- Step7: 设置 ADC_CR0.SEL 为 0x0A, 选择待转换的通道为温度传感器的输出。
- Step8: 设置 ADC_CR0.BUFEN 为 1, 使能输入信号放大器。
- Step9: 设置 ADC_CR0.START 为 1, 启动 ADC 单次转换。
- Step10: 等待 ADC_CR0.START 变为 0, 读取 ADC_result 寄存器以获取 ADC 转换结果。
- Step11: 设置 ADC_CR0.ADCEN 及 BGR_CR.BGR_EN 为 0, 关闭 ADC 模块、BGR 模块。
- Step12: 读取温度传感器校准值, 根据公式计算当前的环境温度。

21.10 ADC 模块寄存器

基地址 0x40002400

寄存器	偏移地址	描述
ADC_CR0	0x004	ADC 配置寄存器0
ADC_CR1	0x008	ADC 配置寄存器1
ADC_CR2	0x00C	ADC 配置寄存器2
ADC_result0	0x030	ADC 通道0转换结果
ADC_result1	0x034	ADC 通道1转换结果
ADC_result2	0x038	ADC 通道2转换结果
ADC_result3	0x03C	ADC 通道3转换结果
ADC_result4	0x040	ADC 通道4转换结果
ADC_result5	0x044	ADC 通道5转换结果
ADC_result6	0x048	ADC 通道6转换结果
ADC_result7	0x04C	ADC 通道7转换结果
ADC_result 8	0x050	ADC 通道8转换结果
ADC_result_acc	0x054	ADC 转换结果累加值
ADC_HT	0x058	ADC 比较上阈值
ADC_LT	0x05C	ADC 比较下阈值
ADC_IFR	0x060	ADC 中断标志寄存器
ADC_ICLR	0x064	ADC 中断清除寄存器
ADC_result	0x068	ADC 转换结果

表 21-1 ADC 寄存器

21.10.1 ADC 配置寄存器 0 (ADC_CR0)

偏移地址 0x004

复位值 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			

位	标记	功能描述
31:16	Reserved	保留
15	StateRst	ADC连续转换状态控制 1: 复位ADC连续转换状态 0: 无效
14	IE	ADC中断控制 1: 使能中断 0: 禁止中断
13	Reserved	保留
12:11	SAM	ADC采样周期选择 00: 4个采样周期 01: 6个采样周期 10: 8个采样周期 11: 12个采样周期
10	BUFEN	ADC输入信号放大器控制 0: 关闭放大器, 外部输入信号与ADC直接相连。 1: 打开放大器, 外部输入信号通过放大器放大后与ADC相连, 用于高阻信号。
9:8	SREF	ADC参考电压选择 00: 内部1.5V 01: 内部2.5V 10: 外部参考电压ExRef (P3.6) 11: 电源电压
7:4	SEL	ADC转换通道选择 (单次转换模式) 0000: 选择通道0输入P2.4 0001: 选择通道1输入P2.6 0010: 选择通道2输入P3.2 0011: 选择通道3输入P3.3 0100: 选择通道4输入P3.4

		<p>0101: 选择通道5输入P3.5 0110: 选择通道6输入P3.6 0111: 选择通道7输入P0.1 1000: 选择通道8输入P0.2 1001: VCC/3 注: ADC_CR0.BUFEN必须为1 1010: 内置温度传感器输出电压 注: ADC_CR0.BUFEN必须为1 1011: 内部基准1.2V输出电压 注: ADC_CR0.BUFEN必须为1</p>
3:2	CLKSEL	<p>ADC时钟选择 00: PCLK时钟 01: PCLK时钟2分频 10: PCLK时钟4分频 11: PCLK时钟8分频</p>
1	START	<p>ADC转换控制 1: 启动ADC转换 0: 停止ADC转换</p>
0	ADCEN	<p>ADC使能控制 1: 使能ADC 0: 禁止ADC</p>

21.10.2 ADC 配置寄存器 1 (ADC_CR1)

偏移地址 0x008

复位值 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				

位	标记	功能描述
31:16	Reserved	保留
15	RACC_CLR	ADC转换结果累加寄存器清零 1: 无作用; 0: ADC转换结果累加寄存器 (ADC_result_acc) 清零。 注: 该bit读数为0, 故操作本寄存器时需要特别注意该bit的值以防误动作。
14	RegCmp	ADC区间比较控制 1: 使能区间比较 0: 禁止区间比较
13	HtCmp	ADC高阈值比较控制 1: 使能高阈值比较 0: 禁止高阈值比较
12	LtCmp	ADC低阈值比较控制 1: 使能低阈值比较 0: 禁止低阈值比较
11	RACC_EN	ADC转换结果自动累加控制 1: 使能ADC转换结果自动累加功能 0: 禁止ADC转换结果自动累加功能
10	CT	ADC转换模式选择 1: 连续转换模式 0: 单次转换模式
9:5	TRIGS1	ADC转换自动触发选择2: 00000: 禁用自动触发ADC转换 00001: Timer0中断, 自动触发ADC转换 00010: Timer1中断, 自动触发ADC转换 00011: Timer2中断, 自动触发ADC转换 00100: LPTimer中断, 自动触发ADC转换 00101: Timer4中断, 自动触发ADC转换

		<p>00110: Timer5中断, 自动触发ADC转换 00111: Timer6中断, 自动触发ADC转换 01000: UART0中断, 自动触发ADC转换 01001: UART1中断, 自动触发ADC转换 01010: LPUART中断, 自动触发ADC转换 01011: VCO中断, 自动触发ADC转换 01100: VC1中断, 自动触发ADC转换 01101: RTC中断, 自动触发ADC转换 01110: PCA中断, 自动触发ADC转换 01111: SPI中断, 自动触发ADC转换 10000: P01中断, 自动触发ADC转换 10001: P02中断, 自动触发ADC转换 10010: P03中断, 自动触发ADC转换 10011: P14中断, 自动触发ADC转换 10100: P15中断, 自动触发ADC转换 10101: P23中断, 自动触发ADC转换 10110: P24中断, 自动触发ADC转换 10111: P25中断, 自动触发ADC转换 11000: P26中断, 自动触发ADC转换 11001: P27中断, 自动触发ADC转换 11010: P31中断, 自动触发ADC转换 11011: P32中断, 自动触发ADC转换 11100: P33中断, 自动触发ADC转换 11101: P34中断, 自动触发ADC转换 11110: P35中断, 自动触发ADC转换 11111: P36中断, 自动触发ADC转换</p> <p>Note:</p> <p>1) TIM4/5/6中断触发ADC自动转换, 除了需要使能TIM4/5/6的相应中断外, 还需要配置Advanced Timer的展频及中断触发选择寄存器TIMX_CR选择可以触发ADC的中断源。</p> <p>2) 触发ADC使用的是各中断标志位的上升沿。如果需要重复触发, 需要清除中断标志。如果不需要进入中断服务程序, 请不要使能NVIC的中断使能。</p>
4:0	TRIGS0	<p>ADC转换自动触发选择2:</p> <p>00000: 禁用自动触发ADC转换 00001: 选择Timer0中断, 自动触发ADC转换 00010: 选择Timer1中断, 自动触发ADC转换 00011: 选择Timer2中断, 自动触发ADC转换 00100: 选择LPTimer中断, 自动触发ADC转换 00101: 选择Timer4中断, 自动触发ADC转换 00110: 选择Timer5中断, 自动触发ADC转换 00111: 选择Timer6中断, 自动触发ADC转换 01000: 选择UART0中断, 自动触发ADC转换</p>

		<p>01001: 选择UART1中断, 自动触发ADC转换 01010: 选择LPUART中断, 自动触发ADC转换 01011: 选择VCO中断, 自动触发ADC转换 01100: 选择VC1中断, 自动触发ADC转换 01101: 选择RTC中断, 自动触发ADC转换 01110: 选择PCA中断, 自动触发ADC转换 01111: 选择SPI中断, 自动触发ADC转换 10000: 选择P01中断, 自动触发ADC转换 10001: 选择P02中断, 自动触发ADC转换 10010: 选择P03中断, 自动触发ADC转换 10011: 选择P14中断, 自动触发ADC转换 10100: 选择P15中断, 自动触发ADC转换 10101: 选择P23中断, 自动触发ADC转换 10110: 选择P24中断, 自动触发ADC转换 10111: 选择P25中断, 自动触发ADC转换 11000: 选择P26中断, 自动触发ADC转换 11001: 选择P27中断, 自动触发ADC转换 11010: 选择P31中断, 自动触发ADC转换 11011: 选择P32中断, 自动触发ADC转换 11100: 选择P33中断, 自动触发ADC转换 11101: 选择P34中断, 自动触发ADC转换 11110: 选择P35中断, 自动触发ADC转换 11111: 选择P36中断, 自动触发ADC转换</p> <p>Note:</p> <p>1) TIM4/5/6中断触发ADC自动转换, 除了需要使能TIM4/5/6的相应中断外, 还需要配置Advanced Timer的展频及中断触发选择寄存器TIMX_CR选择可以触发ADC的中断源。</p> <p>2) 触发ADC使用的是各中断标志位的上升沿。如果需要重复触发, 需要清除中断标志。如果不需要进入中断服务程序, 请不要使能NVIC的中断使能。</p>
--	--	---

21.10.3 ADC 配置寄存器 2 (ADC_CR2)

偏移地址 0x00C

复位值 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							

位	标记	功能描述
31:16	Reserved	保留
15:8	ADCCNT	ADC连续转换次数配置 0: 连续转换1次 1: 连续转换2次 255: 连续转换256次
7	CH7EN	ADC连续转换通道7使能 1: 使能 0: 禁止
6	CH6EN	ADC连续转换通道6使能 1: 使能 0: 禁止
5	CH5EN	ADC连续转换通道5使能 1: 使能 0: 禁止
4	CH4EN	ADC连续转换通道4使能 1: 使能 0: 禁止
3	CH3EN	ADC连续转换通道3使能 1: 使能 0: 禁止
2	CH2EN	ADC连续转换通道2使能 1: 使能 0: 禁止
1	CH1EN	ADC连续转换通道1使能 1: 使能 0: 禁止
0	CH0EN	ADC连续转换通道0使能

		1: 使能 0: 禁止
--	--	----------------

21.10.4 ADC 通道 0 转换结果 (ADC_result0)

偏移地址 0x030

复位值 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				result0											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result0	ADC通道0转换结果

21.10.5 ADC 通道 1 转换结果 (ADC_result1)

偏移地址 0x034

复位值 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				result1											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result1	ADC通道1转换结果

21.10.6 ADC 通道 2 转换结果 (ADC_result2)

偏移地址 0x038

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result2	ADC 通道2转换结果

21.10.7 ADC 通道 3 转换结果 (ADC_result3)

偏移地址 0x03C

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result3	ADC 通道3转换结果

21.10.8 ADC 通道 4 转换结果 (ADC_result4)

偏移地址 0x040

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result4	ADC 通道4转换结果

21.10.9 ADC 通道 5 转换结果 (ADC_result5)

偏移地址 0x044

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result5	ADC 通道5转换结果

21.10.10 ADC 通道 6 转换结果 (ADC_result6)

偏移地址 0x048

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result6	ADC通道6转换结果

21.10.11 ADC 通道 7 转换结果 (ADC_result7)

偏移地址 0x04C

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result7	ADC 通道7转换结果

21.10.12 ADC 通道 8 转换结果 (ADC_result8)

偏移地址 0x050

复位值 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											
Reserved				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result8	ADC 通道8转换结果

21.10.13 ADC 转换结果累加值 (ADC_result_acc)

偏移地址 0x054

复位值 0x00000000

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

位	标记	功能描述
31:20	Reserved	保留
19:0	Result_acc	ADC转换累加值

21.10.14 ADC 比较上阈值 (ADC_HT)

偏移地址 0x058

复位值 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											
Reserved				R/W											

位	标记	功能描述
31:12	Reserved	保留
11:0	HT	ADC转换结果比较上阈值

21.10.15 ADC 比较下阈值 (ADC_LT)

偏移地址 0x05C

复位值 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											
Reserved				R/W											

位	标记	功能描述
31:12	Reserved	保留
11:0	LT	ADC转换结果比较下阈值

21.10.16 ADC 中断标志寄存器 (ADC_IFR)

偏移地址 0x060

复位值 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												CONT	REG_	HHT_	LLT_
												_INTF	INTF	INTF	INTF
												RO	RO	RO	RO

位	标记	功能描述
31:4	Reserved	保留
3	CONT_INTF	连续转换完成标志 1: ADC连续转换完成 0: ADC连续转换未完成
2	REG_INTF	ADC转换结果比较区间标志 1: ADC转换结果位于[ADC_LT , ADC_HT)区间内 0: ADC转换结果位于[ADC_LT , ADC_HT)区间外
1	HHT_INTF	ADC转换结果比较上阈值标志 1: ADC转换结果位于[ADC_HT, 4095]区间内 0: ADC转换结果位于[ADC_HT, 4095]区间外
0	LLT_INTF	ADC转换结果比较下阈值标志 1: ADC转换结果位于[0, ADC_LT)区间内 0: ADC转换结果位于[0, ADC_LT)区间外

21.10.17 ADC 中断清除寄存器 (ADC_ICLR)

偏移地址 0x064

复位值 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												CONT	REG_	HHT_	LLT_
												_INTC	INTC	INTC	INTC
												W0	W0	W0	W0

位	标记	功能描述
31:4	Reserved	保留
3	CONT_INTC	写0清除连续转换完成标志 写1无作用
2	REG_INTC	写0清除ADC转换结果比较区间标志 写1无作用
1	HHT_INTC	写0清除ADC转换结果比较上阈值 写1无作用
0	LLT_INTC	写0清除ADC转换结果比较下阈值标志 写1无作用

21.10.18 ADC 结果 (ADC_result)

偏移地址 0x068

复位值 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				result											
				RO											

位	标记	功能描述
31:12	Reserved	保留
11:0	result	ADC转换结果

22 模拟比较器（VC）

22.1 模拟电压比较器 VC 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“+”输入端电压高于“-”输入端电压时，电压比较器输出为高电平；当“+”输入端电压低于“-”输入端电压时，电压比较器输出为低电平。本产品内部集成的模拟电压比较器 VC 具有以下特性：

- 支持电压比较功能；
- 支持内部 64 阶 VCC 分压（使用分压来源电压需要大于 1.8V）
- 支持 8 个外部输入端口和片内 BGR 输出的参考电压作为电压比较器的输入；
- 支持三种软件可配置的中断触发方式：高电平触发/上升沿触发/下降沿触发；
- 电压比较器的输出可以作为 Base Timer 和 LPTimer 门控端口的输入；
- 电压比较器的输出可以作为 Advanced Timer 的刹车输入或者捕获输入；
- 支持在超低功耗模式下工作，电压比较器的中断输出可以将芯片从超低功耗模式下唤醒；
- 提供软件可配置的滤波时间以增强芯片的抗干扰能力。

22.2 电压比较器框架图

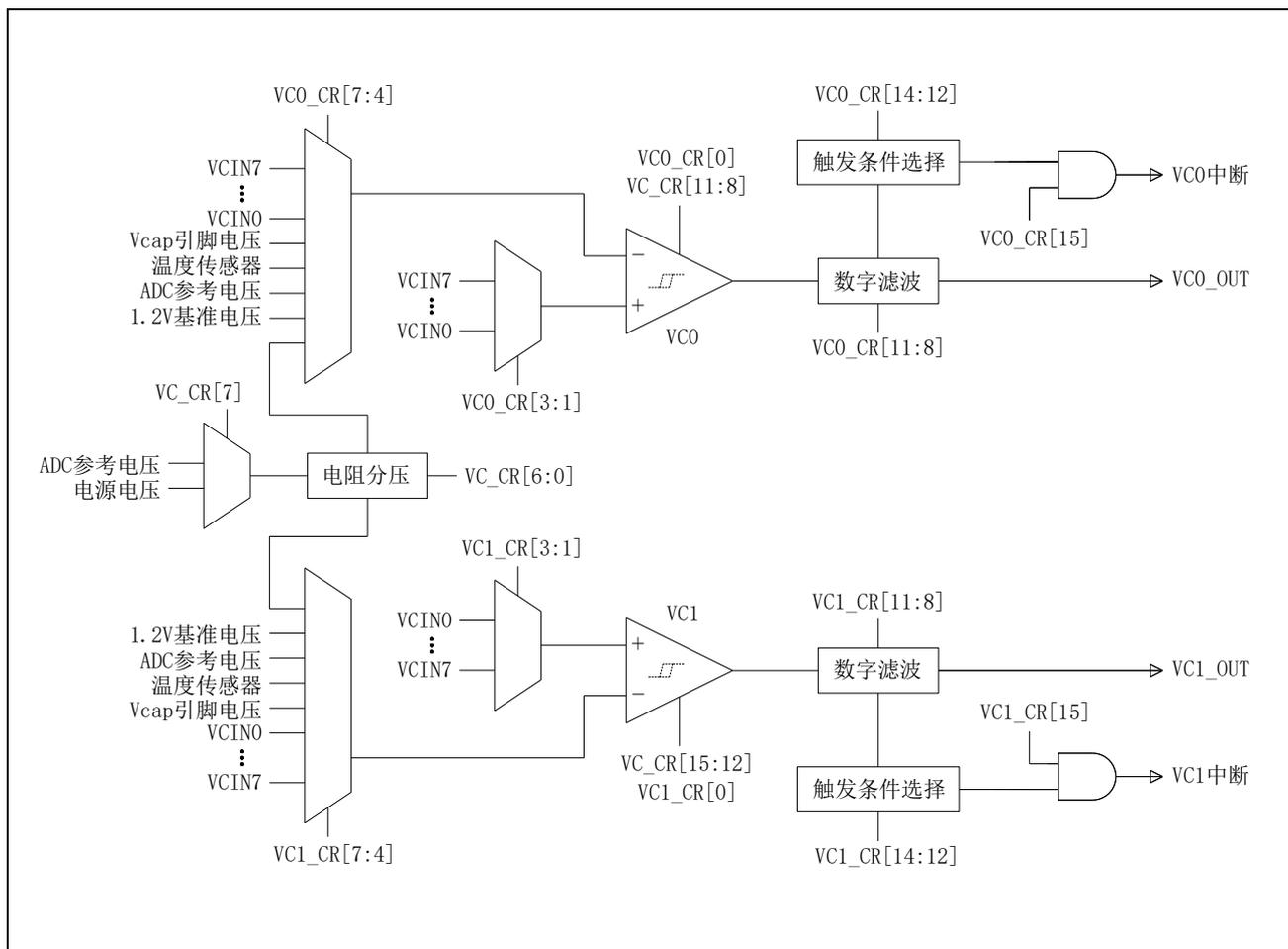


图 22-1 VC 框架图

22.3 建立/响应时间

当使用电压比较器时，从 VC 使能或者 VC 的两端输入电压变化到输出正确的结果的时间由 VC 控制寄存器（VC_CR）中的 BIAS_SEL 控制位决定。

如果选择温度传感器、1.2V 基准电压、ADC 模块参考电压作为比较器的端输入，则需要打开内部的 BGR 模块。内部 BGR 的启动时间大约为 20us，电压比较器需要等待内部 BGR 稳定后才能正常输出。

22.4 滤波时间

在电压比较器固有的建立/响应时间之外，用户可以设置更长的滤波时间来过滤掉系统噪声，比如马达停止时的大电流噪声。

从寄存器 VC_IFR.VCx_Filter 可以读出数字滤波后的信号电平；当 GPIO 功能配置为 VCx_OUT 时，数字滤波后的信号可以从 GPIO 输出以方便测量。

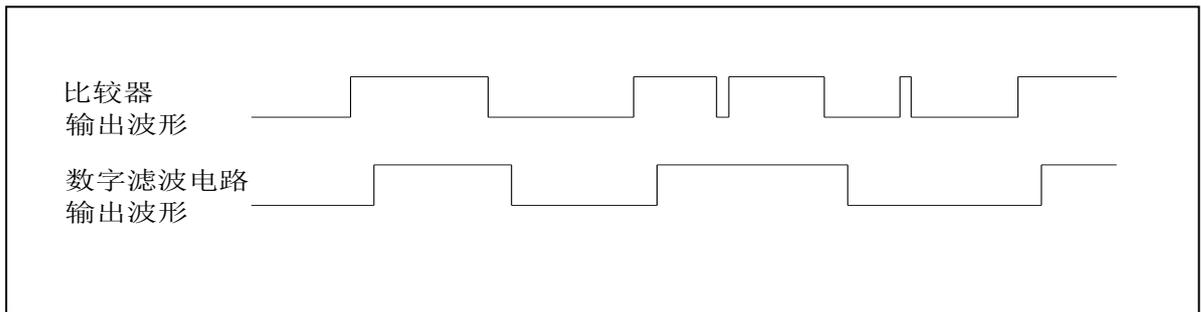


图 22-2 VC 滤波响应时间

22.5 迟滞功能

电压比较器可以选择迟滞功能，迟滞功能使能后的图示如下

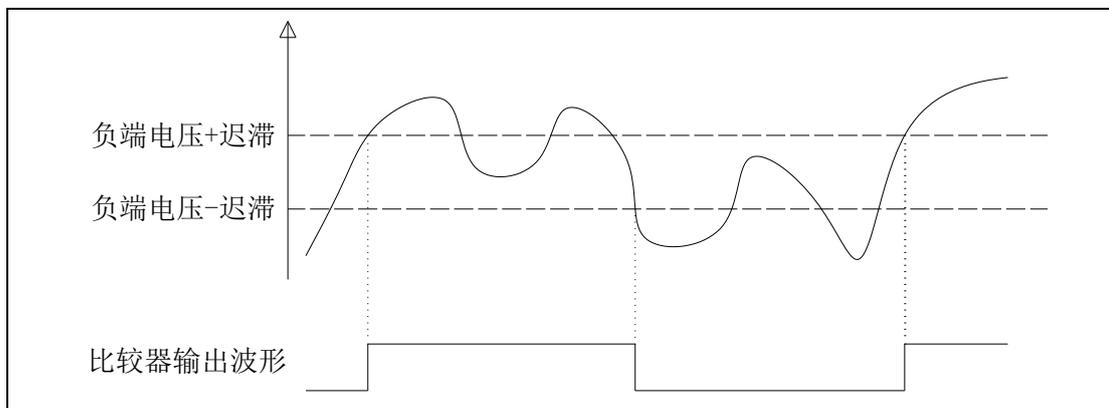


图 22-3 VC 迟滞功能

22.6 VC 寄存器

基地址 0x40002400

寄存器	偏移地址	描述
VC_CR	0x010	VC0/1 配置寄存器0
VC0_CR	0x014	VC0 配置寄存器
VC1_CR	0x018	VC1 配置寄存器
VC0_OUT_CFG	0x01C	VC0 输出配置寄存器
VC1_OUT_CFG	0x020	VC1 输出配置寄存器
VC_IFR	0x024	VC 中断寄存器

表 22-1 VC 寄存器

22.6.1 VC 配置寄存器 (VC_CR)

偏移地址 0x010

复位值 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_REF2P5_SEL	VC_DI_V_EN	VC_DIV										
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									

位	标记	功能描述
31:16	Reserved	保留
15:14	VC1_HYS_SEL	VC1 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
13:12	VC1_BIAS_SEL	VC1 功耗选择 (功耗越大, 响应速度越快) 00:300nA 01:1.2uA 10:10uA(需要电源电压不小于2.8V, 需要手动开启BGR) 11:20uA(需要电源电压不小于2.8V, 需要手动开启BGR) 注: BGR启动时间大约30us
11:9	VC0_HYS_SEL	VC0 迟滞选择: 00:没有迟滞 01:迟滞电压大约10mV 10:迟滞电压大约20mV 11:迟滞电压大约30mV
9:8	VC0_BIAS_SEL	VC0 功耗选择 (功耗越大, 响应速度越快) 00:300nA 01:1.2uA 10:10uA(需要电源电压不小于2.8V, 需要手动开启BGR) 11:20uA(需要电源电压不小于2.8V, 需要手动开启BGR) 注: BGR启动时间大约30us
7	VC_REF2P5_SEL	VC_DIV 参考电压Vref选择 0:VCC 1: ADC_CR0.SREF所选择的参考电压
6	VC_DIV_EN	6位 DAC 使能

		1: 使能
5:0	VC_DIV	6位 DAC 配置 000000: 1/64 Vref 000001: 2/64 Vref 000010: 3/64 Vref 000011: 4/64 Vref ... 111110: 63/64 Vref 111111: Vref

22.6.2 VC0 配置寄存器 (VC0_CR)

偏移地址 0x014

复位值 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			

位	标记	功能描述
31:16	Reserved	保留
15	IE	VC 中断使能 1: 使能; 0: 禁止
14	level	VC 输出信号高电平触发 INT flag
13	rising	VC 输出信号上升沿触发 INT flag
12	Falling	VC 输出信号下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111: 滤波时间大约为28.8ms 110: 滤波时间大约为7.2ms 101: 滤波时间大约为1.8ms 100: 滤波时间大约为450us 011: 滤波时间大约为112us 010: 滤波时间大约为28us 001: 滤波时间大约为14us 000: 滤波时间大约为7us 注意: 滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1: 启动 VC 滤波 0: VC 无滤波
7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input P2.3 0001: select channel 1 input P2.5 0010: select channel 2 input P3.2 0011: select channel 3 input P3.3 0100: select channel 4 input P3.4 0101: select channel 5 input P3.5 0110: select channel 6 input P3.6 0111: select channel 7 input P0.1 1000:电阻分压输出电压

		<p>1001:内置温度传感器输出电压</p> <p>1010:内部基准1.2V输出电压</p> <p>1011:ADC模块的参考电压</p> <p>1100:VCAP引脚的电压</p>
3:1	P_SEL	<p>电压比较器“+”端输入选择</p> <p>000: select channel 0 input P2.3</p> <p>001: select channel 1 input P2.5</p> <p>010: select channel 2 input P3.2</p> <p>011: select channel 3 input P3.3</p> <p>100: select channel 4 input P3.4</p> <p>101: select channel 5 input P3.5</p> <p>110: select channel 6 input P3.6</p> <p>111: select channel 7 input P0.1</p>
0	EN	<p>电压比较器使能</p> <p>1: 使能电压比较器</p> <p>0: 关闭电压比较器</p>

22.6.3 VC1 配置寄存器 (VC1_CR)

偏移地址 0x018

复位值 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	

位	标记	功能描述
31:16	Reserved	保留
15	IE	VC 中断使能 1: 使能; 0: 禁止
14	level	VC 输出信号触发中断选择 1: 使能高电平触发 INT flag 0: 禁止高电平触发 INT flag
13	rising	VC 输出信号触发中断选择 1: 使能上升沿触发 INT flag 0: 禁止上升沿触发 INT flag
12	falling	VC 输出信号触发中断选择 1: 使能下降沿触发 INT flag 0: 禁止下降沿触发 INT flag
11:9	debounce_time	VC 输出滤波时间配置 111: 滤波时间大约为28.8ms 110: 滤波时间大约为7.2ms 101: 滤波时间大约为1.8ms 100: 滤波时间大约为450us 011: 滤波时间大约为112us 010: 滤波时间大约为28us 001: 滤波时间大约为14us 000: 滤波时间大约为7us 注意: 滤波时间的配置只有在 FLTEN=1 时才有效。
8	FLTEN	1: 启动 VC 滤波 0: VC 无滤波
7:4	N_SEL	电压比较器“-”端输入选择 0000: select channel 0 input P2.3 0001: select channel 1 input P2.5 0010: select channel 2 input P3.2

		<p>0011: select channel 3 input P3.3 0100: select channel 4 input P3.4 0101: select channel 5 input P3.5 0110: select channel 6 input P3.6 0111: select channel 7 input P0.1 1000: 电阻分压输出电压 1001: 内置温度传感器输出电压 1010: 内部基准1.2V输出电压 1011: ADC模块的参考电压 1100: VCAP引脚的电压</p>
3:1	P_SEL	<p>电压比较器“+”端输入选择 000: select channel 0 input P2.3 001: select channel 1 input P2.5 010: select channel 2 input P3.2 011: select channel 3 input P3.3 100: select channel 4 input P3.4 101: select channel 5 input P3.5 110: select channel 6 input P3.6 111: select channel 7 input P0.1</p>
0	EN	<p>电压比较器使能 1: 使能电压比较器 0: 关闭电压比较器</p>

22.6.4 VC0 输出配置寄存器 (VC0_OUT_CFG)

偏移地址 0x01C

复位值 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	PCAECI	PCACAP0	INV_PCA	TM3E CLK	LPTIMER3 MG	TIM2 G	TIM1 G	TIM0 G	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

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC0 作为 Advanced Timer 刹车控制 1: 使能; 0: 禁止。
14	TIM6	VC0 filter 结果输出到 TIM6 捕获输入使能 1: 使能; 0: 禁止。
13	INV_TIM6	VC0 filter 结果输出到 TIM6 反向使能 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
12	TIM5	VC0 filter 结果输出到 TIM5 捕获输入使能 1: 使能; 0: 禁止。
11	INV_TIM5	VC0 filter 结果输出到 TIM5 反向使能 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
10	TIM4	VC0 filter 结果输出到 TIM4 捕获输入使能 1: 使能; 0: 禁止。
9	INV_TIM4	VC0 filter 结果输出到 TIM4 反向使能 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
8	PCAECI	VC0 filter 结果输出到 PCA 外部时钟使能控制 1: 使能; 0: 禁止。
7	PCACAP0	VC0 filter 结果输出到 PCA 捕获0使能控制 1: 使能; 0: 禁止。
6	INV_PCA	VC0 filter 结果输出负向到 PCA 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
5	LPTIMECLK	VC0 filter 结果输出到 LPTIMER 外部时钟使能控制 1: 使能; 0: 禁止。
4	LPTIMG	VC0 filter 结果输出到 LPTIMER3 GATE 使能控制 1: 使能; 0: 禁止。
3	TIM2G	VC0 filter 结果输出到 TIM2 GATE 使能控制 1: 使能; 0: 禁止。

2	TIM1G	VC0 filter 结果输出到 TIM1 GATE 使能控制 1: 使能; 0: 禁止。
1	TIM0G	VC0 filter 结果输出到 TIM0 GATE 使能控制 1: 使能; 0: 禁止。
0	INV_Timer	VC0 filter 结果输出负向到各 TIM0/1/2, LPTimer 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。

22.6.5 VC1 输出配置寄存器 (VC1_OUT_CFG)

偏移地址 0x020

复位值 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 ECI	PCACAP1	INV_PCA	TM3E CLK	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

位	标记	功能描述
31:16	Reserved	保留
15	brake	VC1 作为 Advanced Timer 刹车控制 1: 使能; 0: 禁止。
14	TIM6	VC1 filter 结果输出到 TIM6 捕获输入使能 1: 使能; 0: 禁止。
13	INV_TIM6	VC1 filter 结果输出到 TIM6 反向使能 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
12	TIM5	VC1 filter 结果输出到 TIM5 捕获输入使能 1: 使能; 0: 禁止。
11	INV_TIM5	VC1 filter 结果输出到 TIM5 反向使能 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
10	TIM4	VC1 filter 结果输出到 TIM4 捕获输入使能 1: 使能; 0: 禁止。
9	INV_TIM4	VC1 filter 结果输出到 TIM4 反向使能 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。
8	PCA ECI	VC1 filter 结果输出到 PCA 外部时钟使能控制 1: 使能; 0: 禁止。
7	PCACAP1	VC1 filter 结果输出到 PCA 捕获1使能控制 1: 使能; 0: 禁止。
6	INV_PCA	VC1 filter 结果输出负向到 PCA 1: 使能反向; 0: 禁止反向, 输入与VC输出同向。
5	LPTIMECLK	VC1 filter 结果输出到 LPTIMER 外部时钟使能控制 1: 使能; 0: 禁止。
4	LPTIMG	VC1 filter 结果输出到 LPTIMER3 GATE 使能控制 1: 使能; 0: 禁止。
3	TIM2G	VC1 filter 结果输出到 TIM2 GATE 使能控制 1: 使能; 0: 禁止。

2	TIM1G	VC1 filter 结果输出到 TIM1 GATE 使能控制 1: 使能; 0: 禁止。
1	TIM0G	VC1 filter 结果输出到 TIM0 GATE 使能控制 1: 使能; 0: 禁止。
0	INV_Timer	VC1 filter 结果输出负向到各 TIM0/1/2, LPTimer 1: 使能反向; 0: 禁止反向, 输入与 VC 输出同向。

22.6.6 VC 中断寄存器 (VC_IFR)

偏移地址 0x024

复位值 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

位	标记	功能描述
31:4	Reserved	保留
3	VC1_Filter	VC1 Filter 后的状态
2	VC0_Filter	VC0 Filter 后的状态
1	VC1_INTF	VC1 中断标志, 1发生 VC1 中断; 0未发生中断; 写0清除中断标志, 写1无效
0	VC0_INTF	VC0 中断标志, 1发生 VC0 中断; 0未发生中断; 写0清除中断标志, 写1无效

23 低电压检测器 (LVD)

23.1 LVD 简介

LVD 可用于监测 VCC 及芯片引脚的电压。当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号，用户可根据该信号执行一些紧急任务。

LVD 具有以下特性：

- 4 路监测源，VCC、P03、P23、P25；
- 16 阶阈值电压，灵活多用；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

23.2 LVD 框图

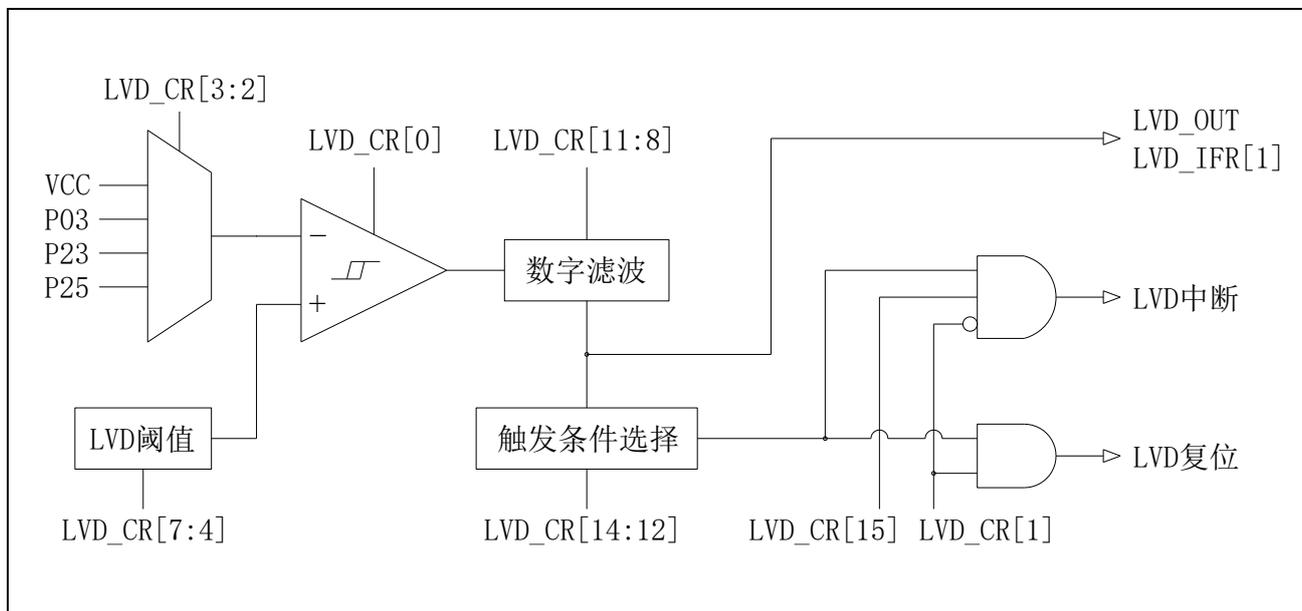


图 23-1 LVD 框图

23.3 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.Debounce_time 的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。

从寄存器 LVD_IFR[1]可以读出数字滤波后的信号电平；当GPIO功能配置为LVD_OUT时，数字滤波后的信号可以从GPIO输出以方便测量。

使能数字滤波模块，滤波示意如下所示：

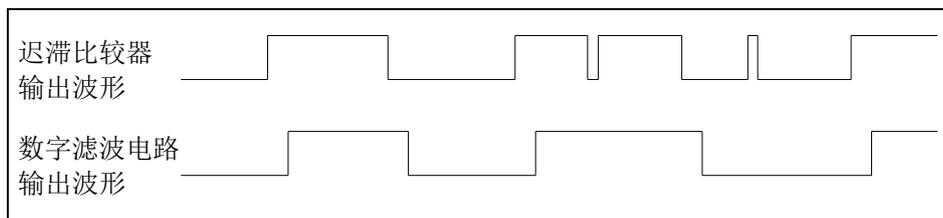


图 23-2 LVD 滤波输出

23.4 迟滞功能

LVD 内置的电压比较器具有迟滞功能，其输出信号会等到输入信号高于或低于阈值电压 20mV 后才发生翻转。迟滞功能可以增强芯片的抗干扰能力，如下图所示。

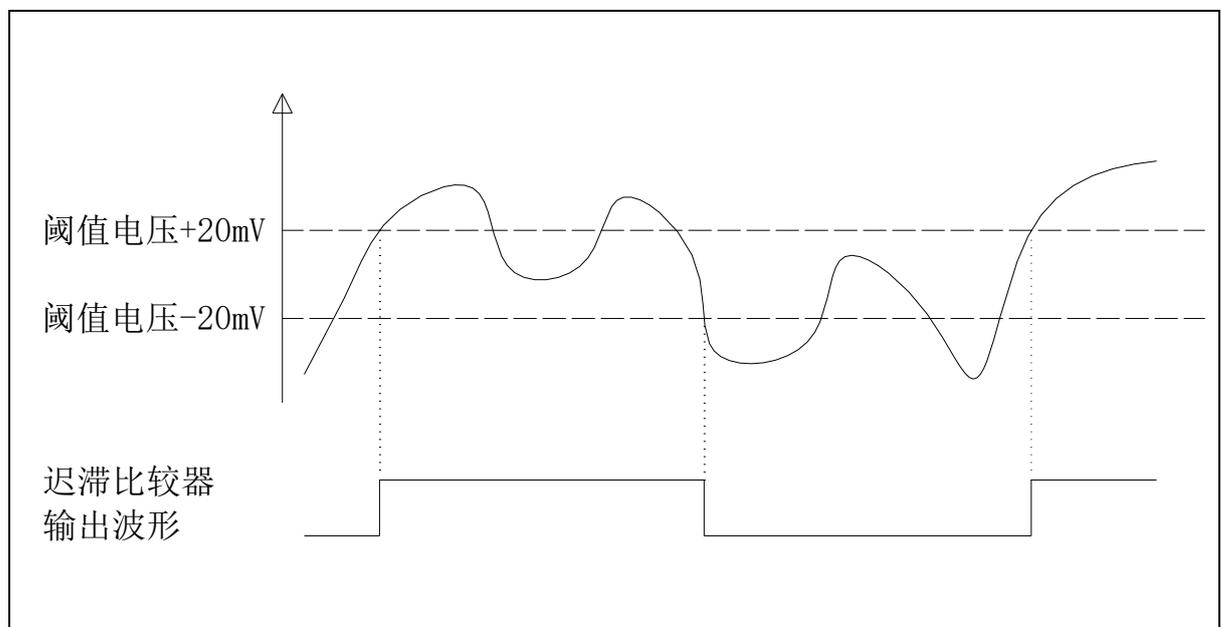


图 23-3 LVD 迟滞响应

23.5 配置示例

23.5.1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。

配置方法如下所示：

Step1: 配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2: 配置 LVD_CR.VTDS，选择 LVD 的阈值电压。

Step3: 配置 LVD_CR.Debounce_time，选择 LVD 滤波时间。

Step4: 配置 LVD_CR.FLTEN，使能 LVD 滤波。

Step5: 设置 LVD_CR.HTEN 为 1，选择高电平触发 LVD 动作。

Step6: 设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位。

Step7: 设置 LVD_CR.LVDEN 为 1，使能 LVD。

23.5.2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。

配置方法如下所示：

Step1: 配置 LVD_CR.Source_sel，选择待监测的电压来源。

Step2: 配置 LVD_CR.VTDS，选择 LVD 的阈值电压。

Step3: 配置 LVD_CR.Debounce_time，选择 LVD 滤波时间。

Step4: 配置 LVD_CR.FLTEN，使能 LVD 滤波。

Step5: 设置 LVD_CR.RTEN 和 LVD_CR.FTEN 为 1，选择电平变化触发 LVD 动作。

Step6: 设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断。

Step7: 设置 LVD_CR.IE 为 1，使能 LVD 中断。

Step8: 使能 NVIC 中断向量表中的 LVD 中断。

Step9: 设置 LVD_CR.LVDEN 为 1，使能 LVD。

Step10: 在中断服务程序中向 LVD_IFR 写入 0x00 以清除中断标志。

23.6 LVD 寄存器

基地址 0x40002400

寄存器	偏移地址	描述
LVD_CR	0x028	LVD 配置寄存器
LVD_IFR	0x02C	LVD 中断标志寄存器

表 23-1 LVD 寄存器

23.6.1 LVD 配置寄存器 (LVD_CR)

偏移地址 0x028

复位值 0x00000100

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	

位	标记	功能描述
31:16	Reserved	保留
15	IE	LVD中断使能 1: 使能; 0: 禁止。
14	HTEN	高电平触发使能 (被监测电压低于阈值电压) 1: 使能; 0: 禁止。
13	RTEN	上升沿触发使能 (被监测电压从高于阈值电压变为低于阈值电压) 1: 使能; 0: 禁止。
12	FTEN	下降沿触发使能 (被监测电压从低于阈值电压变为高于阈值电压) 1: 使能; 0: 禁止。
11:9	Debounce_time	数字滤波时间配置 111: 滤波时间大约为28.8ms 110: 滤波时间大约为7.2ms 101: 滤波时间大约为1.8ms 100: 滤波时间大约为450us

		<p>011: 滤波时间大约为112us 010: 滤波时间大约为28us 001: 滤波时间大约为14us 000: 滤波时间大约为7us 注: 滤波时间仅在FLTEN为1时有效。</p>
8	FLTEN	<p>数字滤波功能配置 1: 使能数字滤波 0: 禁止数字滤波</p>
7:4	VTDS	<p>LVD 监测电压选择 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</p>
3:2	Source_sel	<p>LVD 监测来源选择 11: P2.5端口输入电压 10: P2.3端口输入电压 01: P0.3端口输入电压 00: VCC 电源电压</p>
1	ACT	<p>LVD触发动作选择 1: 系统复位 0: NVIC中断</p>
0	LV DEN	<p>LVD控制 1: 使能LVD 0: 禁止LVD</p>

23.6.2 LVD 中断寄存器 (LVD_IFR)

偏移地址 0x02C

复位值 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	INTF
														RO	R/W

位	标记	功能描述
31:2	Reserved	保留
1	LVD_Filte	LVD Filter 后的状态
0	INTF	LVD中断标志： 1: 发生LVD中断； 0: 未发生中断； 写0清除中断标志，写1无效。

24 模拟其它寄存器

基地址 0x40002400

寄存器	偏移地址	描述
BGR_option	0x078	BGR 控制寄存器

24.1 BGR 配置寄存器 (BGR_CR)

偏移地址 0x000

复位值 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_	BGR
														EN	_EN					
														R/W	R/W					

位	标记	功能描述
31:2	Reserved	保留
1	TS_EN	内部温度传感器使能 1: 启动内部温度传感器 0: 禁用内部温度传感器 注意: TS 需要大约 20us 的启动时间才能稳定。
0	BGR_EN	BGR 使能控制 1: 使能 BGR 0: 禁止 BGR 注意: 1) PERI_CLKEN.ADC 为 1 时, 才可以操作此寄存器。 2) BGR 使能 20us 后, 才能输出稳定的高精度参考电压。BGR 稳定后才可以被其它模块使用, 故用户操作中应加入等待 BGR 稳定的步骤。 3) 当使用 ADC 时, 必须使能 BGR。 4) 当使用 VC 时, 需根据 VC 寄存器的配置决定是否使能 BGR。

25 SWD 调试接口

本系列使用 ARM Cortex-M0+内核，该内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。当 HC32L110 微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

注意：

- SWD 在 DeepSleep 模式下不能工作，请在 Active 和 Sleep 模式下进行调试操作。

25.1 SWD 调试附加功能

本产品使用了 ARM Cortex-M0+ CPU，该内核包含用于高级调试功能的硬件扩展，因此本产品所拥有的调试功能与 Cortex-M0+一致。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

当调试主机与 MCU 相连并进行调试时，将使用调试功能。

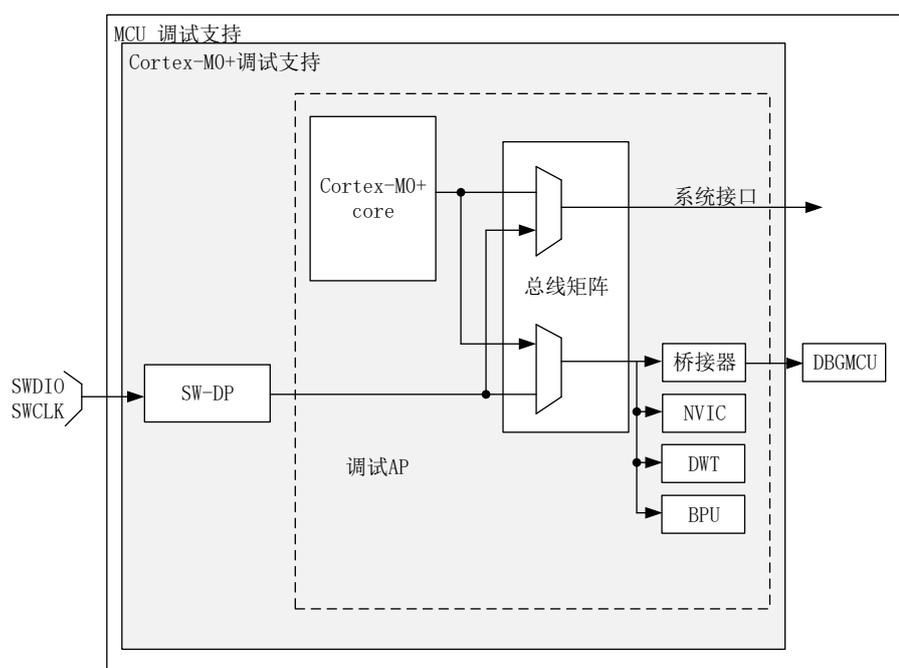


图 25-1 调试支持框图

Cortex®-M0+ 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex®-M0+内核提供集成片上调试支持。它包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发

注：

- 有关 ARM® Cortex®-M0+ 内核支持的调试功能的详细信息，请参见 Cortex®-M0+ 技术参考手册。

25.2 ARM® 参考文档

- Cortex®-M0+技术参考手册 (TRM)
可从 www.infocenter.arm.com 获取。
- ARM® 调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册

25.3 调试端口引脚

25.3.1 SWD 端口引脚

本系列的 SWD 接口，需要用到 2 个引脚，如下表所示。

SWD 端口名称	调试功能	引脚分配
SWCLK	串行时钟	P31
SWDIO	串行数据输入/输出	P27

25.3.2 SW-DP 引脚分配

如果烧录程序时使能了【加密芯片】选项，则上电后 SWD 调试功能被禁止。如果烧录程序时没有使能【加密芯片】选项，则上电后 P27/P31 引脚均被初始化为可被调试器使用的专用引脚。用户可设置 SYSCTRL1.SWD_USE_IO 寄存器来禁止 SWD 引脚的调试功能，SWD 引脚将被释放以用作普通 GPIO。SWD 引脚的配置与功能小结如下表所示：

【加密芯片】选项	SWD_USE_IO 配置	P27/P31 功能
加密	0	NA
加密	1	GPIO
不加密	0	SWD
不加密	1	GPIO

25.3.3 SWD 引脚上的内部上拉

用户软件释放 SW I/O 后，GPIO 控制器便会控制这些引脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入上拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

25.4 SWD 端口

25.4.1 SWD 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（ARM® 建议采用 100k）。这些上拉电阻可在内部配置。无需外部上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

25.4.2 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

位	名称	说明
0	启动	必须为1
1	APnDP	0: DP 访问; 1: AP 访问
2	RnW	0: 写请求; 1: 读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M0+ TRM。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动。

位	名称	说明
0	ACK	001: FAULT 010: WAIT 100: OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

位	名称	说明
0:31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

25.4.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 ARM® 代码，设置为 **0x0BB11477**（相当于 Cortex®-M0+）。

注意：

- 在目标读取此 ID 代码前，SW-DP 状态机是不工作的。
- 在上电复位后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机必须首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 *Cortex®-M0+ TRM* 和 *CoreSight 设计套件 r1p0TRM*。

25.4.4 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。

- 每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

25.4.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器：

A[3:2]	RW	SELECT 寄存器的 CTRLSEL 位	寄存器	注释
00	读取		IDCODE	制造商代码设置为 Cortex [®] -M0+ 的默认 ARM [®] 代码。 0x0BB11477 （标识 SW-DP）
00	写		ABORT	
01	读/写	0	DP-CTRL/STAT	目的： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的持续 时间）
10	读取		READ RESEND	允许从已损坏的调试软件传输中恢复读取数据， 无需重复执行原始 AP 传输。
10	写		SELECT	用于选择当前访问端口和活动的 4 字寄存器窗口
11	读/写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在 执行下个 AP 事务时提供读取 AP 请求的结果）。 此读取缓冲区捕获 AP 中的数据，显示为

前一次读取的结果，无需启动新操作。

25.4.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器：

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

地址	A[3:2] 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： <ul style="list-style-type: none"> – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
0x8	10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 <ul style="list-style-type: none"> – 位 31:24：APSEL：选择当前 AP (select the current AP) – 位 23:8：保留 – 位 7:4：APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 – 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果

25.5 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0+ TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C_DEBUGEN)

25.6 BPU（断点单元）

Cortex®-M0+ BPU 实现提供四个断点寄存器。

25.6.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 ARMv6-M ARM®和 ARM® CoreSight 组件技术参考手册。

25.7 DWT（数据观察点）

Cortex®-M0+ DWT 实现提供了两个观察点寄存器组。

25.7.1 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 ARMv6-M ARM® 中所述。

25.7.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器(DWT_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 ARMv6-M ARM®。

Cortex®-M0+ DWT_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

25.8 MCU 调试组件 (DBG)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

25.8.1 对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

25.8.2 对定时器、看门狗的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

25.9 调试模式模块工作状态控制 (DEBUG_ACTIVE)

复位值 0x0000FFF(仅在 SWD 调试模式下, 此寄存器设置才起作用)

偏移地址: 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

位	标记	功能描述
31:12	Reserved	保留
11	LPTIM	调试时, Timer3 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer3 计数功能 0: 在 SWD 调试界面下, Timer3 正常计数功能
10	Reserved	保留
9	RTC	调试时, RTC 计数功能配置 1: 在 SWD 调试界面下, 暂停 RTC 计数功能 0: 在 SWD 调试界面下, RTC 正常计数功能
8	WDT	调试时, WDT 计数功能配置 1: 在 SWD 调试界面下, 暂停 WDT 计数功能 0: 在 SWD 调试界面下, WDT 正常计数功能
7	PCA	调试时, PCA 计数功能配置 1: 在 SWD 调试界面下, 暂停 PCA 计数功能 0: 在 SWD 调试界面下, PCA 正常计数功能
6	TIM6	调试时, Timer6 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能 0: 在 SWD 调试界面下, Timer 正常计数功能
5	TIM5	调试时, Timer5 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能 0: 在 SWD 调试界面下, Timer 正常计数功能
4	TIM4	调试时, Timer4 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能 0: 在 SWD 调试界面下, Timer 正常计数功能
3	LPTIM	调试时, LpTimer 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能 0: 在 SWD 调试界面下, Timer 正常计数功能
2	TIM2	调试时, Timer2 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能

		0: 在 SWD 调试界面下, Timer 正常计数功能
1	TIM1	调试时, Timer1 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能 0: 在 SWD 调试界面下, Timer 正常计数功能
0	TIM0	调试时, Timer0 计数功能配置 1: 在 SWD 调试界面下, 暂停 Timer 计数功能 0: 在 SWD 调试界面下, Timer 正常计数功能

26 器件电子签名

电子签名存放在闪存存储器模块的系统存储区域，可以通过 SWD 或者 CPU 读取。它包含的芯片识别信息在出厂时编写，用户固件或者外部设备可以读取电子签名，用以自动匹配不同配置的 HC32Fxxx / HC32Lxxx 微控制器。

26.1 产品唯一身份标识（UID）寄存器（80 位）

唯一身份标识符典型应用场景：

- 用作序列号
- 在对内部 Flash 进行编程前将 UID 与软件加密原语和协议结合使用时用作安全密钥以提高 Flash 中代码的安全性
- 激活安全自举过程等

80 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。80 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

基址：0x0010 0E74

偏移地址	描述	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	固定值 FFH	Reserved							
3	固定值 FFH	Reserved							
4	固定值 00H	UID[23:16]							
5	固定值 00H	UID[31:24]							
6	Wafer Number	UID[39:32]							
7	Y Coordinate on the wafer	UID[47:40]							
8	Wafer Lot Number	UID[55:48]							
9		UID[63:56]							
10		UID[71:64]							
11		UID[79:72]							

26.2 产品型号寄存器

0x0010 0C60 ~ 0x0010 0C6F 存储了产品型号的 ASCII 码。如产品型号不足 16 字节，则以 0x00 进行填充。

例：484333324C3133364B38544100000000 所代表的产品型号为 HC32L136K8TA。

26.3 FLASH 容量寄存器

基地址：0x0010 0C70

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

FlashSize[31:16]															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FlashSize[15:0]															
R															

位	标记	功能描述
31:0	FlashSize	产品内置 Flash 的容量，以字节为单位 0x00008000 代表 Flash 容量为 32K Byte

26.4 RAM 容量寄存器

基地址：0x0010 0C74

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

RamSize[31:16]															
R															

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

RamSize[15:0]															
R															

位	标记	功能描述
31:0	RamSize	产品内置 RAM 的容量，以字节为单位 0x00000800 代表 RAM 容量为 2K Byte

26.5 管脚数量寄存器

基地址：0x0010 0C7A

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

PinCount[15:0]															
R															

位	标记	功能描述
15:0	PinCount	产品管脚的数量，以只为单位 0x0020 代表产品管脚数量为 32

27 附录 A SysTick 定时器

27.1 SysTick 定时器简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0 处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询溢出标志。

27.2 设置 SysTick

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

Step1: 配置 SysTick->CTRL. ENABLE 为 0，禁止 SysTick。

Step2: 配置 SysTick->CTRL. CLKSOURCE 及 SYSTICK_CR[27:25]，选择 SysTick 的时钟源。

Step3: 配置 SysTick->LOAD，选择 SysTick 的溢出周期。

Step4: 向 SysTick->VAL 写入任意值，清零 SysTick->VAL 及 SysTick->CTRL. COUNTFLAG。

Step5: 配置 SysTick->CTRL. TICKINT 为 1，使能 SysTick 中断。

Step6: 配置 SysTick->CTRL. ENABLE 为 1，使能 SysTick。

Step7: 在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。

注：Systick 溢出周期为 SysTick->LOAD+1，配置示例如下：

时钟源	SysTick->LOAD	溢出周期
RCH 4M	3999	1ms
XTL 32.768K	327	10.01ms

27.3 SysTick 寄存器

地址	名称	CMSIS 符号	全名
0XE000E010	SYS_CSR	SysTick->CTRL	SysTick控制和状态寄存器
0XE000E014	SYS_RVR	SysTick->LOAD	SysTick重载值寄存器
0XE000E018	SYS_CVR	SysTick->VAL	SysTick当前值寄存器

27.3.1 SysTick 控制和状态寄存器 (CTRL)

位	符号	功能描述	类型	复位值
31:17	Reserved	-	-	-
16	COUNTFLAG	Systick定时器溢出标志 1: Systick定时器发生下溢出。 0: Systick定时器未发生溢出。 读该寄存器, 可清除 COUNTFLAG 标志	RO	0
15:3	Reserved	-	-	-
2	CLKSOURCE	SysTick时钟源选择 1: 内核时钟(HCLK) 0: 参考时钟, 由SYSTICK_CR[27:25]决定	R/W	0
1	TICKINT	SysTick中断使能 1: 使能中断 0: 禁止中断	R/W	0
0	ENABLE	SysTick定时器使能 1: 使能SysTick 0: 禁止 SysTick	R/W	0

27.3.2 SysTick 重载寄存器 (LOAD)

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	RELOAD	SysTick 定时器重载值	RW	-

27.3.3 SysTick 当前值寄存器 (VAL)

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	CURRENT	读取该寄存器，获取SysTick定时器的当前计数值 写任意值到该寄存器，清零该寄存器及COUNTFLAG	RW	-

28 附录 B 文档约定

28.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

RW：读写，软件可以读写这些位。

RO：只读，软件只能读取这些位。

WO：只写，软件只能写入该位。读取该位时将返回无效数据。

W1：只写 1，硬件自动清 0，写 0 无效

R0W1：软件读取该位为 0，写入 1 将该位清零。写入 0 对该位的值无影响。

RW0：软件可以读写该位，写 1 无效，写 0 清除

R1W0：软件读取该位为 1，写入 0 将该位清零。写入 1 对该位的值无影响。

RC：软件可以读取该位。读取该位时，将自动清零。

写入“0”对该位的值无影响。

Res, Reserverd：保留位，必须保持复位值。

28.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

Word：32 位数据。

Half Word：16 位数据。

Byte：8 位数据。

IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。

ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。

AHB：高级高性能总线。

APB：低速外设总线。

DMA：直接存储器访问。

TIM：定时器

版本记录 & 联系方式

版本	修订日期	修订内容摘要
Rev1.0	2018/1/23	HC32L110系列参考手册初版发布。
Rev1.1	2018/5/4	版本更新，修正Flash数据，修改第4章部分描述。
Rev1.2	2018/5/23	修改了时钟切换流程，增加PCA比较捕获功能模块设置。
Rev1.3	2018/11/1	补充第1章功能模块描述，增加第2章描述引脚配置和功能，补充第9章FLASH操作描述，修改第28章电气特性参数，增加第29章和第31章。
Rev1.4	2018/11/26	修改名称：UART2→LPUART。第2.1和2.2节增加“注”。
Rev1.5	2019/2/22	修正以下数据：①ADC特性 ②ESD特性 ③存储器特性中ECFLASH最小值 ④增加封装尺寸 ⑤引脚配置图中加入AVCC/AVSS。
Rev1.6	2019/7/5	修正以下数据：①更正UID地址 ②更正编程模式 ③更新QFN引脚配置图样式。
Rev1.7	2019/12/13	修正以下数据：①典型应用电路图 ②LVD 框图 ③ADC特性单位 ④电压比较器框架图 ⑤标志位寄存器（UARTx_ISR） ⑥外部时钟源特性中XTH和XTL配图与注意事项 ⑦更新I2C 总线（I2C）、串行外设接口（SPI）、通用同步异步收发器（UART）和低功耗同步异步收发器（LPUART）部分描述。
Rev1.8	2020/1/17	修正以下数据：①增加CSP16封装 ②器件电子签名 ③附录A SysTick 定时器 ④循环冗余校验（CRC） ⑤系统控制寄存器1（SYSCTRL1）增加注意项 ⑥CR寄存器（FLASH_CR）复位值 ⑦I2C配置寄存器（I2C_CR）的0位。
Rev1.9	2020/3/6	修正以下数据：①编程模式中增加注意项 ②17.5.1和17.5.2的Step8。
Rev2.0	2020/4/30	修正以下数据：①ADC特性增加VCC/3精度 ②外部时钟源特性中修正笔误 ③内部时钟源特性中RCL振荡器精度。
Rev2.1	2020/5/29	修正以下数据：①17.5中增加Step2和Step3 ②I2Cx改为I2C。
Rev2.2	2020/7/31	修正以下数据：①增加TIM定时器特性和通信接口节 ②EFT等级 ③通用工作条件中内部AHB/APB时钟频率 ④更正笔误 ⑤端口特性中输入特性——端口 P0,P1,P2,P3, RESET中V _{IH} 和V _{IL} 的值。
Rev2.3	2020/9/30	修正以下数据：①7.8.4更新笔误 ②时钟系统描述 ③内部时钟源特性中RCH振荡器精度 ④RESETB引脚特性的V _{IL} 和V _{IH} ⑤增加SPI特性。
Rev2.31	2020/12/31	删除产品特性、引脚配置、封装信息等（相关信息请参考最新的数据手册），修改声明。
Rev2.32	2022/3/9	公司Logo更新。
Rev2.33	2022/8/13	“I2C总线” 章节，I2C操作流程勘误。



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@xhsc.com.cn

网址: <http://www.xhsc.com.cn>

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201210

