



AC7802x 技术参考手册

文档版本： 1.0

发布日期： 2023-03-20

© 2013 - 2022 杰发科技

本文档包含杰发科技的专有信息。未经授权，严禁复制或披露本文档包含的任何信息。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。

修订信息

版本	日期	作者	描述
1.0	2023-03-20	AutoChips	文档初版

版权声明

本参考手册包含杰发科技的机密信息。禁止未经授权使用或披露本手册包含的信息。对因未经杰发科技授权而全部或部分披露此文档内容而给杰发科技带来的任何损失或损害，杰发科技将追究责任。

杰发科技保留对此处任何信息进行更改的权利，此处的信息如有变更，恕不另行通知。杰发科技对使用或依赖此处包含的信息不承担任何责任。

本参考手册的所有信息均“按原样”提供，不提供任何形式的明示、暗示、法定或其他形式的保证。杰发科技明确拒绝对适销性，非侵权性和针对特定用途的适用性方面的所有暗示保证。杰发科技对本手册可能使用、包含或提供的任何第三方软件不提供任何担保，并且用户同意仅向该等第三方寻求与此相关的任何担保索赔。杰发科技对于根据用户规格或为符合特定标准或公开论坛而产生的任何交付物，也不承担任何责任。

文档目录

修订信息	2
版权声明	3
文档目录	4
插图目录	18
表格目录	21
缩略语	26
1 简介	27
1.1 概要	27
1.2 模块概述	27
2 存储器和总线架构	29
2.1 结构框图	29
2.2 功能描述	30
2.2.1 存储器组织	30
2.2.2 内置 SRAM	30
2.2.3 快速 GPIO 存储器映射	31
2.2.4 存储器映射	31
2.2.5 片内 Flash 存储器	32
2.2.6 片内 Flash 存储器读取	32
2.2.7 芯片型号信息	33
2.2.8 芯片 UUID 信息	33
2.2.9 AHB 与 APB 连接桥	33
2.2.10 嵌套中断向量控制器 (NVIC)	33
2.2.11 启动配置	35
2.3 外设地址分配	35
3 复位 (RESET)	37
3.1 特性	37
3.2 结构框图	37
3.3 功能描述	38

3.3.1	上电复位 (POR)	38
3.3.2	系统复位 (System Reset)	39
3.4	寄存器定义.....	40
3.4.1	复位控制寄存器(RESET_CTRL)	40
3.4.2	复位状态寄存器(RESET_STATUS)	42
4	时钟 (Clock)	44
4.1	简介.....	44
4.2	结构框图	44
4.2.1	时钟控制结构框图.....	44
4.3	寄存器定义.....	45
4.3.1	控制寄存器(CKGEN_CTRL)	45
4.3.2	外设时钟使能寄存器 0(CKGEN_PERI_CLK_EN_0).....	47
4.3.3	外设时钟使能寄存器 1(CKGEN_PERI_CLK_EN_1).....	48
4.3.4	外设复位寄存器 0(CKGEN_PERI_SFT_RST0).....	49
4.3.5	外设复位寄存器 1(CKGEN_PERI_SFT_RST1).....	51
5	电源模式 (Power Modes)	53
5.1	简介.....	53
5.2	功能描述	53
5.3	应用说明	53
5.3.1	进入和退出低功耗模式	53
5.3.2	低功耗模式下的模块操作.....	53
6	系统电源管理 (SPM)	56
6.1	简介.....	56
6.2	特性.....	56
6.3	应用说明	56
6.3.1	SPM 电源控制编程指南	56
6.3.2	晶体振荡器 (XOSC) 电源控制	56
6.4	寄存器定义.....	57
6.4.1	电源管理器配置寄存器 0(SPM_PWR_MGR_CFG0).....	57
6.4.2	电源管理器配置寄存器 1 (SPM_PWR_MGR_CFG1)	58

6.4.3	外设休眠应答状态(SPM_PERIPH_SLEEP_ACK_STATUS).....	59
6.4.4	外设休眠应答使能(SPM_EN_PERIPH_SLEEP_ACK)	61
6.4.5	外设唤醒使能寄存器(SPM_EN_PERIPH_WKUP)	62
6.4.6	唤醒状态标志寄存器(SPM_WAKEUP_IRQ_STATUS)	63
7	通用异步收发器 (UART)	66
7.1	简介.....	66
7.2	特性.....	66
7.3	结构框图	68
7.4	功能描述	69
7.4.1	噪声检测 (Noise Detection)	69
7.4.2	波特率描述	70
7.4.3	LIN 功能.....	70
7.4.4	两种电源模式.....	72
7.5	应用说明	73
7.5.1	波特率配置说明	73
7.5.2	UART 配置说明.....	73
7.6	寄存器定义.....	75
7.6.1	RX/TX 数据寄存器(UART_RBR/THR).....	75
7.6.2	分频器低 8 位寄存器(UART_DIV_L).....	76
7.6.3	分频器高 8 位寄存器(UART_DIV_H)	76
7.6.4	控制寄存器 0(UART_LCR0).....	77
7.6.5	控制寄存器 1(UART_LCR1).....	78
7.6.6	FIFO 控制寄存器(UART_FCR).....	79
7.6.7	中断使能寄存器(UART_IER).....	80
7.6.8	线路状态寄存器 0(UART_LSR0)	81
7.6.9	线路状态寄存器 1(UART_LSR1)	83
7.6.10	采样计数器寄存器(UART_SMP_CNT).....	84
7.6.11	保护时间寄存器(UART_GUARD).....	84
7.6.12	休眠使能寄存器(UART_SLEEP_EN).....	85
7.6.13	小数分频器寄存器(UART_DIV_FRAC).....	86

7.6.14	空闲中断使能寄存器(UART_IDLE).....	86
7.6.15	LIN 控制寄存器 (UART_LINCR).....	87
7.6.16	LIN 同步间隔段控制寄存器(UART_BRKLGH).....	88
8	模数转换器 (ADC)	89
8.1	简介.....	89
8.2	特性.....	89
8.3	结构框图	90
8.4	功能描述	91
8.4.1	上电时序	91
8.4.2	工作模式	91
8.4.3	触发方式	98
8.4.4	线性校准	99
8.4.5	模拟监控器	99
8.4.6	状态标志	101
8.4.7	数据对齐方式.....	102
8.4.8	采样转换时间.....	102
8.4.9	温度传感器	103
8.4.10	低功耗模式	103
8.5	应用说明	104
8.5.1	重置和使能	104
8.5.2	ADC 上电延时	105
8.6	寄存器定义.....	105
8.6.1	状态寄存器(ADC_STR).....	106
8.6.2	控制寄存器 0(ADC_CTRL0)	107
8.6.3	控制寄存器 1(ADC_CTRL1)	109
8.6.4	采样时间寄存器 0(ADC_SPT0).....	109
8.6.5	采样时间寄存器 1(ADC_SPT1).....	110
8.6.6	注入组偏移寄存器(ADC_IOFRx)	111
8.6.7	高阈值寄存器(ADC_AMOHR)	111
8.6.8	低阈值寄存器(ADC_AMOLR).....	112

8.6.9	规则组序列配置寄存器 0(ADC_RSQR0).....	113
8.6.10	规则组序列配置寄存器 1(ADC_RSQR1).....	113
8.6.11	规则组序列配置寄存器 2(ADC_RSQR2).....	114
8.6.12	注入组序列配置寄存器(ADC_ISQR).....	114
8.6.13	注入组数据寄存器(ADC_IDRx).....	115
8.6.14	规则组数据寄存器(ADC_RDRx).....	116
8.6.15	增益误差值校准寄存器(ADC_CGV).....	116
8.6.16	偏置误差值校准寄存器(ADC_COV).....	116
8.6.17	规则组转换结束状态寄存器(ADC_REOC).....	117
8.6.18	规则组转换结束中断使能寄存器(ADC_REOCEN).....	118
8.6.19	注入组转换结束状态寄存器(ADC_IEOC).....	118
8.6.20	注入组转换结束中断使能寄存器(ADC_IEOCEN).....	119
8.6.21	规则组序列配置寄存器 3(ADC_RSQR3).....	119
8.6.22	ADC 模拟配置寄存器 0(ADC_CFG0).....	120
8.6.23	ADC 模拟配置寄存器 1(ADC_CFG1).....	120
9	模拟比较器 (ACMP)	122
9.1	简介.....	122
9.2	特性.....	122
9.3	结构框图.....	122
9.4	功能描述.....	123
9.4.1	普通模式.....	123
9.4.2	轮询模式.....	123
9.4.3	轮询模式下霍尔输出.....	124
9.4.4	迟滞.....	124
9.4.5	DAC 输出.....	125
9.4.6	低功耗模式唤醒.....	125
9.5	寄存器定义.....	125
9.5.1	配置寄存器 0(ACMP_CR0).....	126
9.5.2	配置寄存器 1(ACMP_CR1).....	127
9.5.3	配置寄存器 2(ACMP_CR2).....	128

9.5.4	配置寄存器 3(ACMP_CR3).....	128
9.5.5	配置寄存器 4(ACMP_CR4).....	129
9.5.6	数据输出寄存器(ACMP_DR).....	130
9.5.7	状态寄存器(ACMP_SR).....	131
9.5.8	轮询分频器寄存器(ACMP_FD).....	133
9.5.9	霍尔输出 A 设置寄存器(ACMP_OPA).....	133
9.5.10	霍尔输出 B 设置寄存器(ACMP_OPB).....	134
9.5.11	霍尔输出 C 设置寄存器(ACMP_OPC).....	135
9.5.12	DAC 参考源选择寄存器(ACMP_DACSR).....	135
9.5.13	模拟配置寄存器(ACMP_CFG).....	136
10	脉宽调制 (PWM)	137
10.1	简介.....	137
10.2	特性.....	137
10.3	结构框图.....	138
10.4	传统 PWM 功能描述.....	139
10.4.1	时钟源.....	139
10.4.2	计数器.....	139
10.4.3	工作模式.....	140
10.4.4	输入捕获模式.....	141
10.4.5	输出比较模式.....	142
10.4.6	边沿对齐 PWM 模式(EPWM).....	143
10.4.7	中心对齐 PWM 模式(CPWM).....	144
10.4.8	组合模式.....	145
10.4.9	双边沿捕获模式.....	152
10.4.10	正交解码模式.....	153
10.4.11	写保护.....	156
10.4.12	初始化.....	156
10.4.13	极性控制.....	156
10.4.14	输出屏蔽.....	156
10.4.15	软件输出控制.....	157

10.4.16	初始化触发器	157
10.4.17	通道匹配触发器	157
10.4.18	故障控制	157
10.4.19	写缓冲更新的寄存器	158
10.4.20	PWM 同步	159
10.4.21	特性优先级	167
10.4.22	全局时基	168
10.4.23	PWM 中断	168
10.4.24	低功耗模式	168
10.5	寄存器定义	169
10.5.1	初始化寄存器(PWM_INIT)	170
10.5.2	计数器寄存器(PWM_CNT)	171
10.5.3	最大计数值寄存器(PWM_MCVR)	172
10.5.4	通道状态和控制寄存器(PWM_CHnSCR)	172
10.5.5	通道值寄存器(PWM_CHnV)	173
10.5.6	计数器初始值寄存器(PWM_CNTIN)	174
10.5.7	捕获和比较状态寄存器(PWM_STR)	175
10.5.8	功能选择寄存器(PWM_FUNCSEL)	175
10.5.9	同步寄存器(PWM_SYNC)	177
10.5.10	通道输出初始状态寄存器(PWM_OUTINIT)	178
10.5.11	输出屏蔽控制寄存器(PWM_OMCR)	179
10.5.12	模式选择寄存器(PWM_MODESEL)	181
10.5.13	死区插入控制寄存器(PWM_DTSET)	183
10.5.14	外部触发器寄存器(PWM_EXTTRIG)	184
10.5.15	通道输出极性控制寄存器(PWM_CHOPOLCR)	185
10.5.16	故障检测状态寄存器(PWM_FDSR)	187
10.5.17	输入捕获滤波器控制(PWM_CAPFILTER)	188
10.5.18	故障滤波器和故障使能寄存器(PWM_FFAFER)	189
10.5.19	正交解码器接口配置寄存器(PWM_QDI)	190
10.5.20	配置寄存器(PWM_CONF)	191

10.5.21	故障输入极性寄存器(PWM_FLTPOL)	193
10.5.22	同步配置寄存器(PWM_SYNCONF)	194
10.5.23	反相控制寄存器(PWM_INVCR)	196
10.5.24	通道软件输出控制寄存器(PWM_CHOSWCR).....	197
11	脉冲宽度检测定时器(PWDT).....	199
11.1	简介.....	199
11.2	特性.....	199
11.3	结构框图	200
11.4	功能描述	200
11.4.1	脉冲宽度测量功能.....	200
11.4.2	定时器功能	203
11.4.3	中断请求	204
11.4.4	低功耗模式	204
11.5	应用说明	204
11.5.1	脉冲宽度测量功能编程指南	204
11.5.2	定时器功能编程指南	204
11.6	寄存器定义.....	205
11.6.1	初始化寄存器 0(PWDT_INIT0)	205
11.6.2	脉宽计数寄存器(PWDT_NPW).....	206
11.6.3	初始化寄存器 1(PWDT_INIT1)	207
12	周期性中断定时器 (TIMER)	209
12.1	简介.....	209
12.2	特性.....	209
12.3	结构框图	209
12.4	功能描述	210
12.4.1	普通模式	210
12.4.2	链接模式	210
12.4.3	中断	210
12.5	寄存器定义.....	210
12.5.1	定时器模块控制寄存器(TIMER_MCR).....	211

12.5.2	定时器装载值寄存器(TIMER_LDVAL)	211
12.5.3	定时器当前值寄存器(TIMER_CVAL)	212
12.5.4	定时器初始寄存器 (TIMER_INIT)	212
12.5.5	定时器标志寄存器(TIMER_TF)	213
13	采集传输终端 (CTU)	214
13.1	简介	214
13.2	特性	214
13.3	结构框图	215
13.4	功能描述	215
13.4.1	ACMP 输出捕获	215
13.4.2	RTC 捕获	216
13.4.3	ADC 硬件触发	216
13.4.4	PWM 软件同步	216
13.4.5	低功耗模式	216
13.5	寄存器定义	216
13.5.1	配置寄存器 0(CTU_CONFIG0)	216
13.5.2	配置寄存器 1(CTU_CONFIG1)	218
14	通用输入/输出 (GPIO)	220
14.1	简介	220
14.2	特性	220
14.3	结构框图	221
14.4	功能描述	222
14.4.1	外部中断	222
14.4.2	复用功能	224
14.4.3	低功耗模式	226
14.5	应用说明	226
14.5.1	外部输入	226
14.5.2	复用功能	226
14.5.3	开漏输出	226
14.5.4	HIGH_Z 模式	226

14.5.5	GPIO 功能.....	227
14.5.6	编程指南.....	227
14.6	寄存器定义.....	227
14.6.1	端口配置寄存器(GPIO_CR).....	228
14.6.2	端口输入数据寄存器 (GPIO_IDR).....	229
14.6.3	端口输出数据寄存器(GPIO_ODR).....	229
14.6.4	端口置位/复位寄存器(GPIO_BSRR).....	230
14.6.5	端口复位寄存器(GPIO_BRR).....	231
14.6.6	下拉使能寄存器(GPIO_PD).....	231
14.6.7	上拉使能寄存器(GPIO_PU).....	232
14.6.8	驱动能力选择寄存器 (GPIO_E4_E2).....	232
14.6.9	输入使能寄存器(GPIO_IES).....	233
14.6.10	复用功能选择寄存器(GPIO_PINMUX).....	234
14.6.11	外部中断标志暂停寄存器(GPIO_PR).....	234
14.6.12	中断掩码寄存器(GPIO_IMR).....	235
14.6.13	上升沿触发事件配置寄存器(GPIO_RTSR).....	235
14.6.14	下降沿触发事件配置寄存器(GPIO_FTSR).....	236
14.6.15	外部中断寄存器(GPIO_EXTICR).....	236
15	I2C 总线模块 (I2C)	238
15.1	简介.....	238
15.2	特性.....	238
15.3	结构框图.....	239
15.3.1	I2C 信号组成.....	239
15.3.2	波特率组成.....	240
15.3.3	数据流程.....	240
15.4	功能描述.....	241
15.4.1	主机模式.....	241
15.4.2	从机模式.....	242
15.4.3	SMBus.....	242
15.4.4	中断请求.....	243

15.4.5	从机低功耗唤醒	243
15.5	应用说明	243
15.5.1	数据传输	243
15.5.2	应答控制	245
15.6	寄存器定义	245
15.6.1	地址寄存器 0(I2C_ADDR0)	246
15.6.2	地址寄存器 1(I2C_ADDR1)	246
15.6.3	波特率配置寄存器 0(I2C_SAMPLE_CNT)	247
15.6.4	波特率寄存器 1(I2C_STEP_CNT)	247
15.6.5	控制寄存器 0(I2C_CTRL0)	248
15.6.6	控制寄存器 1(I2C_CTRL1)	249
15.6.7	控制寄存器 2(I2C_CTRL2)	250
15.6.8	控制寄存器 3(I2C_CTRL3)	251
15.6.9	状态寄存器 0(I2C_STATUS0)	251
15.6.10	状态寄存器 1(I2C_STATUS1)	253
15.6.11	毛刺滤波配置寄存器(I2C_DGLCFG)	255
15.6.12	数据寄存器(I2C_DATA)	256
15.6.13	起始与停止信号控制寄存器(I2C_STARTSTOP)	256
16	串行外设接口 (SPI)	258
16.1	简介	258
16.2	特性	258
16.3	结构框图	259
16.4	功能描述	259
16.4.1	数据流 & 算法	259
16.4.2	输入输出时序	260
16.4.3	主机 SCK 输出时序设置	262
16.4.4	主机模式故障检测	262
16.4.5	从机低功耗唤醒	263
16.4.6	中断	264
16.5	应用说明	265

16.5.1	主机 CS 连续模式	265
16.5.2	主机 CS 非连续输出	265
16.5.3	从机模式	266
16.6	寄存器定义	266
16.6.1	配置寄存器 0(SPI_CFG0)	266
16.6.2	配置寄存器 1(SPI_CFG1)	267
16.6.3	命令寄存器 (SPI_CMD)	269
16.6.4	状态寄存器(SPI_STATUS)	270
16.6.5	数据寄存器(SPI_DATA)	272
16.6.6	配置寄存器 2(SPI_CFG2)	272
17	看门狗模块 (WDG)	274
17.1	简介	274
17.2	特性	274
17.3	结构框图	274
17.4	功能描述	275
17.4.1	基本看门狗	275
17.4.2	看门狗默认超时行为	275
17.4.3	窗口看门狗	275
17.4.4	低功耗行为	275
17.4.5	调试模式	275
17.5	应用说明	275
17.5.1	配置看门狗	275
17.5.2	刷新看门狗	276
17.5.3	看门狗中断	276
17.6	寄存器定义	276
17.6.1	配置寄存器 0(WDG_CS0)	277
17.6.2	配置寄存器 1(WDG_CS1)	278
17.6.3	计数器寄存器(WDG_CNT)	279
17.6.4	超时值寄存器(WDG_TOVAL)	279
17.6.5	窗口值寄存器(WDG_WIN)	280

18	实时计数器模块 (RTC)	281
18.1	简介.....	281
18.2	特性.....	281
18.3	结构框图	281
18.4	功能描述	281
18.4.1	时钟源选择	281
18.4.2	计时特性	282
18.4.3	RTC 计时信号输出	282
18.4.4	低功耗唤醒	282
18.5	应用说明	282
18.5.1	RTC 基本使用.....	282
18.5.2	RTC 低功耗唤醒	282
18.6	寄存器定义.....	283
18.6.1	控制与状态寄存器(RTC_SC).....	283
18.6.2	模值寄存器(RTC_MOD).....	284
18.6.3	计数器寄存器(RTC_CNT).....	285
18.6.4	预分频器寄存器(RTC_PS).....	285
18.6.5	预分频器计数寄存器(RTC_PSCNT)	286
19	片内 Flash (Embedded Flash)	287
19.1	简介.....	287
19.2	特性.....	287
19.3	结构框图	288
19.4	功能描述	288
19.4.1	片内 Flash 组织.....	288
19.4.2	片内 Flash 保护	289
19.4.3	Flash 命令 ID.....	291
19.5	应用说明	291
19.5.1	页擦除.....	291
19.5.2	块擦除.....	292
19.5.3	整片擦除	293

19.5.4	页编程.....	294
19.5.5	页擦除验证	295
19.5.6	整片擦除验证.....	296
19.5.7	选项字节擦除.....	297
19.5.8	选项字节编程.....	298
19.6	寄存器定义.....	300
19.6.1	解锁 序列寄存器(EFLASH_UKR)	300
19.6.2	全局信息寄存器(EFLASH_GSR)	301
19.6.3	全局控制寄存器(EFLASH_GCR).....	301
19.6.4	命令状态寄存器(EFLASH_CSR)	302
19.6.5	命令控制寄存器(EFLASH_CCR).....	304
19.6.6	命令地址寄存器(EFLASH_CAR).....	305
19.6.7	命令数据寄存器(EFLASH_CDR).....	306
19.6.8	P-flash 写保护寄存器 0 (EFLASH_PWPR0)	307
19.6.9	P-flash 写保护寄存器 1 (EFLASH_PWPR1)	307
19.6.10	D-flash 写保护寄存器 (EFLASH_DWPR).....	308
20	SRAM 错误检测纠正 (ECC_SRAM)	309
20.1	简介.....	309
20.2	特性.....	309
20.3	功能描述	309
20.4	寄存器定义.....	310
20.4.1	控制以及状态寄存器(ECC_SRAM_CTRL)	310
20.4.2	1 bit 错误地址寄存器(ECC_SRAM_ERR1_ADDR)	311
20.4.3	2 bits 错误地址寄存器(ECC_SRAM_ERR2_ADDR).....	312
21	调试	313
21.1	简介.....	313
21.2	特性.....	313

插图目录

图 2-1 系统架构	29
图 3-1 Reset 模块结构框图.....	38
图 4-1 时钟控制结构	45
图 7-1 UART 结构框图.....	68
图 7-2 UART 传送器流程.....	69
图 7-3 UART 接收器流程.....	69
图 7-4 UART 噪声检测.....	69
图 7-5 LIN 帧流程.....	71
图 7-6 运行模式 (Run mode) 和停止模式 (Stop mode)	72
图 7-7 通过 UART 唤醒芯片的典型流程	72
图 7-8 波特率发生器框图.....	73
图 8-1 ADC 结构框图.....	90
图 8-2 ADC 上电时序.....	91
图 8-3 规则组序列.....	92
图 8-4 有效规则组序列	93
图 8-5 注入组序列.....	93
图 8-6 有效注入组序列	93
图 8-7 Mode 1 工作流程.....	93
图 8-8 Mode 2 工作流程.....	94
图 8-9 Mode 3 注入组扫描模式工作流程.....	94
图 8-10 Mode 3 在 ADC 空闲状态下具有注入触发的工作流程	95
图 8-11 Mode 3 注入组间隔模式工作流程.....	95
图 8-12 Mode 4 工作流程.....	95
图 8-13 Mode 5 注入组扫描模式工作流程.....	96
图 8-14 Mode 5 在 ADC 空闲状态下具有注入触发的工作流程	96
图 8-15 Mode 5 注入组间隔模式工作流程.....	97
图 8-16 Mode 6 操作流程.....	97
图 8-17 Mode 7 操作流程.....	98
图 8-18 Mode 8 操作流程.....	98
图 8-19 ADC GEOE 校准转换示意	99
图 8-20 电平触发模式下监控区域.....	100
图 8-21 边沿触发模式下监控区域.....	101
图 8-22 三个标志行为.....	102
图 8-23 ADC 数据在寄存器内的排布	102
图 8-24 ADC 功耗模式切换流程图	104
图 9-1 ACMP 结构框图.....	122
图 9-2 轮询模式工作流程图	124
图 9-3 迟滞工作原理	124
图 9-4 DAC 输出配置示意图	125
图 10-1 PWM 结构框图.....	138
图 10-2 向上计数.....	139
图 10-3 向上-向下计数.....	140
图 10-4 输入捕获模式.....	142
图 10-5 匹配设置输出比较模式	142
图 10-6 匹配清除输出比较模式	143
图 10-7 匹配翻转输出比较模式	143
图 10-8 EPWM 波形	144

图 10-9 CPWM 波形	145
图 10-10 向上计数组合模式输出波形	146
图 10-11 $(CNTIN < CHnV/CH(n+1)V < MCVR) \&(CHnV < CH(n+1)V)$ 条件下输出波形	146
图 10-12 $(CNTIN < CHnV < MCVR)\&(CH(n+1)V = MCVR)$ 条件下输出波形	146
图 10-13 $(CHnV = CNTIN)\&(CNTIN < CH(n+1)V < MCVR)$ 条件下输出波形	147
图 10-14 $(CNTIN < CHnV/ CH(n+1)V < MCVR)$ 且 $(CHnV > CH(n+1)V)$ 条件下输出波形	147
图 10-15 $(CH(n+1)V < CNTIN)\&(CNTIN < CHnV < MCVR)$ 条件下输出波形	147
图 10-16 $(CH(n+1)V > MCVR)$ 且 $(CNTIN < CHnV < MCVR)$ 条件下输出波形.....	148
图 10-17 向上-向下计数区间范围	148
图 10-18 CHn 匹配点 $DIR=1(Up)$, $CH(n+1)$ 匹配点 $DIR=1(Up)$	149
图 10-19 CHn 匹配点 $DIR=1(Up)$, $CH(n+1)$ 匹配点 $DIR=0(Down)$	149
图 10-20 CHn 匹配点 $DIR=0(Down)$, $CH(n+1)$ 匹配点 $DIR=1(Up)$	149
图 10-21 $CHnV$ 匹配点 $DIR=0(Down)$, $CH(n+1)V$ 匹配点 $DIR=0(Down)$	150
图 10-22 互补模式输出	150
图 10-23 死区时间插入	151
图 10-24 $CH(n+1)V$ 在下一周期匹配的输出生波	152
图 10-25 多通道之间相位偏移输出波形	152
图 10-26 双边沿捕获模式图	153
图 10-27 计数和方向编码模式	154
图 10-28 A 相和 B 相编码模式	155
图 10-29 向上计数 PWM counter 溢出	155
图 10-30 向下计数 PWM counter 溢出	156
图 10-31 $HWTRIGMODESEL = 0$ 硬件触发事件	160
图 10-32 软件触发事件	160
图 10-33 边界周期与加载点	161
图 10-34 PWM_MCVR 寄存器同步流程	162
图 10-35 PWM_CNT 寄存器同步流程	163
图 10-36 PWM_OMCR 寄存器同步流程	164
图 10-37 PWM_INVCR 寄存器同步流程	165
图 10-38 PWM_CHOSWCR 寄存器同步流程	166
图 10-39 PWM_CHOPOLCR 寄存器同步流程	167
图 10-40 特性优先级	168
图 11-1 PWDTC 结构框图	200
图 11-2 四种基本测量模式($HALLN=0$)	201
图 11-3 霍尔测量模式($HALLN=1$)	201
图 11-4 两种常见 Hall 安装方式	202
图 11-5 低电平噪音和滤波器示例	202
图 11-6 高电平噪音和滤波器示例	203
图 11-7 PWDTC 计数器和计数错误	203
图 11-8 在 $TIMEN=0$ 期间修改 $TIMLDVAL$	204
图 11-9 $TIMEN=1$ 期间修改 $TIMLDVAL$	204
图 12-1 TIMER 结构框图	209
图 13-1 CTU 结构框图	215
图 14-1 GPIO 结构框图	221
图 15-1 I2C 结构框图	239
图 15-2 START 和 STOP 条件	239
图 15-3 数据传输格式	239
图 15-4 波特率生成	240
图 15-5 发送器数据流程	241
图 15-6 接收器数据流程	241
图 15-7 主机组合模式	242

图 16-1 SPI 系统连接.....	258
图 16-2 SPI 结构框图.....	259
图 16-3 主机数据流.....	259
图 16-4 从机数据流.....	260
图 16-5 CPHA=0 传输格式.....	261
图 16-6 CPHA=1 传输格式.....	262
图 16-7 波特率生成.....	262
图 16-8 在模式故障检测使能时的 SCK 输出时序.....	263
图 16-9 模式故障检测限制.....	263
图 16-10 唤醒序列.....	264
图 16-11 CS 连续模式.....	265
图 17-17-1 WDG 结构框图.....	274
图 18-1 RTC 结构框图.....	281
图 19-1 eflash 和 eflash 控制器结构框图.....	288
图 19-2 eflash 和 eflash 控制器数据流.....	288
图 19-3 页擦除命令操作流程.....	292
图 19-4 块擦除命令操作流程.....	293
图 19-5 整片擦除命令操作流程.....	294
图 19-6 页编程命令操作流程.....	295
图 19-7 页擦除验证命令操作流程.....	296
图 19-8 整片擦除验证命令操作流程.....	297
图 19-9 选项字节擦除命令操作流程.....	298
图 19-10 选项字节编程命令操作流程.....	299

表格目录

表 1-1 AC7802x 模块	27
表 2-1 小端格式存放方式	30
表 2-2 设备存储器映射	31
表 2-3 中断表	33
表 2-4 启动配置	35
表 2-5 外设地址分配表	35
表 3-1 复位 (Reset) 寄存器映射	40
表 3-2 RESET_CTRL 寄存器	40
表 3-3 RESET_STATUS 寄存器	42
表 4-1 时钟寄存器映射	45
表 4-2 CKGEN_CTRL 寄存器	45
表 4-3 CKGEN_PERI_CLK_EN_0 寄存器	47
表 4-4 CKGEN_PERI_CLK_EN_1 寄存器	48
表 4-5 CKGEN_PERI_SFT_RST0 寄存器	49
表 4-6 CKGEN_PERI_SFT_RST1 寄存器	51
表 5-1 低功耗模式下的模块功能	53
表 6-1 SPM 寄存器映射	57
表 6-2 SPM_PWR_MGR_CFG0 寄存器	57
表 6-3 SPM_PWR_MGR_CFG1 寄存器	58
表 6-4 SPM_PERIPH_SLEEP_ACK_STATUS 寄存器	59
表 6-5 SPM_EN_PERIPH_SLEEP_ACK 寄存器	61
表 6-6 SPM_EN_PERIPH_WKUP 寄存器	62
表 6-7 SPM_WAKEUP_IRQ_STATUS 寄存器	63
表 7-1 功能分类和配置	66
表 7-2 典型的波特率及误差率@bclock=32MHz	70
表 7-3 典型的波特率及误差率@bclock=8MHz	70
表 7-4 UART 寄存器映射	75
表 7-5 UART_RBR/THR 寄存器	75
表 7-6 UART_DIV_L 寄存器	76
表 7-7 UART_DIV_H 寄存器	76
表 7-8 UART_LCR0 寄存器	77
表 7-9 UART_LCR1 寄存器	78
表 7-10 UART_FCR 寄存器	79
表 7-11 UART_IER 寄存器	80
表 7-12 UART_LSR0 寄存器	81
表 7-13 UART_LSR1 寄存器	83
表 7-14 UART_SMP_CNT 寄存器	84
表 7-15 UART_GUARD 寄存器	84
表 7-16 UART_SLEEP_EN 寄存器	85
表 7-17 UART_DIV_FRAC 字段	86
表 7-18 UART_IDLE 寄存器	86
表 7-19 UART_LINCR 寄存器	87
表 7-20 UART_BRKLGH 寄存器	88
表 8-1 工作模式配置表	91
表 8-2 不同触发方式下的响应行为	98
表 8-3 模拟监控通道配置	100
表 8-4 ADC 采样率	103

表 8-5 ADC 重置范围	104
表 8-6 ADC 寄存器映射	105
表 8-7 ADC_STR 寄存器.....	106
表 8-8 ADC_CTRL0 寄存器.....	107
表 8-9 ADC_CTRL1 寄存器.....	109
表 8-10 ADC_SPT0 寄存器.....	109
表 8-11 ADC_SPT1 寄存器.....	110
表 8-12 ADC_IOFRx (x= 0 ~ 3) 寄存器.....	111
表 8-13 ADC_AMOHR 寄存器.....	111
表 8-14 ADC_AMOLR 寄存器.....	112
表 8-15 ADC_RSQR0 寄存器.....	113
表 8-16 ADC_RSQR1 寄存器.....	113
表 8-17 ADC_RSQR2 寄存器.....	114
表 8-18 ADC_ISQR 寄存器.....	114
表 8-19 ADC_IDRx (x=0 ~ 3)寄存器.....	115
表 8-20 ADC_RDRx 寄存器.....	116
表 8-21 ADC_CGV 寄存器.....	116
表 8-22 ADC_COV 寄存器.....	116
表 8-23 ADC_REOC 寄存器.....	117
表 8-24 ADC_REOCEN 寄存器.....	118
表 8-25 ADC_IEOC 寄存器.....	118
表 8-26 ADC_IEOCEN 寄存器.....	119
表 8-27 ADC_RSQR3 寄存器.....	119
表 8-28 ADC_CFG0 寄存器.....	120
表 8-29 ADC_CFG1 寄存器.....	120
表 9-1 ACMP 寄存器映射	125
表 9-2 ACMP_CR0 寄存器.....	126
表 9-3 ACMP_CR1 寄存器.....	127
表 9-4 ACMP_CR2 寄存器.....	128
表 9-5 ACMP_CR3 寄存器.....	128
表 9-6 ACMP_CR4 寄存器.....	129
表 9-7 ACMP_DR 寄存器.....	130
表 9-8 ACMP_SR 寄存器.....	131
表 9-9 ACMP_FD 寄存器.....	133
表 9-10 ACMP_OPA 寄存器.....	133
表 9-11 ACMP_OPB 寄存器.....	134
表 9-12 ACMP_OPC 寄存器.....	135
表 9-13 ACMP_DACSR 寄存器.....	135
表 9-14 ACMP_CFG 寄存器.....	136
表 10-1 工作模式配置.....	140
表 10-2 组合模式软件输出控制行为.....	157
表 10-3 故障源编号表.....	158
表 10-4 PWM_CNTIN 寄存器更新缓存.....	158
表 10-5 PWM_CH(n)V 寄存器更新缓存.....	159
表 10-6 PWM_MCVR 寄存器更新缓存.....	159
表 10-7 PWM 模块低功耗模式.....	168
表 10-8 PWM 寄存器映射.....	169
表 10-9 PWM_INIT 寄存器.....	170
表 10-10 PWM_CNT 寄存器.....	171
表 10-11 PWM_MCVR 寄存器.....	172
表 10-12 PWM_CHnSCR 寄存器.....	172

表 10-13 PWM_CHnV 寄存器	173
表 10-14 PWM_CNTIN 寄存器	174
表 10-15 PWM_STR 寄存器	175
表 10-16 PWM_FUNCSEL 寄存器	175
表 10-17 PWM_SYNC 寄存器	177
表 10-18 PWM_OUTINIT 寄存器	178
表 10-19 PWM_OMCR 寄存器	179
表 10-20 PWM_MODESEL 寄存器	181
表 10-21 PWM_DTSET 寄存器	183
表 10-22 PWM_EXTTRIG 寄存器	184
表 10-23 PWM_CHOPOLCR 寄存器	185
表 10-24 PWM_FDSR 寄存器	187
表 10-25 PWM_CAPFILTER 寄存器	188
表 10-26 PWM_FFAFER 寄存器	189
表 10-27 PWM_QDI 寄存器	190
表 10-28 PWM_CONF 寄存器	191
表 10-29 PWM_FLTPOL 寄存器	193
表 10-30 PWM_SYNCONF 寄存器	194
表 10-31 PWM_INVCR 寄存器	196
表 10-32 PWM_CHOSWCR 寄存器	197
表 11-1 可滤波脉冲宽度范围	202
表 11-2 PWDT 中断汇总	204
表 11-3 PWDT 模块低功耗模式	204
表 11-4 PWDT 寄存器映射	205
表 11-5 PWDT_INIT0 寄存器	205
表 11-6 PWDT_NPW 寄存器	206
表 11-7 PWDT_INIT1 寄存器	207
表 12-1 定时器寄存器映射	210
表 12-2 TIMER_MCR 寄存器	211
表 12-3 TIMER_LDVAL 寄存器	211
表 12-4 TIMER_CVAL 寄存器	212
表 12-5 TIMER_INIT 寄存器	212
表 12-6 TIMER_TF 寄存器	213
表 13-1 CTU 模块低功耗模式	216
表 13-2 CTU 寄存器映射	216
表 13-3 CTU_CONFIG0 寄存器	216
表 13-4 CTU_CONFIG1 寄存器	218
表 14-1 GPIO 外部中断和中断处理函数对应关系	223
表 14-2 GPIO 复用功能描述	224
表 14-3 GPIO 寄存器映射	227
表 14-4 GPIO_CR 寄存器	228
表 14-5 GPIO_IDR 寄存器	229
表 14-6 GPIO_ODR 寄存器	229
表 14-7 GPIO_BSRR 寄存器	230
表 14-8 GPIO_BRR 寄存器	231
表 14-9 GPIO_PD 寄存器	231
表 14-10 GPIO_PU 寄存器	232
表 14-11 GPIO_E4_E2 寄存器	232
表 14-12 GPIO_IES 寄存器	233
表 14-13 GPIO_PINMUX 寄存器	234
表 14-14 GPIO_PR 寄存器	234

表 14-15 GPIO_IMR 寄存器	235
表 14-16 GPIO_RTSTR 寄存器	235
表 14-17 GPIO_FTSTR 寄存器	236
表 14-18 GPIO_EXTICR 寄存器	236
表 15-1 I2C 中断汇总	243
表 15-2 I2C 寄存器映射	245
表 15-3 I2C_ADDR0 寄存器	246
表 15-4 I2C_ADDR1 寄存器	246
表 15-5 I2C_SAMPLE_CNT 寄存器	247
表 15-6 I2C_STEP_CNT 寄存器	247
表 15-7 I2C_CTRL0 寄存器	248
表 15-8 I2C_CTRL1 寄存器	249
表 15-9 I2C_CTRL2 寄存器	250
表 15-10 I2C_CTRL3 寄存器	251
表 15-11 I2C_STATUS0 寄存器	251
表 15-12 I2C_STATUS1 寄存器	253
表 15-13 I2C_DGLCFG 寄存器	255
表 15-14 I2C_DATA 寄存器	256
表 15-15 I2C_STARTSTOP 寄存器	256
表 16-1 中断汇总	264
表 16-2 SPI 寄存器映射	266
表 16-3 SPI_CFG0 寄存器	266
表 16-4 SPI_CFG1 寄存器	267
表 16-5 SPI_CMD 寄存器	269
表 16-6 SPI_STATUS 寄存器	270
表 16-7 SPI_DATA 寄存器	272
表 16-8 SPI_CFG2 寄存器	272
表 17-1 WDG 寄存器映射	276
表 17-2 WDG_CS0 寄存器	277
表 17-3 WDG_CS1 寄存器	278
表 17-4 WDG_CNT 寄存器	279
表 17-5 WDG_TOVAL 寄存器	279
表 17-6 WDG_WIN 寄存器	280
表 18-1 RTC 寄存器映射	283
表 18-2 RTC_SC 寄存器	283
表 18-3 RTC_MOD 寄存器	284
表 18-4 RTC_CNT 寄存器	285
表 18-5 RTC_PS 寄存器	285
表 18-6 RTC_PSCNT 寄存器	286
表 19-1 片内 Flash 存储器组织	288
表 19-2 选项字节内容列表	289
表 19-3 读保护设置	290
表 19-4 写保护设置	290
表 19-5 Flash 命令 ID	291
表 19-6 片内 Flash 寄存器映射	300
表 19-7 EFLASH_UKR 寄存器	300
表 19-8 EFLASH_GSR 寄存器	301
表 19-9 EFLASH_GCR 寄存器	301
表 19-10 EFLASH_CSR 寄存器	302
表 19-11 EFLASH_CCR 寄存器	304
表 19-12 EFLASH_CAR 寄存器	305

表 19-13 EFLASH_CDR 寄存器.....	306
表 19-14 EFLASH_PWPR0 寄存器.....	307
表 19-15 EFLASH_PWPR1 寄存器.....	307
表 19-16 EFLASH_DWPR 寄存器.....	308
表 20-1 ECC_SRAM 寄存器映射.....	310
表 20-2 ECC_SRAM_CTRL 寄存器.....	310
表 20-3 ECC_SRAM_ERR1_ADDR 寄存器.....	311
表 20-4 ECC_SRAM_ERR2_ADDR 寄存器.....	312

缩略语

AAI	Auto Address Increment	地址自增模式
AHB	Advanced High Performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	高级外设总线
CKGEN	Clock Generator	时钟产生器
ECC	Error Checking and Correction	错误检测以及纠正
HSE	High Speed External Clock	高速外部时钟
HSI	High Speed Internal Clock	高速内部时钟
LRU	Least Recently Used	最少最近使用
LSI	Low Speed Internal Clock	低速内部时钟
LVR	Low Voltage Reset	低电压检测复位
NMI	Non Maskable Interrupt	不可屏蔽中断
OSC	Oscillator	振荡器
POR	Power On Reset	上电复位
LVD	Low Voltage Detect	低电压检测警告
SPM	System Power Manager	系统电源管理
SRAM	Static Random-Access Memory	静态随机存取存储器
XOSC	External Crystal Oscillator	晶体振荡器

1 简介

1.1 概要

AC7802x 是采用 ARM Cortex™-M0+内核的高性能、低功耗 MCU。

- 频率高达 32MHz
- 工作温度范围支持-40℃~ +125℃
- 工作电压支持 2.7V~5.5V

1.2 模块概述

表 1-1 AC7802x 模块

模块	说明
ARM Cortex™-M0+内核	<ul style="list-style-type: none"> • ARM Cortex™-M0+ 32位MCU 内核 • 高达32 MHz CPU频率
存储器	<ul style="list-style-type: none"> • 高达 32 Kbyte的P-Flash存储器+2Kbyte D-Flash存储器 • 高达 4 Kbyte的SRAM, 带ECC
时钟	<ul style="list-style-type: none"> • 外部晶振或谐振器 (HSE) <ul style="list-style-type: none"> - 范围: 8MHz ~20MHz • 外部输入时钟 <ul style="list-style-type: none"> - ≤ 20MHz • 内置振荡器 <ul style="list-style-type: none"> - 32MHz振荡器 (HSI) - 32kHz振荡器 (LSI)
系统	<ul style="list-style-type: none"> • 1个时钟 (Clock) 模块 • 1个复位 (Reset) 模块 • 1个系统电源管理 (SPM) 模块 • 1个看门狗 (WDG) 模块
模拟	<ul style="list-style-type: none"> • 1个12位模数转换器 (ADC), 支持18路外部通道+1路内部通道 • 1个模拟比较器 (ACMP), 1个6位可输出的数模转换器 (DAC)
定时器	<ul style="list-style-type: none"> • 1个4通道脉宽调制 (PWM) 控制器 • 2个2通道脉宽调制 (PWM) 控制器 • 4个32bit通用定时器 (Timer) • 实时时钟 (RTC) • 1个脉宽检测定时器 (PWDT) • 系统滴答定时器 (SysTick)
通信接口	<ul style="list-style-type: none"> • 1个串行外设接口 (SPI) 模块 • 1个内部集成电路 (I2C) 模块 • 2个通用异步收发器 (UART) 模块, 仅UART0支持 LIN

通用接口	<ul style="list-style-type: none"> • 最多27个通用输入输出控制器 (GPIO) • 不可屏蔽中断 (NMI) • 5个外部中断 (IRQ)
调试接口	<ul style="list-style-type: none"> • 串行调试 (SWD) 接口

2 存储器和总线架构

2.1 结构框图

AC7802x 主系统由以下部分构成：

- 一个主模块单元
 - Cortex™-M0+ 内核 AHB-Lite 总线
- 四个从模块单元
 - 片内 SRAM
 - 片内 Flash 存储器
 - 快速 IO 和 GPIO
 - AHB 到 APB 的桥 (AHB_APB)，它连接所有的 APB 设备。

如图 2-1 所示，这些模块单元都通过一个多级的 AHB 总线架构相互连接。

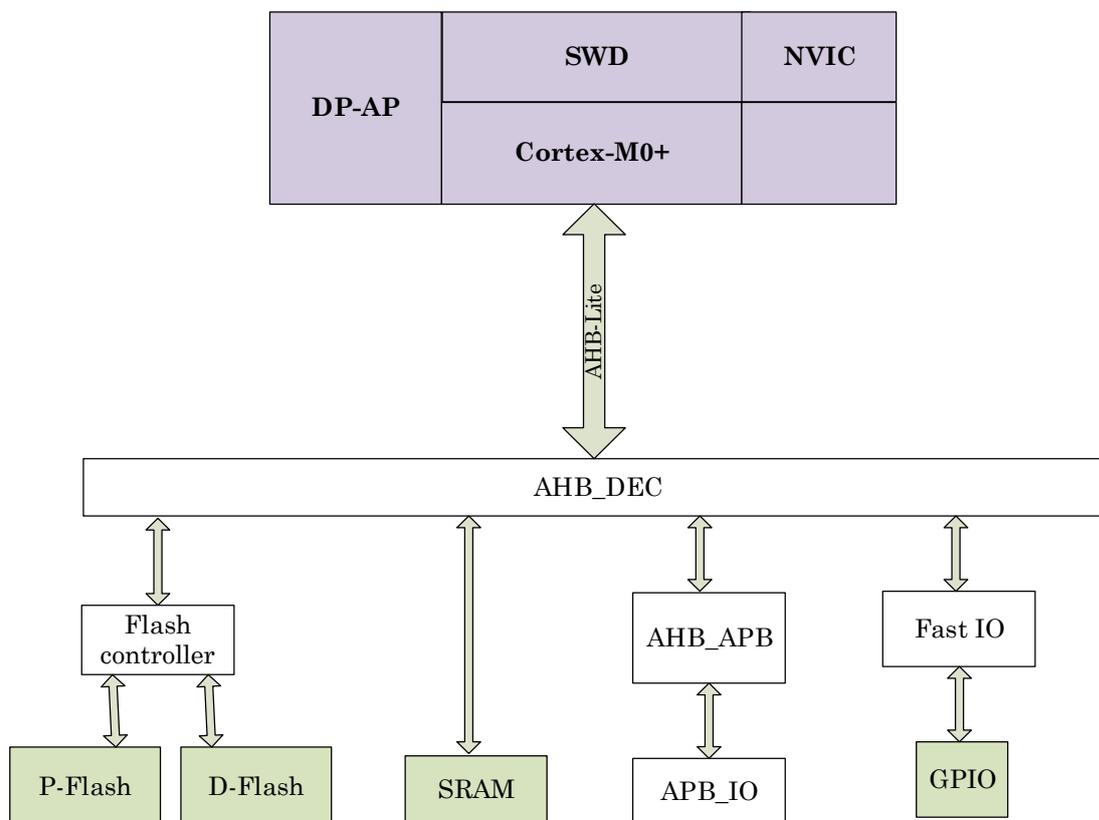


图 2-1 系统架构

AHB-Lite 总线

该总线将 Cortex™-M0+内核的指令和数据请求发出，访问存储器和外设。

AHB_DEC

AHB 译码模块 (AHB Decode)，仲裁后的地址会根据地址范围进行译码，译码后的地址才可以访问不同的 slaver。

总线矩阵

总线矩阵由主模块总线和从模块总线组成。

内部 Flash 只能被 AHB-Lite 总线访问。

AHB 与 APB 连接桥

AHB 与 APB 连接桥 (图 2-1 的 AHB_APB 单元) 在 AHB 和 APB 总线间提供全同步连接，APB 操作频率默认为 AHB 操作频率的 1/2。

DP-AP

DP-AP 即调试访问端口，由 DP (debug port) 和 AP (access port) 组成。

DP 接口模块 (AC7802x 仅支持 SW-DP) 是把外部信号转换成 32bit 的调试总线信号。

AP 接口模块相当于一个总线桥，用于把调试总线命令转换成 AHB 总线上的数据，然后进行传送。

通过 DP 和 AP 的配合，实现 SWD 对 AC7802x 的地址访问。

2.2 功能描述

2.2.1 存储器组织

程序存储器、数据存储器、寄存器和输入输出 (I/O) 接口统一编址在 4G 字节的线性地址空间里。

数据字节在存储器中以小端格式存放。小端是指数据的低位保存在内存的低地址中，而数据的高位保存在内存的高地址中。例如，16bit 宽的数 0x1234 在小端格式 CPU 内存中的存放方式 (假设从地址 0x4000 开始存放) 为：

表 2-1 小端格式存放方式

内存地址	0x4000	0x4001
存放内容	0x34	0x12

外设寄存器的详细映射，请参照各外设章节。可寻址的存储空间分为 8 块，每块的寻址空间大小为 512Mbyte。

2.2.2 内置 SRAM

AC7802x 内置一个 4Kbyte 的静态 SRAM，它可以以字节、半字 (16 位) 或全字 (32 位) 访问。该 SRAM 的起始地址为：0x2000 0000。

2.2.3 快速 GPIO 存储器映射

快速 IO 桥提供了一条更有效地通过 AHB 访问 GPIO 的通道。

每个通道有 4Kbyte 的地址空间。

快速 GPIO 地址范围为：0x20080000 ~ 0x20080FFF。

2.2.4 存储器映射

表 2-2 为 AC7802x 设备存储器映射表，包括两种不同的基于不同启动配置的存储器映射表，分别为：片内 Flash 存储器启动和 SRAM 启动。

所有没有分配片上存储和外设的存储器区域，被称之为“保留”区。

表 2-2 设备存储器映射

0xE010 0000	保留	0xE010 0000	保留
0xE00F FFFF	Cortex™-M0+ 内部外设	0xE00F FFFF	Cortex™-M0+ 内部外设
0xE000 0000		0xE000 0000	
0xDFFF FFFF		0xDFFF FFFF	
0x6000 0000	保留	0x6000 0000	保留
0x5FFF FFFF	保留	0x5FFF FFFF	保留
0x4010 0000		0x4010 0000	
0x400F FFFF		0x400F FFFF	
0x4000 0000	外设 APB 地址	0x4000 0000	外设 APB 地址
0x3FFF FFFF	保留	0x3FFF FFFF	保留
0x2008 1000		0x2008 1000	
0x2008 0FFF		0x2008 0FFF	
0x2008 0000	AHB 快速 IO	0x2008 0000	AHB 快速 IO
0x2007 FFFF	保留	0x2007 FFFF	保留
0x2000 1000		0x2000 1000	
0x2000 0FFF		0x2000 0FFF	
0x2000 0000	AHB SRAM	0x2000 0000	AHB SRAM
0x1FFF FFFF	保留	0x1FFF FFFF	保留
0x0804 3000		0x0804 3000	
0x0804 27FF		0x0804 27FF	
0x0804 2000	保留	0x0804 2000	保留

0x0804 1FFF	保留	0x0804 1FFF	保留
0x0804 0800		0x0804 0800	
0x0804 01FF	选项字节	0x0804 01FF	选项字节
0x0804 0000		0x0804 0000	
0x0802 07FF	D-Flash 存储器	0x0802 07FF	D-Flash 存储器
0x0802 0000		0x0802 0000	
0x0800 7FFF	P-Flash 存储器	0x0800 7FFF	P-Flash 存储器
0x0800 0000		0x0800 0000	
0x07FF FFFF	保留	0x07FF FFFF	保留
0x0000 8000		0x0000 1000	
0x0000 7FFF	Flash 存储器	0x0000 0FFF	AHB SRAM
0x0000 0000		0x0000 0000	
Flash 存储器启动		SRAM 启动	

2.2.5 片内 Flash 存储器

高性能片内 Flash 模块的主要特性如下：

- 高达 32Kbyte 的 Flash 存储器结构
- 存储器组织结构：Flash 存储器由 P-flash 存储块、D-flash 存储块和信息块组成
 - P-flash 存储块容量：主存储块最大为 8 K × 32 bit，每个存储块划分为 64 个 512byte 的页
 - D-flash 存储块容量：2Kbyte，可划分为 256 个 8byte 的页
 - 信息块容量：512byte

Flash 控制器特性为：

- Flash 编程/ 擦除操作
- 读/写保护
- 擦除及空白检查
- 缓存控制器用以提升读效率，最优效率为零等待

2.2.6 片内 Flash 存储器读取

Flash 的指令和数据访问是通过 AHB 总线完成的。

2.2.7 芯片型号信息

芯片型号信息用户可通过 eFlash 读操作接口进行访问。芯片型号信息存放在 0x40002050~0x40002053 连续的空间(1*32Bit)。其中高 8 位为固定 0xFF，低 24 位为芯片型号信息。

2.2.8 芯片 UUID 信息

UUID 共 128Bit，由随机数产生，可作为芯片唯一的识别标识。用户可通过 eFlash 读操作接口进行访问。UUID 信息存放在 0x40002054~0x40002063 连续的空间(4*32Bit)。

2.2.9 AHB 与 APB 连接桥

AHB 与 APB 连接桥用于将 AHB 协议解析成 APB 协议。大多数外设为 APB 接口，详细信息可参考表 2-5 外设地址分配表。

2.2.10 嵌套中断向量控制器 (NVIC)

【特性】

- 32 个可屏蔽中断通道（不包括 16 个 Cortex™-M0+中断）
- 4 个可编程优先等级（使用了 2 位 中断优先级）
- 低延迟的异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

NVIC 和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理新的中断。

NVIC 管理着包括内核异常在内的所有中断。更多关于异常和 NVIC 编程的内容，请参考《ARM Cortex™-M0+ Technical Reference Manual》第 8 章异常和中断。

AC7802x 中断向量表如下：

表 2-3 中断表

中断向量号	优先级	优先级类型	名称	说明
-15	-3	固定	复位 (Reset)	系统复位
-14	-2	固定	不可屏蔽中断 (NMI)	不可屏蔽中断
-13	-1	固定	硬件失效 (HardFault)	用于处理所有的异常
			保留	保留

中断向量号	优先级	优先级类型	名称	说明
			保留	保留
			保留	保留
			保留	保留
-5	0	可设置	SVCcall	执行系统服务调用指令（SVC）引发的异常
			保留	保留
			保留	保留
-2	0	可设置	PendSV	可挂起的系统服务请求
-1	0	可设置	SysTick	系统嘀嗒定时器
0	0	可设置	PWDT0	PWDT0 中断
1	0	可设置	PWM0	PWM0 中断
2	0	可设置	PWM1	PWM1 中断
3	0	可设置	PWM2	PWM2 中断
4	0	可设置	ACMP0	ACMP0 中断
5	0	可设置	UART0	UART0 中断
6	0	可设置	UART1	UART1 中断
			保留	保留
8	0	可设置	WDG	看门狗中断
9	0	可设置	SPI0	SPI0 全局中断
			保留	保留
11	0	可设置	I2C0	I2C0 中断
			保留	保留
17	0	可设置	TIMER_CHANNEL0	定时器通道 0 中断
18	0	可设置	TIMER_CHANNEL1	定时器通道 1 中断
19	0	可设置	TIMER_CHANNEL2	定时器通道 2 中断
20	0	可设置	TIMER_CHANNEL3	定时器通道 3 中断
21	0	可设置	RTC	RTC 中断
22	0	可设置	LVD	LVD 中断
23	0	可设置	SPM	SPM 中断
			保留	保留
25	0	可设置	ADC0	ADC0 中断
26	0	可设置	ECC_SRAM	ECC SRAM 检错中断
27	0	可设置	EXTI0	EXTI 0 中断
28	0	可设置	EXTI1	EXTI 1 中断

中断向量号	优先级	优先级类型	名称	说明
29	0	可设置	EXTI2	EXTI 2 中断
30	0	可设置	EXTI3_8	EXTI 3~8 中断
31	0	可设置	EXTI9_15	EXTI 9~15 中断

2.2.11 启动配置

通过如下配置表，可以选择四种不同的启动模式。

表 2-4 启动配置

PIN 脚名称	BOOT(PA6)	PA1	PA0
eFlash boot	0	x	x
SRAM boot	1	1	0

【注意】 x 表示忽略，不用关注。

在系统复位后，在 8MHz 时钟的第 8 个上升沿，BOOT 配置引脚的值会被锁定。用户需要设置这些引脚来选择需要的启动模式。在锁定前，用户需要保持这些引脚处于稳定状态。

由于固定的存储器映射，代码区始终从地址 0x0000 0000 开始。Cortex™-M0+ CPU 从 AHB-Lite 总线获取复位向量，即启动仅适用于从代码区开始（典型地，从主 Flash 存储器启动）。AC7802x 实现了一种特殊的机制，系统不仅可以从主 Flash 存储器和系统存储器启动，而且可以从 SRAM 启动。

根据选定的启动模式，片内 Flash 存储器可以按照如下方式访问：

- **从片内 Flash 存储器启动：** 片内 Flash 存储器被映射到启动存储空间 (0x0000 0000)，但仍然能够在原有的地址(0x800 0000) 访问它。换言之，片内 Flash 存储器的内容可以在两个地址区域访问：0x0000 0000 或 0x800 0000。
- **从片内 SRAM 启动：** SRAM 被映射到启动存储空间 (0x0000 0000)，但仍然能够在它原有的地址（0x2000 0000）访问它。

2.3 外设地址分配

表 2-5 外设地址分配表

APB 存储映射	基地址	大小（字节）
CKGEN	0x40000000	4K
GPIO	0x40001000 / 0x20080000	4K
片内 Flash 控制器	0x40002000	4K
ADC0	0x40003000	4K
ACMP0	0x40005000	4K
SPM	0x40008000	1K
RTC	0x40008400	1K

APB 存储映射	基地址	大小 (字节)
WDG	0x4000B000	4K
SPI0	0x4000C000	4K
I2C0	0x4000E000	4K
Cortex™-M0+ controller	0x40010000	2K
TIMER	0x40011000	4K
PWM0	0x40013000	4K
PWM1	0x40014000	4K
PWM2	0x40015000	4K
CTU	0x40016000	4K
PWDT0	0x40017000	2K
UART0	0x40018000	4K
UART1	0x4001A000	4K

3 复位 (RESET)

3.1 特性

上电复位 (POR) :

- POR Reset: IC 上电复位

系统复位:

- External Reset: 外部复位脚复位, 低电平有效
- LVR Reset: 低压检测复位, 低电平有效
- Software Reset: Cortex™-M0+ 软件系统复位
- ECC 2 Bit Error Reset: SRAM ECC 检测到 2bit 错误复位, 默认不使能
- Lock up Reset: Cortex™-M0+死锁复位
- Watchdog Reset Normal Mode: 看门狗定时器复位, 正常模式有效
- Watchdog Reset Stop Mode: 看门狗定时器复位, 停止模式有效
- XOSC Loss Reset: 当检测到外部 XOSC 时钟异常时, 将会产生一个可屏蔽的系统复位

每个系统复位源在 [RESET_STATUS](#) 寄存器都有相应的状态位。

3.2 结构框图

模块结构框图如图 3-1 所示。各 Reset 信号默认都是高电平, 任意一信号为低电平, 即产生复位信号。

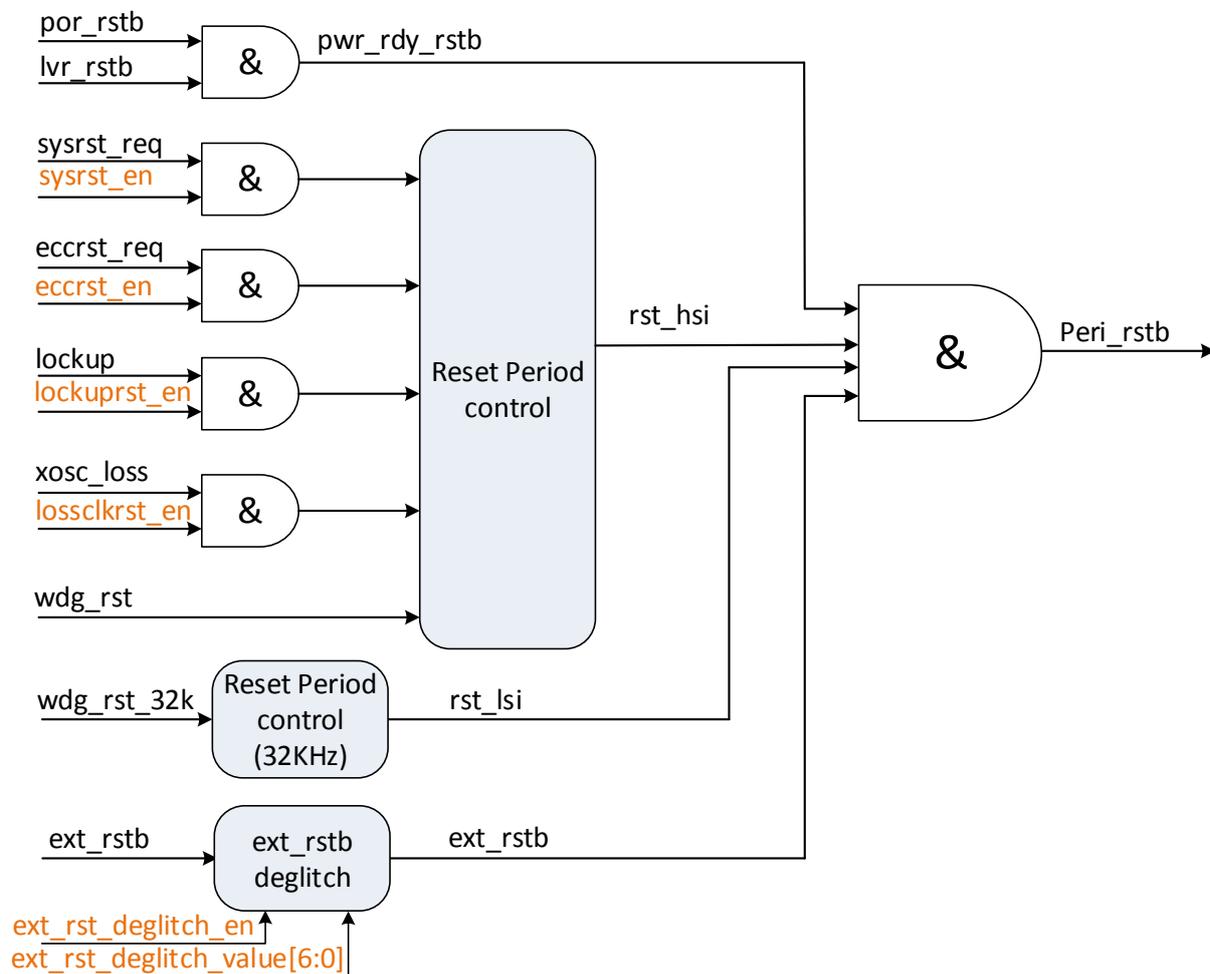


图 3-1 Reset 模块结构框图

3.3 功能描述

3.3.1 上电复位 (POR)

产生 POR 的条件为:

1. MCU 初次上电;
2. 电源电压下降到低于上电复位电压电平 (VPOR)。

随着电源电压上升, 低电压检测 (LVR) 会将 MCU 保持在复位状态, 直到电源电压上升至高于低电压检测 (LVR) 的低阈值 (VLVRL), 具体请参考《ATC_AC7802x_Datasheet_CH》表 6-2 LVD/POR/AVDD 电压告警规格。

3.3.2 系统复位 (System Reset)

系统复位开始时，片上稳压器处于完全稳压状态，系统时钟来源于内部基准时钟。处理器退出复位状态后，执行下列操作：

- 从向量表偏移 0 处读取 SP(SP_main)初始值；
- 从向量表偏移 4 处读取程序计数器 (PC) 初始值；
- 连接寄存器 (LR) 设置为 0xFFFF_FFFF。

3.3.2.1 外部引脚复位 (External Reset)

MCU 专用引脚，用于复位和重启 MCU 所有的模块。由于该引脚是低电平有效，建议在外围 PCB 中加上拉电阻以防止噪声。外部引脚复位功能可以设置寄存器 EXT_RST_EN=0 为屏蔽状态。

3.3.2.2 低压检测复位 (LVR Reset)

集成了一个低压保护系统，以便在电源电压发生变化期间保护存储器内容和控制 MCU 系统状态。该系统由上电复位 POR 电路和 LVR 电路组成，LVR 可以配置为不同的复位基准，由寄存器 LVR_THL 决定复位阈值高档或低档电压。

具体数值，请参考《ATC_AC7802x_Datasheet_CH》5.1.1 DC 特性章节。

3.3.2.3 软件系统复位 (Software Reset)

配置 RESET_CTRL[24]=1 将使能 Cortex™-M0+软件系统复位。将内核发出软件复位请求以产生系统复位。

3.3.2.4 ECC 2 bit 错误复位 (2 Bit Error Reset)

配置 RESET_CTRL[23]=1 将使能 SRAM ECC 2 Bit 错误复位。当 ECC 检测到 2 Bit 错误后，将发出系统复位请求以产生系统复位。

3.3.2.5 死锁复位 (Lock up Reset)

死锁 (Lock up) 表明内核软件出现严重错误。这是由于激活处理器内置系统状态保护硬件后出现无法恢复的异常情况而导致内核锁定的结果。

死锁 (Lock up) 出现时，可自动产生复位以恢复系统。

3.3.2.6 看门狗定时器复位 (Watchdog Reset)

看门狗定时器 (WDG)通过软件定时刷新来对系统进行监控。

如果周期性刷新没有出现，看门狗将发送系统复位。

更多细节的部分，请参考 [17 看门狗模块 \(WDG\)](#)。

3.3.2.7 晶体振荡器 (XOSC) 监控器功能

配置 `CKGEN_CTRL[16]=1` 使能 XOSC 监控系统。使能 XOSC 监控系统后，时钟检测器在 HSE 振荡器启动延迟后启用，并在此振荡器停止时被禁用。如果在 HSE 振荡器时钟上检测到故障，则该振荡器会被自动禁用，同时 XOSC 失效状态触发 NMI 中断以通知 MCU 执行相应的救援操作，如切换内部时钟。或者检测到 XOSC 失效后产生复位信号使系统复位。

3.4 寄存器定义

表 3-1 复位 (Reset) 寄存器映射

CKGEN 基地址: `0x40000000`

地址	名称	宽度 (位)	描述
CKGEN 基地址+0x0C	<code>RESET_CTRL</code>	32	芯片复位控制
CKGEN 基地址+0x10	<code>RESET_STATUS</code>	32	芯片复位状态

3.4.1 复位控制寄存器(RESET_CTRL)

表 3-2 RESET_CTRL 寄存器

RESET_CTRL		芯片复位控制														Reset:0x0f418004	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称					EXT_RST_EN	XOSC_LOSS_RST_EN	CPU_LOCK_UP_RST_EN	CPU_SYS_RST_EN	ECC2_RST_EN								
访问					RW	RW	RW	RW	RW								
Reset					1	1	1	1	0								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称									EXT_RST_DEGLITCH_VALUE							EXT_RST_DEGLITCH_EN	
访问									RW							RW	
Reset									0	0	0	0	0	0	1	0	0

字段	说明
27 EXT_RST_EN	外部复位使能 0: 不能产生 IC 复位 1: 能产生 IC 复位
26 XOSC_LOSS_RST_EN	XOSC 异常复位使能 0: 不能产生 IC 复位 1: 能产生 IC 复位
25 CPU_LOCKUP_RST_EN	CPU 死锁复位使能 0: 不能产生 IC 复位 1: 能产生 IC 复位
24 CPU_SYSRST_EN	CPU 软件系统复位使能 0: 不能产生 IC 复位 1: 能产生 IC 复位
23 ECC2_RST_EN	SRAM ECC 2bit 错误复位使能 0: 不能产生 IC 复位 1: 能产生 IC 复位
22:8 保留	保留
7:1 EXT_RST_DEGLITCH_VALUE	外部复位过滤基数 以 32K 为计数时钟源的周期为基准，如值为 2 时可以过滤掉 1 个周期即 31us 以下复位信号
0 EXT_RST_DEGLITCH_EN	外部复位的毛刺过滤功能使能 0: 禁止 1: 使能

3.4.2 复位状态寄存器(RESET_STATUS)

表 3-3 RESET_STATUS 寄存器

RESET_STATUS		芯片复位状态												Reset:0x8000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CLEAR_RESET_STATUS															
访问	RW															
Reset	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	XOSC_LOSS_RST_STATUS						CPU_LOCKUP_RST_STATUS	CPU_SYSRESET_STATUS	WDG_RESET_STATUS	WDG_32K_RESET_STATUS	ECC2_RESET_STATUS	EXT_RESET_STATUS	LVR_RESET_STATUS	POR_RESET_STATUS		
访问	RO						RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0						0	0	0	0	0	0	0	0	0	0

字段	说明
16 CLEAR_RESET_STATUS	清除复位状态 0: 允许复位状态更新 1: 清除所有复位状态
9 XOSC_LOSS_RST_STATUS	XOSC 异常复位状态 0: 无效 1: 有效
8 保留	保留
7 CPU_LOCKUP_RST_STATUS	CPU 死锁复位状态 0: 无效 1: 有效
6 CPU_SYSRESET_STATUS	CPU 系统复位状态 0: 无效 1: 有效
5 WDG_RESET_STATUS	看门狗正常模式下复位状态 0: 无效 1: 有效
4 WDG_32K_RESET_STATUS	看门狗低功耗模式下复位状态

字段	说明
	0: 无效 1: 有效
3 ECC2_RESET_STATUS	SRAM ECC 2bit 错误复位状态 0: 无效 1: 有效
2 EXT_RESET_STATUS	外部引脚复位状态 0: 无效 1: 有效
1 LVR_RESET_STATUS	LVR 复位状态 0: 无效 1: 有效
0 POR_RESET_STATUS	POR 复位状态, 0: 无效 1: 有效

4 时钟 (Clock)

4.1 简介

时钟控制模块为 MCU 提供时钟源选择。

4.2 结构框图

4.2.1 时钟控制结构框图

时钟模块包含如下时钟源：

- 高速内部时钟 (HSI)：内部 RC 振荡器提供 32MHz 时钟源
- 外部高速时钟 (HSE)：外部 OSC 提供 8MHz ~20MHz 晶振或外部最高 20MHz 的时钟输入
- 低速内部时钟 (LSI)：内部低速 RC OSC 提供 32kHz 时钟源

每个外设都有专用的时钟使能信号来控制时钟的开关，请参阅 [4.3 寄存器定义](#) 章节以了解详细地址。

【注意】

- 系统时钟高达 32MHz
- hclk(AHB)最高可达 32MHz
- pclk(APB)最高可达 16MHz。因此当 hclk 为 32MHz 时，APBCLK_DIV 可以配置为 2 分频。

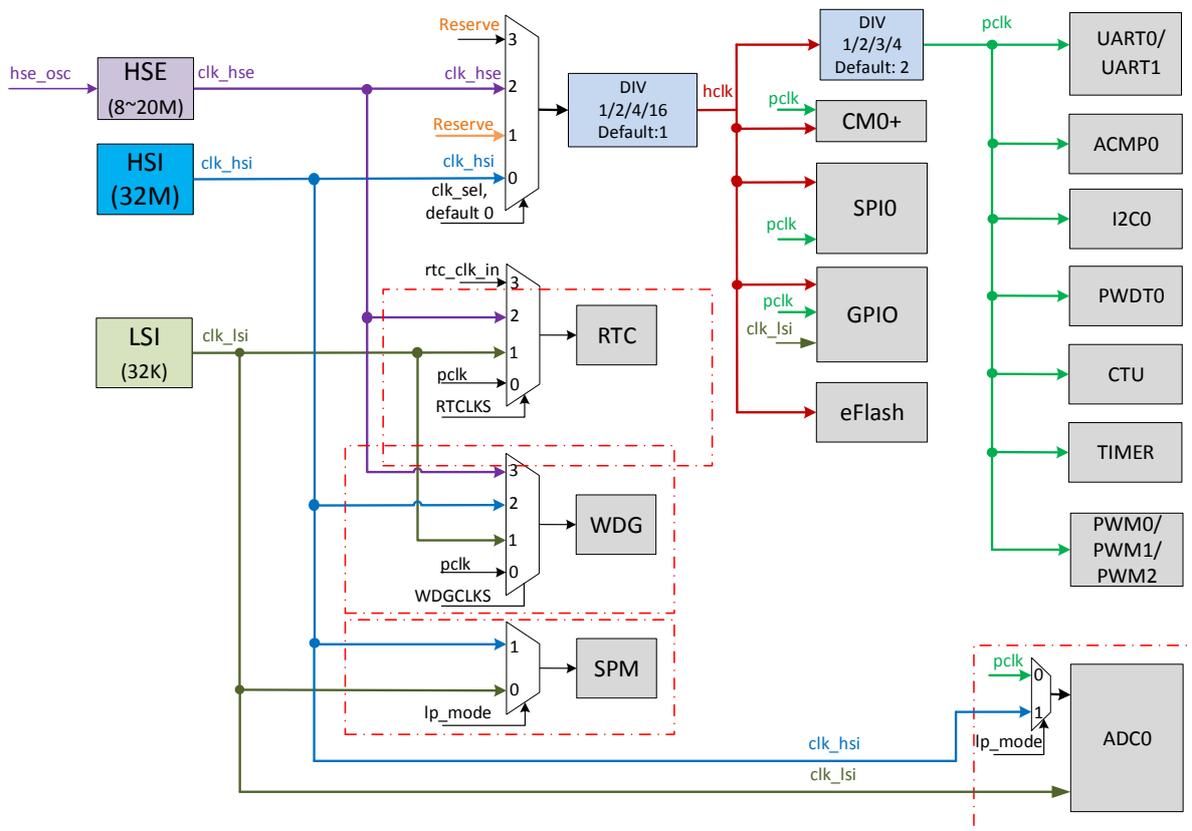


图 4-1 时钟控制结构

4.3 寄存器定义

表 4-1 时钟寄存器映射

CKGEN 基地址: 0x4000000

地址	名称	宽度	描述
CKGEN 基地址+0x00	CKGEN_CTRL	32	时钟控制寄存器
CKGEN 基地址+0x04	CKGEN_PERI_CLK_EN_0	32	外设时钟使能控制 0
CKGEN 基地址+0x08	CKGEN_PERI_CLK_EN_1	32	外设时钟使能控制 1
CKGEN 基地址+0x18	CKGEN_PERI_SFT_RST0	32	外设软件复位控制 0
CKGEN 基地址+0x1C	CKGEN_PERI_SFT_RST1	32	外设软件复位控制 1

4.3.1 控制寄存器(CKGEN_CTRL)

表 4-2 CKGEN_CTRL 寄存器

CKGEN_CTRL 时钟控制寄存器																
Reset: 0x00000100																
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																XO SC_ MO

																	N_EN
访问																	RW
Reset																	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称							APBCLK_DIV				SYSCLK_DIV				SYSCLK_SEL		
访问							RW				RW				RW		
Reset							0	1			0	0			0	0	

字段	说明
31:17 保留	保留
16 XOSC_MON_EN	XOSC 监视器使能 0 : 监视器功能禁用 1 : 监视器功能使能
9:8 APBCLK_DIV	APB 时钟由系统时钟分频产生 00b : 1 分频 01b : 2 分频 10b : 3 分频 11b : 4 分频
5:4 SYSCLK_DIV	系统时钟分频器 00b : 1 分频 01b : 2 分频 10b : 4 分频 11b : 16 分频
3:2 保留	保留
1:0 SYSCLK_SEL	系统时钟源选择 00b : 内部振荡器 HSI 01b : 保留 10b : 外部振荡器 HSE 11b : 保留

4.3.2 外设时钟使能寄存器 0(CKGEN_PERI_CLK_EN_0)

表 4-3 CKGEN_PERI_CLK_EN_0 寄存器

CKGEN_PERI_CLK_EN_0				外设时钟使能控制 0								Reset: 0x02800001				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称							WDG_EN		GPI_O_EN			RTC_EN	TIMER_EN			
访问							RW		RW			RW	RW			
Reset							1		1			0	0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			PWM2_EN	PWM1_EN	PWM0_EN	DT0_EN		I2C0_EN		SPI0_EN					UART1_EN	UART0_EN
访问			RW	RW	RW	RW		RW		RW					RW	RW
Reset			0	0	0	0		0		0					0	1

【说明】写外设寄存器之前需使能对应外设的时钟，否则将产生 Hardfault。

字段	说明
31:26 保留	保留
25 WDG_EN	WDG 时钟使能 0：时钟禁用 1：时钟使能
24 保留	保留
23 GPIO_EN	GPIO AHB 时钟使能 0：时钟禁用 1：时钟使能
22:21 保留	保留
20 RTC_EN	RTC 时钟使能 0：时钟禁用 1：时钟使能
19 TIMER_EN	TIMER 时钟使能 0：时钟禁用 1：时钟使能
18:14 保留	保留

字段	说明
保留	
13 PWM2_EN	PWM2 定时器时钟使能 0：时钟禁用 1：时钟使能
12 PWM1_EN	PWM1 定时器时钟使能 0：时钟禁用 1：时钟使能
11 PWM0_EN	PWM0 定时器时钟使能 0：时钟禁用 1：时钟使能
10 PWDT0_EN	PWDT0 时钟使能 0：时钟禁用 1：时钟使能
9 保留	保留
8 I2C0_EN	I2C0 时钟使能 0：时钟禁用 1：时钟使能
7 保留	保留
6 SPI0_EN	SPI0 时钟使能 0：时钟禁用 1：时钟使能
5:2 保留	保留
1 UART1_EN	UART1 时钟使能 0：时钟禁用 1：时钟使能
0 UART0_EN	UART0 时钟使能 0：时钟禁用 1：时钟使能

4.3.3 外设时钟使能寄存器 1(CKGEN_PERI_CLK_EN_1)

表 4-4 CKGEN_PERI_CLK_EN_1 寄存器

CKGEN_PERI_CLK_EN_1 外设时钟使能控制 1 **Reset:0x00000000**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													ACMP0_EN	ADC0_EN	CTU_EN	
访问													RW	RW	RW	
Reset													0	0	0	

【说明】写外设寄存器之前需使能对应外设的时钟，否则将产生 Hardfault。

字段	说明
31:4 保留	保留
3 ACMP0_EN	ACMP0 时钟使能 0：时钟禁用 1：时钟使能
2 ADC0_EN	ADC0 时钟使能 0：时钟禁用 1：时钟使能
1 CTU_EN	CTU 时钟使能 0：时钟禁用 1：时钟使能
0 保留	保留

4.3.4 外设复位寄存器 0(CKGEN_PERI_SFT_RST0)

表 4-5 CKGEN_PERI_SFT_RST0 寄存器

CKGEN_PERI_SFT_RST0 外设软件复位控制 0 **Reset:0x02900001**

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称							SRST_WDG		SRST_GPIO				SRST_RTC	SRST_TIMER		
访问							RW		RW				RW	RW		
Reset							1		1				1	0		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

名称		SRS T_P WM 2	SR ST_ PW M1	SR ST_ PW M0	SR ST_ PW DT 0		SR ST_ I2C 0		SR ST_ SPI 0			SR ST_ UA RT 1	SR ST_ UA RT 0
访问		RW	RW	RW	RW		RW		RW			RW	RW
Reset		0	0	0	0		0		0			0	1

字段	说明
31:26 保留	保留
25 SRST_WDG	Watch dog 定时器软件复位 0：复位有效 1：复位无效
24 保留	保留
23 SRST_GPIO	GPIO AHB 软件复位 0：复位有效 1：复位无效
22:21 保留	保留
20 SRST_RTC	RTC 软件复位 0：复位有效 1：复位无效
19 SRST_TIMER	TIMER 软件复位 0：复位有效 1：复位无效
18:14 保留	保留
13 SRST_PWM2	PWM2 软件复位 0：复位有效 1：复位无效
12 SRST_PWM1	PWM1 软件复位 0：复位有效 1：复位无效
11 SRST_PWM0	PWM0 软件复位 0：复位有效 1：复位无效

字段	说明
10 SRST_PWDTO	PWDTO 软件复位 0：复位有效 1：复位无效
9 保留	保留
8 SRST_I2C0	I2C0 软件复位 0：复位有效 1：复位无效
7 保留	保留
6 SRST_SPI0	SPI0 软件复位 0：复位有效 1：复位无效
5:2 保留	保留
1 SRST_UART1	UART1 软件复位 0：复位有效 1：复位无效
0 SRST_UART0	UART0 软件复位 0：复位有效 1：复位无效

4.3.5 外设复位寄存器 1(CKGEN_PERI_SFT_RST1)

表 4-6 CKGEN_PERI_SFT_RST1 寄存器

CKGEN_PERI_SFT_RST1											外设软件复位控制 1				Reset:0x00000010		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称												SR ST_ AN A_ RE G	SR ST_ AC MP 0	SR ST_ AD CO	SR ST_ CT U		
访问													RW	RW	RW	RW	
Reset													1	0	0	0	

字段	说明
31:5 保留	保留
4 SRST_ANA_REG	ANA 寄存器软复位 0：复位有效 1：复位无效
3 SRST_ACMP0	ACMP0 软件复位 0：复位有效 1：复位无效
2 SRST_ADC0	ADC0 软件复位 0：复位有效 1：复位无效
1 SRST_CTU	CTU 软件复位 0：复位有效 1：复位无效
0	保留

5 电源模式 (Power Modes)

5.1 简介

本章介绍 AC7802x 各种电源模式及其各个模块在这些模式下的功能。

5.2 功能描述

支持运行 (Run)、休眠 (Sleep) 和停止 (Stop) 模式。运行 (Run)、休眠 (Sleep)、停止 (Stop) 模式下能保持 I/O 状态。

- 运行模式 – CPU 时钟可在全速状态下运行；
- 休眠模式 – CPU 进入休眠模式，系统时钟和总线时钟仍在运行；
- 停止模式 – CPU 进入深度休眠模式，部分模块能够唤醒 CPU。

5.3 应用说明

5.3.1 进入和退出低功耗模式

1. 使用 **SPM_EN_PERIPH_WKUP** 使能需要的唤醒源；
2. 将已使能的模块去能；
3. 调用 **WFI** 指令进入低功耗模式；
4. 处理器通过中断退出低功耗模式；
5. 重新将模块使能。

说明：

1. GPIO/RTC/WDG 不需要去能；
2. Stop 模式下，UART 模块关闭不会影响唤醒。

5.3.2 低功耗模式下的模块操作

下表说明了该芯片处于各低功耗模式时每个模块的功能，表中显示了标准特性及某些例外情况。

表 5-1 低功耗模式下的模块功能

模块	休眠模式	停止模式
CM0+	待机	待机
SRAM	开启	待机

片内 Flash	开启	关闭
I2C	开启	待机 ¹
SPI	开启	待机 ²
WDG	开启	可选开启
PWDT	开启	关闭
UART	开启	待机 ³
TIMER	开启	关闭
PWM	开启	关闭
CTU	开启	关闭
RTC	开启	可选开启
SPM	开启	开启
XOSC	开启	关闭
HSI(32MHz)	开启	关闭
LSI(32kHz)	开启	开启
GPIO	开启	开启 ⁶
ADC	开启	待机 ⁵
LVR	可选开启	可选开启
LVD	可选开启	可选开启 ⁷
ACMP	可选开启	待机 ⁸
DAC	可选开启	待机
T-sensor	可选开启	关闭
DIGLDO	开启	低功耗
FLHLDO	开启	关闭
POR	开启	开启
BG	开启	可选开启 ⁹

【注意】

- 1: 支持停止模式下的地址匹配唤醒。
- 2: 支持停止模式下的从机模式接收和唤醒。
- 3: 支持停止模式下的边沿唤醒(UART RX 管脚低电平直接走 SPM)。
- 4: 支持停止模式下的边沿唤醒，可选择开启过滤器。
- 5: 支持停止模式下的 ADC 唤醒。
- 6: 停止模式 I/O 状态保持，支持所有 GPIO 中断唤醒。
- 7: 支持停止模式下的 LVD Warning 中断唤醒。
- 8: 支持停止模式下的 ACMP 设定电压比较唤醒。
- 9: 在停止模式使能 ADC 唤醒，或者使能 LVR，LVD 则 BG 开启，否则关闭 BG。

【说明】

- 开启：表示模块的 Power 和 Clock 均正常提供。

- 待机：表示模块的 Power 正常，Clock 关闭。
- 关闭：表示模块的 Power 和 Clock 均关闭。

6 系统电源管理 (SPM)

6.1 简介

系统电源管理 SPM (System Power Management) 提供了灵活的系统管理, 包含休眠/唤醒功能、电源域管理和各模块功耗控制。

6.2 特性

- 支持停止 (Stop) 模式下电源管理

6.3 应用说明

6.3.1 SPM 电源控制编程指南

AC7802x 支持停止 (Stop) 模式。

停止 (Stop) 模式下, 各模块电源工作状态以及唤醒源可参考表 5-1。

WFI 指令调用芯片的停止模式, 处理器通过中断指令退出低功耗模式。

编程顺序:

1. 配置唤醒源正常工作, 并能正常产生中断;
2. 设置唤醒源: `SPM_EN_PERIPH_WKUP`;
3. 使能 SPM 电源控制: `PWR_EN`;
4. 执行 WFI 指令。

6.3.2 晶体振荡器 (XOSC) 电源控制

XOSC 默认处于关闭状态。需要时, 通过配置 `SPM_PWR_MGR_CFG1`, 打开或关闭 XOSC。

SPM 寄存器 `SPM_PWR_MGR_CFG1`:

- `XOSC_HSEON`: 外部高速时钟使能。
- `XOSC_HSEBYP`: 外部高速时钟旁路。

当相应的位设置为 1'b1 时, SPM 将按照上电顺序为 XOSC 供电, 可能需要花上一些时间。因此, 在使用之前, 软件需要等待 XOSC 上电完成和时钟就绪。

XOSC 上电状态可通过读取 SPM 寄存器 `SPM_PWR_MGR_CFG1` 来确定。

- `XOSC_RDY`: 外部高速时钟就绪标志。

当芯片从停止模式唤醒时，SPM 将使 XOSC 保持打开或关闭状态，与休眠前的状态相同。

6.4 寄存器定义

表 6-1 SPM 寄存器映射

SPM 基地址：0x40008000

地址	名称	宽度	描述
SPM 基地址 + 0x00	SPM_PWR_MGR_CFG0	32	电源管理器配置寄存器 0
SPM 基地址 + 0x04	SPM_PWR_MGR_CFG1	32	电源管理器配置寄存器 1
SPM 基地址 + 0x0C	SPM_PERIPH_SLEEP_ACK_STATUS	32	外设休眠应答状态
SPM 基地址 + 0x10	SPM_EN_PERIPH_SLEEP_ACK	32	外设休眠应答使能寄存器
SPM 基地址+ 0x14	SPM_EN_PERIPH_WKUP	32	外设唤醒使能寄存器
SPM 基地址+ 0x1C	SPM_WAKEUP_IRQ_STATUS	32	唤醒状态标志寄存器

6.4.1 电源管理器配置寄存器 0(SPM_PWR_MGR_CFG0)

表 6-2 SPM_PWR_MGR_CFG0 寄存器

SPM_PWR_MGR_CFG0 电源管理配置寄存器 0 Reset:0x00000018

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称								EN_IO_S US				EN_LV R	EN_DPWRLV R	EN_LV D	EN_FAS T_ BOOT	PWR_E N	
访问								RW				RW	RW	RW	RW	RW	
Reset								0				1	1	0	0	0	

字段	说明
31:8 保留	保留
7 EN_IO_SUS	在停止模式禁用 IO 0: 在进入停止 (Stop) 时, I/O 状态保持 1: 在进入停止 (Stop) 时, 禁用 I/O
6:5 保留	保留
4 EN_LVR	使能低电压检测复位

字段	说明
	0: 禁用 1: 使能 检测芯片 VCC 电压, 如果 VCC 欠压, 触发 LVR 复位
3 EN_DPWRLVR	使能 LDO 低电压检测 0: 禁用 1: 使能 检测芯片内部 LDO 电压, 如果 LDO 欠压, 同样触发 LVR 复位
2 EN_LVD	使能低电压检测警告 0: 禁用 1: 使能 检测芯片 VCC 电压, 如果 VCC 欠压, 触发 LVD 中断警告
1 EN_FAST_BOOT	使能快速唤醒启动模式 0: 禁用 1: 使能 快速唤醒启动模式: 在休眠过程中, 在接受到唤醒中断时, 芯片会停止休眠时序并立即唤醒
0 PWR_EN	SPM 电源控制使能 0: 禁用 1: 使能 SPM 电源控制

6.4.2 电源管理器配置寄存器 1 (SPM_PWR_MGR_CFG1)

表 6-3 SPM_PWR_MGR_CFG1 寄存器

SPM_PWR_MGR_CFG1				电源管理器配置寄存器 1																Reset:0x00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
名称	XOSC_RDY		XOSC_HSEON	XOSC_HSEBYP																	
访问	RO		RW	RW																	
Reset	0		0	0																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
名称													LVD_THL	LVR_THL							
访问													RW	RW							
Reset													0	0							

字段	说明
31	XOSC 时钟就绪标志

字段	说明
XOSC_RDY	0: 未就绪 1: 就绪
30 保留	保留
29 XOSC_HSEON	外部高速时钟使能 0: 禁用 XOSC 1: 使能 XOSC
28 XOSC_HSEBYP	外部高速时钟旁路 0: 禁用外部时钟旁路振荡器 1: 使用外部时钟旁路振荡器（外部时钟输入）
27:4 保留	保留
3 LVD_THL	LVD 电压阈值 0: 低档 2.9V 1: 高档 4.6V
2 LVR_THL	LVR 电压阈值 0: 低档 2.6V 1: 高档 4.3V
1:0 保留	保留

6.4.3 外设休眠应答状态(SPM_PERIPH_SLEEP_ACK_STATUS)

表 6-4 SPM_PERIPH_SLEEP_ACK_STATUS 寄存器

SPM_PERIPH_SLEEP_ACK_STATUS													外设休眠应答状态			Reset:0xffffffff	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称														EFLASH		ADC0	
访问														RO		RO	
Reset														1		1	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称					UART1		UART0					SPI0		I2C0		ACMP0	
访问					RO		RO					RO		RO		RO	
Reset					1		1					1		1		1	

字段	说明
31:19 保留	保留
18 EFLASH	片内 eFlash 空闲(idle)状态 0: 忙 1: 空闲
17 保留	保留
16 ADC0	ADC0 休眠确认 (sleep ack) 状态 0: 未回 ACK 1: 已回 ACK
15:11 保留	保留
10 UART1	UART1 休眠确认 (sleep ack) 状态 0: 未回 ACK 1: 已回 ACK
9 UART0	UART0 休眠确认 (sleep ack) 状态 0: 未回 ACK 1: 已回 ACK
8:5 保留	保留
4 SPI0	SPI0 休眠确认 (sleep ack) 状态 0: 未回 ACK 1: 已回 ACK
3 保留	保留
2 I2C0	I2C0 休眠确认 (sleep ack) 状态 0: 未回 ACK 1: 已回 ACK
1 保留	保留
0 ACMP0	ACMP0 休眠确认 (sleep ack) 状态 0: 未回 ACK 1: 已回 ACK

6.4.4 外设休眠应答使能(SPM_EN_PERIPH_SLEEP_ACK)

表 6-5 SPM_EN_PERIPH_SLEEP_ACK 寄存器

SPM_EN_PERIPH_SLEEP_ACK 外设休眠应答使能寄存器 Reset:0x00050615

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														EFLASH	ADC0	
访问														RW	RW	
Reset														1	1	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					UART1		UART0						SPI0	I2C0	ACMP0	
访问					RW		RW						RW	RW	RW	
Reset					1		1						1	1	1	

字段	说明
31:19 保留	保留
18 EFLASH	使能 eFlash 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能
17 保留	保留
16 ADC0	使能 ADC0 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能
15:11 保留	保留
10 UART1	使能 UART1 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能
9 UART0	使能 UART0 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能
8:5 保留	保留
4 SPI0	使能 SPI0 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能
3	保留

字段	说明
保留	
2 I2C0	使能 I2C0 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能
1 保留	保留
0 ACMP0	使能 ACMP0 休眠确认等待 (Sleep ACK Waiting) 0: 禁用 1: 使能

6.4.5 外设唤醒使能寄存器(SPM_EN_PERIPH_WKUP)

表 6-6 SPM_EN_PERIPH_WKUP 寄存器

SPM_EN_PERIPH_WKUP												外设唤醒使能寄存器				Reset:0x00028000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
名称													LVD	NMI	GPIO	ADC0			
访问													RW	RW	RW	RW			
Reset													0	0	1	0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
名称	RTC					UART1	UART0					SPI0		I2C0		ACMP0			
访问	RW					RW	RW					RW		RW		RW			
Reset	1					0	0					0		0		0			

字段	说明
31:20 保留	保留
19 LVD	使能 LVD 警告中断 0: 禁用 1: 使能 (产生中断或唤醒需使能)
18 NMI	使能 NMI 中断 0: 禁用 1: 使能 (产生中断或唤醒需使能)
17 GPIO	使能 GPIO 唤醒 0: 禁用 1: 使能

字段	说明
16 ADC0	使能 ADC0 唤醒 0: 禁用 1: 使能
15 RTC	使能 RTC 唤醒 0: 禁用 1: 使能
14:11 保留	保留
10 UART1	使能 UART1 唤醒 0: 禁用 1: 使能
9 UART0	使能 UART0 唤醒 0: 禁用 1: 使能
8:5 保留	保留
4 SPI0	使能 SPI0 唤醒 0: 禁用 1: 使能
3 保留	保留
2 I2C0	使能 I2C0 唤醒 0: 禁用 1: 使能
1 保留	保留
0 ACMP0	使能 ACMP0 唤醒 0: 禁用 1: 使能

6.4.6 唤醒状态标志寄存器(SPM_WAKEUP_IRQ_STATUS)

表 6-7 SPM_WAKEUP_IRQ_STATUS 寄存器

SPM_WAKEUP_IRQ_STATUS	唤醒状态标志寄存器																Reset:0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

名称												OVER_COUNT	LVD	NMI	GPIO	ADC0
访问												R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
Reset												0	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RTC				UART1		UART0				SPI0	I2C0		ACMP0		
访问	R/W1C				R/W1C		R/W1C				R/W1C	R/W1C		R/W1C		
Reset	0				0		0				0	0		0		

字段	说明
31: 21 保留	保留
20 OVER_COUNT	<p>SPM over count 唤醒标志位</p> <p>0: 无效 1: 进入休眠模式前，SPM 会在特定的周期内等待所有外设应答休眠确认。如果有外设 NO ACK，则该标志位置位，并退出休眠模式，产生 SPM 中断。</p> <p>注意：写“1” 清零此位</p>
19 LVD	<p>LVD 中断或唤醒标志位</p> <p>0: 无效 1: 有效</p> <p>注意：写“1” 清零此位</p>
18 NMI	<p>NMI 中断或唤醒标志位</p> <p>0: 无效 1: 有效</p> <p>注意：写“1” 清零此位</p>
17 GPIO	<p>GPIO 唤醒标志位</p> <p>0: 无效 1: 有效</p> <p>注意：写“1” 清零此位</p>
16 ADC	<p>ADC 唤醒标志位</p> <p>0: 无效 1: 有效</p> <p>注意：写“1” 清零此位</p>
15 RTC	<p>RTC 唤醒标志位</p>

字段	说明
	0: 无效 1: 有效 注意：写“1”清零此位
14:11 保留	保留
10 UART1	UART1 唤醒标志位 0: 无效 1: 有效 注意：写“1”清零此位
9 UART0	UART0 唤醒标志位 0: 无效 1: 有效 注意：写“1”清零此位
8:5 保留	保留
4 SPI0	SPI0 唤醒标志位 0: 无效 1: 有效 注意：写“1”清零此位
3 保留	保留
2 I2C0	I2C0 唤醒标志位 0: 无效 1: 有效 注意：写“1”清零此位
1 保留	保留
0 ACMP0	ACMP0 唤醒标志位 0: 无效 1: 有效 注意：写“1”清零此位

7 通用异步收发器 (UART)

7.1 简介

UART (Universal Asynchronous Receiver/Transmitter, 通用异步收发器) 是一种基本的用于串行通信的协议。它主要通过发射器和接收器来实现诸多功能。主要功能由寄存器的 2 个 bit: **LINEN**, **ILEN** 共同组成, 如表 7-1 所示。

表 7-1 功能分类和配置

功能	LINEN
BASIC UART	0
LIN	1

【说明】

1. 只有 UART0 支持软件 LIN 功能。
2. LIN 模式仅支持 8 位数据格式及 16 倍过采样。与此同时, 如果使能自动波特率功能 (LABAUDEN=1), 则同步字段数据 (0x55) 将被丢弃。

7.2 特性

- 全双工, 标准不归零 (NRZ) 格式
- 可编程波特率 (16 位分频器)
 - 支持发送或接收波特率范围 600bps~2Mbps, 波特率误差不超过 1%
- 轮询或中断方式查询状态
 - 传输数据寄存器为空且传送完成
 - 接收数据寄存器已满
 - 接收溢出错误、帧错误、奇偶错误
 - 空闲线路检测
 - 支持 LIN 的分隔符检测, 可选 10 或 11 位 LIN 功能分隔符检测
 - 通过有效的边沿检测将 MCU 从停止 (Stop) 模式唤醒
- 可编程 7/8/9 位数据, 1 或 2 位停止位, 硬件自动生成奇偶校验位
- 可选择传输器输出和接收器输入极性
- 空闲线路检测
- 可生成 13~28 位分隔符

- 轮询方式，波特率最大支持 4M bps；中断方式，波特率最大支持 2M bps（受用户中断回调函数执行效率影响）

7.3 结构框图

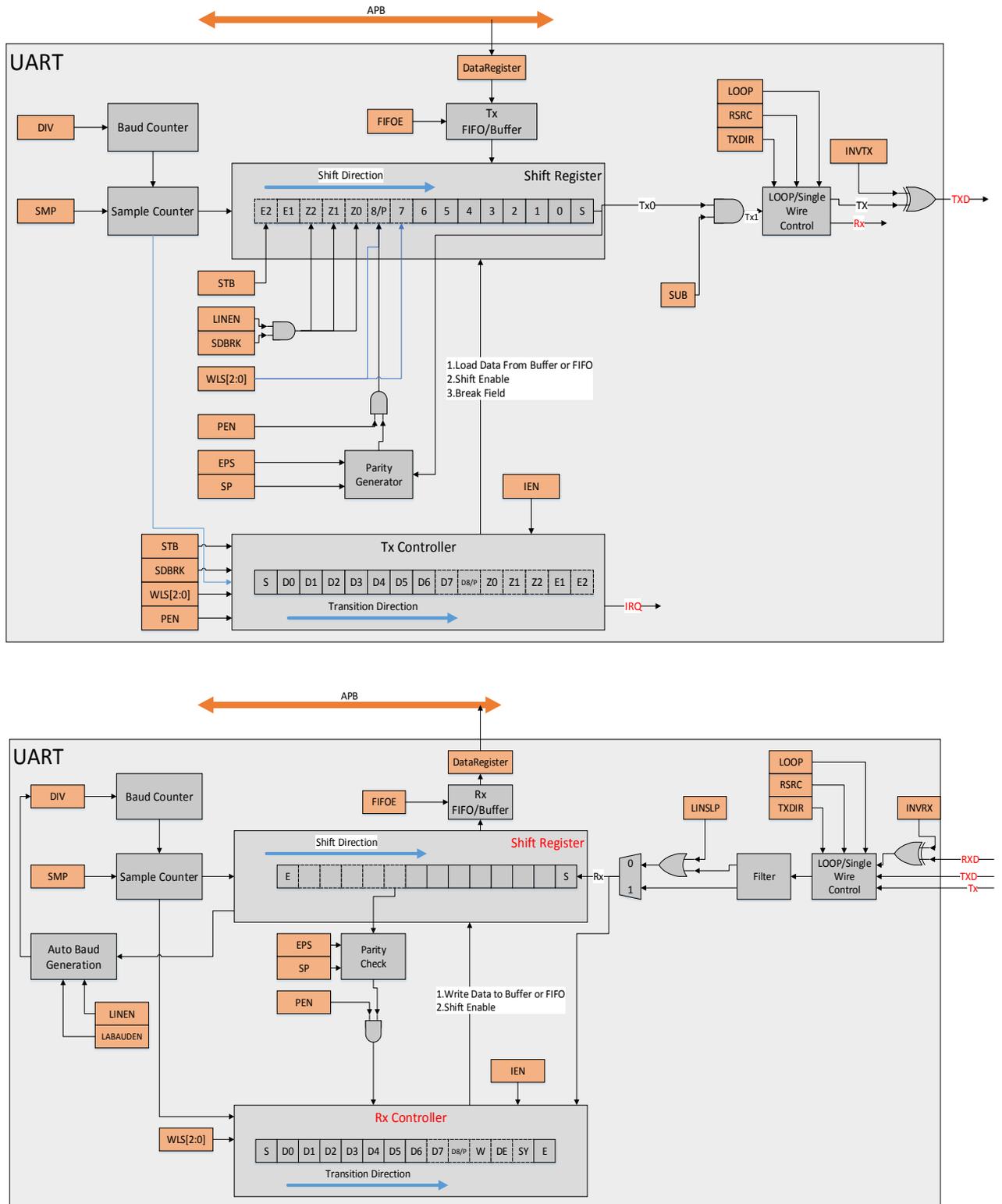


图 7-1 UART 结构框图

7.4 功能描述

UART 功能是逐位发送和接收串行数据。图 7-2 和图 7-3 描述了完整的数据位，包括起始位（start bit）、数据位（data bits）、奇偶位（parity bit）、停止位（stop bits）和保护间隔（guard time）。但 bit6, bit7, bit8, bit9, parity, stop2 位和 guard time 位可由用户配置，详细配置信息请参考 UART_LCR0 和 UART_LCR1 寄存器。一位对应于由波特率控制的一个位时间。

UART 发送和接收时有多种状态。用户最好知道在发送或接收过程中何时产生这些状态。这样，用户可以更好地使用 UART 功能。THRE 和 TC 状态位出现在图 7-2 所示的发送过程中。在全局复位或上电后的初始化状态期间，THRE 和 TC 在 TXEN 设置为 1 后会立即变为 1。但在传输过程中，THRE 会在起始位后立即变为 1，同时 TC 在最后一位后立即变为 1，比如，若 GUARDEN=1，保护时间位也会如此。



图 7-2 UART 传送器流程

如果在接收过程中发生相关事件，如图 7-3 所示，则状态位 DR, OE, PE, FE, BI 和 NE 会在 stop1 位后立即置为 1。

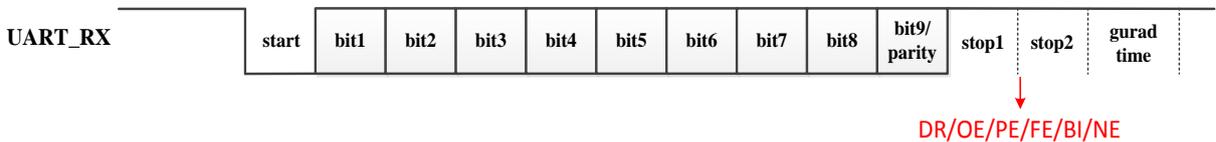


图 7-3 UART 接收器流程

需要指出的是，PE, FE 和 NE 状态仅针对当前接收数据字节，并会在下一个数据接收完成时被自动清除。对于其他状态，如果未通过读取或写入 1 来清除它们，则它们始终保持该值。

7.4.1 噪声检测 (Noise Detection)

对于 NE 状态，当 UART_RX 信号中存在噪声时就会产生此信号。为了检测噪声，UART_RX 在中间位置被采样三次，如图 7-4 所示。

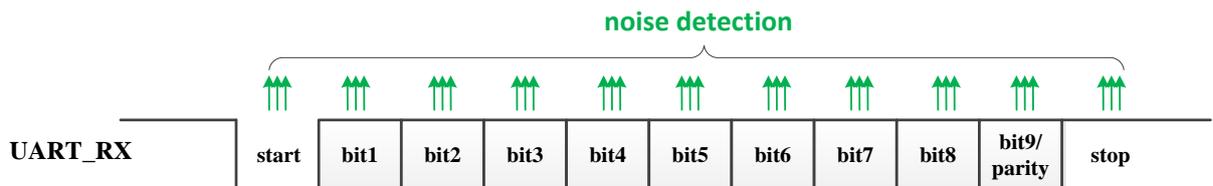


图 7-4 UART 噪声检测

为了更清楚方便地解释，三次采样的值分别叫做 SM1，SM2 和 SM3。如果 SM1，SM2 和 SM3 中有两个以上的“1”作为起始位，则起始位无效且接收器复位成再次接收。除了起始位的这种情况之外，如果 SM1，SM2 和 SM3 的值互不相同，则检测到噪声，NE 状态变为 1。

7.4.2 波特率描述

UART 波特率精度由多个方面确定，包括 UART 时钟、过采样时间等。因此，一些特定的、过高的波特率没有实现或实现起来有大量误差。表 7-2 和表 7-3 分别描述了典型波特率在不同的系统时钟下的配置及相应的误差率。

表 7-2 典型的波特率及误差率@bclock=32MHz

序号	理论值 (bps)	实际值(bps)	DIV_MAN[15:0]	DIV_FRAC[4:0]	过采样次数	误差率
1	600	599.998125	3333	11	16	0.000%
2	2400	2399.97	833	11	16	-0.001%
3	9600	9599.520024	208	11	16	-0.005%
4	19200	19201.92019	104	5	16	0.010%
5	57600	57605.76058	34	23	16	0.010%
6	115200	115107.9137	17	12	16	-0.080%
7	230400	230215.8273	8	22	16	-0.080%
8	460800	460431.6547	4	11	16	-0.080%
9	921600	927536.2319	2	5	16	0.644%
10	1843200	1828571.429	1	3	16	-0.794%

表 7-3 典型的波特率及误差率@bclock=8MHz

序号	理论值 (bps)	实际值 (bps)	DIV_MANTI[15:0]	DIV_FRAC[4:0]	过采样次数	误差率
1	600	599.9925001	833	11	16	0.001%
2	2400	2399.880006	208	11	16	0.005%
3	9600	9598.080384	52	3	16	0.020%
4	19200	19207.68307	26	1	16	0.040%
5	57600	57553.95683	8	22	16	0.080%
6	115200	115107.9137	4	11	16	0.080%
7	230400	231884.058	2	5	16	0.644%
8	460800	457142.8571	1	3	16	0.794%
9	921600	914285.7143	1	3	8	0.794%

7.4.3 LIN 功能

如图 7-5 所示，UART LIN 只是一个软件 LIN，具有传输同步间隔段（break field），同步段（synchronous field）和数据。用户可以在最新的 LIN 协议中了解更多细节。图 7-5 仅描述了一个基

本的帧。除基本 UART 寄存器外，用户还应更加注意 `UART_LINCR` 寄存器。在 UART 模块中，存在一个硬件逻辑单元，用于 LIN 检测，并且当 `LINEN` 配置为 1 时，就启用 LIN 功能。需要注意的是只有 `UART0` 支持 LIN 功能。

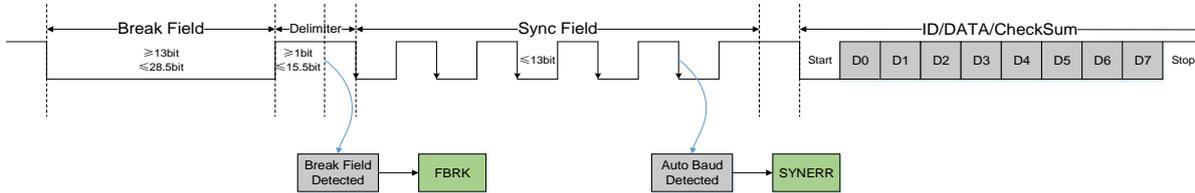


图 7-5 LIN 帧流程

UART-LIN 帧流程如图 7-5 所示，接收过程参考如下步骤：

1. 当 UART 接收到 10 (`LBRKDL=0`) 或 11 (`LBRKDL=1`) 位 0 时，UART LIN 检测逻辑单元将其视为 LIN 帧的有效同步间隔段，LIN 同步间隔段标志 `FBRK` 由硬件设置为 1，表示出现了有效的 LIN 同步间隔段。需要指出的是，LIN 同步间隔段不是数据，也不存储在 UART 接收数据寄存器或 FIFO 中。
2. 在同步间隔段之后，同步段即 `0x55` 当做正常的的数据接收。如果 `LABAUDEN` 配置为 1，则在同步段期间自动波特率检测开始运行，并且自动波特率检测操作在图 7-5 所示的第五个下降沿之后完成。但数据 `0x55` 不会存储到 RX 数据寄存器或 FIFO 中。若 `LABAUDEN` 配置为 0，则不执行自动波特率检测，数据 `0x55` 存储在 RX 数据寄存器或 FIFO 中。波特率同步功能支持 $-50\% \sim +100\%$ 的主机波特率偏差，否则可能会出现同步段错误 `SYNERR`。
3. 在同步段之后，数据都是由 UART 接收。

为避免在发生异常情况时模块暂停，引入超时机制，如图 7-5 所示：

1. LIN 从机接收到的同步间隔段最大为 28.5 位时间，超过该值则认为无效的同步间隔段。
2. 对于同步间隔段分隔符 (Break delimiter) 超过 15.5 位时间会导致模块重置接收器和 LIN 检测逻辑。
3. 如果同步段出错，硬件置位 `UART_LSR1` 的 `SYNERR` 位，如果使能 `UART_LINCR` 的 `SYNERRIE` 位，当同步出错时候，硬件产生同步段出错中断。

UART-LIN 传输过程参考如下步骤：

1. 作为软件 LIN，如何传输同步间隔段是关键步骤。当用户想要传输同步间隔段时，用户应先检查 `UART_LSR0[THRE]` 的状态。发送的同步间隔段长度取决于寄存器 `UART_BRKLN` 的配置。寄存器 `UART_BRKLN` 的取值为 0-15 分别对应 13-28 bit 的同步间隔段。此时如果 `UART_LSR0[THRE]` 值为 1 时，用户向寄存器 `UART_LINCR[SDBRK]` 写入 1 就可以发送同步间隔段。发送完成 `UART_LINCR[SDBRK]` 将由硬件自动清 0。
2. 将同步段 (`0x55`) 和其他后续数据写入 `UART_RBR/THR` 数据寄存器并作为正常数据发送。

UART-LIN 休眠唤醒功能：

1. 用户可设置 `UART_LINCR[LINSLP]` 为 1，使 LIN 处于休眠状态，该状态下不能接收数据；

- 当 LIN 处于休眠状态下，UART RX 上收到大于 150µs 的低电平唤醒信号，LIN 将退出休眠状态，如果使能 `UART_LINCR [LINWAKIE]`，则会触发 `UART_LSR1 [LINWAK]` 中断，然后可以正常接收数据；用户也可以设置 `LINSLP` 为 0，使其退出休眠状态。

7.4.4 两种电源模式

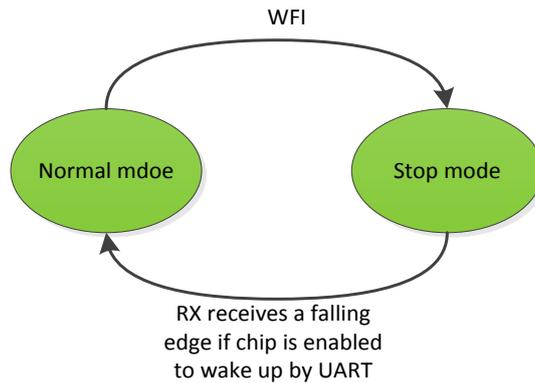


图 7-6 运行模式 (Run mode) 和停止模式 (Stop mode)

本节介绍在两种电源模式下 UART 的状态。如图 7-6 所示，用户可以执行 WFI 指令使芯片进入停止 (Stop) 模式，芯片功耗将明显降低，UART 模块的寄存器配置保持，唤醒后不需要重新配置。

在停止模式下，如果芯片被 UART 唤醒，则当 UART 接收下降沿时，芯片可以唤醒至正常模式。由于唤醒流程所需的时间，UART 通常可以在 5ms 的总时间之后立即接收数据。详细而言，TX1 可以发送 0xFF 作为下降沿，实际上其他数据也可以。特别地，如果 TX1 在唤醒流程中向 RX2 发送一些数据，则数据将在 RX2 中丢失。

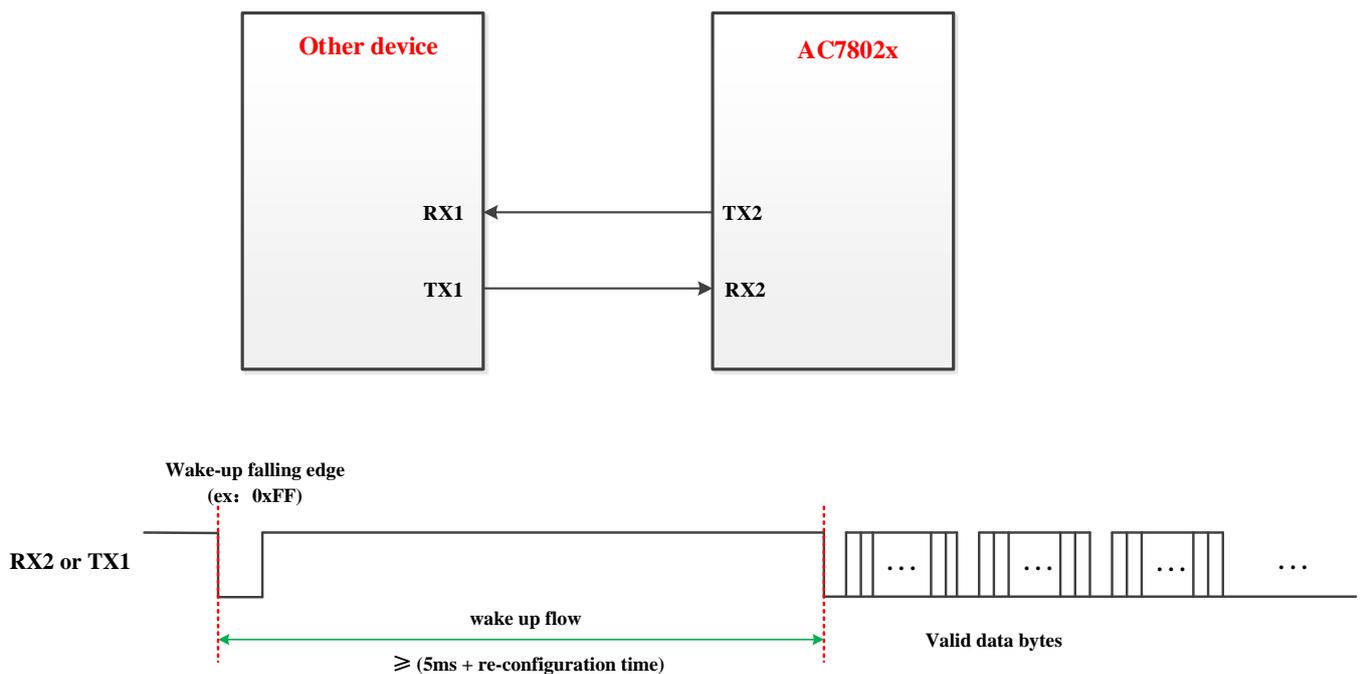


图 7-7 通过 UART 唤醒芯片的典型流程

7.5 应用说明

7.5.1 波特率配置说明

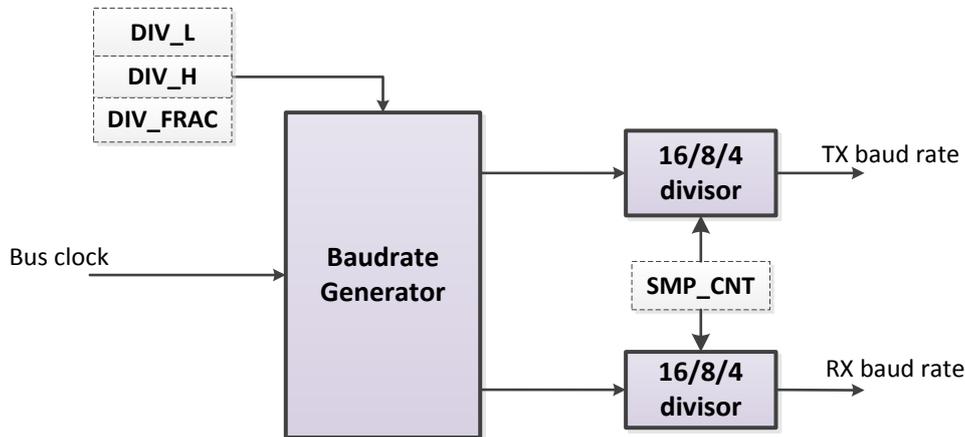


图 7-8 波特率发生器框图

如图 7-8 所示，波特率相关的配置寄存器如下所示：UART_SMP_CNT/UART_DIV_H/ UART_DIV_L /UART_DIV_FRAC。在寄存器映射表中描述了详细的信息。对于配置，如下公式用于波特率配置。

$$Baudrate = \frac{f_{clk}}{DIV * SMP_CNT}$$

在上面公式中，DIV = {UART_DIV_H, UART_DIV_L} ; SMP_CNT = UART_SMP_CNT。

例如：如果用户希望在 8 倍采样模式下以 24MHz 总线频率获得波特率 230400 bps，则可以按如下方式获得 DIV 配置。因此，UART_DIV_L=13, UART_DIV_H=0, UART_DIV_FRAC=32 * 0.0208=1。

$$DIV = \frac{24000000}{Baudrate * SMP_CNT} = \frac{24000000}{230400 * 8} = 13.0208$$

结合上面给出的例子，可以清晰地解释 UART_DIV_FRAC[4: 0]。在上面的表达式中，DIV 不是整数。如果用户删除小数部分，波特率的准确度将降低。特别是在大波特率条件下，丢弃 DIV 小数部分可能会将精度降低到较低水平，这样正常的数据传输可能会出错。

为了保持高精度，UART_DIV_FRAC[4: 0] 配置为 DIV 小数部分。由于宽度为 5 位，UART_DIV_FRAC 取值范围是 0 至 31。因此，32 乘以 DIV 小数部分生成 UART_DIV_FRAC[4: 0] 的配置值。

7.5.2 UART 配置说明

配置步骤：

1. TX/RX 数据存储模式：UART_FCR

2. 波特率： UART_DIV_L/UART_DIV_H/ UART_DIV_FRAC
 3. 数据格式： UART_LCR0/UART_LCR1
-



注意

务必注意 SUB 位。

4. 功能配置： UART_LINCR/UART_IDLE 等。
-



注意

本步骤对不同功能来说是可选的。

5. 中断使能： UART_IER
 6. 收发器使能： UART_LCR1[TXEN] / UART_LCR1[RXEN]
 7. 发送或接收数据： UART_THR/UART_RBR
-



注意

此步骤实际上在正常的发送或接收数据过程中。

【说明】

1. 对于 LIN 功能，数据格式必须配置为 8 位，没有奇偶校验，16 次过采样。
2. 对于 LIN 功能，当 LABAUDEN = 0 时，将接收 sync field 数据（0x55）并将其存储到 FIFO 或 RX 寄存器中，当 LABAUDEN = 1 时，将接收 sync field 数据（0x55），且不存储到 FIFO 或 RX 寄存器中。

7.6 寄存器定义

表 7-4 UART 寄存器映射

UART0 基地址: 0x40018000

UART1 基地址: 0x4001A000

地址	名称	宽度	描述
UARTx 基地址+0x00	UART_RBR/THR	32	TX 保持寄存器 /RX 缓冲区寄存器
UARTx 基地址+0x04	UART_DIV_L	32	分频器低 8 位
UARTx 基地址+0x08	UART_DIV_H	32	分频器高 8 位
UARTx 基地址+0x0C	UART_LCR0	32	UART 辅助控制寄存器 0
UARTx 基地址+0x10	UART_LCR1	32	UART 辅助控制寄存器 1
UARTx 基地址+0x14	UART_FCR	32	FIFO 控制寄存器
UARTx 基地址+0x1C	UART_IER	32	中断使能寄存器
UARTx 基地址+0x20	UART_LSR0	32	状态寄存器 0
UARTx 基地址+0x24	UART_LSR1	32	状态寄存器 1
UARTx 基地址+0x28	UART_SMP_CNT	32	UART 采样计数寄存器
UARTx 基地址+0x34	UART_GUARD	32	保护时间 (Guard time) 添加寄存器
UARTx 基地址+0x3C	UART_SLEEP_EN	32	休眠使能寄存器
UARTx 基地址+0x44	UART_DIV_FRAC	32	小数分频器寄存器
UARTx 基地址+0x58	UART_IDLE	32	空闲中断使能寄存器
UARTx 基地址+0x5C	UART_LINCR	32	软件 LIN 控制寄存器 注: UART1 不支持 LIN, 没有该寄存器
UARTx 基地址+0x60	UART_BRKLN	32	软件 LIN 同步间隔段控制寄存器 注: UART1 不支持 LIN, 没有该寄存器

7.6.1 RX/TX 数据寄存器(UART_RBR/THR)

表 7-5 UART_RBR/THR 寄存器

UART_RBR/THR		RX/TX 数据寄存器																Reset: 0x00000000														
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
名称																																
访问																																
Reset																																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
名称								RBR/THR																								
访问								RW																								
Reset								0																								

字段	说明
8: 0 RBR/THR	RX/TX 数据寄存器 通过访问该寄存器可以读取接收到的数据，且发送数据可以写入该寄存器。数据长度不超过 9 位。

7.6.2 分频器低 8 位寄存器(UART_DIV_L)

表 7-6 UART_DIV_L 寄存器

UART_DIV_L		分频器低 8 位寄存器								Reset: 0x00000001							
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称									DIV_L								
访问									RW								
Reset									1								

字段	说明
7: 0 DIV_L	波特率分频器 分频器低 8 位

7.6.3 分频器高 8 位寄存器(UART_DIV_H)

表 7-7 UART_DIV_H 寄存器

UART_DIV_H		分频器高 8 位寄存器								Reset: 0x00000000							
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称									DIV_H								
访问									RW								
Reset									0								

字段	说明
7: 0	波特率分频器
DIV_H	分频器高 8 位

7.6.4 控制寄存器 0(UART_LCR0)

表 7-8 UART_LCR0 寄存器

UART_LCR0																控制寄存器 0						Reset: 0x00000000					
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16											
名称																											
访问																											
Reset																											
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
名称							SUB						SP		EPS		PEN		STB		WLS1_WLS0						
访问							RW						RW		RW		RW		RW		RW						
Reset							0						0		0		0		0		0						



注意

一定要将 SUB 位配置为 0，否则 tx 在任何时候都发送 ‘0’。

字段	说明
6	设置 Break
SUB	0: 没有效果 1: SOUT 信号被强制进入 "0" 状态
5	奇偶校验位
SP	0: 没有效果 1: 根据 EPS 和 PEN 的状态，奇偶校验位强制进入自定义状态 如果 EPS = 1 & PEN = 1，奇偶校验位的设置(发送)值和检查值为 0。 如果 EPS = 0 & PEN = 1，奇偶校验位的设置(发送)值和检查值为 1。
4	选择偶校验
EPS	0: 发送并检查奇校验 1: 发送并检查偶校验
3	使能奇偶校验
PEN	0: 不传输和检查奇偶性 1: 传输和检查奇偶性

字段	说明
2 STB	<p>STOP 位个数</p> <p>0: 始终添加一个 STOP 位 1: 每个字符发送后添加两个 STOP 位, 除非添加 1 个 STOP 位时, 字符长度等于 5</p>
1: 0 WLS1_WLS0	<p>选择字长</p> <p>与 UART_LCR1 的 WLS2 组合为 WLS[2:0]</p> <p>000: 7 位 001: 7 位 010: 7 位 011: 8 位 100: 9 位 ... 111: 9 位</p>

7.6.5 控制寄存器 1(UART_LCR1)

表 7-9 UART_LCR1 寄存器

UART_LCR1		控制寄存器 1										Reset: 0x00000000					
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称									INVT X	INVR X	WLS2	LOOP				TXEN	RXEN
访问									RW	RW	RW	RW				RW	RW
Reset									0	0	0	0				0	0

字段	说明
7 INVTX	<p>确定是否反转 tx 输出, 包括 idle, break, data 位, start 位, stop 位</p> <p>0: 不反转 tx 输出 1: 反转 tx 输出</p>
6 INVRX	<p>确定是否反转 RX 输入, 包括 idle, break, data 位, start 位, stop 位</p> <p>0: 不反转 rx 输入 1: 反转 rx 输入</p>
5 WLS2	<p>确定 9 位数据模式是否可用</p> <p>0: 不可用 1: 可用</p>

字段	说明
4 LOOP	<p>循环</p> <p>0: 供用户正常使用 1: 控制 uart 进入循环模式(可以用来测试 uart 自身)</p>
1 TXEN	<p>UART 发射器使能</p> <p>0: 禁用 1: 使能</p>
0 RXEN	<p>UART 接收器使能</p> <p>0: 禁用 1: 使能</p>

7.6.6 FIFO 控制寄存器(UART_FCR)

表 7-10 UART_FCR 寄存器

UART_FCR	FIFO 控制寄存器																Reset: 0x00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
名称																					
访问																					
Reset																					
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
名称																	FIFOE				
访问																	RW				
Reset																	0				

字段	说明
0 FIFOE	<p>使能 FIFO</p> <p>0: 禁用 RX 和 TX FIFO 1: 使能 RX 和 TX FIFO</p>

7.6.7 中断使能寄存器(UART_IER)

表 7-11 UART_IER 寄存器

UART_IER		中断使能寄存器																复位值: 0x00000000						
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
名称																								
访问																								
Reset																								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
名称								ETXDF		EOEBI	ENE	EFE	EPE	ETC	ETXE	ERXNE								
访问								RW		RW	RW	RW	RW	RW	RW	RW								
Reset								0		0	0	0	0	0	0	0								

字段	说明
8 ETXDF	发送寄存器或发送 FIFO 满中断使能 0: 禁用 1: 使能
6 EOEBI	溢出错误或分隔符错误中断使能 0: 禁用 1: 使能
5 ENE	噪声错误中断使能 0: 禁用 1: 使能
4 EFE	帧错误中断使能 0: 禁用 1: 使能
3 EPE	奇偶校验错误中断使能 0: 禁用 1: 使能
2 ETC	发送完成中断使能 0: 禁用 1: 使能
1 ETXE	发送数据寄存器为空中断使能 0: 禁用 1: 使能

字段	说明
	注意: fifoe=1 表示 fifo 为空 fifoe=0 表示数据寄存器为空
0 ERXNE	接收数据寄存器非空中断使能
	0: 禁用 1: 使能
	注意: fifoe=1 表示 fifo 非空 fifoe=0 表示数据寄存器非空

7.6.8 线路状态寄存器 0(UART_LSR0)

表 7-12 UART_LSR0 寄存器

UART_LSR0		线路状态寄存器 0																Reset: 0x00000020						
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
名称																								
访问																								
Reset																								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
名称								TXDF	NE	TC	THRE	BI	FE	PE	OE	DR								
访问								RO	R/W1 C	RO	RO	R/W1 C	R/W1 C	R/W1 C	R/W1 C	RO								
Reset								0	0	0	1	0	0	0	0	0								



SUB NE/PE/FE 错误仅对目前的字节数据而言。与此同时, OE/BI 将一直存在, 直到其被清除。

字段	说明
8 TXDF	发送数据寄存器或者发送 FIFO 满标志
	0: 发送数据寄存器(fifoe=0)或者发送 FIFO (fifoe=1)不满 1: 发送数据寄存器(fifoe=0)或者发送 FIFO (fifoe=1)满
	注意: 该标志反映了数据和发送状态。
7 NE	噪声错误标志
	0: 不存在噪声错误 1: 存在噪声错误
	注意: 写 1 将此标志清除为 0。

字段	说明
6 TC	<p>传输完成标志</p> <p>0: TX FIFO(fifoe=1) 或 TX 寄存器(fifoe=0) 非空, 或发送端还没有完成数据移位。 1: TX FIFO(fifoe=1) 或 TX 寄存器(fifoe=0)为空, 发送端完成数据移位。</p> <p>注意: 上电默认值为 0, 只有 TXEN 为 1 后, TC 才起作用。将数据写入 TX FIFO(fifoe=1)/TX 寄存器(fifoe=0) 以将该标志清除为 0。对于 LIN 功能, 设置 SDBRK 位也可以将此标志清除为 0。</p>
5 THRE	<p>TX 保持寄存器或 TX FIFO 空标志</p> <p>0: 只要 TX FIFO 内容不为空, 或 TX 保持寄存器不为空 (废弃 FIFO), 执行复位操作。 1: 只要 TX FIFO 内容为空, 或 TX 保持寄存器为空 (废弃 FIFO), 执行置位操作。</p> <p>注意: 将数据写入 TX FIFO(fifoe=1)/TX 寄存器(fifoe=0)以将此标志清除为 0。</p>
4 BI	<p>分隔符错误标志</p> <p>0: 无分隔符错误 1: 产生分隔符错误。如果禁用 FIFO, 只要 SIN 保持在 0 状态 超过一个传输时间 (START 位 + DATA 位 + PARITY + STOP 位)时, 该位被置位。当中断发生时, 只有一个零字符被加载到 FIFO 或 TX 保持寄存器中。</p> <p>注意: 写 1 将此标志清除为 0。</p>
3 FE	<p>帧错误标志</p> <p>0: 无帧错误 1: 产生帧错误。如果接收数据没有一个有效的 STOP 位, 该位被置位。</p> <p>注意: 写 1 将此标志清除为 0。</p>
2 PE	<p>奇偶错误标志</p> <p>0: 无奇偶错误 1: 产生奇偶错误。没有接收到有效校验位, 该位被置位。</p> <p>注意: 写 1 将此标志清除为 0。</p>
1 OE	<p>溢出错误标志</p> <p>0: 无接收溢出错误 1: 产生接收溢出错误。如果禁用 FIFO, 如果在 RX 移位寄存器的新数据覆盖先前内容之前, CPU 未读取 RX 缓冲区, 则该位将置 1。如果使能 FIFO, 当 RX FIFO 已满且 RX 移位寄存器变满时, 会发生溢出错误。一旦发生这种情况, 就会设置 OE。然后移位寄存器中的字符被覆盖, 但不会传输到 FIFO。</p> <p>注意: 写 1 将此标志清除为 0。</p>

字段	说明
0 DR	<p>数据就绪标志</p> <p>0: 接收数据未就绪 1: 接收数据已经就绪。由 RX 缓冲区变满或 RX FIFO 非空进行置位(至少有一个字节被传输到 FIFO 中)。</p> <p>注意: 读数据寄存器 UART_RBR/THR, 或如果使能 FIFO, 读完所有 FIFO, 会自动清除此标志位为 0.</p>

7.6.9 线路状态寄存器 1(UART_LSR1)

表 7-13 UART_LSR1 寄存器

UART_LSR1		线路状态寄存器 1														Reset: 0x000000E0		
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称													UART _IDLE	LINW AK		FBRK	SYNE RR	IDLE
访问													RO	RO		R/W1 C	R/W1 C	R/W1 C
Reset													1	0		0	0	0

字段	说明
5 UART_IDLE	<p>UART_IDLE</p> <p>0: UART 正在工作中 1: UART 未工作, 也就是说, 发射器和接收器不工作或已完成数据传输或接收</p>
4 LINWAK	<p>LIN 唤醒标志</p> <p>0: 还没接收到唤醒信号 1: 已经接收到唤醒信号</p>
2 FBRK	<p>LIN Break 发生标志</p> <p>0: LIN 没有检测到 LIN 帧中的 break 字段 1: LIN 检测到 LIN 帧中的 break 字段</p> <p>注意: 写 1 将此标志清除为 0.</p>

字段	说明
1 SYNERR	<p>LIN 同步段错误标志</p> <p>0: 没有错误 1: 存在错误</p> <p>注意: 写 1 将此标志清除为 0</p>
0 IDLE	<p>IDLE 标志</p> <p>0: 尚未检测到空闲线路 1: 检测到空闲线路</p> <p>接收器已经接收到数据, 该数据后面为一个至少保持一个字节数据时间的高电平。IDLE 状态标志必须先使能空闲线路检测 ILEN 才起作用。</p> <p>注意: 写 1 将此标志清除为 0。</p>

7.6.10 采样计数器寄存器(UART_SMP_CNT)

表 7-14 UART_SMP_CNT 寄存器

UART_SMP_CNT		采样计数器寄存器														Reset: 0x00000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称																SMP_CNT		
访问																RW		
Reset																0		

字段	说明
1: 0 SMP_CNT	<p>UART 采样计数器</p> <p>00: 基于 16*baud_pulse, baud_rate = 系统时钟频率/16/{DLH, DLL} 01: 基于 8*baud_pulse, baud_rate = 系统时钟频率/8/{DLH, DLL} 10: 基于 4*baud_pulse, baud_rate = 系统时钟频率/4/{DLH, DLL} 11: 保留</p>

7.6.11 保护时间寄存器(UART_GUARD)

表 7-15 UART_GUARD 寄存器

UART_GUARD		保护时间寄存器														Reset: 0x0000000F		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		

访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称													GUAR D_EN	GUARD_CNT			
访问													RW	RW			
Reset													0	F			



添加保护时间有助于消除每个字节的累积误差，因此，通过使用具有保护时间的小数分频器来提高波特率的准确性是很重要的。

字段	说明
4 GUARD_EN	保护间隔时间添加使能信号 0: 禁用 1: 使能
3: 0 GUARD_CNT	保护间隔计数值 0~15: 0 ~ 15 位时间

7.6.12 休眠使能寄存器(UART_SLEEP_EN)

表 7-16 UART_SLEEP_EN 寄存器

UART_SLEEP_EN 休眠使能寄存器 Reset: 0x00000000

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

字段	说明
0 SLEEP_EN	休眠功能使能 0: 不处理睡眠模式指示信号 1: 当芯片进入休眠模式时，根据软件初始设置，激活硬件流程控制。当芯片唤醒时释放硬件流程。

7.6.13 小数分频器寄存器(UART_DIV_FRAC)

表 7-17 UART_DIV_FRAC 字段

UART_DIV_FRAC		小数分频器寄存器																Reset: 0x00000000														
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
名称																																
访问																																
Reset																																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DIV_FRAC															
名称													DIV_FRAC																			
访问													RW																			
Reset													0																			

字段	说明
4: 0	小数分频器
DIV_FRAC	如果实际的分频器为 135.65，则 DIV_FRAC 为 $0.65 * 32 = [20.8] = 21$ ，并且 DIV_L=135。

7.6.14 空闲中断使能寄存器(UART_IDLE)

表 7-18 UART_IDLE 寄存器

UART_IDLE		空闲中断使能寄存器																Reset: 0x00000000														
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
名称																																
访问																																
Reset																																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IDLEIE															
名称									ILEN				IDLEIE																			
访问									RW				RW																			
Reset									0				0																			

字段	说明
7	空闲线路检测使能
ILEN	0: 禁用 1: 使能
4	IDLE 中断使能
IDLEIE	0: 禁用 1: 使能

7.6.15 LIN 控制寄存器 (UART_LINCR)

表 7-19 UART_LINCR 寄存器

UART_LINCR																LIN 控制寄存器								Reset: 0x00000000								
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																
名称																																
访问																																
Reset																																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
名称									LINE	LBRK	LBRK	SDBR	LABA	SYNE	LINW	LINS																
									N	IE	DL	K	UDEN	RRIE	AKIE	LP																
访问									RW																							
Reset									0	0	0	0	0	0	0	0																

字段	说明
7 LINEN	LIN 模式使能 0: 禁用 1: 使能
6 LBRKIE	LIN Break 检测中断使能 0: 禁用 1: 使能
5 LBRKDL	LIN Break 中断检测长度 0: 10 位 1: 11 位
4 SDBRK	LIN 发送 Break 使能 0: 禁用 1: 使能发送 Break。Break 长度由寄存器 BRKLGH 决定 注意：由软件设置，并在 break 发送完成后由 MCU 内部硬件清除。
3 LABAUDEN	同步段自动波特率使能 0: 0x55 不用于自动波特率检测 1: 0x55 用于自动波特率检测
2 SYNERRIE	同步段错误中断使能 0: 禁用同步字节错误中断 1: 使用同步字节错误中断
1 LINWAKIE	LIN 唤醒中断使能 0: 禁用 LIN 唤醒中断 1: 使用 LIN 唤醒中断

字段	说明
0 LINSLP	LIN 休眠状态 0: LIN 退出休眠状态 1: LIN 进入休眠状态

7.6.16 LIN 同步间隔段控制寄存器(UART_BRKLGH)

表 7-20 UART_BRKLGH 寄存器

UART_BRKLGH		LIN 同步间隔段控制寄存器														Reset: 0x00000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称															BRKLGH			
访问															RW			
Reset															0			

字段	说明
3: 0 BRKLGH	LIN 发送 Break 长度 0000: 13bits 同步间隔段 0001: 14bits 同步间隔段 1111: 28bits 同步间隔段

8 模数转换器 (ADC)

8.1 简介

ADC 是一种 12 位逐次逼近型模拟数字转换器，拥有 18 路外部通道和 1 路内部通道，支持单次、连续、扫描或间断转换多种模式。模拟监控器特性允许应用程序监测输入电压是否超出设定的电压范围。

8.2 特性

- 12 位分辨率
- 通道输入电压范围： $AVSS < V_{in} < AVDD$
- 最大转换速率：250Ksps
- 参考电压支持 AVDD 和外部 VREF+/VREF-
- 通道：
 - 18 路外部通道
 - 1 路内部监控通道，分别可测量 1 路内部温度传感器 (T-Sensor) 和 1 路带隙基准电压 (Bandgap)
- 每路通道可单独配置采样时间。
- 转换序列分为 规则组 (regular group) 和 注入组 (injection group)
 - 规则组：最多可配置 21 个通道
 - 注入组：最多可配置 4 个通道
- 8 种操作模式 (方便起见，称为 mode x, x=1~8)
 - 规则组单通道单次转换(mode1)
 - 规则组单通道连续转换(mode2)
 - 规则组扫描+注入组扫描模式多通道单次转换(mode3 注入组扫描模式)
 - 规则组扫描+注入组间隔模式多通道单次转换(mode3 注入组间隔模式)
 - 规则组扫描+自动触发注入组扫描模式多通道单次转换(mode4)
 - 规则组扫描+注入组扫描模式多通道连续转换(mode5 注入组扫描模式)
 - 规则组扫描+注入组间隔模式多通道连续转换(mode5 注入组间隔模式)
 - 规则组扫描+自动触发注入组扫描模式多通道连续转换(mode6)
 - 规则组子组扫描模式转换(mode7)

- 注入组扫描模式转换(mode8)
- 通过内部软件触发或外部硬件触发启动 ADC
- 模拟监控器功能:
 - 配置为单个或所有通道电压检查
 - 监控通道电压是否低于低阈值或高于高阈值
- 中断:
 - 规则组各通道转换结束(EOCx, End Of Conversion, x: 0~20)
 - 注入组各通道转换结束(IEOCx, x: 0~3)
 - 模拟监控器事件(AMO,AAMO,NAMO)

8.3 结构框图

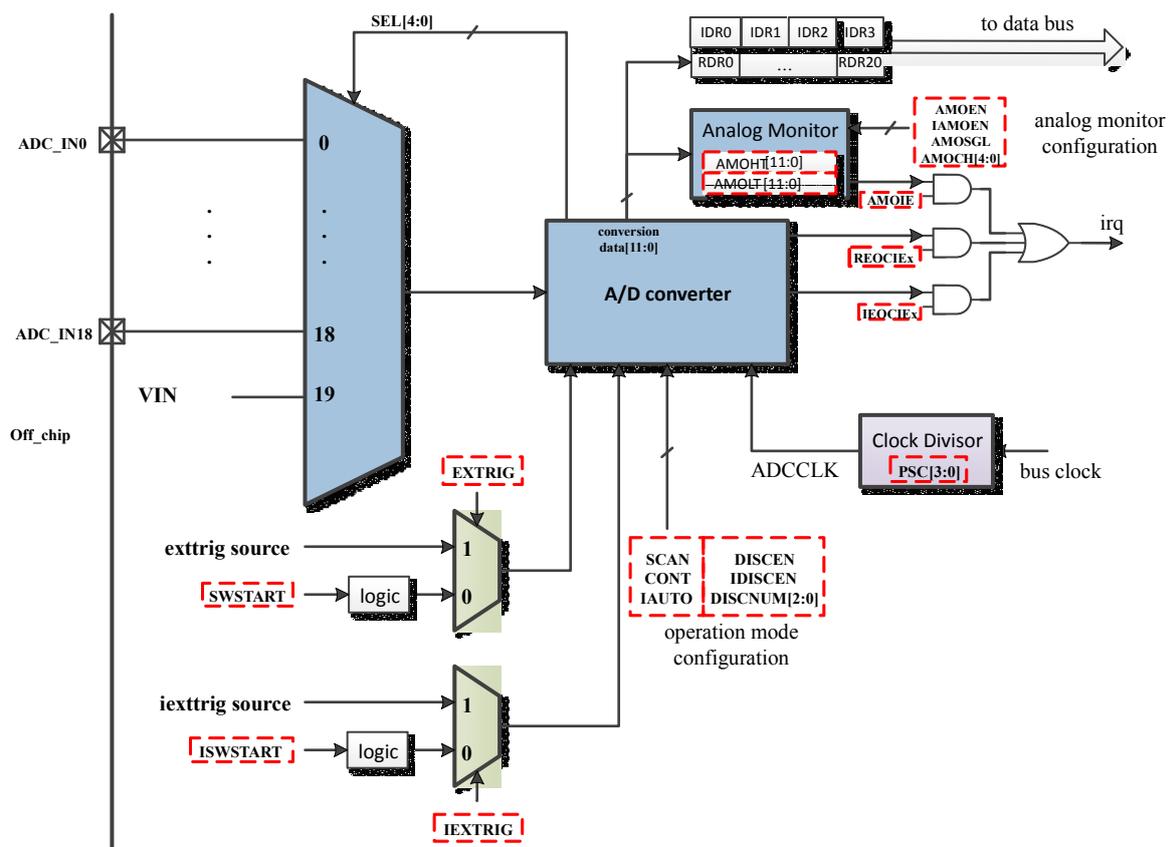


图 8-1 ADC 结构框图

8.4 功能描述

ADC 主要由 ADC 转换器单元 (converter unit)，输入通道选择器 (input channel selector)，时钟分频器 (clock divisor) 和模拟监控器 (analog monitor) 等组成。如图 8-1 所示，A/D 转换器单元工作在 ADC 时钟，简称 ADCCLK，其他电路单元工作在总线时钟。

下面介绍一个典型的操作流程。

ADC 首先上电，然后通过内部 `ADC_CTRL0[SWSTART]` 或外部触发源触发 ADC，该触发来源于其它模块。触发后 ADC 转换器单元开始工作，并将选择信号发送至输入通道选择器，根据规则或注入组通道序列逐个选择所需的通道。在一个通道完成转换后，转换结果将根据当前转换通道所属的组存储到 `ADC_RDRx` 或 `ADC_IDRx` 中，并且产生相应的 `ADC_REOC[EOCx]` 或 `ADC_IEOC[IEOCx]` 标志置位。模拟监控器工作时，如果发生相应的事件则会出现相关的状态标志。需要指出的是，不同的操作模式存在一些差异，详细信息将在后面进行说明。

8.4.1 上电时序

在开始所有功能之前，ADC 首先上电，然后有效的触发器可以启动 ADC 以基于配置的模式工作。上电时序如下图所示。

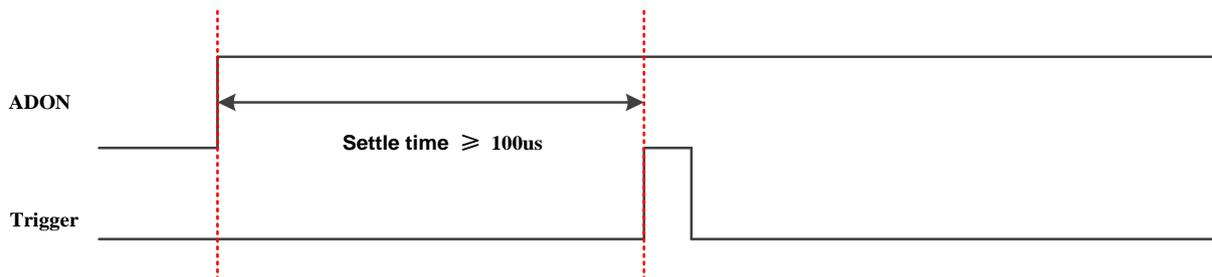


图 8-2 ADC 上电时序

如图 8-2 所示，将 `ADC_CTRL1 [ADON]` 位置为 1 以控制上电过程。在 ADON 置位后，A/D 转换器单元上电等待时间不应低于 $100\mu\text{s}$ 。

8.4.2 工作模式

根据实际应用可以灵活使用不同的转换模式，上电和有效触发后 ADC 工作于以下模式之一。

表 8-1 工作模式配置表

工作模式	MODE_BITS	触发源	转换序列
mode1	5'b0000x	规则触发	规则组单通道单次转换
mode2	5'b0100x	规则触发	规则组单通道连续转换

工作模式	MODE_BITS	触发源	转换序列
mode3 (注入组扫描模式)	5'b10000 (INTERVAL=0)	规则/注入触发	规则组扫描+注入组扫描模式 多通道单次转换
mode3 (注入组间隔模式)	5'b10000 (INTERVAL=1)	规则/注入触发	规则组扫描+注入组间隔模式 多通道单次转换
mode4	5'b10001	规则触发+自动注入触发	规则组扫描+注入组扫描模式 多通道单次转换
mode5 (注入组扫描模式)	5'b11000 (INTERVAL=0)	规则/注入触发	规则组扫描+注入组扫描模式 多通道连续转换
mode5 (注入组间隔模式)	5'b11000 (INTERVAL=1)	规则/注入触发	规则组扫描+注入组间隔模式 多通道连续转换
mode6	5'b11001	规则触发+自动注入触发	规则组扫描+注入组扫描模式 多通道连续转换
mode7	5'b1x10x	规则触发	规则组子组扫描模式转换
mode8	5'b1x01x	注入触发	注入组子组扫描模式转换

注：MODE_BITS = {SCAN, CONT, DISCEN, IDISEN, IAUTO}

在描述每个模式操作流程之前，有必要引入一些术语，例如规则组，注入组等。

ADC 输入通道共有 19 个，它们被称为 ch0~ch19（没有 ch17），其中 ch0~ch18 是外部输入通道，对应表 14-2 中的 ADC_IN0~ADC_IN18，ch19 对应于内部电压监测通道。

ADC 模块包含一个规则组和一个注入组。规则组和注入组均可配置转换零个或多个 ADC 输入通道。同一组或不同组中的输入通道配置，除了最大数量外，没有额外限制，即任意一个输入通道，如 ch0，可以配置到规则组或注入组，或两个组都配置，甚至一个组中配置多个 ch0 都可以。

规则组和注入组可通过不同的触发源进行触发。同一个组中的通道会按顺序进行转换（mode 1 和 2 除外）。

规则组转换的结果会依次存放于 ADC_RSQR0、ADC_RSQR1、ADC_RSQR2 等寄存器，规则组由 RSQ0 至 RSQ20 的最多 21 个通道组成。

例如，如果 RSQ0~RSQ11 分别设置为 9,8,12,1,5,4,7,3,13,2,0,0 则将规则组按图 8-3 所示进行排列。

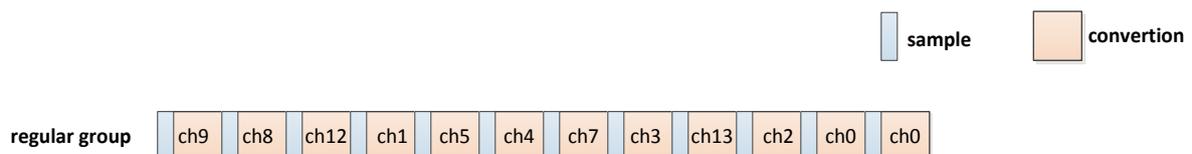


图 8-3 规则组序列

如果 RSQL 设置为 8(Length=9)，则最后 3 个通道将无效且无法转换。因此，有效的规则组序列如图 8-4 所示。

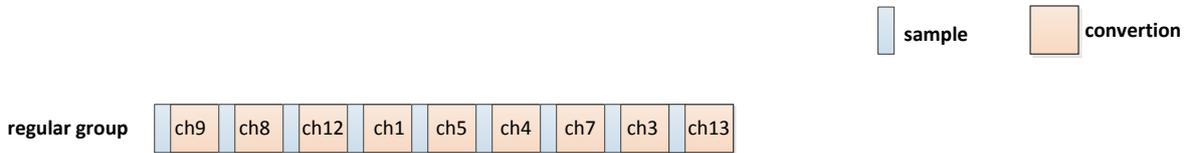


图 8-4 有效规则组序列

以同样的方式，注入组转换的结果会依次存放于 ADC_ISQR0~ADC_ISQR3。基于 ADC_ISQR 寄存器，注入组由最多 4 个通道组成，顺序依次为 ISQ0 至 ISQ3。

例如，如果 ISQ0~ISQ3 分别设置为 12,7,13,2，则将注入组按图 8-5 所示进行排列。

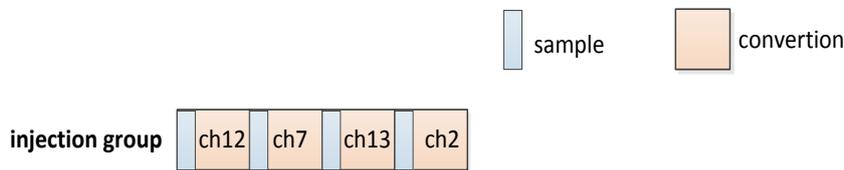


图 8-5 注入组序列

如果 ISQL 设置为 2(Length=3)，则最后 1 个通道将无效并且不会被转换。因此，有效注入组序列如下图所示。

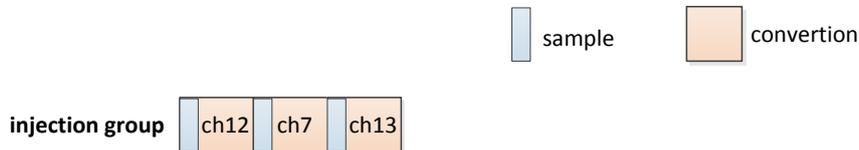


图 8-6 有效注入组序列

显然，规则组触发和注入组触发是开始转换规则组和注入组序列的相应信号。该触发源自 ADC 框图中所示的内部 ADC_CTRL0[SWSTART]或外部触发源。当 ADC 处于常规组通道转换过程中时，规则触发无效。基于此基本介绍，每种模式的详细描述如下。

8.4.2.1 Mode 1

此模式仅转换规则组中的第一个通道，无论 RSQL 为任意值。模式按表 8-1 进行配置后，有效触发可使 ADC 工作在此模式。

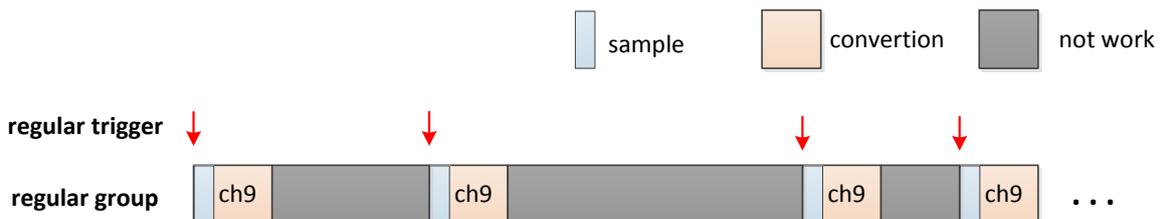


图 8-7 Mode 1 工作流程

如上图所示，规则组中的第一个通道在有效的规则触发后转换一次。然后 ADC 进入空闲状态，直到下一次有效规则触发带来的下一次转换。

8.4.2.2 Mode 2

此模式连续转换规则组中的第一个通道，无论 RSQL 为任意值。模式按表 8-1 进行配置后，有效触发可以使 ADC 在此模式下工作。

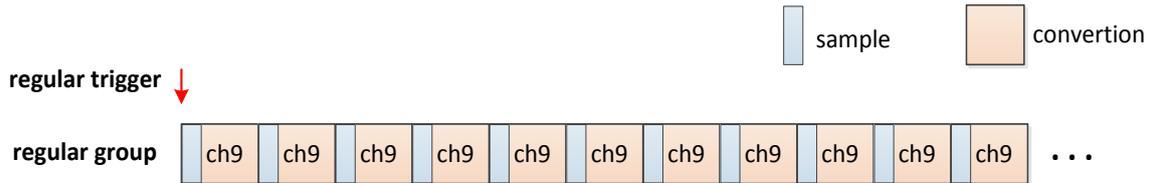


图 8-8 Mode 2 工作流程

如上图所示，在有效的规则触发后，规则组第一个通道将不断转换，除非，断电/复位或者更改 ADC 工作模式。

8.4.2.3 Mode 3

8.4.2.3.1 interval bit=0, 注入组为扫描模式

此模式转换规则组通道和注入组通道。有效的规则和注入组通道长度分别由 RSQL 和 ISQL 决定。使用表 8-1 中的模式配置，有效触发可使 ADC 在此模式下工作。例如，RSQL 设置为 6(Length=7)，ISQL 设置为 2(Length=3)，一个典型操作如下图所示。第一笔规则触发转换规则组中的 7 个通道。当 ADC 转换规则组中的 ch1 时，此时产生注入触发，在 ch1 转换结束后将切换至转换 3 个注入组通道，在所有注入组通道转换完成后，自动切换回规则组通道 ch5 继续转换，完成有效的规则通道转换后，ADC 将运行至空闲状态，直至下一次触发到来。

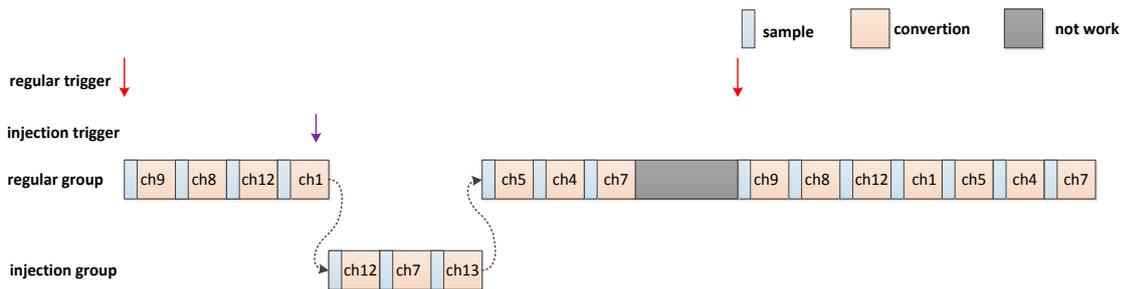


图 8-9 Mode 3 注入组扫描模式工作流程

如果在 ADC 空闲时发生注入触发，ADC 将完成有效注入组通道的转换，如图 8-10 所示。

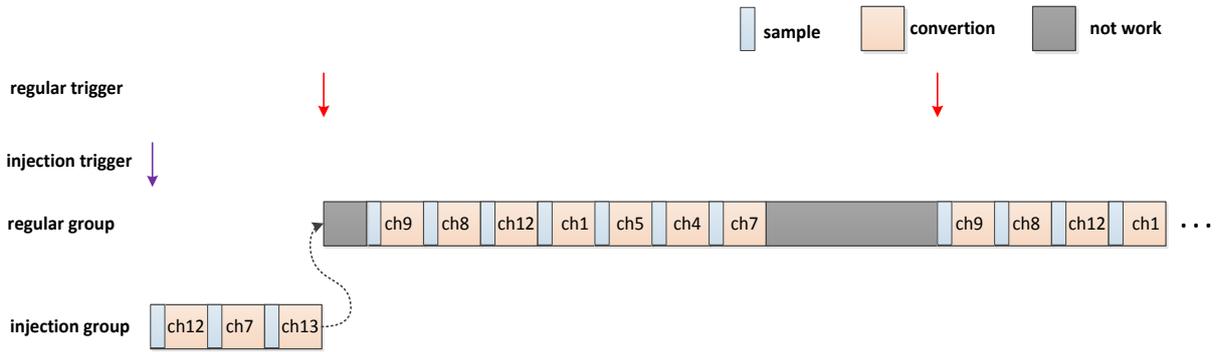


图 8-10 Mode 3 在 ADC 空闲状态下具有注入触发的工作流程

8.4.2.3.2 interval bit=1, 注入组为间隔模式

与图 8-9 的区别在于，产生一次注入触发只会转换注入组序列的一个通道，下一次再发生注入触发，注入组序列的下一通道进行转换。

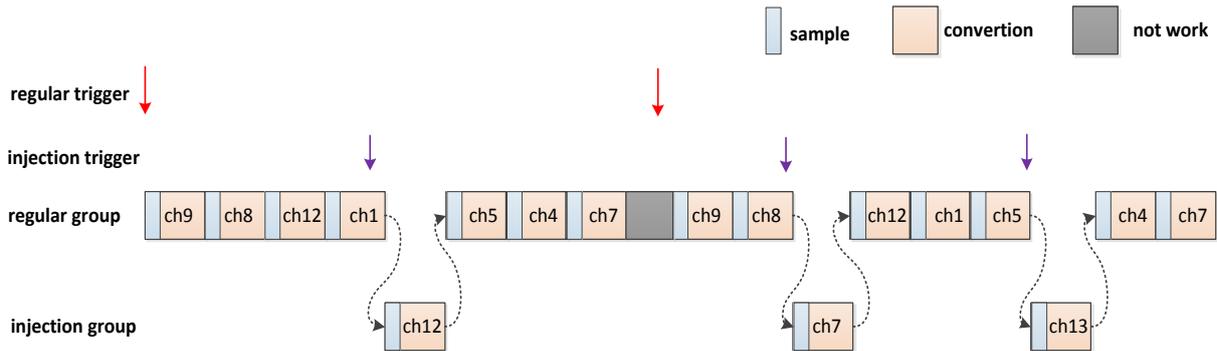


图 8-11 Mode 3 注入组间隔模式工作流程

8.4.2.4 Mode 4

此模式触发后将自动按照规则组通道先转换，后转换注入组通道。有效的规则组通道和注入组通道分别由 RSQL 和 ISQL 决定。使用表 8-1 中的模式配置，有效触发可使 ADC 在此模式下工作。例如，RSQL 设置为 6，ISQL 设置为 2。典型操作如下图所示。规则触发器开始转换前 7 个规则组通道，然后自动转换 3 个注入组通道。在总共 10 个通道均完成完全转换后，ADC 将运行至空闲状态，直到下一个有效的规则触发。

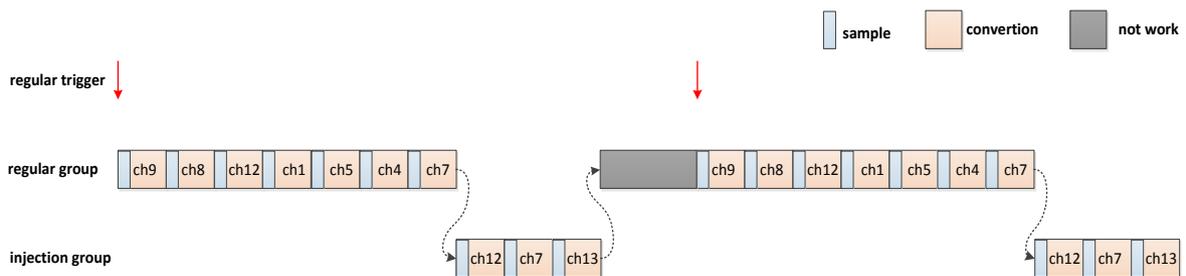


图 8-12 Mode 4 工作流程

8.4.2.5 Mode 5

8.4.2.5.1 interval bit=0, 注入组为扫描模式

与 Mode3 区别在于此模式为连续转换。有效的规则和注入组通道长度分别由 RSQL 和 ISQL 决定，但与 Mode 3 不同，该模式下使用连续转换。使用表 8-1 中的模式配置，有效触发可使 ADC 在此模式下工作。此模式的一个关键特性是单个规则触发可以使 ADC 始终工作，除了掉电、复位或模式更改。例如，RSQL 设置为 6，ISQL 设置为 2。一个典型操作如下图所示。在规则触发后，ADC 按规则组通道顺序工作，如果发生注入触发，则在注入组通道上工作。

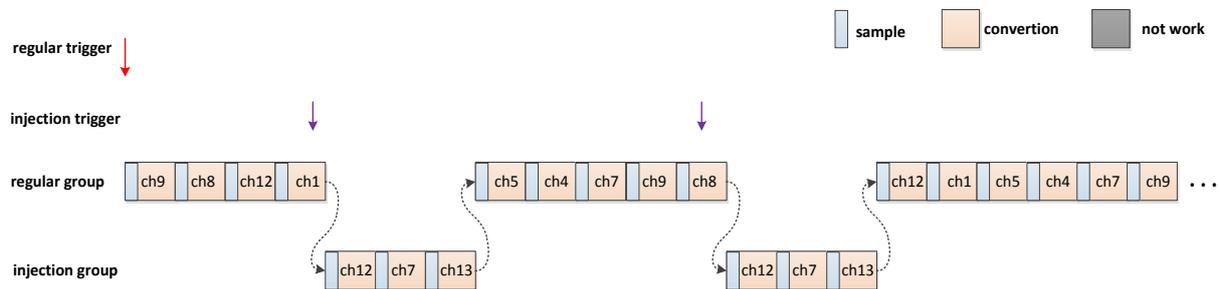


图 8-13 Mode 5 注入组扫描模式工作流程

特别地，如果在 ADC 空闲时发生注入触发，ADC 将首先完成有效注入组通道的转换，如下图所示。

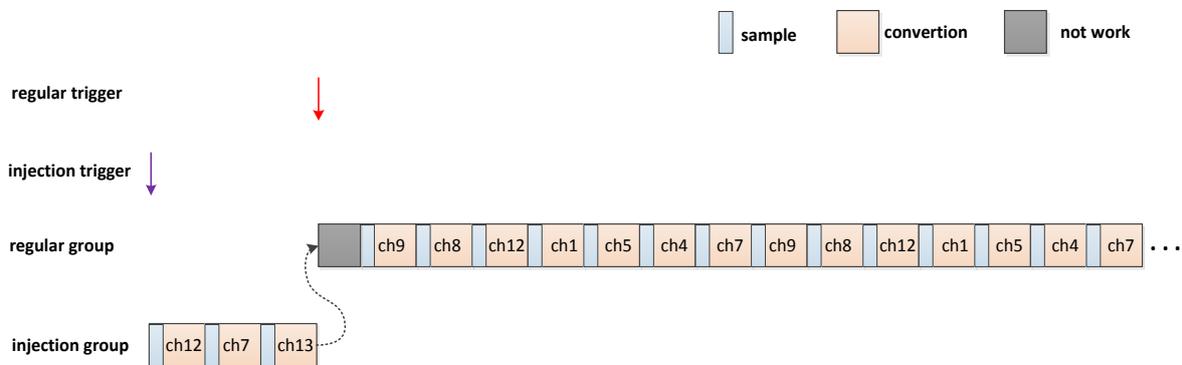


图 8-14 Mode 5 在 ADC 空闲状态下具有注入触发的工作流程

8.4.2.5.2 interval bit=1, 注入组为间隔模式

与图 8-13 的区别在于，产生一次注入触发只会转换注入组序列的一个通道，下一次再发生注入触发，注入组序列的下一通道进行转换。

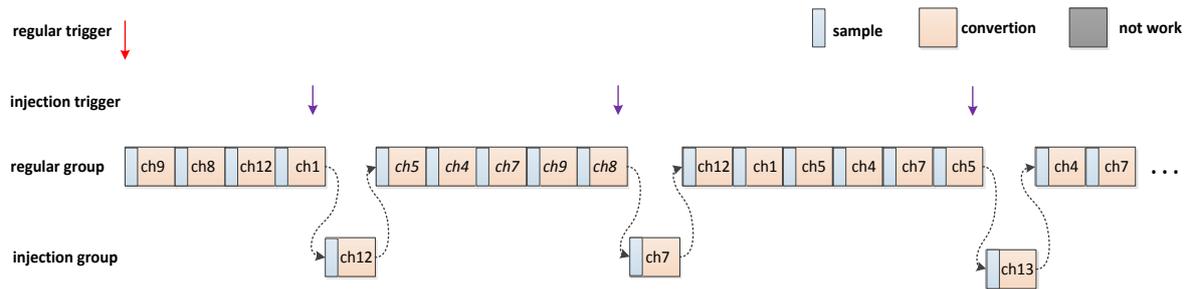


图 8-15 Mode 5 注入组间隔模式工作流程

8.4.2.6 Mode 6

与 Mode4 区别在于此模式为连续转换。有效的规则和注入组通道长度分别由 RSQL 和 ISQL 决定，但与 Mode 4 不同，该模式下使用连续转换。使用表 8-1 中的模式配置，有效的规则触发可以使 ADC 在此模式下工作。此模式的一个关键特性是单个规则触发可以使 ADC 始终工作，除了掉电，复位或模式更改。例如，RSQL 设置为 6，ISQL 设置为 2，操作流程如下图所示。ADC 在规则组通道上按顺序工作，然后在规则组转换完成后转换注入组通道。

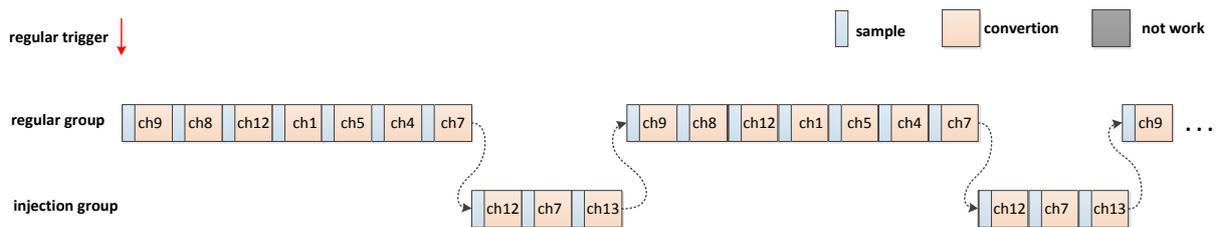


图 8-16 Mode 6 操作流程

8.4.2.7 Mode 7

此模式仅转换规则组通道。有效的规则组通道由 RSQL 决定。使用表 8-1 中的模式配置，ADC 可以在此模式下工作。依据 DISCNUM 将有效的规则通道分成若干子组。

例如，RSQL 设置为 6，DISCNUM 设置为 1。

第一次规则触发：ch9, ch8;

第二次规则触发：ch12, ch1;

第三次规则触发：ch5, ch4;

第四次规则触发：ch7;

因此，实际的转换流程如下图所示。

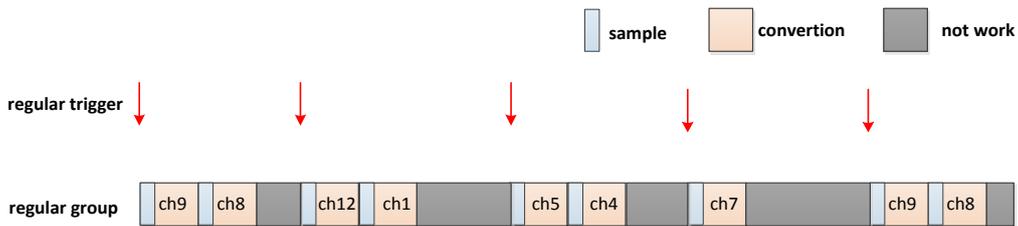


图 8-17 Mode 7 操作流程

8.4.2.8 Mode 8

该模式仅转换注入组通道。有效的注入通道组通道由 ISQL 决定。使用表 8-1 中的模式配置，ADC 可以在此模式下工作。每次触发只转换一个通道。例如，ISQL 设置为 2。

- 第一次注入触发：ch12；
- 第二次注入触发：ch7；
- 第三次注入触发：ch13；
- 第四次注入触发：ch12；

...

因此，实际的转换流程如下图所示。

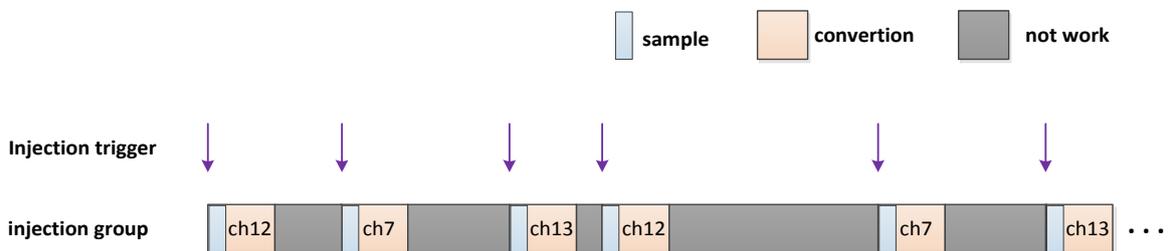


图 8-18 Mode 8 操作流程

8.4.3 触发方式

根据不同的触发方式可以组合出以下 7 种情景。

表 8-2 不同触发方式下的响应行为

触发方式	响应行为
触发规则组	规则组转换
触发注入组	注入组转换
在规则组转换期间产生规则触发	规则组持续转换，第二次触发事件不响应
在规则组转换期间产生注入触发	等待规则组的当前通道转换完才切换到注入组，注入组转换完后切换到原来的规则组继续执行（规则组序列未转换完的情况下）

触发方式	响应行为
在注入组转换期间产生注入触发	注入组持续转换，第二次注入触发事件不响应
在注入组转换期间产生规则触发	在注入转换期间产生一规则事件，注入转换不会被打断，但是规则序列将在注入序列结束后被执行
规则触发和注入触发同一时刻产生	注入组先转换，完成后再转换规则组

8.4.4 线性校准

AC7802x 的 ADC 模块支持线性校准计算功能。该方法主要应用于中等精度 SAR ADC，由于通道数目多，开关个数多，导致实际应用时会产生非理想因素带来的性能下降，主要表现在静态性能的恶化。码值校准的目的是将非理想的 ADC 传递函数曲线通过测量、换算，优化成接近理想传递函数的曲线，进而提高静态性能。该过程如图 8-19 所示。

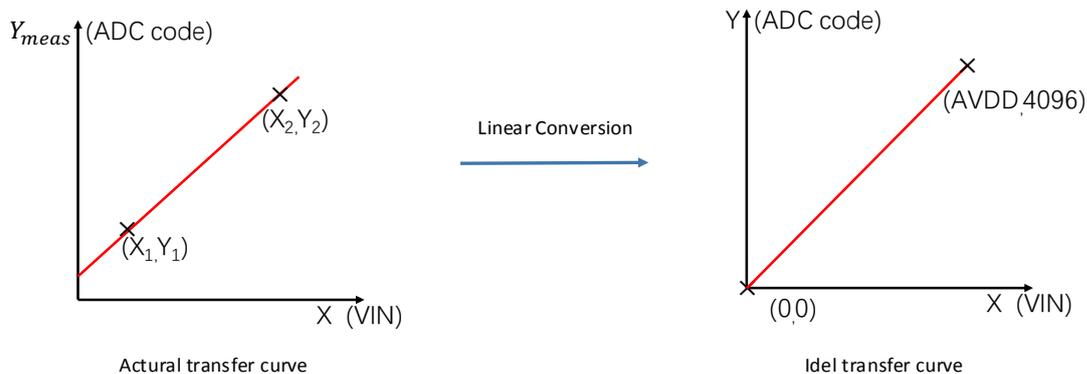


图 8-19 ADC GEOE 校准转换示意

需要注意的是，该校正无法改变 ADC 自身传递曲线，只能对传递曲线进行线性移位优化，故无法优化动态性能。

AC7802x 系列芯片在出厂时会在 5V 供电及 5V 外部参考及常温条件下进行线性校准，以消除芯片生产时带来的线性误差。

在进行生产校准时会得到 ADC 的 GE 及 OE 的补偿值，将该值写入 flash 中，并在芯片上电后，芯片自动将该值加载到 ADC_CGV 及 ADC_COV 寄存器。但由于 flash 中的 GE OE 值都只有 9bit（最高位 bit8 是符号位），和这两个寄存器中的位数不同，而芯片在读取 flash 中的 GE OE 值时只会将这 9 个 bit 读取到对应寄存器的 bit0-8，不会进行符号位扩展填充，因此需要在上电后通过软件将这两个寄存器的值读出，进行相应的移位操作后，再写回这两个寄存器。该操作在上电后仅需执行一次。

8.4.5 模拟监控器

模拟监控器支持电平触发监控事件和边沿触发监控事件模式，当监控通道的电压值超出阈值范围时，模拟监控器将产生监控事件。模拟监控器需要监测的通道可通过 AMOEN, IAMOEN, AMOSGL 和 AMOCH 位来配置。

阈值通过 AMOHR 和 AMOLR 寄存器进行配置。这两个寄存器各个域（AMOHT/ AMOLT/ AMOHO/ AMOLO）配置的值是用于硬件比较的 ADC code。注意，阈值比较时使用的是 ADC 原始数据，设置数据对齐或注入组偏移不影响比较结果。

表 8-3 模拟监控通道配置

模拟监控通道	{AMOEN,IAMOEN,AMOSGL}	工作模式	注释
无	3'b00x	所有模式	-
所有注入组通道	3'b010	mode3/4/5/6/8	-
所有规则组通道	3'b100	除了 mode8	-
所有通道	3'b110	所有模式	-
单注入组通道	3'b011	mode3/4/5/6/8	转换序列必须包含由 AMOCH [4: 0]指定的注入通道
单规则组通道	3'b101	mode1 ~ 7	转换序列必须包含由 AMOCH [4: 0]指定的规则通道
单规则组或注入组通道	3'b111	所有通道	转换序列必须包含由 AMOCH [4: 0]指定的规则或注入通道

8.4.5.1 电平触发模式

设置 AMOMODE=0，模拟监控器工作于电平触发模式。

如果被监控通道的电压大于高阈值 AMOHT 或小于低阈值 AMOLT，则模拟监控器将 AMO 标志设置为 1，如果 AMOIE 被配置为 1，则产生中断。

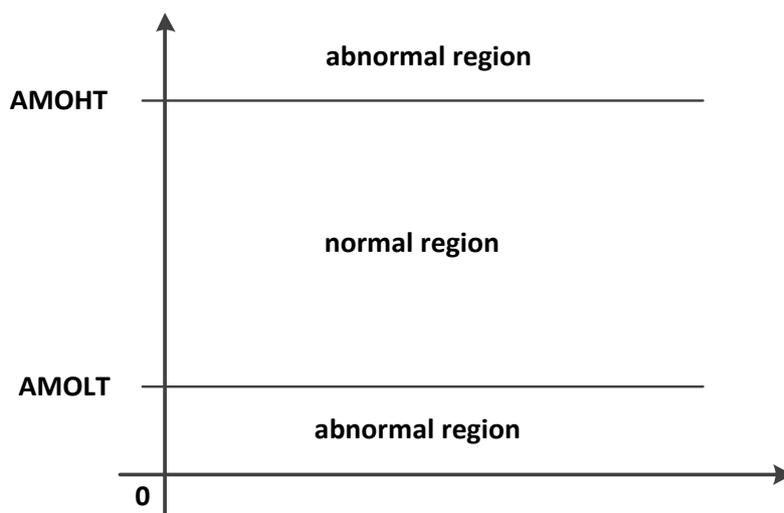


图 8-20 电平触发模式下监控区域

8.4.5.2 边沿触发模式

设置 AMOMODE=1，模拟监控器工作于边沿触发模式。

当监控通道电压从正常区域（低阈值 AMOLT 和高阈值 AMOHT 之间）到异常区域（低于低阈值 AMOLT 或高于高阈值 AMOHT）时则产生一次监控异常事件，模拟监控器将 AAMO 标志设置为 1，如果 AMOIE 被配置为 1，则产生监控事件中断。

当监控通道电压从异常区域（低于低阈值 AMOLT 或高于高阈值 AMOHT）到正常区域（AMOHT-AMOHO 和 AMOLT+AMOLO 之间），则产生一次监控恢复事件，模拟监控器将 NAMO 标志设置为 1，如果 AMOIE 被配置为 1，则产生监控事件中断。边界值 = [高阈值-高偏移值，低阈值+低偏移值]，即[AMOHT-AMOHO, AMOLT+AMOLO]。



边沿触发模式只支持在监控单通道时使用，监控多通道无法区分不同通道触发的异常和恢复中断。

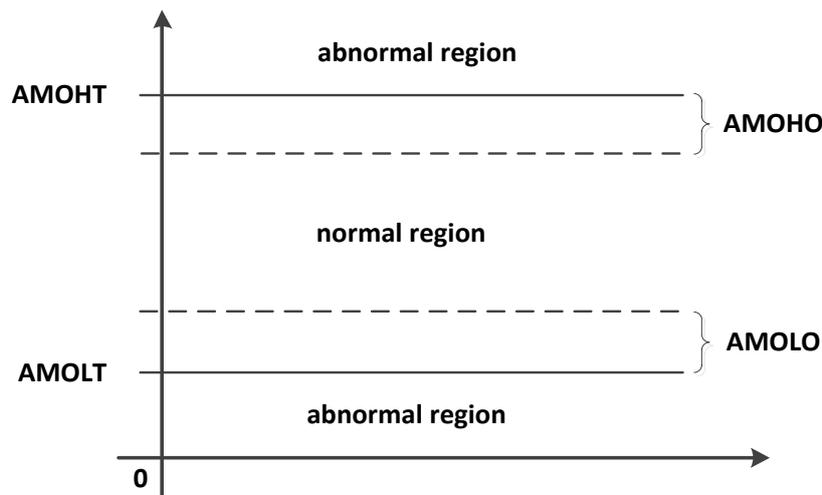


图 8-21 边沿触发模式下监控区域

8.4.6 状态标志

ADC 有三种转换状态标志位：EOCx、IEOCx 和 AMO（边沿触发模式使用 AAMO 和 NAMO）。EOCx 标志表示规则组对应序列通道的转换结束。IEOCx 标志表示注入组对应序列通道的转换结束。AMO 标志标识是否发生模拟监控器事件。模拟监控器事件表示当前的转换结果是否高于当前配置的高阈值或低于当前配置的低阈值。对于不同模式，EOCx 和 IEOCx 标志的行为相同，AMO 标志在所有模式下产生的时刻相同。假设 ch5 小于 AMOLT，ch7 大于 AMOHT，并且模拟监控器被配置为检查所有通道，如包括规则组通道和注入组通道等，以下描述基于以上假设。

对于任意 mode，在相应序列通道完成转换时生成 EOCx 或 IEOCx 标志，同时生成 AMO 标志。有关三个标志的详细信息（基于模式 6），如下图所示。

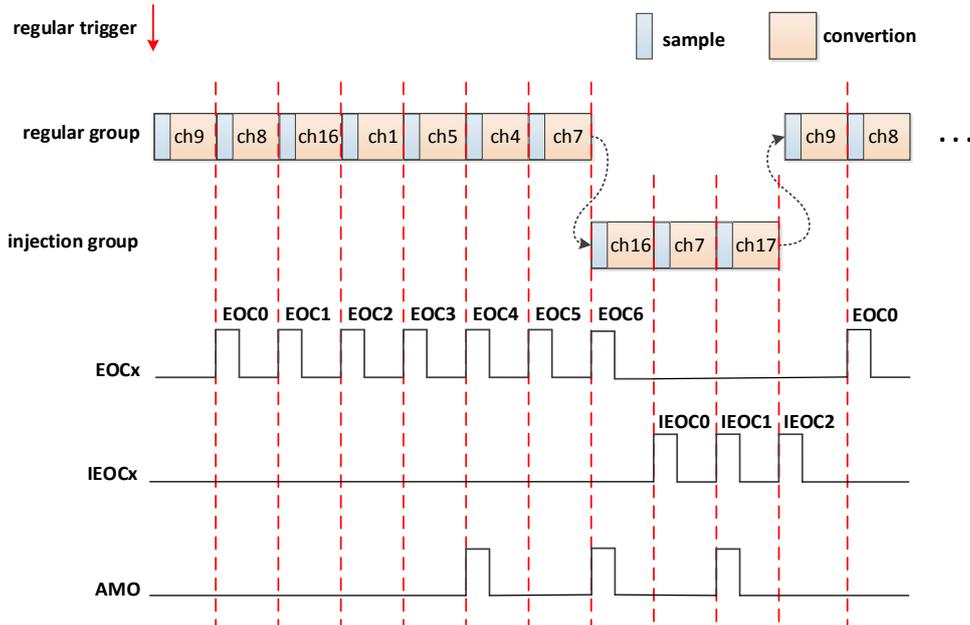


图 8-22 三个标志行为

8.4.7 数据对齐方式

数据对齐功能用于控制 ADC 转换结果数据在数据寄存器中的排布，可选择左对齐或右对齐方式。两种对齐方式下的数据摆放如下图所示。需要注意，如果出现注入组 data register 减去 ADC_IOFRx 后变为负值的情况，符号位会被舍弃。



图 8-23 ADC 数据在寄存器内的排布

8.4.8 采样转换时间

ADC 需要使用若干个 ADC_CLK 周期对输入电压采样，即对 ADC 内部电路进行充电，使其达到外部输入信号的电平，完成采样之后才能进行模拟到数字的转换。采样周期个数可通过 ADC_SPT 寄存器中的 SPT[2:0]位配置。每个通道可以分别用不同的时间进行采样。

总转换时间公式： $(SPT + 12) * \text{ADC 周期} + 5 \text{ 个 APB 周期}$

例：当 APB=16MHz, ADCCLK=8MHz, SPT=17 ADCCLK, 总转换时间= $(17+12)/8+(5/16) \approx 4\mu\text{s}$ 。

ADC 的时钟在 MCU 的正常工作状态下是使用 pclk，可参考图 4-1。ADC 支持的 pclk 最高为 16M，而 ADC 的转换时钟，即 ADC_CLK 最高支持 8M。

下表列出最高时钟配置下 ADC 各个 SPT 配置对应的采样率。注意，AC7802x ADC 支持的最高采样率为 250Ksps，如果配置采样率高于该值，可能会导致采样结果精度显著减低。

表 8-4 ADC 采样率

SPT	采样周期(8M)	采样时间(μs)	转换周期(8M)	总转换周期(16M)	采样率(kHz)
0	9	1.125	12	47	340.43
1	7	0.875	12	43	372.09
2	17	2.125	12	63	253.97
3	33	4.125	12	95	168.42
4	64	8	12	157	101.91
5	140	17.5	12	309	51.78
6	215	26.875	12	459	34.86
7	5	0.625	12	39	410.26

8.4.9 温度传感器

温度传感器可以用来测量器件周围的温度(T_A)，与 ADC 内部直接连接，通过 ADC 把传感器输出电压转换成数字量。

内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

温度计算公式：温度($^{\circ}\text{C}$) = $\{(V_{\text{TEMP25}} - V_{\text{SENSE}}) / \text{Slope}\} + 25$

V_{TEMP25} : 25 $^{\circ}\text{C}$ 时的电压数值

V_{SENSE} : 当前温度电压数值

Slope: 温度传感器的平均斜率(单位为 $\text{mV}/^{\circ}\text{C}$)

具体可参考数据手册的电气特性章节中 V_{TEMP25} 和 Slope 的实际值。

8.4.10 低功耗模式

ADC 提供两种功耗模式：一种是正常模式，另一种是低功耗模式。ADC 时钟在低功耗模式下工作频率较低，以降低功耗。当 MCU 进入 stop 模式时，可以使 ADC 进入低功耗模式。低功耗模式下的 ADC 模拟监控器事件可将 MCU 从停止模式唤醒至正常模式。

ADC 功耗模式切换流程如下图所示。

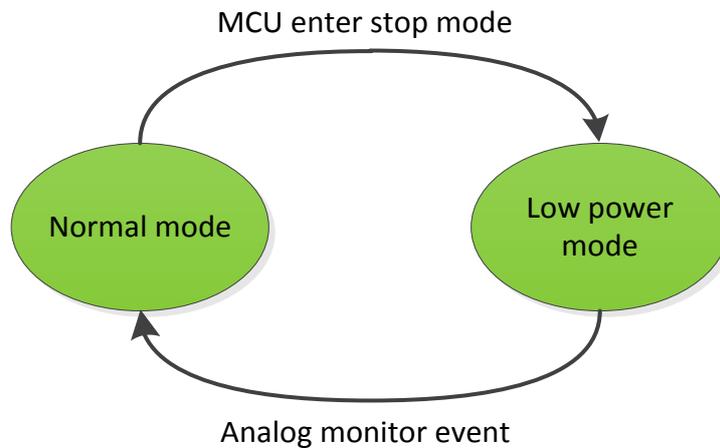


图 8-24 ADC 功耗模式切换流程图

如果配置 ADC 在 MCU STOP 模式下工作（SPM 中配置 ADC 作为唤醒源），有以下情况需要注意：

1. MCU 在进入 STOP 模式时，ADC 的工作时钟会自动切换到 HSI(32MHz)，由于 ADC 的最高工作时钟是 16MHz，因此需要在进入 STOP 模式之前，将 ADC 工作时钟分频调整到 2 分频以上；
2. 在 SPM 中配置 ADC 作为唤醒源之后，需要保证 ADC 的时钟已经使能，Reset 信号已关闭
3. STOP 模式下，ADC 的固定唤醒间隔是 12ms，每次进行转换时仅转换一个通道，即如果是 mode 3，则在收到硬件触发后，每隔 12ms 依次转换被触发组（规则组或注入组）中的一个通道，直到完成该组配置的所有通道。
4. MCU 退出 STOP 模式后，ADC 的工作时钟自动切回原时钟。

8.5 应用说明

8.5.1 重置和使能

本节介绍了 ADC 的 3 种重置范围，为了更好的使用 ADC，需要了解这 3 种重置范围。

表 8-5 ADC 重置范围

编号	重置信号	ADC 重置效果	描述
1	将 CKGEN_PERI_SFT_RST1 中的 SRST_ADC0 置位	重置整个 ADC，包括所有寄存器	ADC 所有寄存器值均不会保留。如果需要重新使用 ADC，需要进行重新配置，也需要在 ADON 置为 1 后等待超过 100μs，再触发转换。
2	将 ADON 置为 0	重置 ADC 内部状态，并重置 RDR/IDR 寄存器	ADC 配置相关寄存器会保留。如果需要重新使用 ADC，只需要将 ADON 置为 1，并发送触发信号。

3	修改 ADC 工作模式或修改 RSQR、ISQR、RSQL、ISQL 中任意一个	仅重置 ADC 内部状态，不会影响配置寄存器	ADC 配置相关寄存器会保留。 如果需要重新使用 ADC，只需要发出触发信号。
---	--	------------------------	--

关于使能 ADC 模块，需要在 CKGEN 模块中进行以下操作：

1. 使能 ADC 时钟
2. 关闭 ADC 重置信号

相关寄存器配置请参考 4.3.2 及 4.3.5。

8.5.2 ADC 上电延时

如 8.4.1 章节所述，在 ADC 模块第一次上电时，需要有约 100 μ s 的时间用于模块内部信号的稳定，在系统处于低功耗模式（Stop）时，当 ADC 转换结束，则 ADC 会自动断电，如果配置 ADC 为此模式下工作，则当 ADC 收到触发转换的信号时，ADC 存在硬件自动上电的过程，此时 ADC 也存在 100 μ s 的上电延时。

8.6 寄存器定义

表 8-6 ADC 寄存器映射

ADC0 基地址 = 0x40003000

Analog 基地址 = 0x40008800

地址	名称	宽度	描述
ADCx 基地址+0x0	ADC_STR	32	状态寄存器
ADCx 基地址+0x4	ADC_CTRL0	32	控制寄存器 0
ADCx 基地址+0x8	ADC_CTRL1	32	控制寄存器 1
ADCx 基地址+0xC	ADC_SPT0	32	采样时间寄存器 0
ADCx 基地址+0x10	ADC_SPT1	32	采样时间寄存器 1
ADCx 基地址+0x14	ADC_IOFR0	32	注入组偏移寄存器 0
ADCx 基地址+0x18	ADC_IOFR1	32	注入组偏移寄存器 1
ADCx 基地址+0x1C	ADC_IOFR2	32	注入组偏移寄存器 2
ADCx 基地址+0x20	ADC_IOFR3	32	注入组偏移寄存器 3
ADCx 基地址+0x24	ADC_AMOHR	32	模拟监控器高阈值寄存器
ADCx 基地址+0x28	ADC_AMOLR	32	模拟监控器低阈值寄存器
ADCx 基地址+0x2C	ADC_RSQR0	32	规则组序列配置寄存器 0
ADCx 基地址+0x30	ADC_RSQR1	32	规则组序列配置寄存器 1
ADCx 基地址+0x34	ADC_RSQR2	32	规则组序列配置寄存器 2
ADCx 基地址+0x38	ADC_ISQR	32	注入组序列配置寄存器
ADCx 基地址+0x3C	ADC_IDR0	32	注入组数据寄存器 0

地址	名称	宽度	描述
ADCx 基地址+0x40	ADC_IDR1	32	注入组数据寄存器 1
ADCx 基地址+0x44	ADC_IDR2	32	注入组数据寄存器 2
ADCx 基地址+0x48	ADC_IDR3	32	注入组数据寄存器 3
ADCx 基地址+0x4C	ADC_RDR0	32	规则组数据寄存器 0
ADCx 基地址+0x50	ADC_RDR1	32	规则组数据寄存器 1
ADCx 基地址+0x54	ADC_RDR2	32	规则组数据寄存器 2
ADCx 基地址+0x58	ADC_RDR3	32	规则组数据寄存器 3
ADCx 基地址+0x5C	ADC_RDR4	32	规则组数据寄存器 4
ADCx 基地址+0x60	ADC_RDR5	32	规则组数据寄存器 5
ADCx 基地址+0x64	ADC_RDR6	32	规则组数据寄存器 6
ADCx 基地址+0x68	ADC_RDR7	32	规则组数据寄存器 7
ADCx 基地址+0x6C	ADC_RDR8	32	规则组数据寄存器 8
ADCx 基地址+0x70	ADC_RDR9	32	规则组数据寄存器 9
ADCx 基地址+0x74	ADC_RDR10	32	规则组数据寄存器 10
ADCx 基地址+0x78	ADC_RDR11	32	规则组数据寄存器 11
ADCx 基地址+0x7C	ADC_RDR12	32	规则组数据寄存器 12
ADCx 基地址+0x80	ADC_RDR13	32	规则组数据寄存器 13
ADCx 基地址+0x84	ADC_RDR14	32	规则组数据寄存器 14
ADCx 基地址+0x88	ADC_RDR15	32	规则组数据寄存器 15
ADCx 基地址+0x8C	ADC_RDR16	32	规则组数据寄存器 16
ADCx 基地址+0x90	ADC_RDR17	32	规则组数据寄存器 17
ADCx 基地址+0x94	ADC_RDR18	32	规则组数据寄存器 18
ADCx 基地址+0x98	ADC_RDR19	32	规则组数据寄存器 19
ADCx 基地址+0x9C	ADC_RDR20	32	规则组数据寄存器 20
ADCx 基地址+0xCC	ADC_CGV	32	增益误差值校准寄存器
ADCx 基地址+0xD0	ADC_COV	32	偏置误差值校准寄存器
ADCx 基地址+0xD4	ADC_REOC	32	规则组转换结束状态寄存器
ADCx 基地址+0xD8	ADC_REOCEN	32	规则组转换结束中断使能寄存器
ADCx 基地址+0xDC	ADC_IEOC	32	注入组转换结束状态寄存器
ADCx 基地址+0xE0	ADC_IEOCEN	32	注入组转换结束中断使能寄存器
ADCx 基地址+0xE4	ADC_RSQR3	32	规则组序列配置寄存器 3
Analog 基地址+0x40	ADC_CFG0	32	模拟配置寄存器 0
Analog 基地址+0x44	ADC_CFG1	32	模拟配置寄存器 1

8.6.1 状态寄存器(ADC_STR)

表 8-7 ADC_STR 寄存器

ADC_STR

ADC 状态寄存器

Reset: 0x00000010

位	31~7	6	5	4	3	2	1	0
名称		AAMO	NAMO	IDLE				AMO
访问		R/W0C	R/W0C	RO				R/W0C
Reset		0	0	1				0

字段	说明
6 AAMO	模拟监控器事件发生(边沿触发模式使用) 0: 没有异常事件 1: 发生异常事件, 写 0 清除
5 NAMO	模拟监控恢复事件发生标志(边沿触发模式使用) 0: 没有恢复事件 1: 发生恢复事件, 写 0 清除
4 IDLE	ADC 空闲状态标志 0: ADC 处于非空闲状态 1: ADC 处于空闲状态
0 AMO	模拟监控异常事件发生标志(电平触发模式使用) 0: 没有异常事件 1: 发生异常事件, 写 0 清除

8.6.2 控制寄存器 0(ADC_CTRL0)

表 8-8 ADC_CTRL0 寄存器

ADC_CTRL0		ADC 控制寄存器 0										Reset: 0x00000000					
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	SW STA RT	ISW STA RT						INT ER VAL	AM OM OD E	AL IG N	IEX TT RIG	EX TT RIG		AM OIE			
访问	RW	RW						RW	RW	R W	RW	RW		RW			
Reset	0	0						0	0	0	0	0		0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	SCA N	CO NT	DIS CE N	IDI SC EN	IA U T O	DISCNUM[2: 0]			AM OE N	IA M O E N	AM OS GL	AMOCH[4: 0]					
访问	RW	RW	RW	RW	R W	R W	R W	RW	RW	R W	RW	RW	R W	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

字段	说明
31 SWSTART	规则通道软件触发 写 1 触发，读为 0
30 ISWSTART	注入通道软件触发 写 1 触发，读为 0
24 INTERVAL	间隔模式(仅在 Mode3/5 使用) 0: 注入组为扫描模式 1: 注入组为间隔模式
23 AMOMODE	模拟监控器触发模式 0: 电平触发模式 1: 边沿触发模式
22 ALIGN	数据对齐 0: 右对齐 1: 左对齐
21 IEXTTRIG	注入组触发源选择 0: 内部 (软件触发) 1: 外部
20 EXTTRIG	规则组触发源选择 0: 内部 (软件触发) 1: 外部
18 AMOIE	AMO 中断功能使能 0: 禁用 1: 使能
15: 11 Modes control bits	ADC 工作模式 详细配置参考表 8-1
10: 8 DISCNUM	Mode 7 规则组子组长度 0 ~ 7: mode 7 子组长度, 分别为 1~8
7: 5 Analog monitor control bits	模拟监控通道配置 详细配置参考表 8-3
4: 0 AMOCH	模拟监控通道 当模拟监控器配置为仅检测单个通道时, 指定被监测的通道

字段	说明
29: 0	通道采样时间
SPTx(x=10 ~ 19)	SPTx 的编号: 0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道
	SPTx 编码含义如下: 000b: 9 ADCCLK 001b: 7 ADCCLK 010b: 17 ADCCLK 011b: 33 ADCCLK 100b: 64 ADCCLK 101b: 140 ADCCLK 110b: 215 ADCCLK 111b: 5 ADCCLK

8.6.5 采样时间寄存器 1(ADC_SPT1)

表 8-11 ADC_SPT1 寄存器

ADC_SPT1		ADC 采样时间寄存器 1														Reset: 0x00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称			SPT9[2: 0]			SPT8[2: 0]			SPT7[2: 0]			SPT6[2: 0]			SPT5[2:0]		
访问			RW														
Reset			0														
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称			SPT4[2: 0]			SPT3[2: 0]			SPT2[2: 0]			SPT1[2: 0]			SPT0[2: 0]		
访问			RW														
Reset			0														

字段	说明
29: 0	通道采样时间
SPTx (x=0 ~ 9)	SPTx 的编号: 0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道 SPTx 编码含义如下: 000b: 9 ADCCLK 001b: 7 ADCCLK 010b: 17 ADCCLK 011b: 33 ADCCLK 100b: 64 ADCCLK 101b: 140 ADCCLK 110b: 215 ADCCLK 111b: 5 ADCCLK

8.6.6 注入组偏移寄存器(ADC_IOFRx)

表 8-12 ADC_IOFRx (x= 0 ~ 3) 寄存器

ADC_IOFRx (x= 0 ~ 3)				ADC 注入组偏移寄存器												Reset: 0x00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称					IOFR												
访问					RW												
Reset					0												

字段	说明
11: 0	注入组偏移值
IOFR	用于将注入组转换结果减去该寄存器对应值。注入组通道转换的数据值 IDR 已经减去了在 ADC_IOFR 寄存器中定义的偏移量。

8.6.7 高阈值寄存器(ADC_AMOHR)

表 8-13 ADC_AMOHR 寄存器

ADC_AMOHR				ADC 高阈值寄存器												Reset: 0x00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	AMOHO																

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

字段	说明
27: 16 AMOHO	模拟监控器高阈值对应的恢复偏移值 定义高阈值的偏移值
11: 0 AMOHT	模拟监控器的高阈值 定义高阈值

8.6.8 低阈值寄存器(ADC_AMOLR)

表 8-14 ADC_AMOLR 寄存器

ADC_AMOLR				ADC 低阈值寄存器												Reset: 0x00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
名称					AMOLO														
访问					RW														
Reset					0														
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
名称					AMOLT														
访问					RW														
Reset					0														

字段	说明
27: 16 AMOLO	模拟监控器低阈值对应的恢复偏移值 定义低阈值的偏移值
11: 0 AMOLT	模拟监控器的低阈值 定义低阈值

8.6.9 规则组序列配置寄存器 0(ADC_RSQR0)

表 8-15 ADC_RSQR0 寄存器

ADC_RSQR0 ADC 规则组序列配置寄存器 0 Reset: 0x000FFFFF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									RSQL[4: 0]				RSQ15[4: 0]			
访问									RW				RW			
Reset									0				1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RSQ14[4: 0]				RSQ13[4: 0]				RSQ12[4: 0]							
访问	RW				RW				RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
24: 20 RSQL	规则组的长度 0 ~ 20: 定义规则组长度 1~21 长度必须小于实际有效的规则组序列长度的值。大于 20 的值无效
19: 0 RSQx(x=12~15)	规则组通道选择 0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道 其他值未使用。

8.6.10 规则组序列配置寄存器 1(ADC_RSQR1)

表 8-16 ADC_RSQR1 寄存器

ADC_RSQR1 ADC 规则组序列配置寄存器 1 Reset: 0x3FFFFFFF

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	RSQ11[4: 0]				RSQ10[4: 0]				RSQ9[4: 0]							
访问	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RSQ8[4: 0]				RSQ7[4: 0]				RSQ6[4: 0]							
访问	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
29: 0	规则组通道选择
RSQx(x=6~11)	0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道 其他值未使用。

8.6.11 规则组序列配置寄存器 2(ADC_RSQR2)

表 8-17 ADC_RSQR2 寄存器

ADC_RSQR2		ADC 规则组序列配置寄存器 2												Reset: 0x3FFFFFFF			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称			RSQ5[4: 0]					RSQ4[4: 0]					RSQ3[4: 0]				
访问			RW					RW					RW				
Reset			1	1	1	1	1	1	1	1	1	1	1	1	1	1	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称			RSQ2[4: 0]				RSQ1[4: 0]				RSQ0[4: 0]						
访问			RW				RW				RW						
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

字段	说明
29: 0	规则组通道选择
RSQx(x=0~5)	0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道 其他值未使用。

8.6.12 注入组序列配置寄存器(ADC_ISQR)

表 8-18 ADC_ISQR 寄存器

ADC_ISQR		ADC 注入组序列配置寄存器												Reset: 0x00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称											ISQL[1: 0]		ISQ3[4: 0]				
访问											RW						
Reset											0	0	1	1	1	1	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称			ISQ2[4: 0]				ISQ1[4: 0]				ISQ0[4: 0]						
访问			RW														
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

字段	说明
21: 20 ISQL	注入组长度 0 ~ 3: 定义注入组长度 1~4 注意: 该长度必须小于实际有效的注入组序列数。
19: 0 ISQx(x=0 ~ 3)	注入组通道选择 0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道 其他值未使用。

8.6.13 注入组数据寄存器(ADC_IDRx)

表 8-19 ADC_IDRx (x=0 ~ 3)寄存器

ADC_IDRx (x=0 ~ 3)	ADC 注入组数据寄存器																Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	IDR																
访问	RO																
Reset	0																

字段	说明
15: 0 IDR	注入组数据寄存器 注意: 1. ADC_IDR 已经减去了在 ADC_IOFR 寄存器中定义的偏移量。 2. 不同对齐方式下的数据摆放参考 8.4.7

8.6.14 规则组数据寄存器(ADC_RDRx)

表 8-20 ADC_RDRx 寄存器

ADC_RDRx (x=0 ~ 20)				ADC 规则组数据寄存器								Reset: 0x00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RDR															
访问	RO															
Reset	0															

字段	说明
15: 0	规则组数据寄存器
RDR	注意, 不同对齐方式下的数据摆放参考 8.4.7

8.6.15 增益误差值校准寄存器(ADC_CGV)

表 8-21 ADC_CGV 寄存器

ADC_CGV				ADC 增益误差值校准值寄存器								Reset: 0x00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称					GE1											
访问					RW											
Reset					0x0											
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					GE0											
访问					RW											
Reset					0x0											

字段	说明
27: 16	内部通道增益误差
GE1	补码存储, 最高位为符号位。
11: 0	外部通道增益误差
GE0	补码存储, 最高位为符号位。

8.6.16 偏置误差值校准寄存器(ADC_COV)

表 8-22 ADC_COV 寄存器

ADC_COV ADC 偏置误差值校准值寄存器 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称											OE1					
访问											RW					
Reset											0x0					
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称											OE0					
访问											RW					
Reset											0x0					

字段	说明
26: 16 OE1	内部通道偏置误差校准值 补码存储，最高位为符号位。
10: 0 OE0	外部通道偏置误差校准值 补码存储，最高位为符号位。

8.6.17 规则组转换结束状态寄存器(ADC_REOC)

表 8-23 ADC_REOC 寄存器

ADC_COV ADC 规则组转换结束状态寄存器 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称											EO C20	EO C19	EO C18	EO C17	EO C16	
访问											R/ W1 C	R/ W1 C	R/ W1 C	R/ W1 C	R/ W1 C	
Reset											0x0	0x0	0x0	0x0	0x0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	EO C15	EO C14	EO C13	EO C12	EO C11	EO C10	EO C9	EO C8	EO C7	EO C6	EO C5	EO C4	EO C3	EO C2	EO C1	EO C0
访问	R/ W1 C															
Reset	0x0															

字段	说明
20:0 EOCx(x=0~20)	规则组转换结束标志 0: 规则组序列转换未结束，或未开始转换 1: 规则组序列转换结束 对应 bit 写 1 清除标志。

8.6.18 规则组转换结束中断使能寄存器(ADC_REOCEN)

表 8-24 ADC_REOCEN 寄存器

ADC_REOCEN ADC 规则组转换结束中断使能寄存器 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称												RE OC IE2 0	RE OC IE1 9	RE OC IE1 8	RE OC IE1 7	RE OC IE1 6
访问												RW	RW	RW	RW	RW
Reset												0x0	0x0	0x0	0x0	0x0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RE OC IE1 5	RE OC IE1 4	RE OC IE1 3	RE OC IE1 2	RE OC IE1 1	RE OC IE1 0	RE OC IE9	RE OC IE8	RE OC IE7	RE OC IE6	RE OC IE5	RE OC IE4	RE OC IE3	RE OC IE2	RE OC IE1	RE OC IE0
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
20:0 REOCIE _x (x=0~20)	规则组转换结束标志中断使能 0: 规则组序列转换结束标志不触发中断 1: 规则组序列转换结束标志触发中断

8.6.19 注入组转换结束状态寄存器(ADC_IEOC)

表 8-25 ADC_IEOC 寄存器

ADC_IEOC ADC 注入组转换结束状态寄存器 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													IE OC 3	IE OC 2	IE OC 1	IE OC 0
访问													R/ W1 C	R/ W1 C	R/ W1 C	R/ W1 C
Reset													0x0	0x0	0x0	0x0

字段	说明
3:0 IEOC _x (x=0~3)	注入组转换结束标志

字段	说明
	0: 注入组序列转换未结束, 或未开始转换
	1: 注入组序列转换结束
	对应 bit 写 1 清除标志。

8.6.20 注入组转换结束中断使能寄存器(ADC_IEOCEN)

表 8-26 ADC_IEOCEN 寄存器

ADC_IEOCEN		ADC 注入组转换结束中断使能寄存器												Reset: 0x00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称													IE OC IE3	IE OC IE2	IE OC IE1	IE OC IE0	
访问													RW	RW	RW	RW	
Reset													0x0	0x0	0x0	0x0	

字段	说明
3:0 IEOCIE _x (x=0~3)	注入组转换结束标志中断使能
	0: 注入组序列转换结束标志不触发中断
	1: 注入组序列转换结束标志触发中断

8.6.21 规则组序列配置寄存器 3(ADC_RSQR3)

表 8-27 ADC_RSQR3 寄存器

ADC_RSQR3		ADC 规则组序列配置寄存器 3												Reset: 0x01FFFFFF				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称									RSQ20[4: 0]				RSQ19[4: 0]					
访问									RW				RW					
Reset									1	1	1	1	1	1	1	1	1	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称	RSQ18[4: 0]				RSQ17[4: 0]				RSQ16[4: 0]									
访问	RW				RW				RW									
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

字段	说明
24: 0 RSQx(x=16~20)	规则组通道选择 0~18: 外部通道 0~18 (没有通道 17) 19: 内部电压通道 其他值未使用

8.6.22 ADC 模拟配置寄存器 0(ADC_CFG0)

表 8-28 ADC_CFG0 寄存器

ADC_CFG0			ADC 模拟配置寄存器 0													Reset: 0x02492000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			VR EF _SE L													
访问			RW													
Reset			0													
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										VB UF _E N						
访问										RW						
Reset										0						

字段	说明
29 VREF_SEL	参考源选择 0: 使用 VREF+/VREF-信号作为参考源 1: 使用 AVDD 作为参考源
6 VBUF_EN	Bandgap 和 T-sensor 内部通道使能 0: 关闭内部通道 1: 打开内部通道

8.6.23 ADC 模拟配置寄存器 1(ADC_CFG1)

表 8-29 ADC_CFG1 寄存器

ADC_CFG1			ADC 模拟配置寄存器 1													Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	MON_SEL															
访问	RW	RW														
Reset	0	0														

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																
访问																
Reset																

字段	说明
31:30 MON_SEL	内部电压信号通道选择 00b: Bandgap 01b: T-sensor 其他值未使用。

9 模拟比较器 (ACMP)

9.1 简介

模拟比较器提供一个用于比较两个模拟输入电压的电路，模拟多路复用器提供一个用于从 9 路通道中选择模拟输入信号的电路，其中 1 路通道由 6 位数字模拟转换器 (DAC) 提供，其他通道由外部输入提供。轮询模式和霍尔输出(Hall)功能专为电机应用而设计。

9.2 特性

- 1 个模拟比较器，1 个 DAC 模块
- 片上 6 位数字模拟转换器 (DAC)，可从 VDD 或内部带隙基准电压 (Bangap) 中选择基准电压
- 可配置迟滞，支持 10/20/40mV
- 可在比较器输出上升沿、下降沿或上升/下降沿时产生中断
- 最多 9 个可选择比较器输入(ACMP_IN0~ACMP_IN7 以及内部 DAC)
- 支持停止 (Stop) 模式唤醒
- 支持轮询模式
- 支持 ACMP 比较结果输出
- 支持 DAC 输出
- 支持霍尔(Hall) 输出

9.3 结构框图

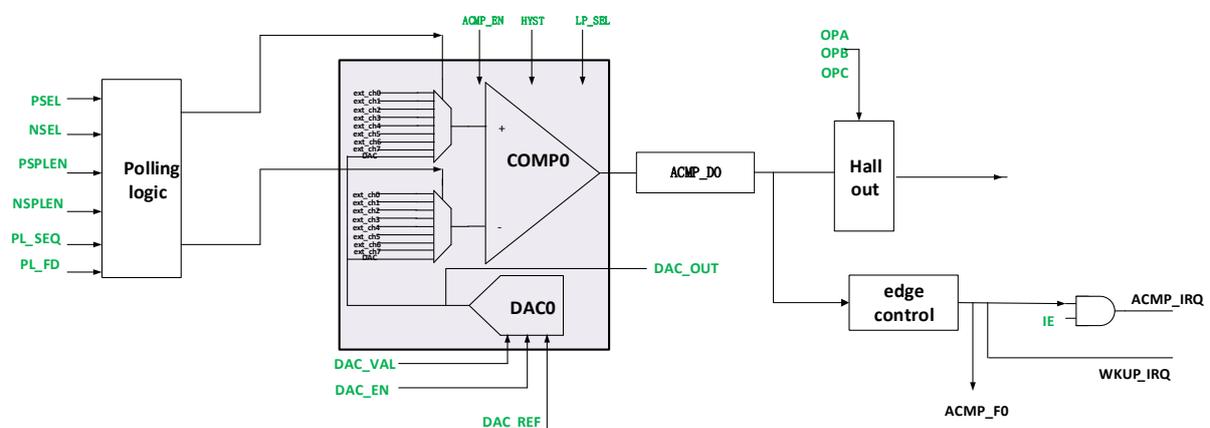


图 9-1 ACMP 结构框图

9.4 功能描述

ACMP 模块就功能而言由两部分组成：数模转换器 (DAC)和比较器 (CMP)。

DAC 包含一个 64 级 DAC（数模转换器）和相关的控制逻辑。通过配置 ACMP_CR2[DACREF]，DAC 可选择 Vdd 或片上带隙基准源作为 DAC 输入源。在 DAC 使能后，将 ACMP_CR2[DACVAL]中设置的数据转换为步进式模拟输出，馈入输入端进行比较。DAC 电压也可同时输出到外部引脚。

ACMP 可以实现正输入和负输入的模拟比较，然后提供一个数字输出和相关的中断。模拟比较器的正负输入均可从 9 个通用输入中选择。

9.4.1 普通模式

普通模式下，正输入和负输入固定所选中通道进行比较，比较结果以数字输出呈现。正输入比负输入电压高则输出 1，反之输出 0。

只要输出出现设置的有效边沿变化，SR 的状态位就会变为有效值。

如果 IE 置位，则发生中断。

ACMP 输出由总线时钟同步生成比较结果，以便 CPU 能读出比较结果。

数据寄存器根据比较结果而改变，因此它可以用做一个跟踪标志，连续指示输入的电压变化。

9.4.2 轮询模式

轮询模式下，可通过相关逻辑动态切换比较器正输入或负输入的输入通道。轮询序列在 ACMP_CR4[PLSEQ]中定义，切换频率由 ACMP_FD[PLFD]控制。ACMP_CR3[PSPLEN]和 ACMP_CR3[NSPLEN]是轮询模式的使能位这两个 bit 无法同时使能，如果都使能不会触发轮询模式，因此，软件必须确保只使能上述两个字段其中之一。

这里提供一个有关轮询模式的示例。正输入轮询，轮询频率为 source_clk/67，外部通道 1-4 和 DAC 输出作轮询，负输入选择外部通道 0，下降沿触发中断。

步骤 1: ACMP_CR0[IE] = 1'b1, ACMP_CR0[MOD] = 2'b00;

步骤 2: ACMP_CR2[DACEN] = 1'b1, ACMP_CR2[DACVAL]=value;

步骤 3: ACMP_CR3[PSPLEN] = 1'b1, ACMP_CR3[NSPLEN] = 1'b0;

步骤 4: ACMP_FD[PLFD] = 2'b01, ACMP_CR4[PLSEQ] = 9'b100011110, ACMP_CR1[NSEL] = 4'b0000;

步骤 5: ACMP_CR0[EN] = 1'b1。

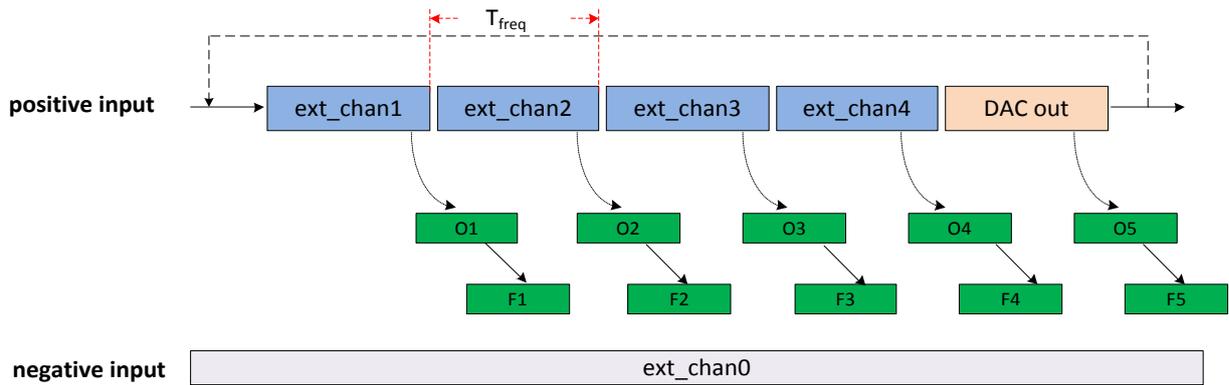


图 9-2 轮询模式工作流程图

9.4.3 轮询模式下霍尔输出

ACMP 有三个 hall 输出: Hall A Output, Hall B Output 和 Hall C Output, 这些信号连接到芯片内部的 PWD 模块。hall 输出与轮询功能配合, 可获取到无传感电机的 hall 位置 (通过电机反电动势检测电机转子所在位置)。每个 hall 输出都可以通过轮询模式选择 9 个通道其中之一。

例如, 若轮询模式 $ACMP_CR4[PLSEQ] = 9'b000001110$, 则轮询顺序为: 外部输入 1 -> 外部输入 2 -> 外部输入 3。

设置 $ACMP_OPA[OPASEL] = 4'b0010$, 则 Hall A Output 为 $ACMP_DR[O2]$ 。

9.4.4 迟滞

ACMP 支持一种迟滞模式, 如下所示:

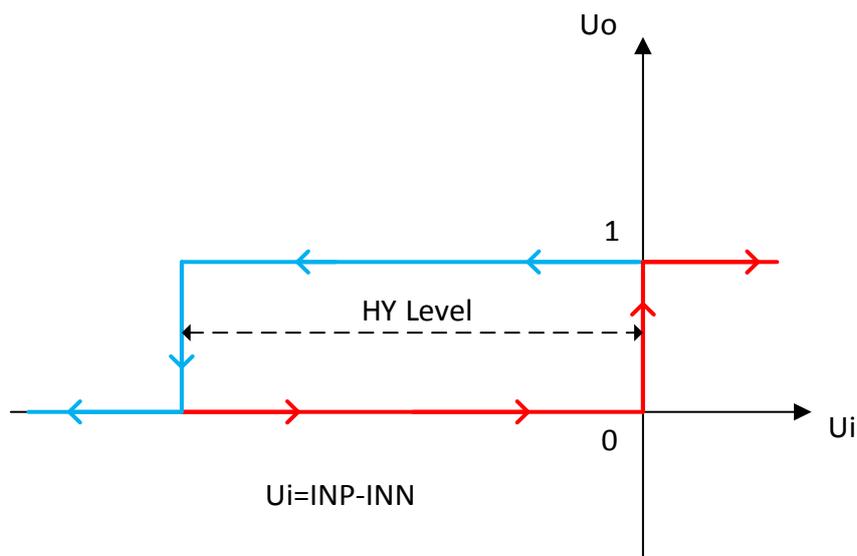


图 9-3 迟滞工作原理

9.4.5 DAC 输出

DAC 支持输出方式支持以下三种:

- 仅输出到 ACMP，用作内部使用，不输出到外部引脚
- 不经过内部缓冲放大器输出到外部引脚
- 通过内部缓冲放大器输出到外部引脚

以上使用方式相关的寄存器配置如下图所示:

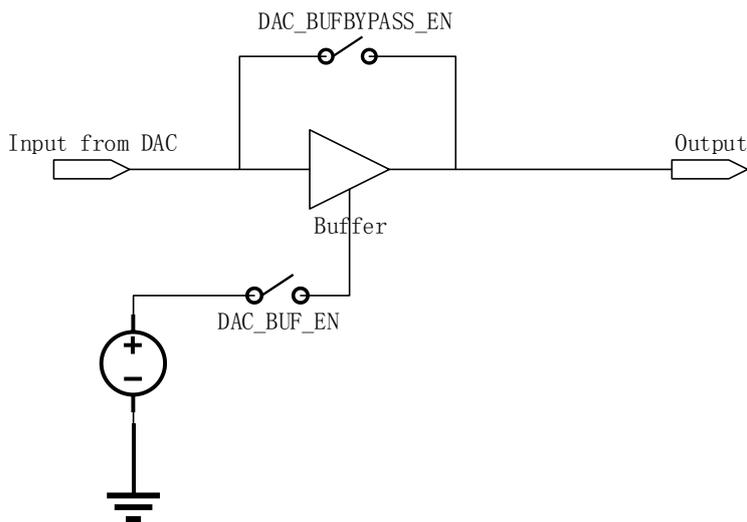


图 9-4 DAC 输出配置示意图

9.4.6 低功耗模式唤醒

在 stop 模式下，ACMP 输出的有效边沿触发中断，可将 MCU 从低功耗模式唤醒。通过向 WPF 写 1 来清除唤醒标志位。



唤醒功能仅作用于普通模式，轮询模式在低功耗模式下不工作。

9.5 寄存器定义

表 9-1 ACMP 寄存器映射

ACMP0 基地址: 0x40005000

Analog 基地址: 0x40008800

地址	名称	宽度	描述
----	----	----	----

ACMPx 基地址+0x0	ACMP_CR0	32	配置寄存器 0
ACMPx 基地址+0x4	ACMP_CR1	32	配置寄存器 1
ACMPx 基地址+0x8	ACMP_CR2	32	配置寄存器 2
ACMPx 基地址+0xC	ACMP_CR3	32	配置寄存器 3
ACMPx 基地址+0x10	ACMP_CR4	32	配置寄存器 4
ACMPx 基地址+0x14	ACMP_DR	32	数据输出寄存器
ACMPx 基地址+0x18	ACMP_SR	32	状态寄存器
ACMPx 基地址+0x1C	ACMP_FD	32	轮询分频器寄存器
ACMPx 基地址+0x20	ACMP_OPA	32	hall A 输出设置寄存器
ACMPx 基地址+0x24	ACMP_OPB	32	hall B 输出设置寄存器
ACMPx 基地址+0x28	ACMP OPC	32	hall C 输出 设置寄存器
ACMPx 基地址+0x2C	ACMP_DACSR	32	DAC 参考源选择寄存器
Analog 基地址+0x30	ACMP_CFG	32	模拟配置寄存器

9.5.1 配置寄存器 0(ACMP_CR0)

表 9-2 ACMP_CR0 寄存器

ACMP_CR0		配置寄存器 0										Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称								EN					IE	OUTEN	OPE	MOD
访问								RW					RW	RW	RW	RW
Reset								0					0	0	0	0

字段	说明
7 EN	ACMP 使能 0: 禁用 1: 使能
4 IE	中断使能 0: 禁用 1: 使能
3 OUTEN	比较结果输出到外部 PIN 0: 禁用 1: 使能
2 OPE	hall 输出使能 0: 禁用 1: 使能

字段	说明
1: 0 MOD	中断触发模式 00b: 输出下降沿中断 01b: 输出上升沿中断 10b: 输出下降沿中断(注: 00 和 10 是相同的配置) 11b: 输出下降沿或上升沿中断

9.5.2 配置寄存器 1(ACMP_CR1)

表 9-3 ACMP_CR1 寄存器

ACMP_CR1		配置寄存器 1														Reset:00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									PSEL				NSEL			
访问									RW				RW			
Reset									0				0			

字段	说明
7: 4 PSEL	正输入选择 0000b: 外部输入 0 0001b: 外部输入 1 0010b: 外部输入 2 0011b: 外部输入 3 0100b: 外部输入 4 0101b: 外部输入 5 0110b: 外部输入 6 0111b: 外部输入 7 1000b: DAC 输出 其他值为无效值
3: 0 NSEL	负输入选择 0000b: 外部输入 0 0001b: 外部输入 1 0010b: 外部输入 2 0011b: 外部输入 3 0100b: 外部输入 4 0101b: 外部输入 5 0110b: 外部输入 6 0111b: 外部输入 7 1000b: DAC 输出

字段	说明
	其他值为无效值

9.5.3 配置寄存器 2(ACMP_CR2)

表 9-4 ACMP_CR2 寄存器

ACMP_CR2		配置寄存器 2										Reset:00000000					
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称									DACEN		DACVAL						
访问									RW		RW						
Reset									0		0						

字段	说明
7 DACEN	DAC 使能 0: 禁用 1: 使能
5: 0 DACVAL	DAC 输出电压值

9.5.4 配置寄存器 3(ACMP_CR3)

表 9-5 ACMP_CR3 寄存器

ACMP_CR3		配置寄存器 3										Reset:00000000						
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称									PSPLEN		NSPLEN							
访问									RW		RW							
Reset									0		0							

字段	说明
7 PSPLEN	使能正输入为轮询模式 0: 禁用 1: 使能

字段	说明
3 NSPLEN	<p>注意：PSPLEN 和 NSPLEN 无法同时使能。PSPLEN 和 NSPLEN 都使能不会触发轮询模式。</p> <p>使能负输入为轮询模式</p> <p>0: 禁用 1: 使能</p> <p>注意：PSPLEN 和 NSPLEN 无法同时使能。PSPLEN 和 NSPLEN 都使能不会触发轮询模式。</p>

9.5.5 配置寄存器 4(ACMP_CR4)

表 9-6 ACMP_CR4 寄存器

ACMP_CR4	配置寄存器 4																Reset:00000000							
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
名称																								
访问																								
Reset																								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
名称								PLSEQ																
访问								RW																
Reset								0																

字段	说明
8: 0 PLSEQ	<p>轮询通道序列设置</p> <p>Bit0~8 分别代表外输输入通道 0~7 和 DAC 通道</p> <p>0: 禁用相应的通道 1: 使能相应的通道</p>

9.5.6 数据输出寄存器(ACMP_DR)

表 9-7 ACMP_DR 寄存器

ACMP_DR		数据输出寄存器										Reset:00000000					
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称							O	O8	O7	O6	O5	O4	O3	O2	O1	O0	
访问							RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	
Reset							0	0	0	0	0	0	0	0	0	0	
字段	说明																
9	正常模式输出																
0	0: 正常模式比较结果为 0 1: 正常模式比较结果为 1																
8	轮询模式通道 8 输出																
O8	0: DAC 通道比较结果为 0 1: DAC 通道比较结果为 1																
7	轮询模式通道 7 输出																
O7	0: 外部通道 7 比较结果为 0 1: 外部通道 7 比较结果为 1																
6	轮询模式通道 6 输出																
O6	0: 外部通道 6 比较结果为 0 1: 外部通道 6 比较结果为 1																
5	轮询模式通道 5 输出																
O5	0: 外部通道 5 比较结果为 0 1: 外部通道 5 比较结果为 1																
4	轮询模式通道 4 输出																
O4	0: 外部通道 4 比较结果为 0 1: 外部通道 4 比较结果为 1																
3	轮询模式通道 3 输出																
O3	0: 外部通道 3 比较结果为 0 1: 外部通道 3 比较结果为 1																
2	轮询模式通道 2 输出																
O2																	

	0: 外部通道 2 比较结果为 0 1: 外部通道 2 比较结果为 1
1 O1	轮询模式通道 1 输出 0: 外部通道 1 比较结果为 0 1: 外部通道 1 比较结果为 1
0 O0	轮询模式通道 0 输出 0: 外部通道 0 比较结果为 0 1: 外部通道 0 比较结果为 1

9.5.7 状态寄存器(ACMP_SR)

表 9-8 ACMP_SR 寄存器

ACMP_SR	状态寄存器																Reset:00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称						WPF	F	F8	F7	F6	F5	F4	F3	F2	F1	F0	
访问						R/W1C	R/W1										
Reset						0	0	0	0	0	0	0	0	0	0	0	

字段	说明
10 WPF	低功耗模式唤醒中断标志 0: 未触发低功耗模式唤醒标志 1: 触发低功耗模式唤醒标志 写 1 清除该标志
9 F	正常模式中断标志 0: 未触发正常模式边沿标志 1: 触发正常模式边沿标志 写 1 清除该标志
8 F8	轮询模式通道 8 中断标志 0: 未触发 DAC 通道边沿标志 1: 触发 DAC 通道边沿标志 写 1 清除该标志

字段	说明
7 F7	轮询模式通道 7 中断标志 0: 未触发外部通道 7 边沿标志 1: 触发外部通道 7 边沿标志 写 1 清除该标志
6 F6	轮询模式通道 6 中断标志 0: 未触发外部通道 6 边沿标志 1: 触发外部通道 6 边沿标志 写 1 清除该标志
5 F5	轮询模式通道 5 中断标志 0: 未触发外部通道 5 边沿标志 1: 触发外部通道 5 边沿标志 写 1 清除该标志
4 F4	轮询模式通道 4 中断标志 0: 未触发外部通道 4 边沿标志 1: 触发外部通道 4 边沿标志 写 1 清除该标志
3 F3	轮询模式通道 3 中断标志 0: 未触发外部通道 3 边沿标志 1: 触发外部通道 3 边沿标志 写 1 清除该标志
2 F2	轮询模式通道 2 中断标志 0: 未触发外部通道 2 边沿标志 1: 触发外部通道 2 边沿标志 写 1 清除该标志
1 F1	轮询模式通道 1 中断标志 0: 未触发外部通道 1 边沿标志 1: 触发外部通道 1 边沿标志 写 1 清除该标志
0 F0	轮询模式通道 0 中断标志 0: 未触发外部通道 0 边沿标志 1: 触发外部通道 0 边沿标志 写 1 清除该标志

9.5.8 轮询分频器寄存器(ACMP_FD)

表 9-9 ACMP_FD 寄存器

ACMP_FD		轮询分频器寄存器														Reset:00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称															PLFD		
访问															RW		
Reset															0	0	

字段	说明
1: 0	轮询模式分频器
PLFD	此分频器控制轮询通道的切换频率
	00b: source_clk/171
	01b: source_clk/67
	10b: source_clk/47
	11b: source_clk/34

9.5.9 霍尔输出 A 设置寄存器(ACMP_OPA)

表 9-10 ACMP_OPA 寄存器

ACMP_OPA		Hall A 输出设置寄存器														Reset:00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称													OPASEL				
访问													RW				
Reset													0				

字段	说明
3: 0	Hall A 输出 设置
OPASEL	0000b: 轮询通道 0
	0001b: 轮询通道 1
	0010b: 轮询通道 2

字段	说明
	0011b: 轮询通道 3
	0100b: 轮询通道 4
	0101b: 轮询通道 5
	0110b: 轮询通道 6
	0111b: 轮询通道 7
	1000b: 轮询 DAC
	其他值为无效值

9.5.10 霍尔输出 B 设置寄存器(ACMP_OPB)

表 9-11 ACMP_OPB 寄存器

ACMP_OPB		Hall B 输出设置寄存器														Reset:00000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	OPBSEL	
名称																OPBSEL		
访问																RW		
Reset																0		

字段	说明
3: 0	Hall B 输出设置
OPBSEL	0000b: 轮询通道 0
	0001b: 轮询通道 1
	0010b: 轮询通道 2
	0011b: 轮询通道 3
	0100b: 轮询通道 4
	0101b: 轮询通道 5
	0110b: 轮询通道 6
	0111b: 轮询通道 7
	1000b: 轮询 DAC
	其他值为无效值

9.5.11 霍尔输出 C 设置寄存器(ACMP_OPC)

表 9-12 ACMP_OPC 寄存器

ACMP_OPC																Hall C 输出设置寄存器				Reset:00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
名称																							
访问																							
Reset																							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
名称													OPBSEL										
访问													RW										
Reset													0										

字段	说明
3: 0	Hall C 输出设置
OPBSEL	0000b: 轮询通道 0 0001b: 轮询通道 1 0010b: 轮询通道 2 0011b: 轮询通道 3 0100b: 轮询通道 4 0101b: 轮询通道 5 0110b: 轮询通道 6 0111b: 轮询通道 7 1000b: 轮询 DAC 其他值为无效值

9.5.12 DAC 参考源选择寄存器(ACMP_DACSR)

表 9-13 ACMP_DACSR 寄存器

ACMP_DACSR																DAC 参考源选择寄存器				Reset:00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
名称																							
访问																							
Reset																							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
名称													DACREF										
访问													RW										
Reset													0										

字段	说明
0 DACREF	DAC 参考选择
	0: DAC 选择带隙作为参考 1: DAC 选择 Vdd 作为参考

9.5.13 模拟配置寄存器(ACMP_CFG)

表 9-14 ACMP_CFG 寄存器

ACMP_CFG	ACMP 模拟配置寄存器												Reset:01400000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称									HYS_SEL		LPF_FREQ_SEL				DA C_B UF_ EN	DAC_B UFBYP ASS_E N
访问									RW	RW	RW	RW			RW	RW
Reset									0	1	0	0			0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称																
访问																
Reset																

字段	说明
23: 22 HYS_SEL	模拟比较器 0 迟滞电压选择 00b: 无迟滞 01b: 10mV 10b: 20mV 11b: 40mV
21: 20 LPF_FREQ_SEL	低通滤波器选择 00b: 500kHz 01b: 1MHz 10b: 2MHz 11b: 无低通滤波
17 DAC_BUF_EN	DAC 内部缓冲放大器使能 0: 关闭 DAC 内部缓冲放大器供电 1: 开启 DAC 内部缓冲放大器电源
16 DAC_BUFBYPASS_EN	跳过 DAC 内部缓冲放大器配置 0: DAC 输出要经过内部缓冲放大器 1: DAC 输出不经过内部缓冲放大器

10 脉宽调制 (PWM)

10.1 简介

PWM 模块是一个多功能的定时器，支持输入捕获、输出比较、正交解码和 PWM 信号生成。PWM 的计数功能是通过一个 16 位的计数器产生的。该 MCU 设备包含 3 个 PWM 模块，其中 PWM0、PWM1 模块支持 2 通道，PWM2 模块支持 4 通道。

10.2 特性

PWM 特性包括：

- PWM 时钟源为总线时钟
- 16 位预分频器支持 1 至 65536 分频
- 16 位计数器
 - 它可以为一个自由运行、没有限制的计数器，或一个有初值和终值的计数器
 - 支持向上、向上-向下两种计数方式
- 每个通道都可以配置为输入捕获、输出比较或边沿对齐 PWM 模式、中心对齐 PWM 模式
- 在输入捕获模式下，捕获可以发生在上升沿、下降沿或上升沿/下降沿
- 输入捕获模式下，所有通道均可以选择输入滤波器
- 在输出比较模式下，可以在匹配时设置、清除或者翻转输出
- 每对通道都可以组合起来生成一个 PWM 信号，并且能够独立控制 PWM 信号的上升沿和下降沿
- PWM 通道可以采用具有同等输出或者互补输出的成对工作方式
- 死区插入可用于每一对组合通道，互补/非互补模式均可插入死区
- 可生成匹配触发
- 软件控制 PWM 输出
- 输出屏蔽可设置通道为无效状态
- 每个 PWM 模块的故障控制最多支持 3 路故障输入
- 每个通道的极性可配置
- 每个通道产生一个中断
- 计数器溢出时产生中断
- 当检测到故障条件时，产生中断

- 同步加载写缓冲 PWM 寄存器
- 关键寄存器支持写保护
- 用于脉冲和周期宽度测量的双边沿捕获
- 支持正交解码（AB 相输入引脚映射到每个 PWM 模块的 CH0 和 CH1）
- 支持全局时基
- 支持 PWM 输出波形相位偏移

10.3 结构框图

PWM 每通道使用一个输入/输出（I/O）引脚、CH_n（PWM 通道（n）），其中 n 是通道编号（PWM0、PWM1 中 n = 0/1 为两通道模块，PWM2 中 n = 0/1/2/3 为四通道模块）。

下图为 PWM 结构图。PWM 的核心部分为 16 位计数器，具有可编程的初始值和最终值，其计数可以是向上或向上-向下。

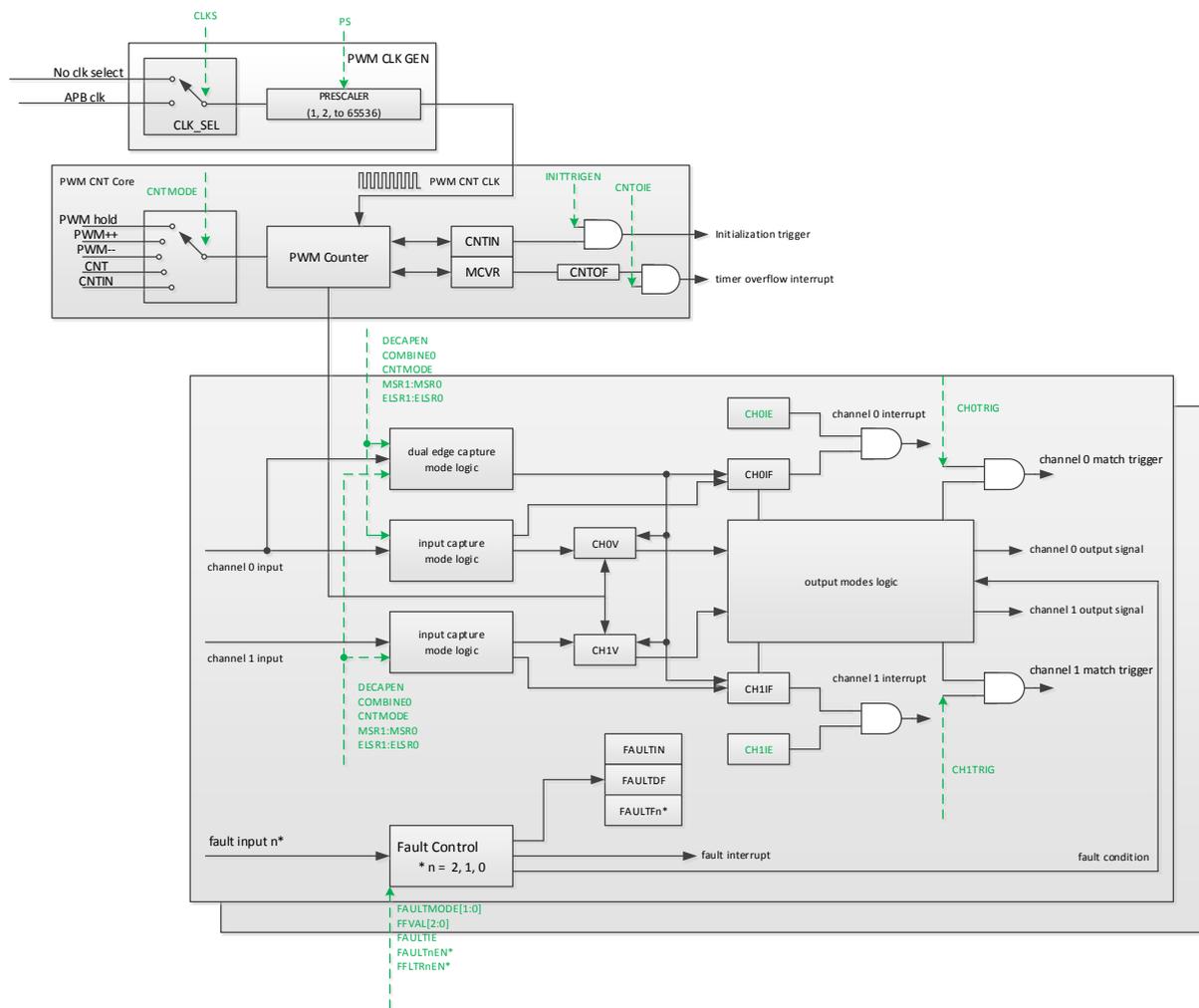


图 10-1 PWM 结构框图

10.4 传统 PWM 功能描述

10.4.1 时钟源

`PWM_INIT` 寄存器中的 `CLKSRC` 位选择 PWM 计数器的时钟源为 APB 时钟或者禁用 PWM 计数器。MCU 复位后，`CLKSRC = 0`，因此没有选择时钟源。通过将 `CLKSRC` 位写入 0 来禁用 PWM 计数器不会影响 PWM 计数器值或其他寄存器。

10.4.2 计数器

PWM 包含一个 16 位计数器，用于通道输入或输出模式。PWM 计数器时钟是由预分频器分频的选定时钟。PWM 计数器具有以下工作模式：

- 向上计数
- 向上-向下计数
- 正交解码模式

10.4.2.1 向上计数

当 `QDIEN=0` 且 `CNTMODE=0` 时，为向上计数模式。`CNTIN` 定义计数的起始值，`MCVR` 定义计数的终值，如下图所示。`CNTIN` 的值加载到 PWM 计数器中，计数器递增，直到达到 `MCVR` 的值，此时计数器重新加载 `CNTIN` 的值。向上计数模式 PWM 周期为 $(MCVR - CNTIN + 0x0001) \times$ PWM 计数器时钟周期。当 PWM 计数器从 `MCVR` 变为 `CNTIN` 时，`CNTOF` 位置 1。

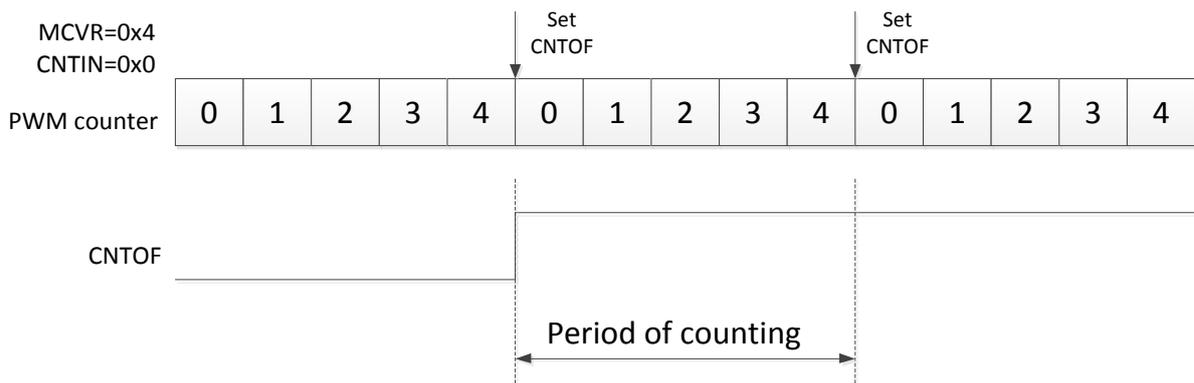


图 10-2 向上计数

10.4.2.2 向上-向下计数

当 `QDIEN=0` 且 `CNTMODE=1` 时，选择向上-向下计数。`CNTIN` 定义计数的起始值，`MCVR` 定义计数的终值。`CNTIN` 的值被加载至 PWM 计数器中，并且计数器递增直到达到 `MCVR` 的值，此时计数器递减，直到它返回到 `CNTIN` 的值，然后重新开始上下计数。

向上-向下计数时的 PWM 周期为 $2 \times (MCVR - CNTIN) \times \text{PWM 计数器时钟周期}$ 。当 PWM 计数器从 MCVR 变为 MCVR - 1 时，CNTOF 位置 1，如下图所示。

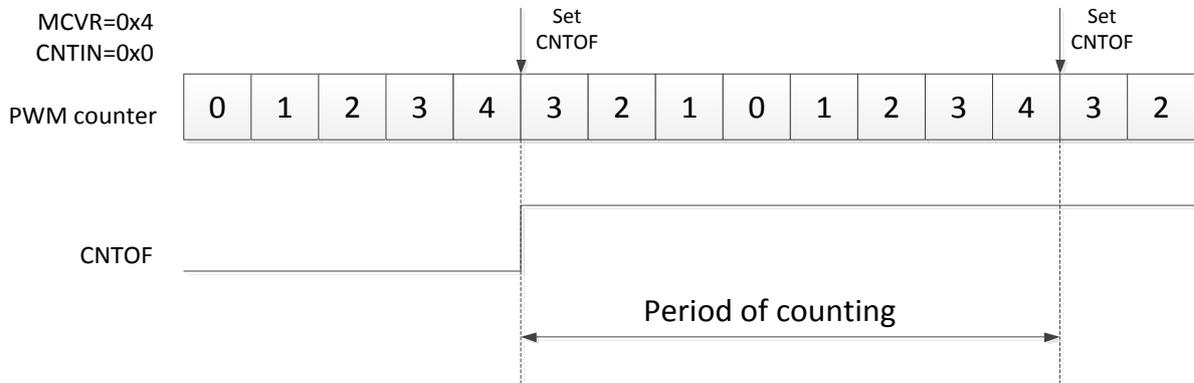


图 10-3 向上-向下计数

10.4.3 工作模式

PWM 可配置为输入捕获、输出比较或边沿对齐 PWM 模式、中心对齐 PWM 模式、组合模式、正交解码模式，详细配置参考以下表格。

表 10-1 工作模式配置

QDIEN	DECAPEN	COMBINE	CNTMODE	MSR1: MSR0	ELSR1: ELSR0	工作模式	配置
0	0	0	0	0 0	0 1	输入捕获	上升沿捕获
0	0	0	0	0 0	1 0		下降沿捕获
0	0	0	0	0 0	1 1		上升或下降沿捕获
0	0	0	0	0 1	0 1	输出比较	匹配时翻转输出
0	0	0	0	0 1	1 0		匹配时清除输出
0	0	0	0	0 1	1 1		匹配时设置输出
0	0	0	0	1 X	1 0	边沿对齐 PWM	High-true 脉冲(匹配时清除输出)
0	0	0	0	1 X	X 1		Low-true 脉冲(匹配时设置输出)
0	0	0	1	X X	1 0	中心对齐 PWM	High-true 脉冲(匹配时清除输出)
0	0	0	1	X X	X 1		Low-true 脉冲(匹配时设置输出)
0	0	1	0	X X	1 0	向上计数组合模式	Channel(n) 匹配时设置输出
0	0	1	0	X X	X 1		Channel(n+1) 匹配时清除输出, Channel (n)匹配时清除输出, Channel (n+1) 匹配时设置输出
0	0	1	1	X X	1 0	向上-向下计数组合模式	Channel(n) 匹配时设置输出
0	0	1	1	X X	X 1		Channel(n+1) 匹配时清除输出, Channel (n)匹配时清除输出, Channel (n+1) 匹配时设置输出
0	1	0	0	X 0	0 1	双边沿单次捕获	上升沿
0	1	0	0	X 0	1 0		下降沿
0	1	0	0	X 0	1 1		上升沿或下降沿
0	1	0	0	X 1	0 1	双边沿持续捕获	上升沿
0	1	0	0	X 1	1 0		下降沿
0	1	0	0	X 1	1 1		上升沿或下降沿
1	X	X	X	X X	X X	正交解码	QDIEN=1使能正交解码, 正交解码器模式优先于其他模式

10.4.4 输入捕获模式

当通道输入出现选定的边沿时，PWM 计数器的当前值会被捕获到 **PWM_CHnV** 寄存器中。同时，**CHnIF** 位置 1。如果由 **CHnIE** = 1 使能，则产生通道中断。当通道配置为输入捕获时，**PWMx_CHn** 引脚为边沿敏感输入。**ELSnR1:ELSnR0** 控制位决定何种边沿（下降沿或上升沿）触发输入捕获事件。注意，可以被正确检测到的输入信号最大频率为 1/4 总线时钟，这是满足信号采样的奈奎斯特准则所必需的条件。在输入捕捉模式下，忽略写入 **CHnV** 寄存器操作。如下图，通道配置检测上升沿捕获。

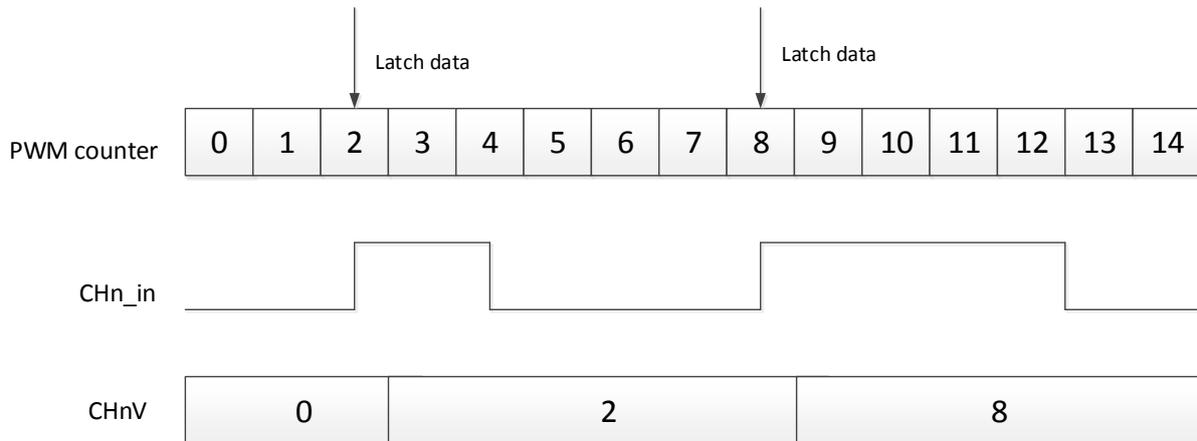


图 10-4 输入捕获模式

对于 PWM 模块的所有通道，均支持捕获滤波器来过滤输入信号。滤波器为 5 位计数器，可通过寄存器 **CHnCAPFVAL** (n = 0,1,2,3) 进行配置。当 **CHnCAPFVAL** [4: 0] = 0 时，滤波器功能被禁用，如果 **CHnCAPFVAL** [4: 0] ≠ 00000，输入信号将被延迟 (**CHnCAPFVAL** [4: 0] x 4) 个总线时钟，然后才传输到边沿检测器。

通道输入滤波器中计数器的时钟是总线时钟的 4 分频。

10.4.5 输出比较模式

在输出比较模式下，PWM 可以生成具有可编程位置、极性、持续时间和频率的定时脉冲。当计数器与输出比较通道的 **CHnV** 值匹配时，可以设置、清除或翻转通道 n 输出。当通道最初配置为翻转 (Toggle) 模式时，通道保持输出先前的值，直到发生第一个输出比较事件。如果在通道 n 匹配时 (PWM 计数器 = **CHnV**) **CHnIE** = 1，则 **CHnIF** 位置 1，产生通道 n 中断。

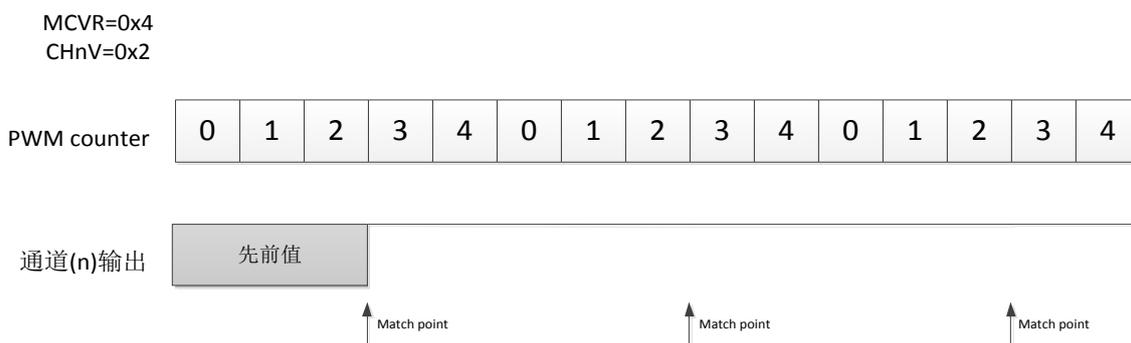


图 10-5 匹配设置输出比较模式

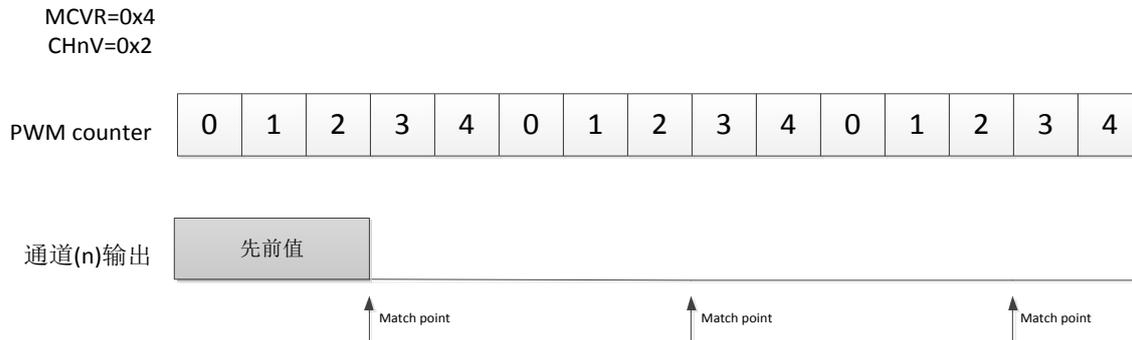


图 10-6 匹配清除输出比较模式

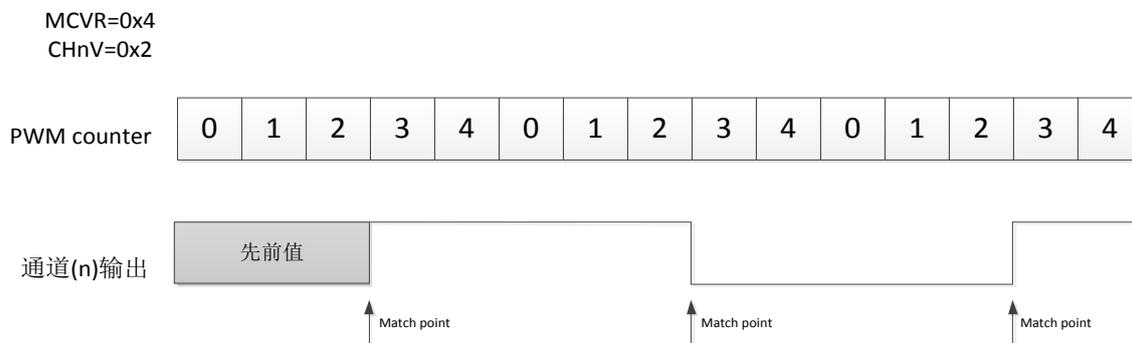


图 10-7 匹配翻转输出比较模式

10.4.6 边沿对齐 PWM 模式(EPWM)

周期=(MCVR-CNTIN + 0x0001) × PWM 计数器时钟周期

脉冲宽度=(CHnV + 0x0001 - CNTIN) × PWM 计数器时钟周期

ELSnR1: ELSnR0=1: 0, 通道(n)输出在 CNTIN 值加载到 PWM 计数器时为高电平, 在通道(n)匹配 (PWM 计数器 = CHnV) 时为低电平。

ELSnR1: ELSnR0=X: 1, 通道(n)输出在 CNTIN 值加载到 PWM 计数器时为低电平, 在通道(n)匹配 (PWM 计数器 = CHnV) 时为高电平。

CHnIE = 1, 通道(n)匹配 (PWM 计数器= CHnV) 时, CHnIF 位置 1 且产生通道(n)中断。

这种类型的 PWM 信号称为边沿对齐, 是因为所有 PWM 信号的前沿与周期的开端对齐, 这对于 PWM 内的所有通道都是相同的。

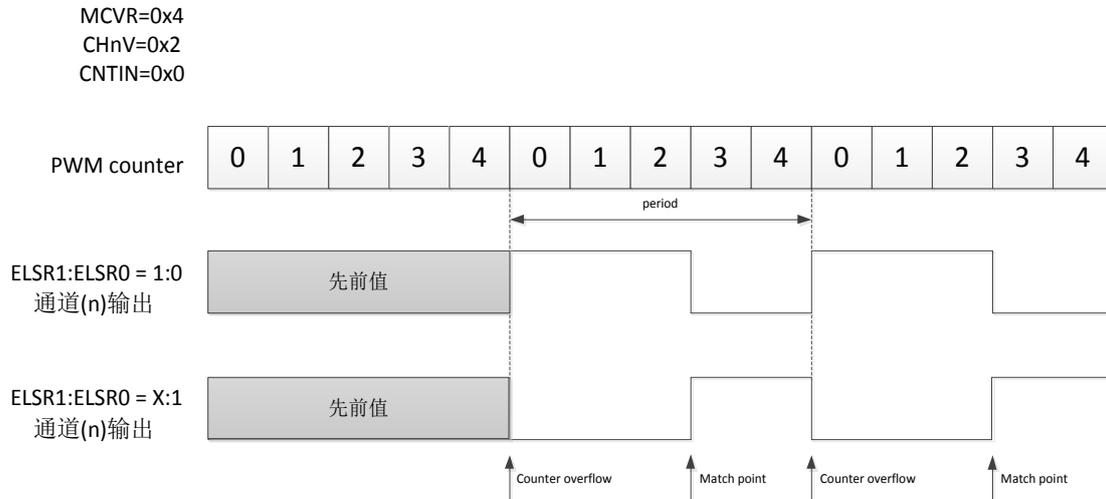


图 10-8 EPWM 波形

ELSnR1: ELSnR0=1: 0, 如果(CHnV = 0x0000), 则通道(n)输出为 0%占空比 EPWM 信号; 如果(CHnV >= MCVR), 则通道(n)输出为 100%占空比 EPWM 信号, ELSnR1: ELSnR0=X: 1 则反之。

10.4.7 中心对齐 PWM 模式(CPWM)

周期=2 × (MCVR - CNTIN) × PWM 计数器时钟周期

脉冲宽度=2 × (CHnV - CNTIN) × PWM 计数器时钟周期

在 CPWM 模式下, PWM 计数器向上计数直到达到 MCVR, 然后向下计数直至达到 CNTIN。

ELSnR1: ELSnR0=1: 0, 通道(n)输出在向下计数时与通道(n)匹配 (PWM 计数器 = CHnV) 下为高电平, 在向上计数时与通道(n)匹配 (PWM 计数器 = CHnV) 的情况下为低电平。

ELSnR1: ELSnR0=X: 1, 通道(n)输出在向下计数时与通道(n)匹配 (PWM 计数器 = CHnV) 时为低电平, 在向上计数时与通道(n)匹配 (PWM 计数器 = CHnV) 的情况下为高电平。

CHnIE=1, 当 PWM 计数减少或 PWM 计数增加, 在通道(n)匹配 (PWM 计数器=CHnV) 时, CHnIF 位置 1 且通道(n)中断产生。

这种类型的 PWM 信号被称为中心对齐, 因为所有通道的脉冲宽度中心与 CNTIN 的值对齐。

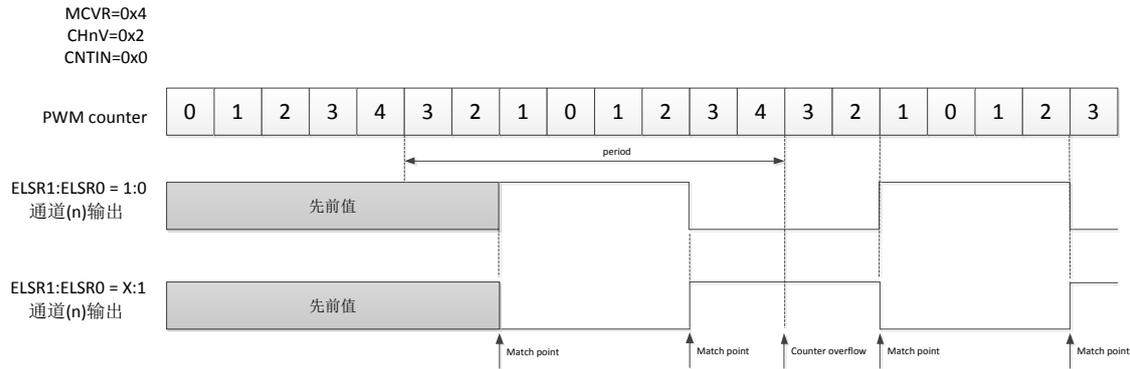


图 10-9 CPWM 波形

ELSnR1: ELSnR0=1: 0, 如果(CHnV = 0x0000), 则通道(n)输出为 0%占空比 CPWM 信号; 如果 (CHnV >= MCVR), 则通道(n)输出为 100%占空比 CPWM 信号, ELSnR1: ELSnR0=X: 1 则反之。

10.4.8 组合模式

组合模式下, 将偶数通道 (n) 和相邻的奇数通道 (n + 1) 组合以在通道 (n) 输出中产生 PWM 信号。

根据计数方式的不同, 可分为向上计数组合模式和向上-向下计数组合模式。

ELSnR1: ELSnR0=1: 0, 在通道(n)匹配时(PWM 计数器=CH(n)V)为高电平; 在通道(n+1)匹配时为低电平。

ELSnR1: ELSnR0=X: 1, 在通道(n)匹配时(PWM 计数器=CH(n)V)为低电平; 在通道(n+1)匹配时为高电平。

CH(n)IE=1, 通道(n)匹配时(PWM 计数器=CH(n)V), CH(n)IF 位置 1 且产生通道(n)中断。

CH(n+1)IE=1, 通道(n+1) 匹配时(PWM 计数器=CH(n+1)V), CH(n+1)IF 位置 1 且产生通道(n+1)中断。



ELSnR1 和 ELSnR0 位不用于控制 channels (n) 和 (n+1) 输出。

10.4.8.1 向上计数组合模式

周期=(MCVR - CNTIN + 0x0001) × PWM 计数器时钟周期

脉冲宽度 = |CH(n+1)V - CH(n)V| × PWM 计数器时钟周期

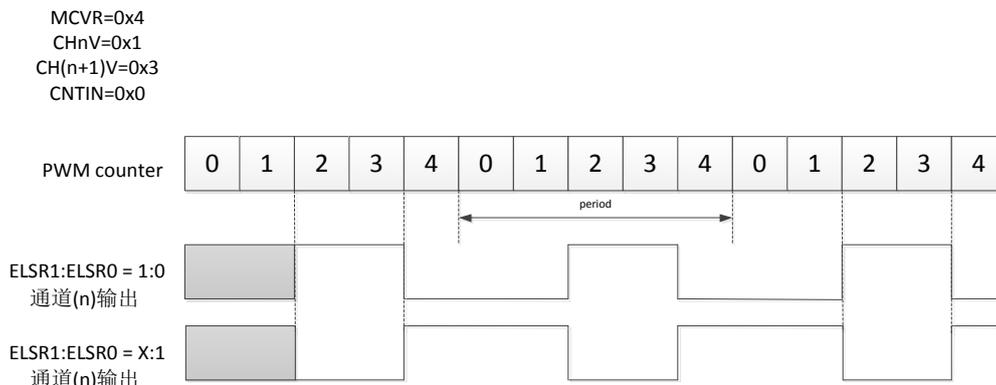


图 10-10 向上计数组合模式输出波形

以下展示了向上计数组合模式的各种条件下 PWM 信号输出波形:

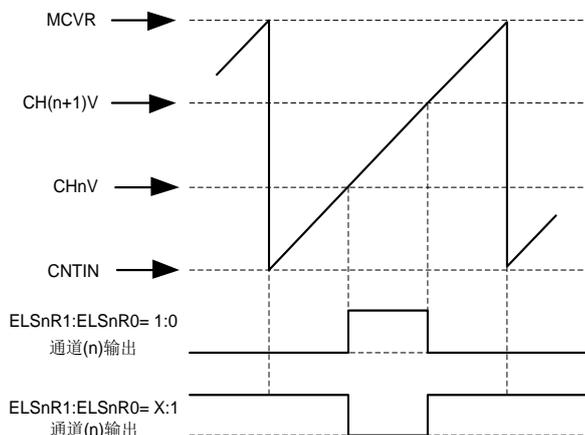


图 10-11 (CNTIN < CHnV/CH(n+1)V < MCVR) & (CHnV < CH(n+1)V)条件下输出波形

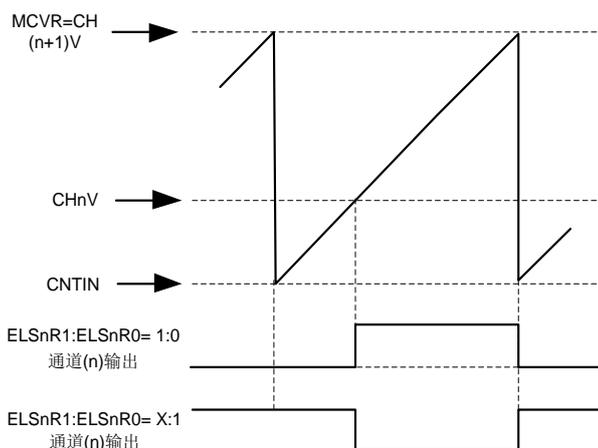


图 10-12 (CNTIN < CHnV < MCVR) & (CH(n+1)V = MCVR)条件下输出波形

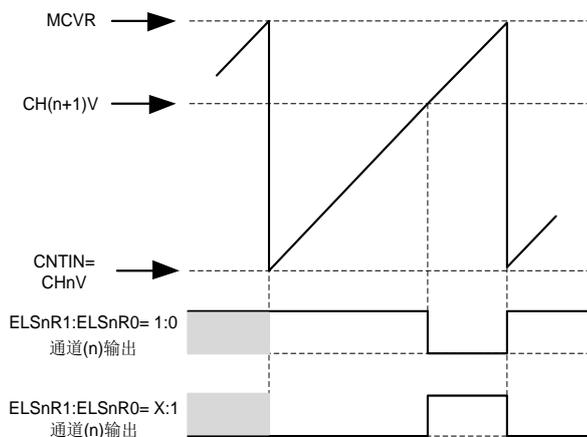


图 10-13 ($CH_nV = CNTIN$) & ($CNTIN < CH(n+1)V < MCVR$) 条件下输出波形

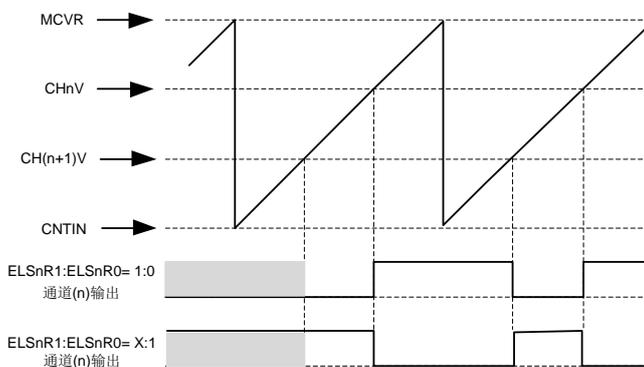


图 10-14 ($CNTIN < CH_nV / CH(n+1)V < MCVR$) 且 ($CH_nV > CH(n+1)V$) 条件下输出波形

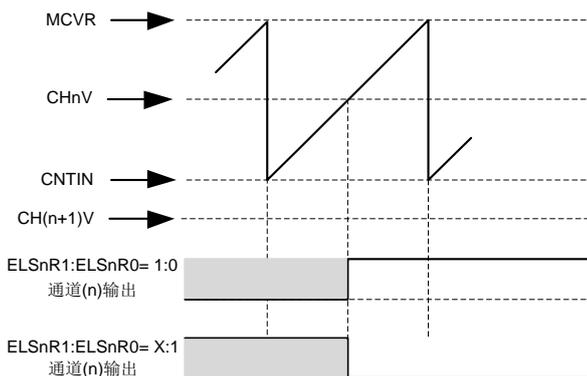


图 10-15 ($CH(n+1)V < CNTIN$) & ($CNTIN < CH_nV < MCVR$) 条件下输出波形

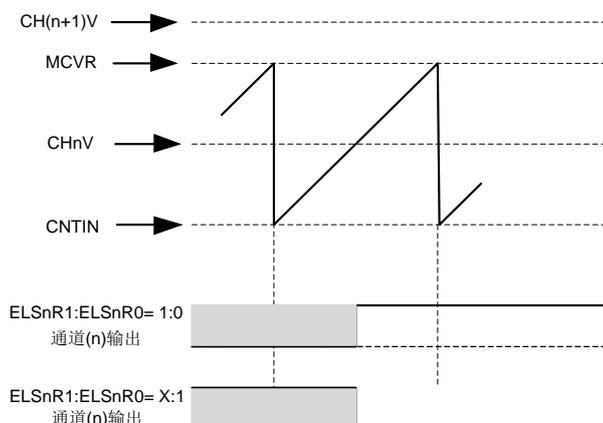


图 10-16 (CH(n+1)V > MCVR)且(CNTIN < CHnV < MCVR)条件下输出波形

10.4.8.2 向上-向下计数组合模式

周期 = $2 \times (MCVR - CNTIN) \times \text{PWM 计数器时钟周期}$

在向上-向下计数过程中，通道在向上计数产生一次匹配，向下计数也会产生一次匹配。为了便于控制通道输出，提供了匹配生效点设置功能，可通过设置匹配生效点 **CHSCR[DIR]**是在向上计数或向下计数过程中作用。因为匹配生效点依赖于计数方向，需要清晰定义向上计数和向下计数区间范围，如下图：

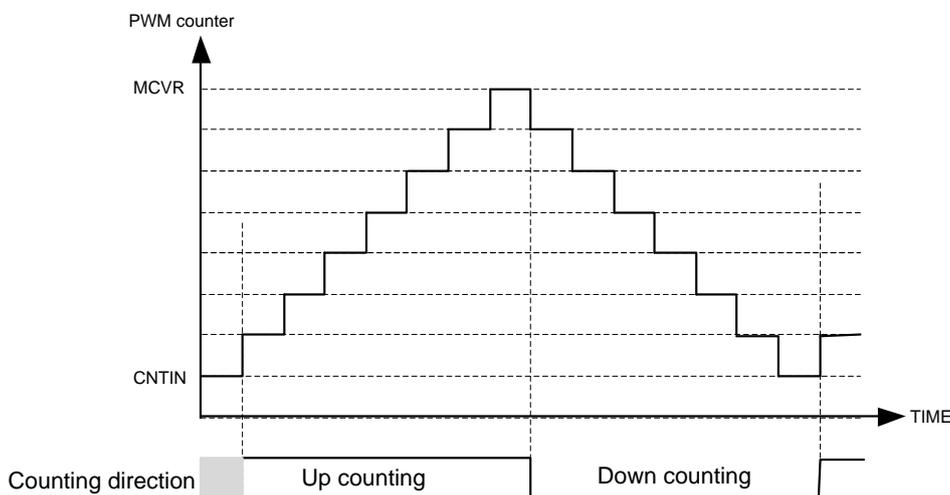


图 10-17 向上-向下计数区间范围

说明

首次计数周期 CNTIN 值为向上计数区间，后续周期 CNTIN 值都为向下计数区间。

2 个匹配点可以组合出 4 种情况:

MCVR=0x4
 CHnV=0x1
 CH(n+1)V=0x3
 CNTIN=0x0

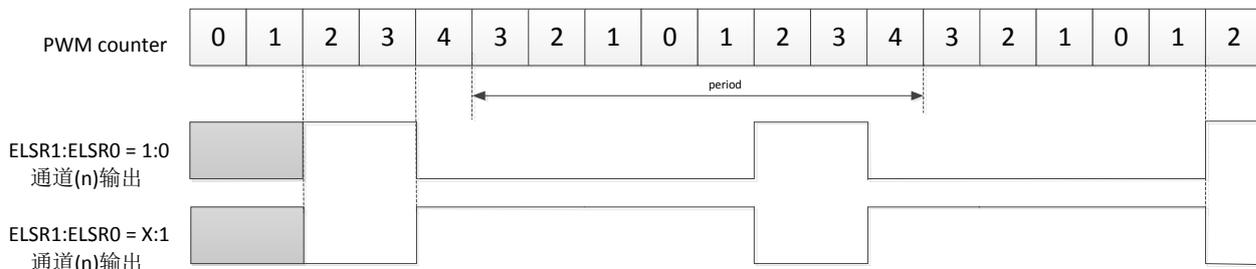


图 10-18 CHn 匹配点 DIR=1(Up), CH(n+1)匹配点 DIR=1(Up)

MCVR=0x4
 CHnV=0x1
 CH(n+1)V=0x3
 CNTIN=0x0

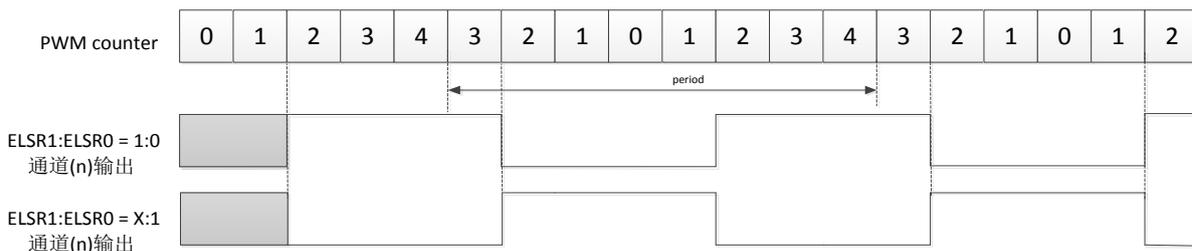


图 10-19 CHn 匹配点 DIR=1(Up), CH(n+1)匹配点 DIR=0(Down)

MCVR=0x4
 CHnV=0x1
 CH(n+1)V=0x3
 CNTIN=0x0

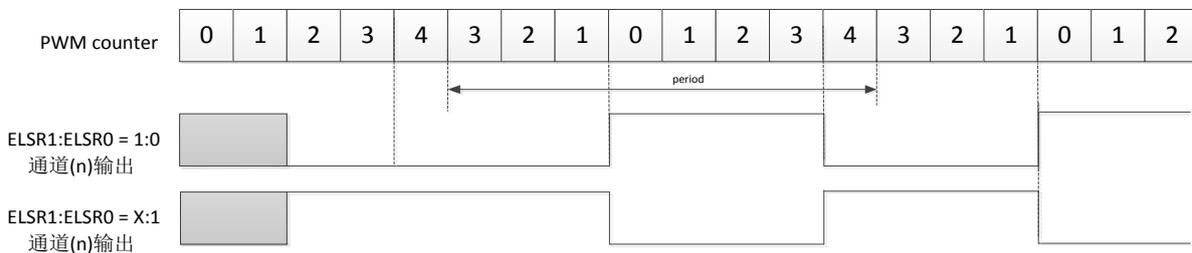


图 10-20 CHn 匹配点 DIR=0(Down), CH(n+1)匹配点 DIR=1(Up)

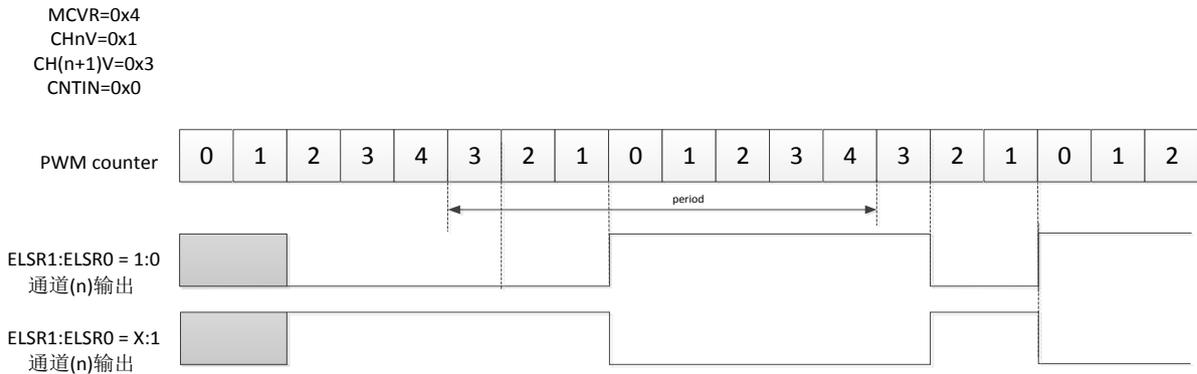


图 10-21 CHnV 匹配点 DIR=0(Down), CH(n+1)V 匹配点 DIR=0(Down)

10.4.8.3 互补功能

组合模式下支持互补功能。使能互补 $PAIRnCOMPEN=1$ ，通道 (n+1) 输出和通道 (n) 输出电平相反。

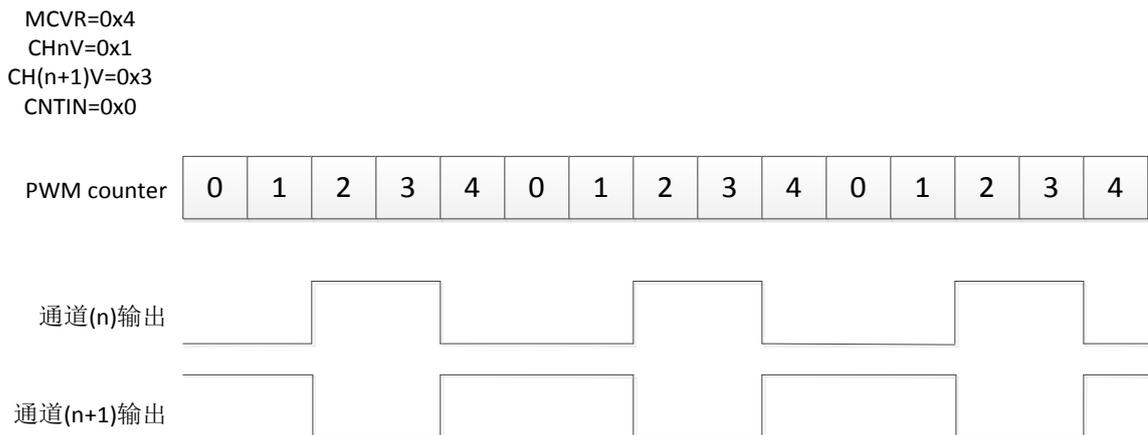


图 10-22 互补模式输出

10.4.8.4 死区时间插入

当 $DTEN = 1$ 且 $DTVAL[5: 0]$ 为非零时，使能死区插入。 PWM_DTSET 寄存器定义了可用于所有 PWM 通道的死区延迟。 $DTPSC[1: 0]$ 位定义总线时钟的预分频器， $DTVAL [5: 0]$ 位定义死区模数，即死区预分频器时钟数。死区延迟插入确保没有两个互补信号（通道 (n) 和 通道(n+1)）同时驱动活动状态。

如果 $CH(n)POL = 0$ ， $CH(n+1)POL = 0$ ，并且使能死区，那么当出现通道 (n) 匹配（PWM 计数器= 通道(n)计数值）时，通道 (n) 输出保持低电平状态，直到死区延迟结束，通道 (n) 输出置位。类似地，当发生通道(n+1)匹配(PWM 计数器 = CH(n+1)V)时，通道 (n+1) 输出保持低电平状态，直到死区延迟结束，通道 (n+1) 输出置位。

如果 $CH(n)POL = 1$ ， $CH(n+1)POL = 1$ ，并且使能死区，则当出现通道(n)匹配(PWM 计数器 = CH(n)V)时，通道 (n) 输出保持高电平状态，直到死区延迟结束，通道 (n) 输出清零时。类似地，当发生

通道 (n+1)匹配(PWM 计数器 = CH(n+1)V)时, 通道 (n+1)输出保持高电平状态, 直到死区延迟结束, 通道 (n+1)输出清零时。

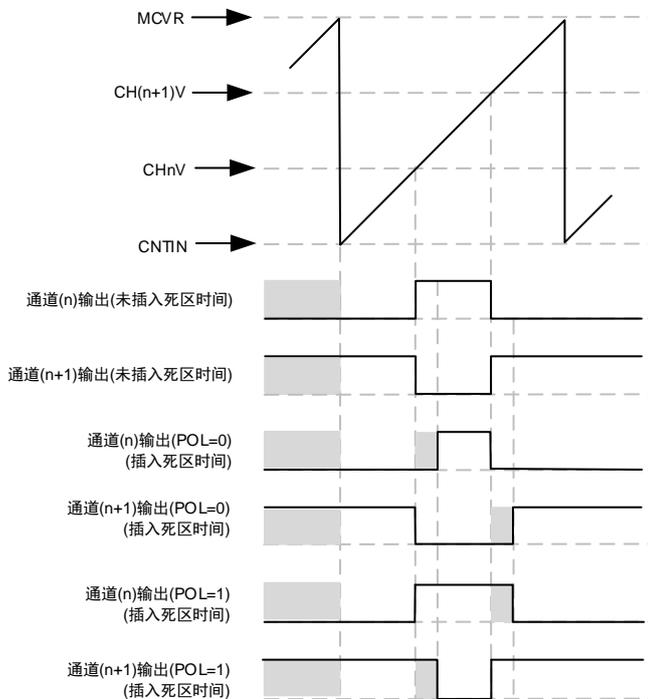


图 10-23 死区时间插入

10.4.8.5 反相

反相功能用于将通道(n)和通道(n+1)输出信号进行交换。在如下情况选择反相操作:

$PAIR(n)INVEN_ = 1, n=0,1,2,3。$

10.4.8.6 移相

在某些情况下, 通道(n+1)匹配会发生在下一个 PWM 周期。当 $(CNTIN < CHnV < MCVR) \& (CNTIN < CH(n+1)V < MCVR)$ 时, $CHnV > CH(n+1)V$ 时会发生此种情景。

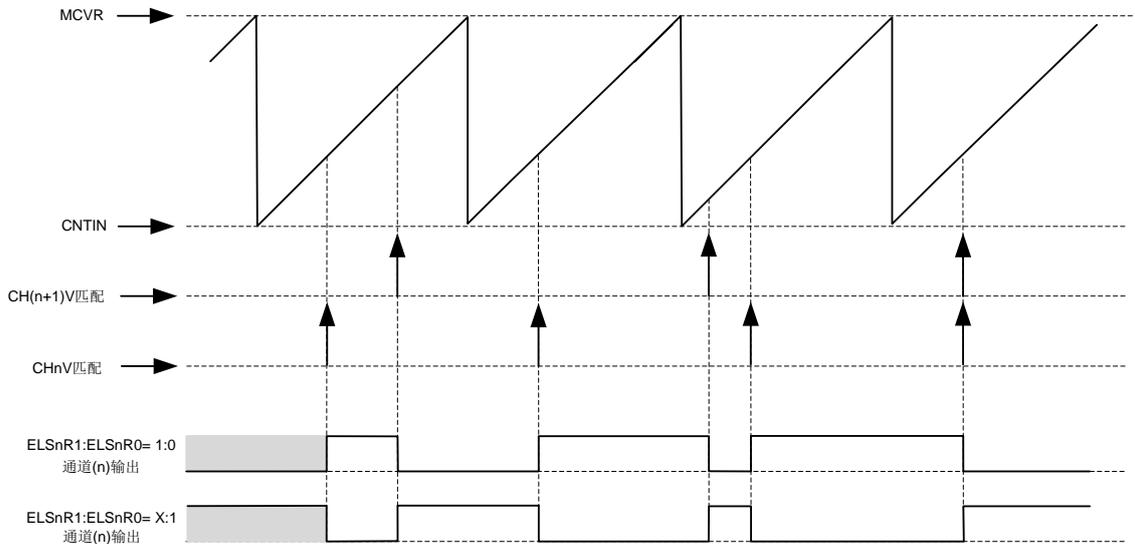


图 10-24 CH(n+1)V 在下一周期匹配的输出生形

基于以上特性，当 PWM 模块多个通道以组合模式输出时，可设定同一个 PWM 模块不同通道对之间的相位偏移量。此种行为有利于产生合适的车灯控制信号，通道输出边沿不重合的信号便于消除噪声。

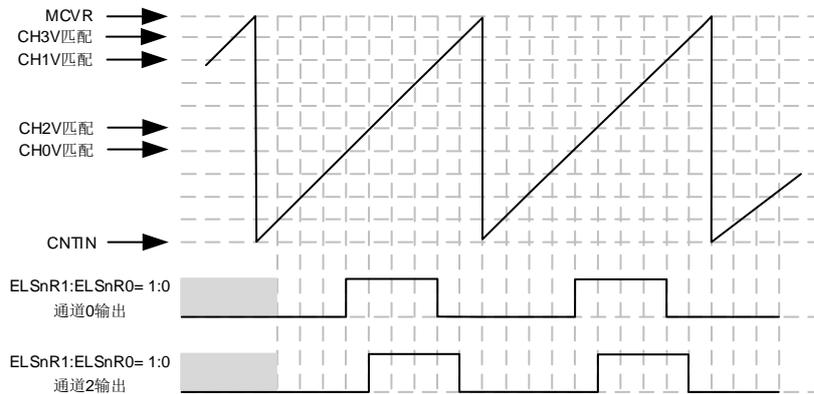


图 10-25 多通道之间相位偏移输出波形

10.4.9 双边沿捕获模式

DECAPEN = 1，则选择双边沿捕捉模式。该模式可用于测量通道(n)输入端脉冲宽度或信号周期。在此模式下，仅使用通道(n)输入，忽略通道(n+1)输入。当 n 为 0 或 2 时，通道(n)滤波器在此模式下有效。ELSnR1: ELSnR0 位选择通道(n)捕获的边沿，ELSn+1R1: ELSn+1R0 位选择通道(n+1)捕获的边沿。如果 ELSnR1: ELSnR0 和 ELSn+1R1: ELSn+1R0 位都选择相同的边沿，则为周期测量；如果这些位选择不同的边沿，则为脉冲宽度测量。

如果在通道(n)输入处检测到通道(n)位选择的边沿，则 CH(n)IF 置位并生成通道(n)中断（如果 CH(n)IE = 1）。如果在通道(n)输入且(CH(n)IF = 1)时检测到通道 (n+1)位选择的边沿，则 CH(n+1)IF 置位且生成通道(n+1)中断（如果 CH(n+1)IE = 1）。双边沿模式不支持同时使能通道(n)和通道(n+1)中断。

当在通道 (n)输入处检测到通道(n)选择的边沿时, PWM_CH(n)V 寄存器存储 PWM 计数器的值。在通道(n)输入处检测到通道(n+1)选择的边沿时, PWM_CH(n+1)V 寄存器存储 PWM 计数器的值。在此模式下, 当读取 PWM_CH(n)V 和 PWM_CH(n+1)V 寄存器时, 唯一的要求是必须在 CH(n+1)V 之前读取 CH(n)V。

如下图, 通道 (n)选择上升沿捕获, 通道 (n+1)选择下降沿捕获, 用于测量脉宽。

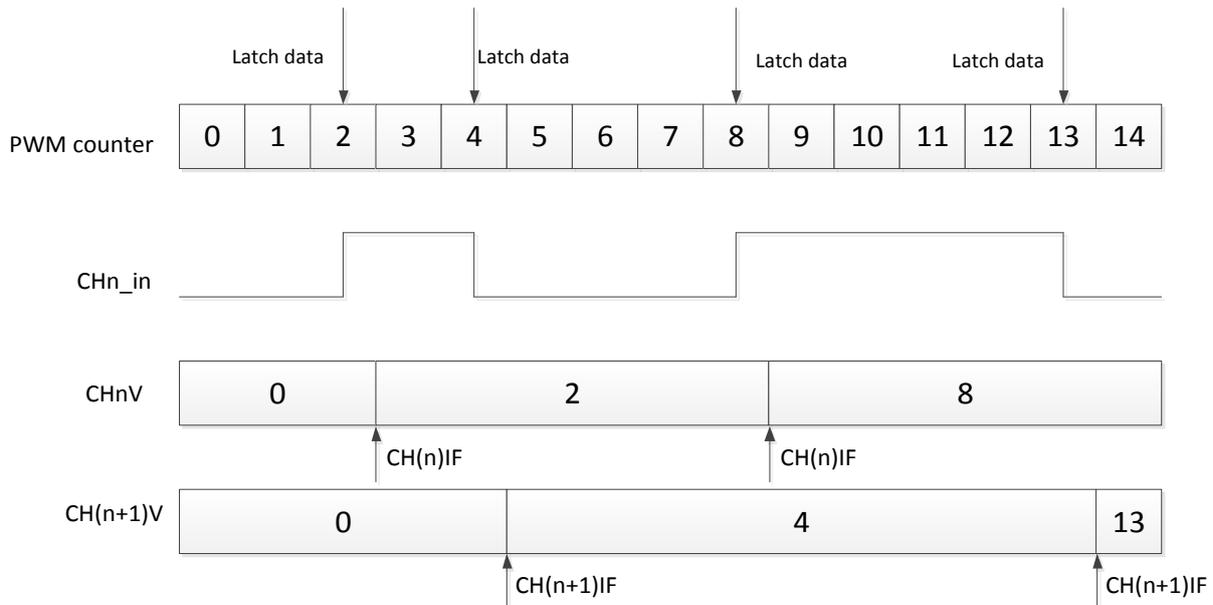


图 10-26 双边沿捕获模式图

10.4.10 正交解码模式

QDIEN = 1 使能正交解码器模式。正交解码器模式采用输入信号相位 A 和相位 B 控制 PWM 计数器递增和递减, 其中信号相位 A 通过 PWM 模块的 CH0 通道输入, 信号相位 B 通过 PWM 模块的 CH1 通道输入。每个输入信号相位 A 和相位 B 都有一个滤波器, 该滤波器和通道输入中使用的滤波器是相同的, 相位 A 的滤波值由 **CH0CAPFVAL**[4:0]位定义, 相位 B 的滤波值由 **CH1CAPFVAL** [4:0]位定义, 值为 0 时禁用滤波器(**CH(m)CAPFVAL**[4:0]位在 **PWM_CAPFILTER** 寄存器中)。

PHAPOL 位选择 A 相输入的极性, **PHBPOL** 位选择 B 相输入的极性。**QUADMODE** 选择正交解码器使用的编码模式。如果 **QUADMODE** = 1, 则使能计数和方向编码模式, 参见下图。在该模式下, 相位 B 的输入表示计数方向, 相位 A 的输入定义计数频率。当相位 A 输入信号有上升沿时, PWM 计数器将被更新。

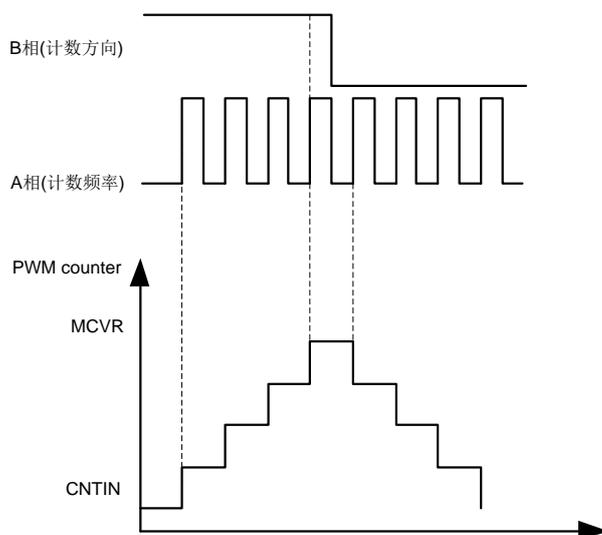


图 10-27 计数和方向编码模式

如果 **QUADMODE** = 0，则启用相位 A 和相位 B 编码模式，参见下图。在这种模式下，相位 A 和 B 信号之间的关系表示计数方向，相位 A 和 B 信号定义计数频率。当相位 A 或相位 B 信号有边沿时，PWM 计数器将被更新。

如果 **PHAPOL** = 0 & **PHBPOL** = 0，则 PWM 计数器递增发生在以下情况：

- A 相信号上升沿时，B 相信号为低电平；
- B 相信号上升沿时，A 相信号为高电平；
- B 相信号下降沿时，A 相信号为低电平；
- A 相信号下降沿时，B 相信号为高电平。

PWM 计数器递减发生在以下情况：

- A 相信号下降沿时，B 相信号为低电平；
- B 相信号下降沿时，A 相信号为高电平；
- B 相信号上升沿时，A 相信号为低电平；
- A 相信号上升沿时，B 相信号为高电平。

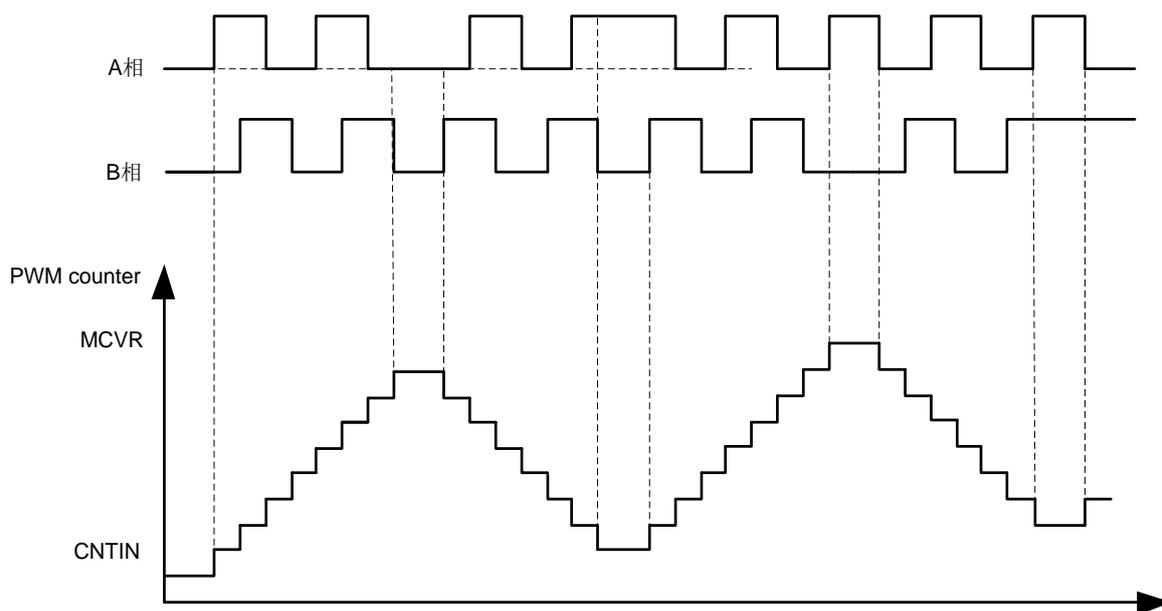


图 10-28 A 相和 B 相编码模式

下图显示了向上计数时 PWM 计数器溢出。当 PWM 计数器从 MCVR 更改为 CNTIN 时，设置 CNTOF 和 CNTOFDIR 位。CNTOF 位表示发生了 PWM 计数器溢出，CNTOFDIR 指示 PWM 计数器在向上计数时发生溢出。

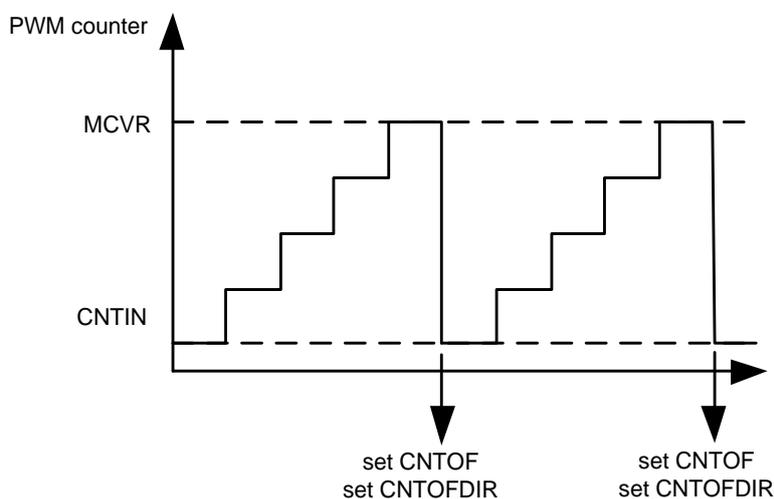


图 10-29 向上计数 PWM counter 溢出

下图显示了向下计数时 PWM 计数器溢出。当 PWM 计数器从 CNTIN 更改为 MCVR 时，将设置 CNTOF 位和清除 CNTOFDIR 位。CNTOF 位表示发生了 PWM 计数器溢出，CNTOFDIR 指示 PWM 计数器在向下计数时发生溢出。

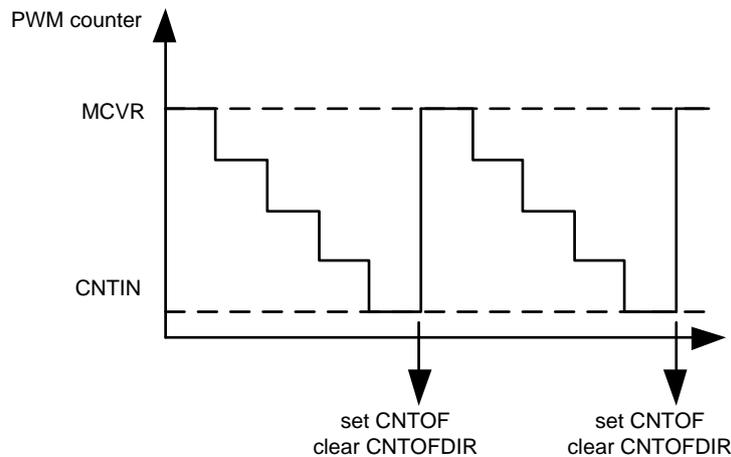


图 10-30 向下计数 PWM counter 溢出

10.4.11 写保护

在一些安全等级要求高的应用场景(如电机控制)，错误修改寄存器配置会引起系统异常，容易导致电机损坏。写保护功能可提供对某些关键寄存器位写入保护。用户在完成初始化配置后，可通过设置 `PWM_FDSR[WPEN] = 1` 激活写保护功能，避免人为或其它异常引起的对关键寄存器位误操作。设置 `PWM_FUNCSEL[WPDIS] = 1` 关闭写保护功能，可重新写入。支持写保护功能的位在 10.4.24 寄存器定义章节中进行说明。

10.4.12 初始化

向 `PWM_FUNCSEL[INIT]` 位写入 1 时，初始化将强制通道(n)输出为 `PWM_OUTINIT[CHnOIV]` 位的值。初始化特性只能在禁用 PWM 计数器时设置使用，当 PWM 计数器开始工作时 `PWM_FUNCSEL[INIT]` 位自动清除，初始化输出一直保持直至 PWM 通道开始接管控制。

10.4.13 极性控制

`CHnPOL` 位选择通道(n) 输出极性：

- `CHPOLCR[CHnPOL] = 0`，通道(n) 输出极性为高，逻辑 1 为有效状态，逻辑 0 为无效状态
- `CHPOLCR[CHnPOL] = 1`，通道(n) 输出极性为低，逻辑 0 为有效状态，逻辑 1 为无效状态

10.4.14 输出屏蔽

输出屏蔽可用于通过软件强制通道输出为各自的无效状态。如果 `CHnOMEN = 1`，则通道(n)输出强制为通道的无效状态 (`PWM_CHOPOLCR[CHnPOL]` 位值)。

10.4.15 软件输出控制

软件输出控制可在 PWM 输出过程中强制通道输出软件定义值。**CH(n)SWEN** 位使能软件输出控制，**CH(n)SWCV** 设置通道强制输出值。非组合模式，软件输出功能每个通道可单独设置，组合模式软件输出功能如下表。

表 10-2 组合模式软件输出控制行为

CH(n)SWEN	CH(n+1)SWEN	CH(n)SWCV	CH(n+1)SWCV	Channel(n) 输出	Channel(n+1) 输出
0	X ⁽¹⁾	X ⁽¹⁾	X ⁽¹⁾	关闭软件控制功能	关闭软件控制功能
1	X ⁽¹⁾	0	0	0	0
1	X ⁽¹⁾	0	1	0	1
1	X ⁽¹⁾	1	0	1	0
1	X ⁽¹⁾	1	1	1	0 或 1 ⁽²⁾

注(1): X 为 0 或 1 任意值。

注(2): PAIRnCOMPEN 位为 0 时，输出为 1；PAIRnCOMPEN 位为 1 时，输出为 0。

10.4.16 初始化触发器

如果 INITTRIGEN = 1，则在以下情况中 PWM 计数器更新为 PWM_CNTIN 寄存器值时，生成触发。

- PWM 计数器通过所选计数模式自动更新为 PWM_CNTIN 寄存器值
- 对 PWM_CNT 寄存器执行写操作
- PWM 计数器同步

10.4.17 通道匹配触发器

CHnTRIG = 1，在发生通道(n) 匹配(PWM 计数器 = CH(n)V)时，PWM 产生通道触发。通道触发输出提供用于片上模块的触发信号。

在向上-向下组合模式，通道值在向上计数产生一次匹配，向下计数也会产生一次匹配。为了便于控制通道匹配触发，可通过设置匹配生效点 PWM_CHnSCR[DIR]是在向上计数还是向下计数时作用。

10.4.18 故障控制

PWM 提供 3 个故障输入源：1 个来自芯片内部，2 个来自外部管脚输入。

FERnEN 位使能故障输入 n，FFnEN 位使能故障输入 n 滤波器。FFVAL 位选择已使能的每个故障输入滤波器的值。

如果故障控制和故障输入 n 已使能，且在故障输入 n 信号上检测到有效边沿，则表明已出现故障状况且 FAULTDFn 位已置位。FAULTDF 位是 FAULTDFn[2:0]位的逻辑或(OR)。

如果已使能故障控制(**FAULTMODE** [1:0] \neq 0:0)、使能故障输入且已检测到故障，则强制 PWM 通道输出为各自安全值：

- 通道(n)输出采用 CHnPOL 的值
- 通道(n+1)采用 CH(n+1)POL 的值

当(**FAULTDF** = 1)且(**FAULTIE** = 1)时，生成故障中断。

表 10-3 故障源编号表

故障输入编号	故障源	说明
FAULT0	ACMP0_OUT	内部故障输入
FAULT1	PWMx_FLT0	外部引脚故障输入
FAULT2	PWMx_FLT1	外部引脚故障输入

10.4.18.1 自动故障清除

如果选择自动故障清除模式 (**FAULTMODE**[1: 0]=1: 1)，则当故障输入信号(**FAULTIN**)为零且新的 PWM 周期开始时，被故障控制禁用的通道输出将恢复正常输出。

10.4.18.2 手动故障清除

如果选择手动故障清除 (**FAULTMODE** [1: 0]=0: 1 或 1: 0)，则当 **FAULTDF** 位被清除且新的 PWM 周期开始时，被故障控制禁用的通道输出将恢复正常输出。

10.4.18.3 故障输入极性

FLTnPOL 位选择故障输入 n 的极性：

- **FLTnPOL** = 0，故障 n 输入极性为高，故障输入 n 处的逻辑 1 代表一个故障
- **FLTnPOL** = 1，故障 n 输入极性为低，故障输入 n 处的逻辑 0 代表一个故障

10.4.19 写缓冲更新的寄存器

10.4.19.1 PWM_CNTIN 寄存器更新缓存

表 10-4 PWM_CNTIN 寄存器更新缓存

条件	寄存器更新时刻
CLKSRC = 0	往 PWM_CNTIN 寄存器采取写入操作时
CLKSRC \neq 0 & PWMSYNCEN = 0	往 PWM_CNTIN 采取写入操作之后的下一个系统时钟周期
CLKSRC \neq 0 & PWMSYNCEN = 1	参考 PWM_CNTIN 寄存器同步 章节

10.4.19.2 PWM_CH(n)V 寄存器更新缓存

表 10-5 PWM_CH(n)V 寄存器更新缓存

条件	寄存器更新时刻
CLKSRC = 0	往 PWM_CH(n)V 寄存器采取写入操作时
CLKSRC ≠ 0 & PWMSYNCEN = 0	<ul style="list-style-type: none"> • 如果选择的模式为 EPWM，则 PWM 计数器从 MCVR 更改为 CNTIN 之后更新。 • 如果选择的模式为 CPWM，则 PWM 计数器从 MCVR 更改为(MCVR - 0x0001)之后更新。
CLKSRC ≠ 0 & PWMSYNCEN = 1	参考 10.4.20.7 PWM_CH(n)V 和 PWM_CH(n+1)V 寄存器同步 章节

10.4.19.3 PWM_MCVR 寄存器更新缓存

表 10-6 PWM_MCVR 寄存器更新缓存

条件	寄存器更新时刻
CLKSRC = 0	往 PWM_MCVR 寄存器采取写入操作时
CLKSRC ≠ 0 & PWMSYNCEN = 0	<ul style="list-style-type: none"> • 如果选择的模式为 EPWM，则 PWM 计数器从 MCVR 更改为 CNTIN 之后更新。 • 如果选择的模式为 CPWM，则 PWM 计数器从 MCVR 更改为(MCVR - 0x0001)之后更新。
CLKSRC ≠ 0 & PWMSYNCEN = 1	参考 10.4.20.4 PWM_MCVR 寄存器同步 章节

10.4.20 PWM 同步

PWM 同步功能提供一个机制可将 PWM_MCVR, PWM_CNTIN, PWM_CHnV, PWM_OMCR, PWM_INVCR 和 PWM_CHOSWCR 寄存器对应的缓存值更新到寄存器，并设置 PWM 计数器为 PWM_CNTIN 寄存器值。

10.4.20.1 硬件触发器

当 TRIGn = 1 时，使能 PWM 模块的三个硬件触发信号输入，其中 n (0,1 或 2) 分别对应于每个输入信号。硬件触发器输入 n 由总线时钟同步。

- 硬件触发器 0: ACMP_OUT 信号作为 PWM 同步触发源 0;
- 硬件触发器 1: CTU 模块的 PWMTRIG 控制位作为 PWM 同步硬件触发源 1;
- 硬件触发器 2: PWM 通道 0 输出作为同步的硬件触发源 2; (其中 PWM0 & PWM2 模块的触发源为 PWM1_CH0_OUT 信号, PWM1 模块的触发源为 PWM0_CH0_OUT 信号)

如果 HWTRIGMODESEL = 0, 在使能的硬件触发输入处检测到上升沿时, 将启动与硬件触发的 PWM 同步。当写入 0 或检测到触发 n 事件时, TRIGn 位被清零。

在这种情况下，如果启用了两个或更多硬件触发器（例如，TRIG0 和 TRIG1 = 1），并且仅发生触发 1 事件，则仅清零 TRIG1 位。如果触发事件与写设置 TRIGn 位一起发生，则启动同步，但由于写操作，TRIGn 位会保持置 1。

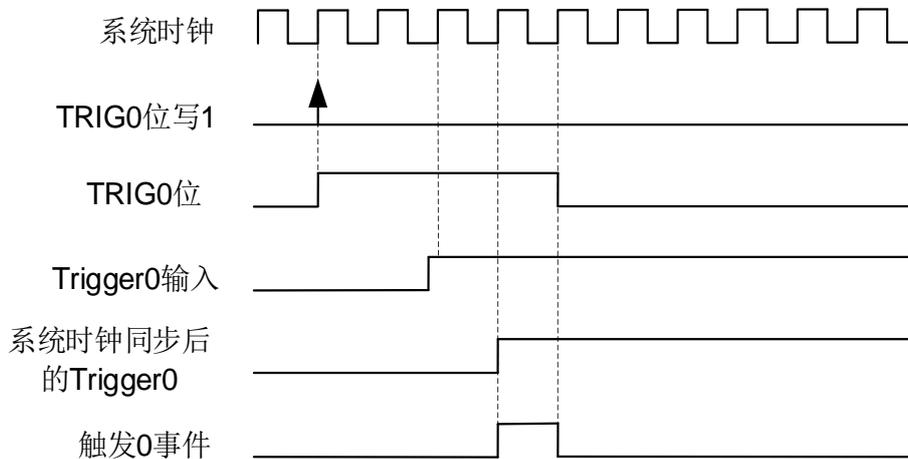


图 10-31 HWTRIGMODESEL = 0 硬件触发事件

10.4.20.2 软件触发器

当 PWM_SYNC[SWSYNC]位写入 1 时，会发生软件触发事件。当 SWSYNC 位写入 0 时，或当由软件事件启动的 PWM 同步完成时，该位被清零。

如果同时发生另一个软件触发事件（通过将另一个 1 写入 SWSYNC 位），则由先前软件触发事件启动的 PWM 同步结束，新的 PWM 同步开始，SWSYNC 位保持为 1。

SYNCMODE = 1，则根据 CNTVSWSYNC 位，SWSYNC 位也会被 PWM 清零。如果 CNTVSWSYNC=0，则软件触发事件发生后，在下一个选定的加载点清除 SWSYNC 位。如果 CNTVSWSYNC = 1，然后在软件触发事件发生时清除 SWSYNC 位。

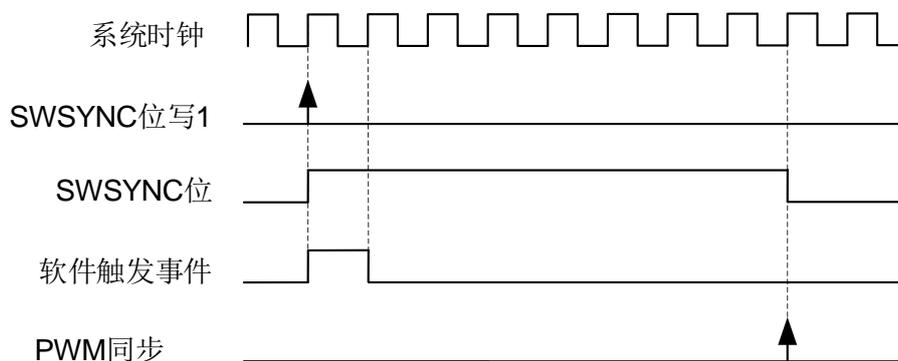


图 10-32 软件触发事件

10.4.20.3 边界周期和加载点

向上计数模式下，边界周期定义为计数器变为其初始值(CNTIN)的时候。向上-向下计数模式下，边界周期则定义为计数器从向下计数变为向上计数的时候以及从向上计数变为向下计数的时候。下图显示了寄存器的边界周期和加载点：

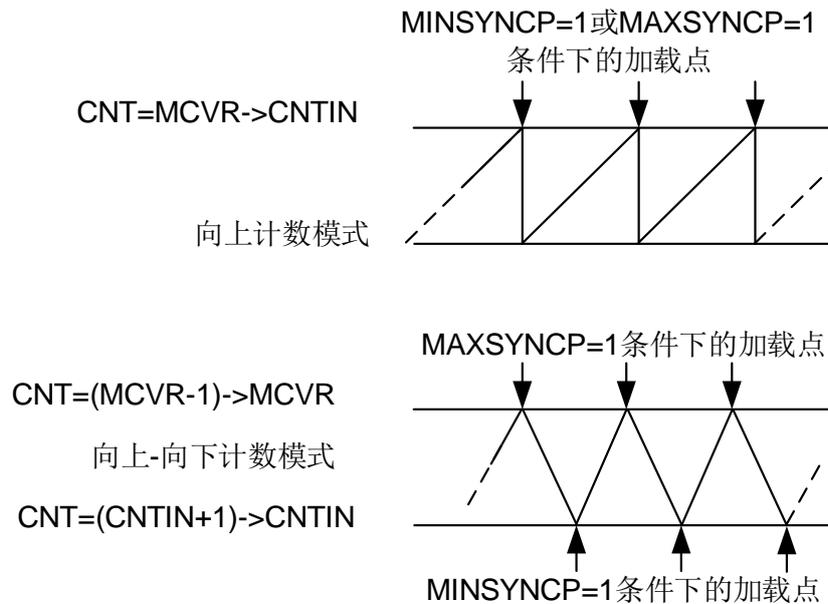


图 10-33 边界周期与加载点

向上计数模式中，MINSYNCP 或 MAXSYNCP 其中一位为 1，则使能加载点。向上-向下计数模式下，由 MINSYNCP 和 MAXSYNCP 位选择加载点。在这两种计数模式中，如果 MINSYNCP 和 MAXSYNCP 都不是 1，则边界周期不用作寄存器更新的加载点，即使有触发信号也不会产生寄存器同步（CNTVSWSYNC=0 条件下）。有关详细信息，请参见以下各节中的寄存器同步说明。

10.4.20.4 PWM_MCVR 寄存器同步

PWMSYNCEN = 1 使能 PWM_MCVR 寄存器同步功能，同步时将其缓存值更新至 PWM_MCVR 寄存器。

MCVR 寄存器同步流程图如下：

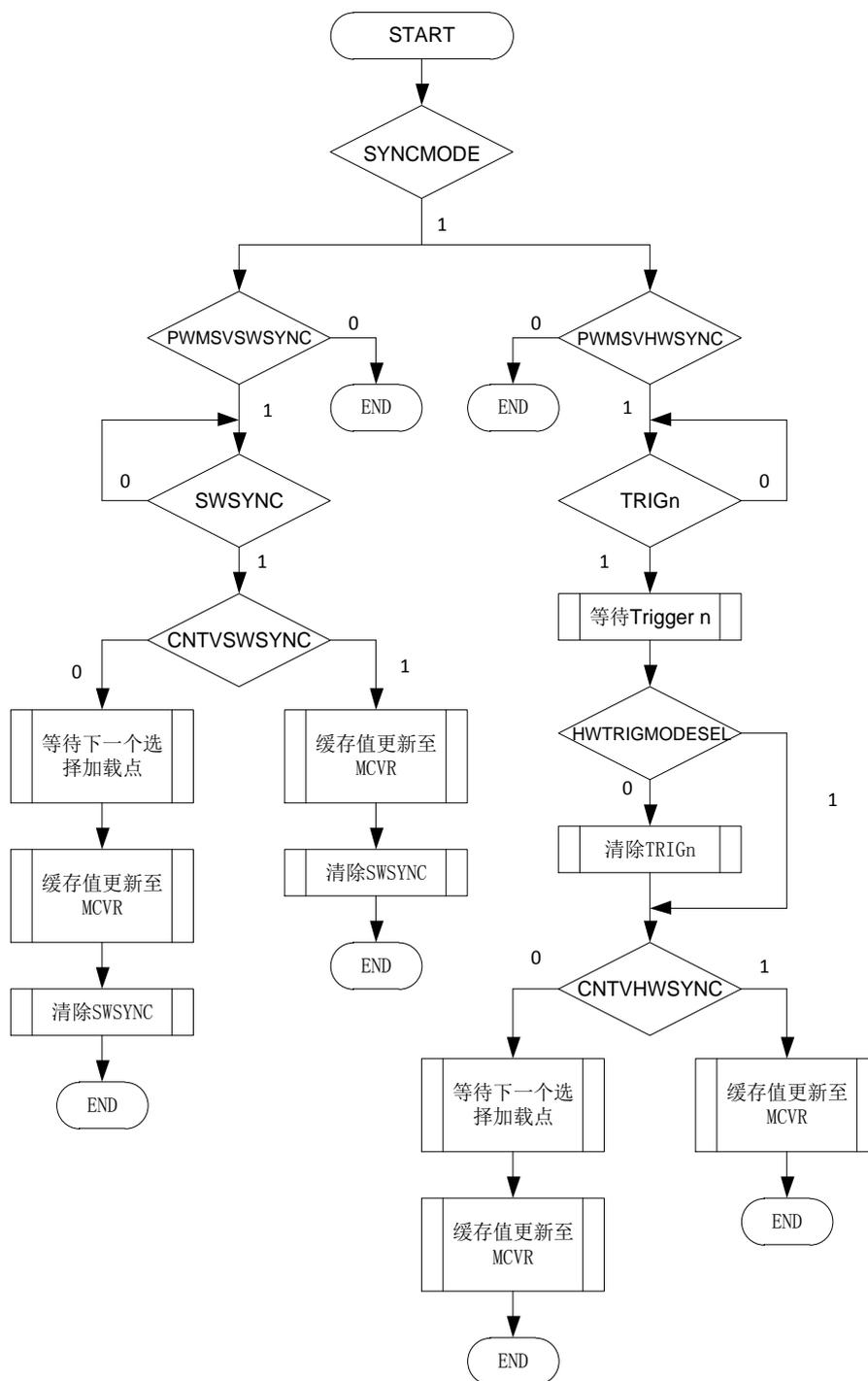


图 10-34 PWM_MCVR 寄存器同步流程

10.4.20.5 PWM_CNT 寄存器同步

PWM_CNT 寄存器同步功能可以在 PWM 周期中的特定点重新开始生成 PWM。通道输出强制为各自的初始值，PWM_CNT 寄存器强制为 PWM_CNTIN 寄存器定义的初始计数值。PWM_CNT 寄存器同步流程图如下：

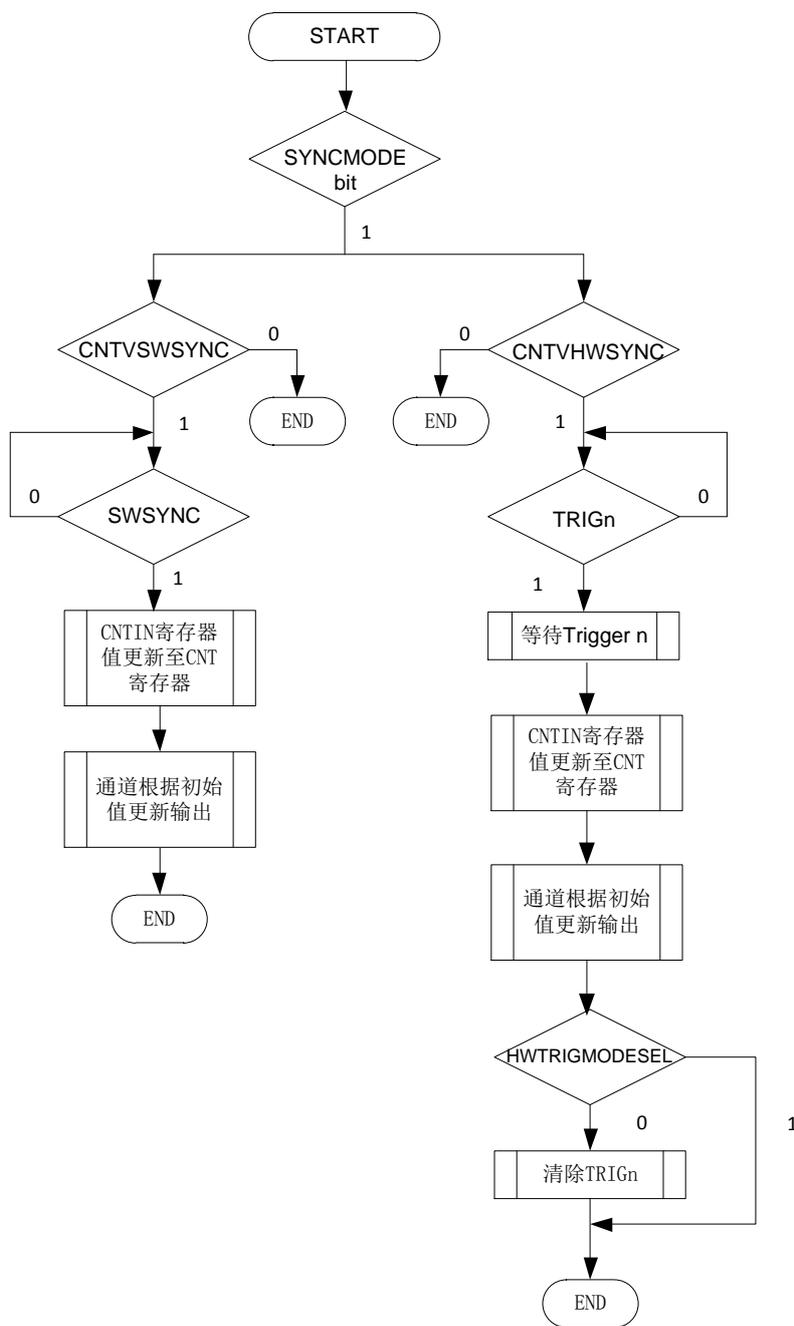


图 10-35 PWM_CNT 寄存器同步流程

10.4.20.6 PWM_CNTIN 寄存器同步

PWMSYNCEN = 1、SYNCMODE = 1 且 CNTINC = 1，将使能这种同步。同步机制与 PWM_MCVR 寄存器同步相同，参照图 10-34 PWM_MCVR 寄存器同步流程。

10.4.20.7 PWM_CH(n)V 和 PWM_CH(n+1)V 寄存器同步

PWMSYNCEN = 1 且 PAIR(n)SYNCEN = 1, 则会启用该同步。同步机制与 PWM_MCVR 寄存器同步流程相同, 参照图 10-34 PWM_MCVR 寄存器同步流程。

10.4.20.8 PWM_OMCR 寄存器同步

PWM_OMCR 寄存器同步将其缓存值更新至 PWM_OMCR 寄存器。参照以下流程图:

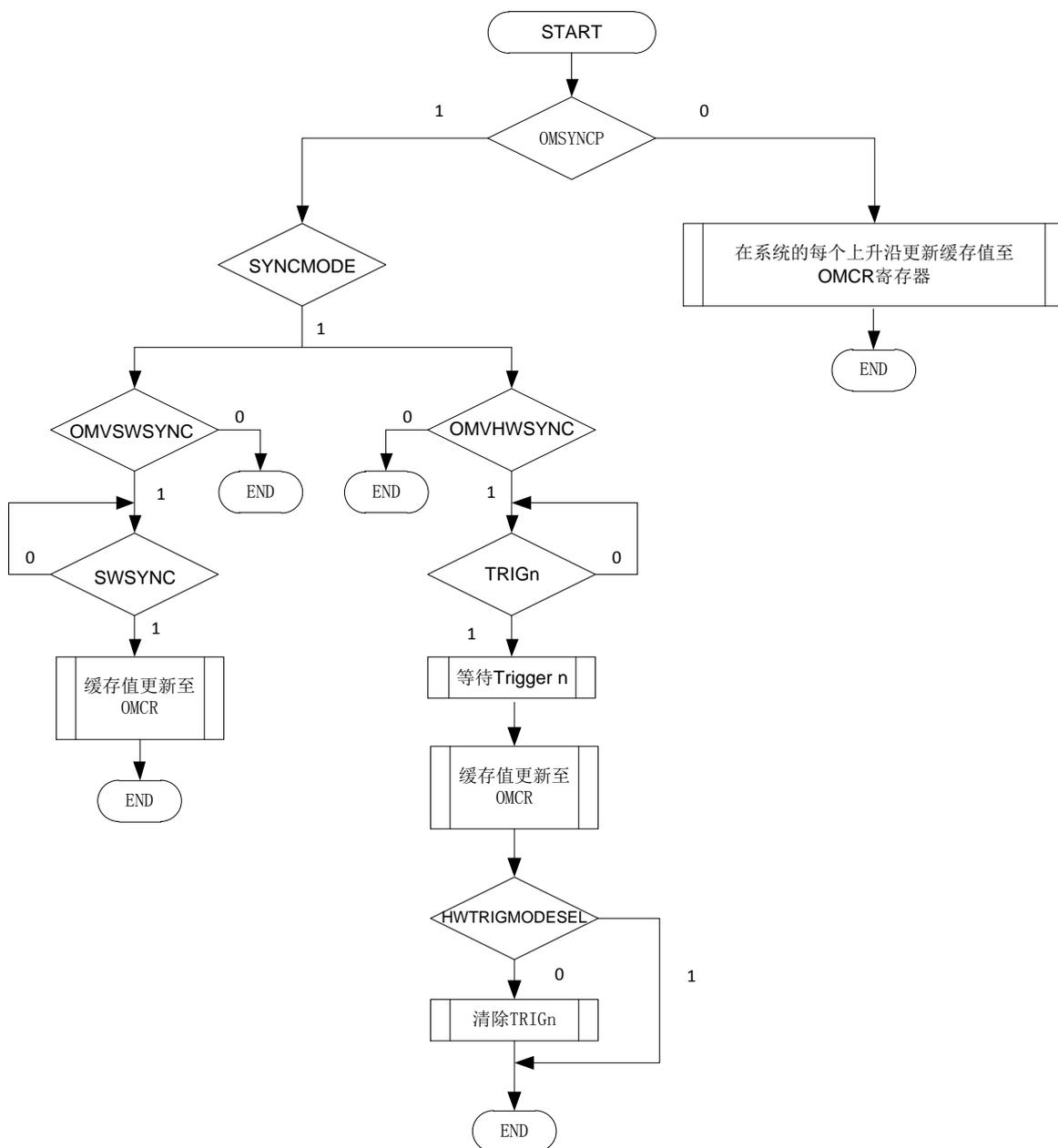


图 10-36 PWM_OMCR 寄存器同步流程

10.4.20.9 PWM_INVCR 寄存器同步

PWM_INVCR 寄存器同步将其缓存值更新至 PWM_INVCR 寄存器。参照以下流程图：

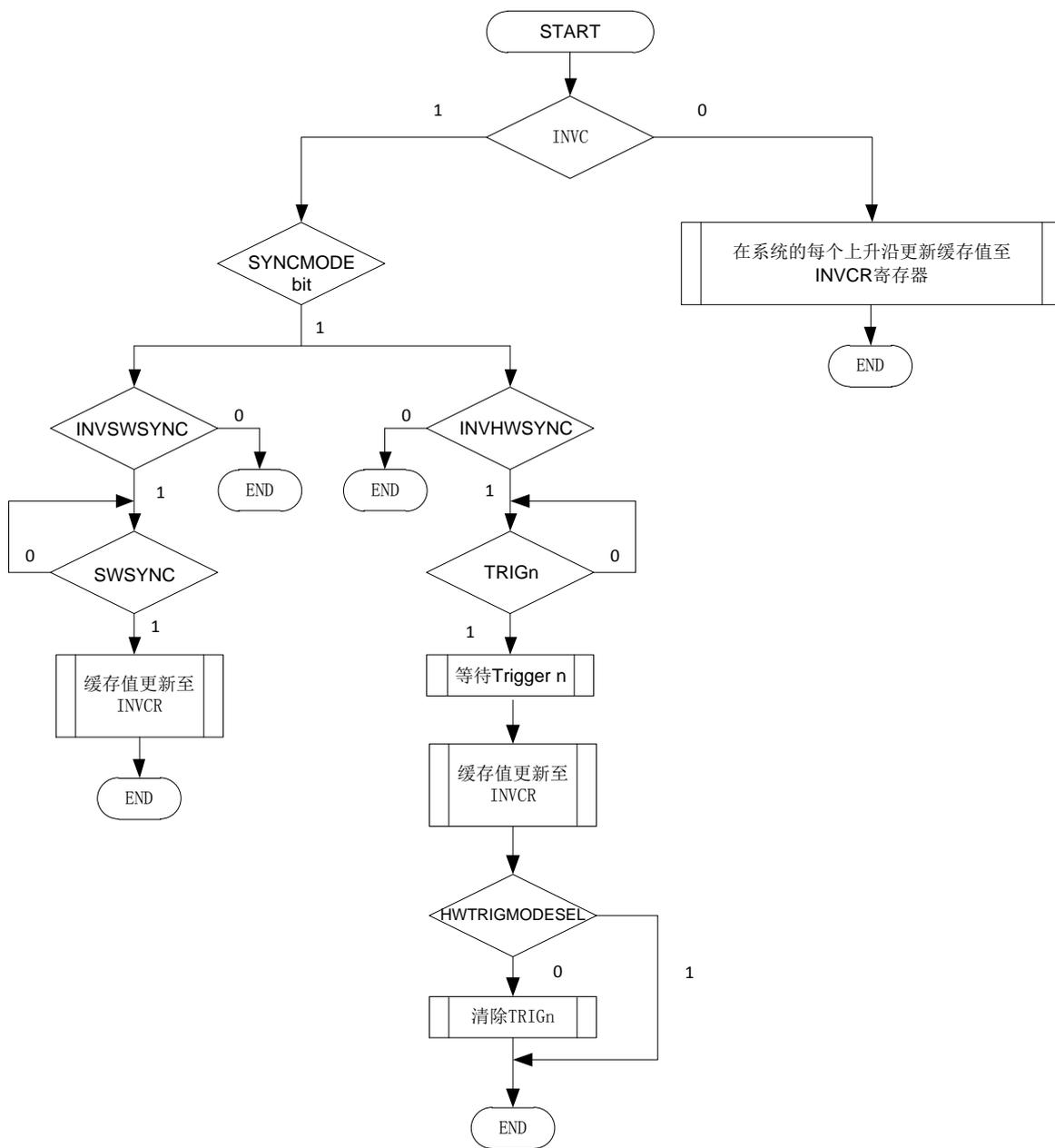


图 10-37 PWM_INVCR 寄存器同步流程

10.4.20.10 PWM_CHOSWCR 寄存器同步

PWM_CHOSWCR 寄存器同步将其缓存值更新至 PWM_CHOSWCR 寄存器。参照以下流程图：

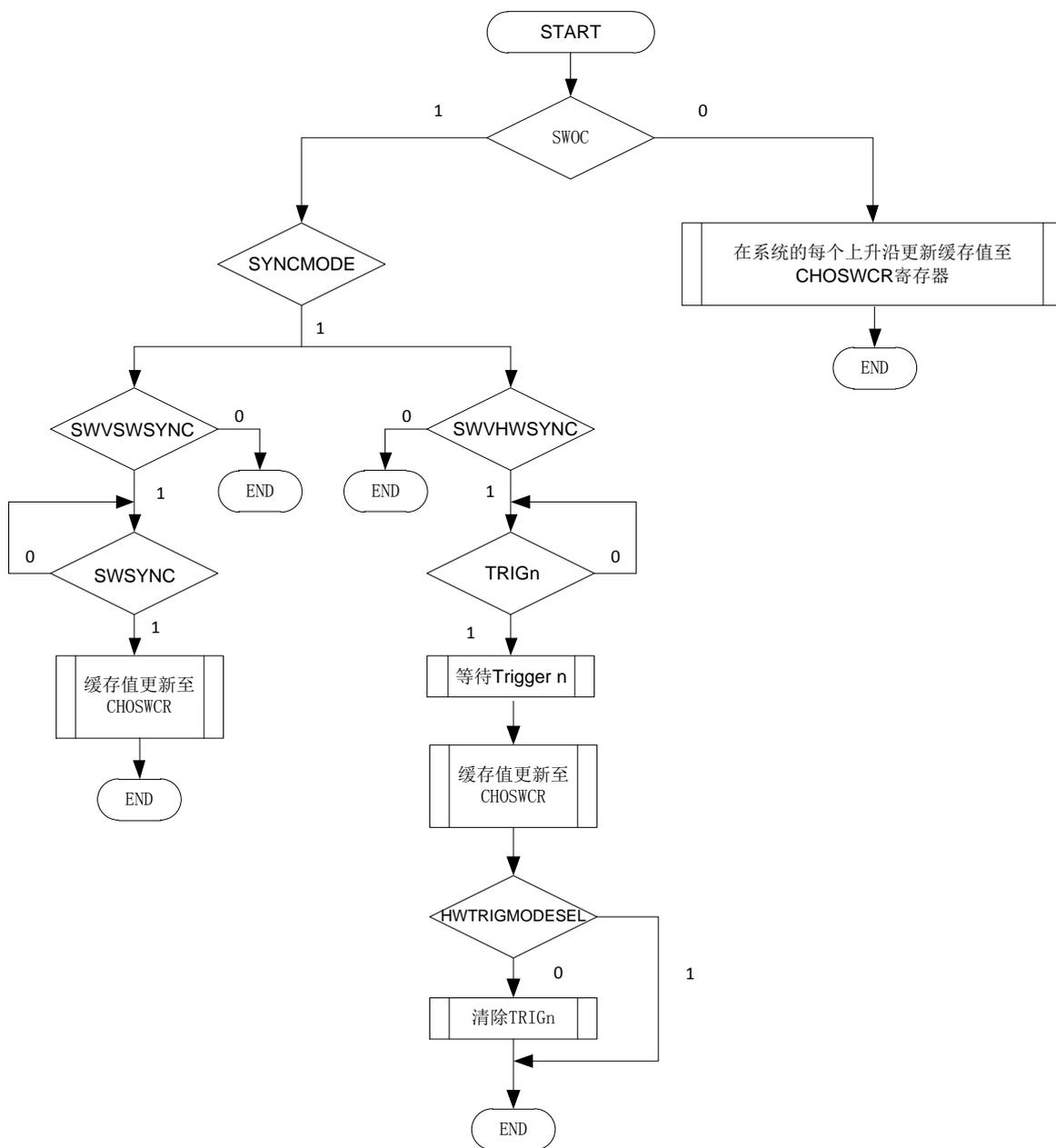


图 10-38 PWM_CHOSWCR 寄存器同步流程

10.4.20.11 PWM_CHOPOLCR 寄存器同步

PWM_CHOPOLCR 寄存器同步将其缓存值更新至 PWM_CHOPOLCR 寄存器。参照以下流程图：

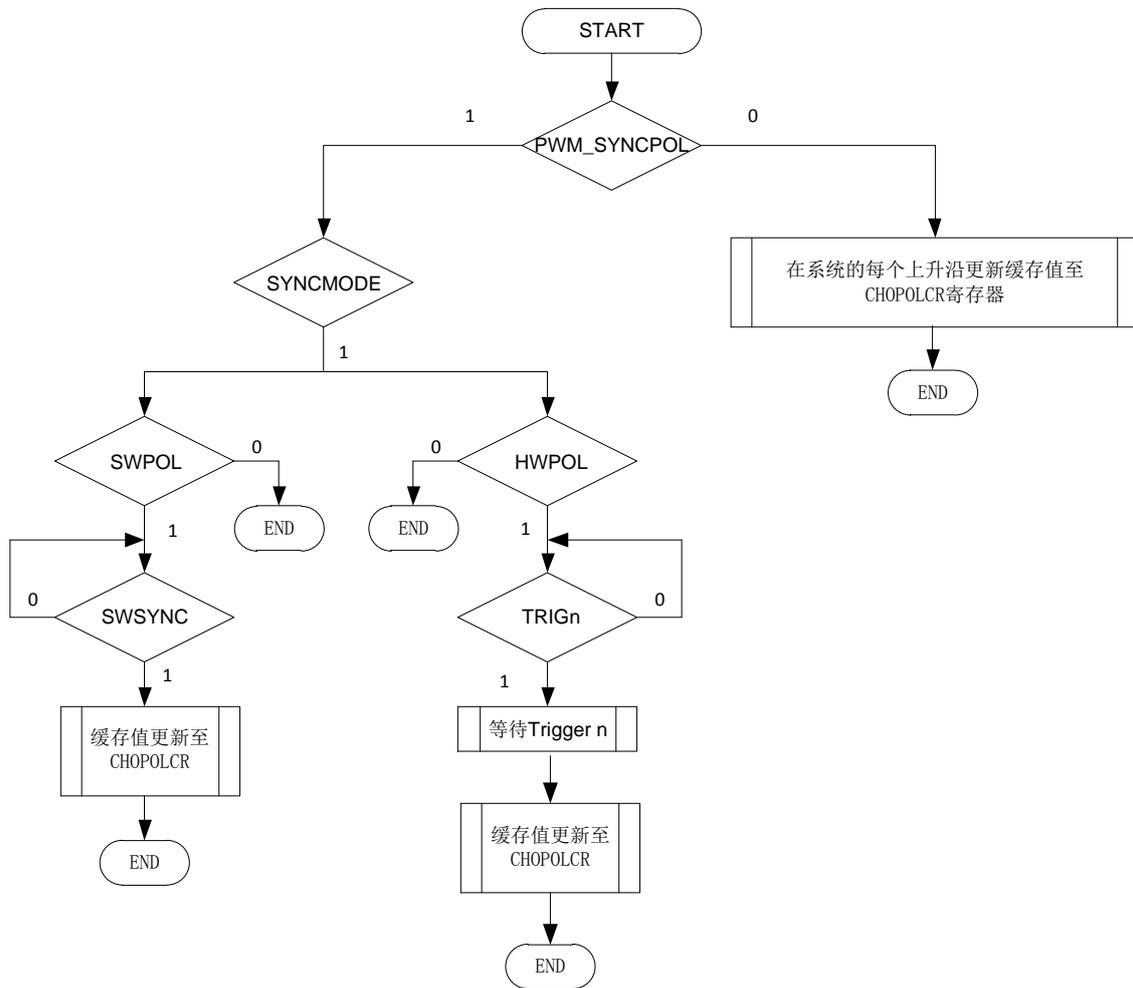


图 10-39 PWM_CHOPOLCR 寄存器同步流程

10.4.21 特性优先级

下图展示了生成通道(n)和(n+1)输出信号时所用特性的优先级。

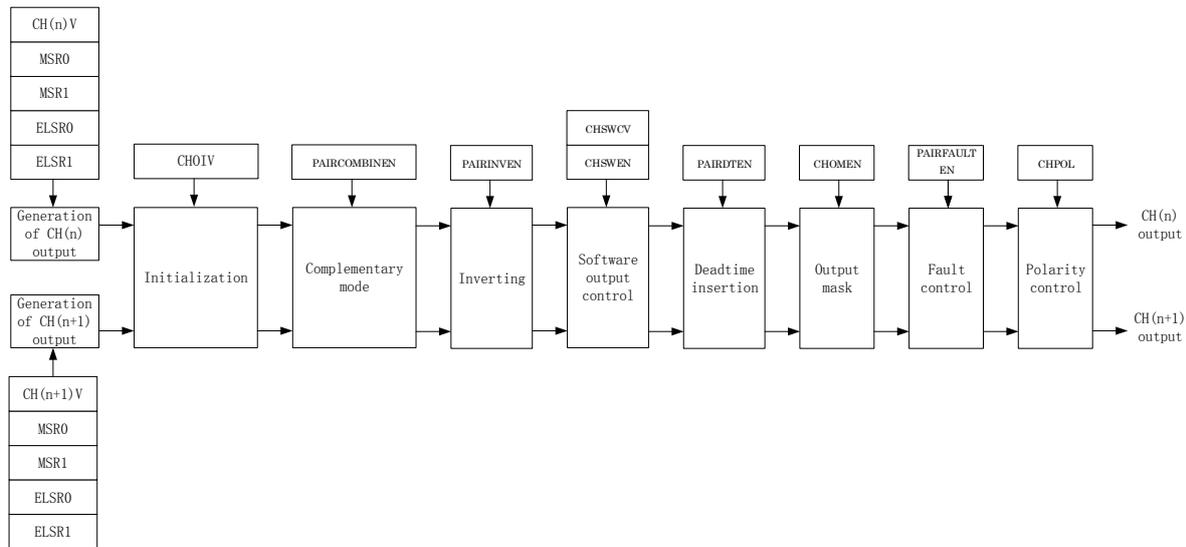


图 10-40 特性优先级

10.4.22 全局时基

全局时基(GTB)允许在同一时刻启动芯片上的多个 PWM 模块同时工作，达到多个 PWM 模块同步开始计数的目的。

GTB 功能可通过 `PWM_CONF` 寄存器中的 `GTBEEN` 和 `GTBEOUT` 位实现。

要使能 GTB 特性，请对每个参与的 PWM 模块执行以下步骤：

1. 复位 PWM 模块；
2. 配置 PWM 模块，但不开启 PWM 时钟，即向 `PWM_INIT[CLKSRC]` 写入 00b；
3. 向 `PWM_CONF [GTBEEN]` 写入 1，同时向 `PWM_CONF [GTBEOUT]` 写入 0；
4. 在 `PWM_INIT [CLKSRC]` 中选择 PWM 计数器时钟源；
5. 向 `PWM_CNT` 寄存器中写入任意值复位 PWM 计数器；
6. `PWM_CONF [GTBEOUT]` 写入 1，启动多个 PWM 模块同时工作。

10.4.23 PWM 中断

- 当 `CNTOIE=1` 且 `CNTOF=1` 时，产生计数溢出中断。
- 当 `CHnIE=1` 且 `CHnIF = 1` 时，产生通道(n) 中断
- 当 `FAULTIE =1` 且 `FAULTDF = 1` 时，产生故障中断

10.4.24 低功耗模式

表 10-7 PWM 模块低功耗模式

模式	唤醒源	备注
休眠模式	中断	模块正常工作，可通过中断唤醒 MCU
停止模式	----	模块被关闭

10.5 寄存器定义

表 10-8 PWM 寄存器映射

PWM0 基地址 = 0x40013000

PWM1 基地址 = 0x40014000

PWM2 基地址 = 0x40015000

地址	名称	宽度	描述
PWMx 基地址+0x00	PWM_INIT	32	初始化寄存器
PWMx 基地址+0x04	PWM_CNT	32	计数器寄存器
PWMx 基地址+0x08	PWM_MCVR	32	最大计数值寄存器
PWMx 基地址+0x0C	PWM_CH0SCR	32	Channel (0) 状态和控制寄存器
PWMx 基地址+0x10	PWM_CH0V	32	Channel (0)值
PWMx 基地址+0x14	PWM_CH1SCR	32	Channel (1) 状态和控制寄存器
PWMx 基地址+0x18	PWM_CH1V	32	Channel (1) 值
PWMx 基地址+0x1C	PWM_CH2SCR	32	Channel (2) 状态和控制寄存器
PWMx 基地址+0x20	PWM_CH2V	32	Channel (2) 值
PWMx 基地址+0x24	PWM_CH3SCR	32	Channel (3) 状态和控制寄存器
PWMx 基地址+0x28	PWM_CH3V	32	Channel (3) 值
PWMx 基地址+0x4C	PWM_CNTIN	32	计数器初始值寄存器
PWMx 基地址+0x50	PWM_STR	32	捕获和比较状态寄存器
PWMx 基地址+0x54	PWM_FUNCSEL	32	功能选择寄存器
PWMx 基地址+0x58	PWM_SYNC	32	同步寄存器
PWMx 基地址+0x5C	PWM_OUTINIT	32	通道输出的初始状态寄存器
PWMx 基地址+0x60	PWM_OMCR	32	输出屏蔽控制寄存器
PWMx 基地址+0x64	PWM_MODESEL	32	模式选择寄存器
PWMx 基地址+0x68	PWM_DTSET	32	死区设置寄存器
PWMx 基地址+0x6C	PWM_EXTTRIG	32	外部触发器
PWMx 基地址+0x70	PWM_CHOPOLCR	32	通道输出极性控制寄存器
PWMx 基地址+0x74	PWM_FDSR	32	故障检测状态寄存器
PWMx 基地址+0x78	PWM_CAPFILTER	32	输入捕获滤波器控制
PWMx 基地址+0x7C	PWM_FFAFER	32	故障滤波和使能寄存器
PWMx 基地址+0x80	PWM_QDI	32	正交解码控制和状态寄存器
PWMx 基地址+0x84	PWM_CONF	32	配置寄存器
PWMx 基地址+0x88	PWM_FLTPOL	32	故障输入极性寄存器

地址	名称	宽度	描述
PWM _x 基地址+0x8C	PWM_SYNCONF	32	同步配置寄存器
PWM _x 基地址+0x90	PWM_INVCR	32	反相控制寄存器
PWM _x 基地址+0x94	PWM_CHOSWCR	32	通道软件输出控制寄存器

【说明】上表中，x=0，1，2。

10.5.1 初始化寄存器(PWM_INIT)

表 10-9 PWM_INIT 寄存器

PWM_INIT								PWM 初始化寄存器								Reset:00000000							
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
名称									CLKPSC[15: 8]														
访问									RW														
Reset									0														
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
名称	CLKPSC[7: 0]								CN TOF	CN TOIE	CN TM ODE		CLK SRC										
访问	RW								R/W OC	RW	RW		RW										
Reset	0								0				0										

字段	说明
23: 8 CLKPSC	<p>PWM CLK 预分频器</p> <p>时钟分频 = CLKPSC + 1。</p> <p>在将新值更新为寄存器位后，新的预分频因子会影响下一个总线时钟周期的时钟源。该字段为写保护。它只能在 FUNCSEL[WPDIS] = 1 时写入。</p>
7 CNTOF	<p>定时器溢出标志</p> <p>0： PWM 计数器未溢出 1： PWM 计数器溢出</p> <p>当 PWM 计数器值达到 MCVR 寄存器中值时置位。当 CNTOF 置 1 时，通过读取 INIT 寄存器，然后将 0 写入 CNTOF 位可清除定时器溢出标志。如果在读和写操作之间发生另一次 PWM 溢出，则清除操作无效，CNTOF 仍保持置位。</p>
6 CNTOIE	<p>定时器溢出中断使能</p> <p>0： 禁用 CNTOF 中断，使用软件轮询 1： 使能 CNTOF 中断，当 CNTOF 置位时，产生中断</p> <p>使能 PWM 溢出中断</p>

字段	说明
5 CNTMODE	<p>计数器工作模式</p> <p>0：PWM 计数器以向上计数模式工作 1：PWM 计数器以向上-向下计数模式工作</p> <p>选择计数器工作模式。该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时写入。</p>
3 CLKSRC	<p>时钟源选择</p> <p>0：没有选择任何时钟。禁用 PWM 计数器 1：总线时钟</p> <p>选择 PWM 计数器时钟源。该字段为写保护，它仅在 FUNCSEL[WPDIS] = 1 时可写。</p>

10.5.2 计数器寄存器(PWM_CNT)

表 10-10 PWM_CNT 寄存器

PWM_CNT		PWM 计数器值																Reset:00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称	COUNT																	
访问	RW																	
Reset	0																	

字段	说明
15: 0 COUNT	<p>PWM 计数器的值</p> <p>CNT 寄存器包含 PWM 计数器值。Reset 清除 CNT 寄存器。将任何值写入 COUNT 都会使用其初始值 CNTIN 更新计数器。</p>

10.5.3 最大计数值寄存器(PWM_MCVR)

表 10-11 PWM_MCVR 寄存器

PWM_MCVR		最大计数值寄存器										Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	MCVR															
访问	RW															
Reset	0															

字段	说明
	最大计数值寄存器
15: 0 MCVR	MCVR 寄存器包含 PWM 计数器的模数值。当 PWM 计数值达到 MCVR 值后，溢出标志 (CNTOF) 在下一个时钟置起，计数器的下一个值 取决于所选的计数方法。写入 MCVR 寄存器会将值锁存到缓冲区中。根据从写缓冲区更新的寄存器，MCVR 寄存器使用其写缓冲区的值进行更新。在写入 MCVR 寄存器之前，通过写入 CNT 来初始化 PWM 计数器。

10.5.4 通道状态和控制寄存器(PWM_CHnSCR)

表 10-12 PWM_CHnSCR 寄存器

PWM_CHnSCR		Channel (n) 状态和控制寄存器										Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									CHIF	CHIE	MSR1	MSR0	ELSR1	ELSR0	DIR	
访问									WOC	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	

字段	说明
7 CHIF	通道中断标志 在通道上发生事件时由硬件置位。通过读取 CHSCR 寄存器，然后将 CHIF 位写 0 来清除 CHIF 位。向 CHIF 中写 1 不起作用。如果在读取和写入操作间发生另一个事件，则清除操作无效，CHIF 仍保持置位，此时 CHIF 的中断请求也不会因为写清除操作而丢失。
0:	没有发生通道事件

字段	说明
	1: 通道事件已经发生
6 CHIE	通道中断使能 0: 禁用通道中断 1: 使能通道中断 使能通道中断
5 MSR1	通道模式选择寄存器 1 用于通道逻辑的进一步选择, 其功能取决于通道模式。该字段为写保护。它只能在 FUNCSEL[WPDIS] = 1 时可写入。
4 MSR0	通道模式选择寄存器 0 用于进一步选择通道逻辑, 其功能取决于通道模式。该字段为写保护。它只能在 FUNCSEL[WPDIS] = 1 时可写入。
3 ELSR1	边沿或 电平选择寄存器 1 ELSR1 和 ELSR0 的功能依赖于通道模式。该字段写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写入。
2 ELSR0	边沿或 电平选择寄存器 0 ELSR1 和 ELSR0 的功能依赖于通道模式。该字段写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写入。
1 DIR	匹配点方向 0: 在向下计数过程中匹配点生效 1: 在向上计数过程中匹配点生效 仅用于向上-向下计数组合模式, 用于选择匹配生效点方向。该字段写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写入。

10.5.5 通道值寄存器(PWM_CHnV)

表 10-13 PWM_CHnV 寄存器

PWM_CHnV	Channel (n) 值																Reset:00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	CHCVAl																
访问	RW																
Reset	0																

字段	说明
15: 0 CHCVAL	通道计数值 这些寄存器包含输入模式的捕获 PWM 计数器值或输出模式的匹配值。在输入捕获、捕获测试和双边沿捕获模式下，忽略对 CHnV 寄存器的任何写入操作。在输出模式下，写入 CHnV 寄存器会将值锁存在缓冲区中。根据从写缓冲区更新的寄存器，使用其写缓冲区的值更新 CHnV 寄存器。

10.5.6 计数器初始值寄存器(PWM_CNTIN)

表 10-14 PWM_CNTIN 寄存器

PWM_CNTIN		计数器初始值																Reset:00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称		CNTINIT																
访问		RW																
Reset		0																

字段	说明
15: 0 CNTINIT	计数器初始值 计数器初始值寄存器包含 PWM 计数器的初始值。写入 CNTIN 寄存器会将值锁存至缓冲区中。根据从写缓冲区更新的寄存器，使用其写缓冲区的值更新 CNTIN 寄存器。当最初选择 PWM 时钟时，通过向时钟源选择位 CLKSRC 写入非 0 值，PWM 计数器以值 0x0000 开始。为避免这种行为，在第一次写入选择 PWM 时钟前，将新值写入 CNTIN 寄存器，然后通过向 CNT 寄存器中写入任意值，来初始化 PWM 计数器。

10.5.7 捕获和比较状态寄存器(PWM_STR)

表 10-15 PWM_STR 寄存器

PWM_STR																捕获和比较状态寄存器				Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
名称																								
访问																								
Reset																								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
名称													CH3SF	CH2SF	CH1SF	CH0SF								
访问													R/W0	R/W0	R/W0	R/W0								
Reset													C	C	C	C								
Reset													0	0	0	0								

字段	说明
3 CH3SF	通道 3 状态标志 0：没有发生通道事件 1：已发生通道事件
2 CH2SF	通道 2 状态标志 0：没有发生通道事件 1：已发生通道事件
1 CH1SF	通道 1 状态标志 0：没有发生通道事件 1：已发生通道事件
0 CH0SF	通道 0 状态标志 0：没有发生通道事件 1：已发生通道事件

10.5.8 功能选择寄存器(PWM_FUNCSEL)

表 10-16 PWM_FUNCSEL 寄存器

PWM_FUNCSEL																功能选择				Reset:00000004				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16								
名称																								
访问																								
Reset																								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

PWM_FUNCSEL

功能选择

Reset:00000004

名称		FA UL TI E	FAULTM ODE		PW MS YN C	WP DI S	INI T	PWM SYNC EN
访问		RW	RW		RW	RW	RW	RW
Reset		0	0		0	1	0	0

字段	说明
7 FAULTIE	<p>错误中断使能</p> <p>0: 禁用故障控制中断 1: 使能故障控制中断</p> <p>当 PWM 检测到故障, 并使能 PWM 故障控制时, 允许产生中断。</p>
6: 5 FAULTMODE	<p>故障控制模式</p> <p>00: 所有通道禁用故障控制。 01: 仅对偶数通道使能故障控制 (通道: 0, 2), 所选模式为手动故障清除 10: 所有通道均支持故障控制, 所选模式为手动故障清除 11: 所有通道均支持故障控制, 所选模式为自动故障清除</p> <p>定义 PWM 故障控制模式。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
3 PWMSYNC	<p>PWM 同步模式</p> <p>0: 没有限制, MCVR, CHnV, OMCR 和 PWM 计数器同步可以使用软件和硬件触发器 1: 软件触发器只能由 MCVR 和 CHnV 同步使用, 硬件触发只能由 OMCR 和 PWM 计时器同步使用。</p> <p>选择 MCVR, CHnV, OMCR 和 PWM 计数器同步可以使用的触发器。</p>
2 WPDIS	<p>禁用写保护</p> <p>0: 使能写保护 1: 禁用写保护</p> <p>只允许写 1, 写 0 无效。当写保护使能时, 具有写保护功能的位不能被写入。当写保护禁用时, 具有写保护功能的位可以写入。WPDIS 位是 WPEN 位取反。当 WPEN 位写入 1 时, 可清除 WPDIS, 即使能写保护。当 WPDIS 位写入 1 时, 可清除 WPEN, 即禁用写保护。</p>
1 INIT	<p>初始化通道输出</p> <p>当 INIT 位写入 1 时, 根据 OUTINIT 寄存器中相应位的状态, 初始化通道输出。将 INIT 位置 0 不起作用。INIT 位始终读为 0。</p>
0 PWMSYNCEN	<p>PWM 同步功能使能</p> <p>0: 禁用寄存器同步功能</p>

字段	说明
	1: 使能寄存器同步功能
	该字段写保护，它仅能在 FUNCSEL[WPDIS] = 1 时可写。

10.5.9 同步寄存器(PWM_SYNC)

表 10-17 PWM_SYNC 寄存器

PWM_SYNC		同步寄存器										Reset:00000000					
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称					PW M_ SY NC PO L				SW SY NC	TRI G2	TRI G1	TRI G0	OM SY NC P			MA XS YN CP	MIN SY NC P
访问					RW				RW	RW	RW	RW	RW			RW	RW
Reset					0				0	0	0	0	0			0	0

字段	说明
11 PWM_SYNCPOL	<p>PWM_SYNCPOL</p> <p>0: CHOPOLCR 寄存器在总线时钟的所有上升沿使用其缓冲区的值进行更新 1: CHOPOLCR 寄存器仅通过 PWM 同步单元更新其缓冲区的值</p> <p>选择何时使用缓冲区的值更新 CHOPOLCR 寄存器。</p>
7 SWSYNC	<p>PWM 同步软件触发器</p> <p>0: 未选择软件触发 1: 选择软件触发</p> <p>选择软件触发器作为 PWM 同步触发器。将 1 写入 SWSYNC 位时，会发生软件触发。</p>
6 TRIG2	<p>PWM 同步硬件触发器 2</p> <p>0: 触发禁用 1: 触发使能</p> <p>PWM 通道 0 输出作为 PWM 同步硬件触发源，输入信号检测到上升沿时，产生硬件触发。</p>
5 TRIG1	<p>PWM 同步硬件触发器 1</p>

字段	说明
	0: 触发禁用 1: 触发使能 CTU 模块的 PWMTRIG 控制位作为 PWM 同步硬件触发源，输入信号中检测到上升沿时，产生硬件触发。
4 TRIG0	PWM 同步硬件触发器 0 0: 触发禁用 1: 触发使能 ACMP0 输出作为同步的硬件触发源，输入信号中检测到上升沿时，产生硬件触发。
3 OMSYNCP	输出掩码同步 0: OMCR 寄存器在总线时钟的所有上升沿使用其缓冲区的值进行更新 1: OMCR 寄存器仅通过 PWM 同步单元更新其缓冲区的值 选择何时使用缓冲区的值更新 OMCR 寄存器。
1 MAXSYNCP	使能最大加载点 0: 禁用最大加载点 1: 使能最大加载点 选择 PWM 同步的最大加载点。如果 MAXSYNCP 为 1，则所选加载点为 PWM 计数器达到其最大值（MCVR 寄存器）时。
0 MINSYNCP	使能最小加载点 0: 禁用最小加载点 1: 使能最小加载点 选择 PWM 同步的最小加载点。如果 MINSYNCP 为 1，则所选加载点为 PWM 计数器达到其最小值（CNTIN 寄存器）时。

10.5.10 通道输出初始状态寄存器(PWM_OUTINIT)

表 10-18 PWM_OUTINIT 寄存器

PWM_OUTINIT		通道输出的初始状态																Reset:00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

PWM_OUTINIT 通道输出的初始状态 Reset:00000000

名称		CH 3OI V	CH 2O IV	CH 1O IV	CH0O IV
访问		RW	R W	RW	RW
Reset		0	0	0	0

字段	说明
3 CH3OIV	<p>通道 3 输出初始值</p> <p>0: 初始值为 0 1: 初始值为 1</p> <p>进行初始化时, 强制进入通道输出的值</p>
2 CH2OIV	<p>通道 2 输出初始值</p> <p>0: 初始值为 0 1: 初始值为 1</p> <p>进行初始化时, 强制进入通道输出的值</p>
1 CH1OIV	<p>通道 1 输出初始值</p> <p>0: 初始值为 0 1: 初始值为 1</p> <p>进行初始化时, 强制进入通道输出的值</p>
0 CH0OIV	<p>通道 0 输出初始值</p> <p>0: 初始值为 0 1: 初始值为 1</p> <p>进行初始化时, 强制进入通道输出的值</p>

10.5.11 输出屏蔽控制寄存器(PWM_OMCR)

表 10-19 PWM_OMCR 寄存器

PWM_OMCR 输出屏蔽控制寄存器 Reset:00000000

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

PWM_OMCR

输出屏蔽控制寄存器

Reset:00000000

名称	CH3 OMEN	CH2 OMEN	CH1 OMEN	CH0 OMEN
访问	RW	RW	RW	RW
Reset	0	0	0	0

字段	说明
3 CH3OMEN	通道 3 输出屏蔽 0：通道输出没有被屏蔽，继续正常运行 1：通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
2 CH2OMEN	通道 2 输出屏蔽 0：通道输出没有被屏蔽，继续正常运行 1：通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
1 CH1OMEN	通道 1 输出屏蔽 0：通道输出没有被屏蔽，继续正常运行 1：通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽
0 CH0OMEN	通道 0 输出屏蔽 0：通道输出没有被屏蔽，继续正常运行 1：通道输出被屏蔽，强制进入非活动状态 定义通道输出是否被屏蔽或取消屏蔽

10.5.12 模式选择寄存器(PWM_MODESEL)

表 10-20 PWM_MODESEL 寄存器

PWM_MODESEL			PWM 模式选择										Reset:00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		PAIR1FAULTEN	PAIR1SYNCCEN	PAIR1DTEEN	PAIR1DTEEN	PAIR1DECAPEEN	PAIR1COMBINEEN	PAIR1COMBINEEN		PAIR1ROFAULTEEN	PAIR1ROSYNCCEN	PAIR1RODTEEN	PAIR1RODTEEN	PAIR1RODTEEN	PAIR1RODTEEN	PAIR1RODTEEN
访问		RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0		0	0	0	0	0	0	0

字段	说明
14 PAIR1FAULTEN	故障控制使能 (n = 2) 0: 禁用此通道对中的故障控制 1: 使能此通道对中的故障控制 使能通道 (n) 和 (n+1) 的故障控制。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。
13 PAIR1SYNCCEN	同步使能 (n = 2) 0: 禁用此通道对中的 PWM 同步 1: 使能此通道对中的 PWM 同步 使能寄存器 CH(n)V 和 CH(n+1)V 的 PWM 同步。
12 PAIR1DTEEN	死区使能 (n = 2) 0: 禁用此通道对中的死区插入 1: 使能此通道对中的死区插入 使能通道 (n) 和 (n+1) 的死区插入, 该字段写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。
11 PAIR1DECAP	双边沿捕获模式捕获 (n = 2) 0: 双边沿捕获无效

字段	说明
	1: 双边沿捕获有效 根据通道 (n) 输入事件和双边沿捕获位的配置, 使能 PWM 计时器值的捕获。该字段仅在 DECAPEN = 1 时适用。如果选择双边沿捕获单发模式, 且发生捕获通道(n+1)的事件, 则硬件将自动清除 DECAP 位。
10 PAIR1DECAPEN	双边沿捕获模式使能 (n = 2) 0: 该通道对中的双沿捕捉模式禁用 1: 该通道对中的双沿捕捉模式使能。 在通道(n) 和 (n+1)中支持双边沿捕获模式。该位在双边沿捕获模式下重新配置 MSnR0, ELSnR1:ELSnR0 和 ELS(n+1) R1:ELS(n+1)R0 位的功能。它只能在 FUNCSEL[WPDIS] = 1 时可写入。
9 PAIR1COMPEN	Channel (n) 互补 (n = 2) 0: 通道(n+1) 输出和通道(n)输出相同 1: 通道(n+1) 输出是通道(n)输出互补 使能组合通道的互补模式。在互补模式下, 通道 (n+1) 输出是通道(n) 输出取反。该字节为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。
8 PAIR1COMBINEN	组合通道 (n = 2) 0: 通道(n)和(n+1) 是独立的 1: 通道(n)和(n+1) 是组合的 使能通道 (n) 和 (n+1)的组合功能。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。
6 PAIROFAULTEN	故障控制使能 (n = 0) 0: 禁用此通道对中的故障控制 1: 使能此通道对中的故障控制 使能通道 (n) 和 (n+1)的故障控制。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。
5 PAIROSYNCEN	同步使能 (n = 0) 0: 禁用此通道对中的 PWM 同步 1: 使能此通道对中的 PWM 同步 使能寄存器 CH(n)V 和 CH(n+1)V 的 PWM 同步。
4 PAIRODTEN	死区使能 (n = 0) 0: 禁用此通道对中的死区插入 1: 使能此通道对中的死区插入

字段	说明
	使能通道 (n)和(n+1)的死区插入，该字段写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
3 PAIR0DECAP	<p>双边沿捕获模式捕获 (n = 0)</p> <p>0: 双边沿捕获无效 1: 双边沿捕获有效</p> <p>根据通道 (n) 输入事件和双边沿捕获位的配置，使能 PWM 计时器值的捕获。该字段仅在 DECAPEN = 1 时适用。如果选择双边沿捕获单发模式，且发生捕获通道(n+1)的事件，则硬件将自动清除 DECAP 位。</p>
2 PAIR0DECAPEN	<p>双边沿捕获模式使能 (n = 0)</p> <p>0: 该通道对中的双沿捕捉模式禁用 1: 该通道对中的双沿捕捉模式使能</p> <p>在通道(n) 和 (n+1)中支持双边沿捕获模式。该位在双边沿捕获模式下重新配置 MSnR0, ELSnR1:ELSnR0 和 ELS(n+1) R1:ELS(n+1)R0 位的功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写入。</p>
1 PAIR0COMPEN	<p>Channel (n)互补 (n = 0)</p> <p>0: 通道(n+1) 输出和通道(n)输出相同 1: 通道(n+1) 输出是通道(n)输出互补</p> <p>使能组合通道的互补模式。在互补模式下，通道 (n+1) 输出是通道(n) 输出取反。该字节为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
0 PAIR0COMBINEN	<p>组合通道 (n = 0)</p> <p>0: 通道(n)和(n+1) 是独立的 1: 通道(n)和(n+1) 是组合的</p> <p>使能通道 (n) 和 (n+1)的组合功能。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>

10.5.13 死区插入控制寄存器(PWM_DTSET)

表 10-21 PWM_DTSET 寄存器

PWM_DTSET	死区插入控制																Reset:00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

PWM_DTSET

死区插入控制

Reset:00000000

名称		DTPSC	DTVAL
访问		RW	RW
Reset		0	0

字段	说明
----	----

7: 6
DTPSC
死区预分频值
0x: 总线时钟 1 分频
10: 总线时钟 4 分频
11: 总线时钟 16 分频

选择总线时钟分频因子。死区计数器使用此预分频时钟，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

死区值

选择死区计数器的死区插入值。死区计数器由总线时钟的缩放版本计时，请参考 [10.4.8.4 死区时间插入](#)。

5: 0
DTVAL
死区插入值 = (DTPSC x DTVAL)。DTVAL 选择插入的死区计数值，如下所示：
当 DTVAL 为 0 时，没有插入任何计数
当 DTVAL 为 1 时，插入 1 个计数值
当 DTVAL 为 2 时，插入 2 个计数
这种模式可能持续到 63 个计数，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

10.5.14 外部触发器寄存器(PWM_EXTTRIG)

表 10-22 PWM_EXTTRIG 寄存器

PWM_EXTTRIG

PWM 外部触发器

Reset:00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称							TRIGF	INITRIGEN						CH3TRIG	CH2TRIG	CH1TRIG	CH0TRIG
访问							R/WOC	RW						RW	RW	RW	RW
Reset							0	0						0	0	0	0

字段	说明
9 TRIGF	<p>通道触发标志</p> <p>0: 没有生成通道触发 1: 产生通道触发</p> <p>任意某个通道触发时置 1。通过读取 EXTTRIG，然后将 0 写入 TRIGF 可清除。如果在清除序列完成之前生成另一个通道触发，则清除操作无效，TRIGF 仍保持置位。</p>
8 INITTRIGEN	<p>使能初始化触发器</p> <p>0: 禁用初始化触发 1: 使能初始化触发</p> <p>当 PWM 计数器等于 CNTIN 寄存器时，允许产生触发。</p>
3 CH3TRIG	<p>通道 3 触发使能</p> <p>0: 禁用通道触发 1: 使能通道触发</p> <p>当 PWM 计数器等于 CHnV 寄存器时，允许产生通道触发。</p>
2 CH2TRIG	<p>通道 2 触发使能</p> <p>0: 禁用通道触发 1: 使能通道触发</p> <p>当 PWM 计数器等于 CHnV 寄存器时，允许产生通道触发。</p>
1 CH1TRIG	<p>通道 1 触发使能</p> <p>0: 禁用通道触发 1: 使能通道触发</p> <p>当 PWM 计数器等于 CHnV 寄存器时，允许产生通道触发。</p>
0 CH0TRIG	<p>通道 0 触发使能</p> <p>0: 禁用通道触发 1: 使能通道触发</p> <p>当 PWM 计数器等于 CHnV 寄存器时，允许产生通道触发。</p>

10.5.15 通道输出极性控制寄存器(PWM_CHOPOLCR)

表 10-23 PWM_CHOPOLCR 寄存器

PWM_CHOPOLCR	通道输出极性控制寄存器																Reset:00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

PWM_CHOPOLCR

通道输出极性控制寄存器

Reset:00000000

名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称													CH3 POL	CH2P OL	CH1 POL	CH0 POL
访问													RW	RW	RW	RW
Reset													0	0	0	0

字段	说明
3 CH3POL	<p>通道 3 极性</p> <p>0：通道极性为高电平有效 1：通道极性为低电平有效</p> <p>定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
2 CH2POL	<p>通道 2 极性</p> <p>0：通道极性为高电平有效 1：通道极性为低电平有效</p> <p>定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
1 CH1POL	<p>通道 1 极性</p> <p>0：通道极性为高电平有效 1：通道极性为低电平有效</p> <p>定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
0 CH0POL	<p>通道 0 极性</p> <p>0：通道极性为高电平有效 1：通道极性为低电平有效</p> <p>定义通道输出的极性，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>

10.5.16 故障检测状态寄存器(PWM_FDSR)

表 10-24 PWM_FDSR 寄存器

PWM_FDSR		故障检测状态寄存器										Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称									FA UL TD F	WP EN	FA UL TI N			FA UL TD F2	FA UL TD F1	FAU LTD F0
访问									R/ W0 C	RW	RO			R/ W0 C	R/ W0 C	R/W 0C
Reset									0	0	0			0	0	0

字段	说明
7 FAULTDF	<p>故障检测标志</p> <p>0: 未检测到故障 1: 检测到故障</p> <p>FAULTDF j 位的逻辑或结果, 其中 j = 2, 1, 0。在 FAULTDF 置位的情况下, 当故障输入端不存在故障条件时, 读取故障状态寄存器, 向 FAULTDF 写入 0 可清零 FAULTDF。如果在清除序列完成前在一个已使能的故障输入中检测到另一个故障, 则清除操作无效, FAULTDF 仍保持置位状态。当 FAULTDF j 位全部清除后, FAULTDF 也会被清 0。</p>
6 WPEN	<p>写保护使能</p> <p>0: 禁用写保护, 写保护位可写 1: 使能写保护, 写保护位不可写</p> <p>只允许写 1, 写 0 无效。当写保护使能时, 具有写保护功能的位不能被写入。当写保护禁用时, 具有写保护功能的位可以写入。WPEN 位是 WPDIS 位取反。当 WPDIS 位写入 1 时, 可清除 WPEN, 即禁用写保护。当 WPENS 位写入 1 时, 可清除 WPDIS, 即使能写保护。</p>
5 FAULTIN	<p>故障输入</p> <p>0: 使能故障输入的逻辑或为 0 1: 使能故障输入的逻辑或为 1</p> <p>表示当使能故障控制后, 在其过滤器 (若已启用过滤器) 后使能的故障输入的逻辑或。</p>
2 FAULTDF2	<p>故障检测标志 2</p>

字段	说明
	<p>0：未检测到故障</p> <p>1：检测到故障</p> <p>在 FAULTDF2 置 1 时，读取 FDSR 寄存器然后往 FAULTDF2 写 0 可清除。向 FAULTDF2 中写 1 不起作用。当 FAULTDF 位被清 0 时，FAULTDF2 位也会被清 0。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除操作无效，FAULTDF2 仍保持置位状态。</p>
1 FAULTDF1	<p>故障检测标志 1</p> <p>0：未检测到故障</p> <p>1：检测到故障</p> <p>在 FAULTDF1 置 1 时，读取 FDSR 寄存器然后往 FAULTDF1 写 0 可清除。向 FAULTDF1 中写 1 不起作用。当 FAULTDF 位被清 0 时，FAULTDF1 位也会被清 0。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除操作无效，FAULTDF1 仍保持置位状态。</p>
0 FAULTDF0	<p>故障检测标志 0</p> <p>0：未检测到故障</p> <p>1：检测到故障</p> <p>在 FAULTDF0 置 1 时，读取 FDSR 寄存器然后往 FAULTDF0 写 0 可清除。当 FAULTDF 位被清 0 时，FAULTDF0 位也会被清 0。如果在清除序列完成前在相应的的故障输入中检测到另一个故障，则清除无效，FAULTDF0 将保持置位状态。</p>

10.5.17 输入捕获滤波器控制(PWM_CAPFILTER)

表 10-25 PWM_CAPFILTER 寄存器

PWM_CAPFILTER												输入捕获滤波器控制				Reset:00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
名称													CH3CAPFVAL[4: 1]						
访问													RW						
Reset													0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
名称	CH3CAPFVAL[0: 0]	CH2CAPFVAL				CH1CAPFVAL				CH0CAPFVAL									
访问	RW	RW				RW				RW									
Reset	0	0				0				0									

字段	说明
19: 15 CH3CAPFVAL	通道 3 输入滤波器

字段	说明
	选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。
14: 10 CH2CAPFVAL	通道 2 输入滤波器 选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。
9: 5 CH1CAPFVAL	通道 1 输入滤波器 选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。
4: 0 CH0CAPFVAL	通道 0 输入滤波器 选择用于通道输入的滤波器值，当值为 0 时禁用该滤波器。

10.5.18 故障滤波器和故障使能寄存器(PWM_FFAFER)

表 10-26 PWM_FFAFER 寄存器

PWM_FFAFER		故障滤波器和故障使能寄存器										Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	FFVAL									FF	FF	FF		FE	FE	FERO
访问	RW									2E	1E	OE		R2	R1	EN
Reset	0									N	N	N		EN	EN	EN
访问										RW	RW	RW		RW	RW	RW
Reset										0	0	0		0	0	0

字段	说明
15: 8 FFVAL	故障输入滤波器 选择用于故障输入的滤波器值，当值为 0 时禁用故障滤波器。
6 FF2EN	故障输入 2 滤波器使能 0：禁用故障输入滤波器 1：使能故障输入滤波器 启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
5 FF1EN	故障输入 1 滤波器使能 0：禁用故障输入滤波器 1：使能故障输入滤波器 启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。
4	故障输入 0 滤波器使能

字段	说明
FFOEN	<p>0：禁用故障输入滤波器 1：使能故障输入滤波器</p> <p>启用故障输入滤波器，该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
2 FER2EN	<p>故障输入 2 使能</p> <p>0：禁用故障输入 1：使能故障输入</p> <p>使能故障输入。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
1 FER1EN	<p>故障输入 1 使能</p> <p>0：禁用故障输入 1：使能故障输入</p> <p>使能故障输入。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
0 FER0EN	<p>故障输入 0 使能</p> <p>0：禁用故障输入 1：使能故障输入</p> <p>使能故障输入。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>

10.5.19 正交解码器接口配置寄存器(PWM_QDI)

表 10-27 PWM_QDI 寄存器

PWM_QDI																正交解码器接口配置寄存器							Reset:00000004						
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16													
名称																													
访问																													
Reset																													
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0													
名称											PH AP OL	PH BP OL	QU AD MO DE	QU AD IR	CN TO FDI R	QDI EN													
访问											RW	RW	RW	RO	RO	RW													
Reset											0	0	0	1	0	0													

字段	说明
5 PHAPOL	A 相 输入极性

字段	说明
	<p>0: 正常极性。在识别 A 相输入信号的上升沿和下降沿之前, 该信号未反相</p> <p>1: 反转极性。在识别 A 相输入信号的上升沿和下降沿之前, 该信号反相</p> <p>选择正交解码器 A 相输入的极性。</p>
4 PHBPOL	<p>B 相 输入极性</p> <p>0: 正常极性。在识别 B 相输入信号的上升沿和下降沿之前, 该信号未反相</p> <p>1: 反转极性。在识别 B 相输入信号的上升沿和下降沿之前, 该信号反相</p> <p>选择正交解码器 B 相输入的极性</p>
3 QUADMODE	<p>正交解码器模式</p> <p>0: A 相和 B 相编码模式</p> <p>1: 计数和方向编码模式</p> <p>选择正交解码器模式下使用的编码模式。</p>
2 QUADIR	<p>正交解码器模式下的 PWM 计数器方向</p> <p>0: 计数方向为降低 (PWM 计数器递减).</p> <p>1: 计数方向为增加 (PWM 计数器递增)</p> <p>表示计数方向。</p>
1 CNTOFDIR	<p>正交解码器模式下的定时器溢出方向</p> <p>0: CNTOF 位设置在计数的底部。PWM 计数器递减, PWM 计数器从其最小值 (CNTIN 寄存器) 变为最大值 (MCVR 寄存器)</p> <p>1: CNTOF 位设置在计数的顶部。PWM 计数器递增, PWM 计数器从其最大值 (MCVR 寄存器) 变为最小值 (CNTIN 寄存器)</p> <p>表示 CNTOF 位是否设置在计数的顶部或底部。</p>
0 QDIEN	<p>正交解码器模式使能</p> <p>0: 禁用正交解码器模式</p> <p>1: 使能正交解码器模式</p> <p>使能正交解码器模式。在此模式下, A 相和 B 相输入信号控制 PWM 计数器方向。正交解码器模式优先于其他模式。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>

10.5.20 配置寄存器(PWM_CONF)

表 10-28 PWM_CONF 寄存器

PWM_CONF	配置														Reset:00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

PWM_CONF

配置

Reset:00000000

名称																	EVENT3P SC	EVENT2P SC	EVENT1P SC	EVENT0PSC
访问																	RW	RW	RW	RW
Reset																	0	0	0	0
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
名称							GT BE OU T	GT BE EN	CNTOFNUM											
访问							RW	RW	RW											
Reset							0	0	0											

字段	说明
23: 22 EVENT3PSC	通道 3 输入事件预分频 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
21: 20 EVENT2PSC	通道 2 输入事件预分频 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
19: 18 EVENT1PSC	通道 1 输入事件预分频 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
17: 16 EVENT0PSC	通道 0 输入事件预分频 00: 1 个输入事件触发一次捕获 01: 2 个输入事件触发一次捕获 10: 4 个输入事件触发一次捕获 11: 8 个输入事件触发一次捕获
10 GTBEOUT	全局时基输出 0: 禁用全局时基信号生成 1: 使能全局时基信号生成 使能到其它 PWM 的全局时基信号生成。
9 GTBEEN	全局时基使能

字段	说明
6: 0 CNTOFNUM	<p>0: 禁用外部全局时基</p> <p>1: 使能外部全局时基</p> <p>CNTOF 频率</p> <p>选择计数器溢出次数与 CNTOF 位置 1 次数之间的比率</p> <p>CNTOFNUM = 0: 每个计数器溢出都设置 CNTOF 位</p> <p>CNTOFNUM = 1: CNTOF 位设置为第一次计数器溢出, 但不是下一次溢出</p> <p>CNTOFNUM = 2: CNTOF 位设置为第一次计数器溢出, 但不会设置为接下来的两次溢出</p> <p>CNTOFNUM = 3: CNTOF 位设置为第一次计数器溢出, 但不会设置为接下来的三次溢出</p>

10.5.21 故障输入极性寄存器(PWM_FLTPOL)

表 10-29 PWM_FLTPOL 寄存器

PWM_FLTPOL																PWM 故障输入极性			Reset:00000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16					
名称																					
访问																					
Reset																					
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
名称														FL T2 PO L	FL T1 PO L	FLT0 POL					
访问														RW	RW	RW					
Reset														0	0	0					

字段	说明
2 FLT2POL	<p>故障输入 2 极性</p> <p>0: 故障输入极性为高电平有效。故障输入处的 1 表示一个故障。</p> <p>1: 故障输入极性为低电平有效。故障输入处的 0 表示一个故障。</p> <p>定义故障输入极性。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
1 FLT1POL	<p>故障输入 1 极性</p> <p>0: 故障输入极性为高电平有效。故障输入处的 1 表示一个故障。</p> <p>1: 故障输入极性为低电平有效。故障输入处的 0 表示一个故障。</p> <p>定义故障输入极性。该字段为写保护, 它只能在 FUNCSEL[WPDIS] = 1 时可写。</p>
0 FLT0POL	<p>故障输入 0 极性</p>

字段	说明
----	----

- 0: 故障输入极性为高电平有效。故障输入处的 1 表示一个故障。
- 1: 故障输入极性为低电平有效。故障输入处的 0 表示一个故障。

定义故障输入极性。该字段为写保护，它只能在 FUNCSEL[WPDIS] = 1 时可写。

10.5.22 同步配置寄存器(PWM_SYNCONF)

表 10-30 PWM_SYNCONF 寄存器

PWM_SYNCONF										同步配置							Reset:00000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
名称										HW POL	SW POL	SW VH W SY NC	IN VH WS Y NC	OM VH W SY NC	P W MS VH W SY NC	CNTV HWS YNC			
访问										R W	R W	R W	R W	R W	R W	R W			
Reset										0	0	0	0	0	0	0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
名称				SWV SWS YNC	INVS WSY NC	OMV SWS YNC	PW MSV SWS YNC	C NT VS W SY N C	SY NC MO DE			SW OC	IN VC		CN TI NC		HWT RIGM ODES EL		
访问				RW	RW	RW	RW	R W	RW		R W	R W		R W		RW			
Reset				0	0	0	0	0	0		0	0		0		0			

字段	说明
----	----

22 通道 CHPOLCR 同步由硬件触发器激活
HWPOL

- 0: 硬件触发器不会激活 CHPOLCR 寄存器同步
- 1: 硬件触发器激活 CHPOLCR 寄存器同步

21 通道 CHPOLCR 同步由软件触发器激活
SWPOL

- 0: 软件触发器不会激活 CHPOLCR 寄存器同步
- 1: 软件触发器激活 CHPOLCR 寄存器同步

20 软件输出控制同步由硬件触发器激活
SWVHWSYNC

- 0: 硬件触发器不会激活 CHOSWCR 寄存器同步
- 1: 硬件触发器激活 CHOSWCR 寄存器同步

字段	说明
19 INVHWSYNC	通过硬件触发器激活反相控制同步 0: 硬件触发器不会激活 INVCR 寄存器同步 1: 硬件触发器激活 INVCR 寄存器同步
18 OMVHWSYNC	通过硬件触发器激活输出屏蔽同步 0: 硬件触发器不会激活 OMCR 寄存器同步 1: 硬件触发器激活 OMCR 寄存器同步
17 PWMSVHWSYNC	通过硬件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步 0: 硬件触发器不会激活 MCVR, CNTIN 和 CHV 寄存器同步 1: 硬件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步
16 CNTVHWSYNC	通过硬件触发器激活 PWM 计数器同步 0: 硬件触发器不会激活 PWM 计数器同步 1: 硬件触发器激活 PWM 计数器同步
12 SWVSWSYNC	通过软件触发器激活软件输出控制同步 0: 软件触发器不会激活 CHOSWCR 寄存器同步。 1: 软件触发器激活 CHOSWCR 寄存器同步
11 INVSWSYNC	通过软件触发器激活反相控制同步 0: 软件触发器不会激活 INVCR 寄存器同步。 1: 软件触发器激活 INVCR 寄存器同步
10 OMVSWSYNC	通过软件触发器激活输出掩码同步 0: 软件触发器不会激活 OMCR 寄存器同步 1: 软件触发器激活 OMCR 寄存器同步
9 PWMSVSWSYNC	通过软件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步 0: 软件触发器不会激活 MCVR, CNTIN 和 CHV 寄存器同步 1: 软件触发器激活 MCVR, CNTIN 和 CHV 寄存器同步。
8 CNTVSWSYNC	通过软件触发器激活 PWM 寄存器同步 0: 软件触发器不会激活 PWM 计数器同步 1: 软件触发器激活 PWM 计数器同步
7 SYNCMODE	同步模式 0: 选择传统 PWM 同步 1: 选择增强 PWM 同步 选择 PWM 同步模式
5 SWOC	CHOSWCR 寄存器同步

字段	说明
	0: 在总线时钟的所有上升沿, 使用其缓冲区的值更新 CHOSWCR 寄存器 1: 通过 PWM 同步, 使用其缓冲区的值更新 CHOSWCR 寄存器
4 INVC	INVCR 寄存器同步 0: 在总线时钟的所有上升沿, 使用其缓冲区的值更新 INVCR 寄存器 1: 通过 PWM 同步, 使用其缓冲区的值更新 INVCR 寄存器
2 CNTINC	CNTIN 寄存器同步 0: 在总线时钟的所有上升沿, 使用其缓冲区的值更新 CNTIN 寄存器 1: 通过 PWM 同步, 使用其缓冲区的值更新 CNTIN 寄存器
0 HWTRIGMODESEL	硬件触发模式 0: 当检测到硬件触发器 j 时, PWM 清零 TRIGj 位, 其中 j = 0, 1, 2. 1: 当检测到硬件触发器 j 时, PWM 不会清零 TRIGj 位, 其中 j = 0, 1, 2.

10.5.23 反相控制寄存器(PWM_INVCR)

表 10-31 PWM_INVCR 寄存器

PWM_INVCR		PWM 反相控制寄存器														Reset:00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																	
访问																	
Reset																	
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称															PAI	PAI	
															R1I	R0I	
															NV	NVE	
															EN	N	
访问															RW	RW	
Reset															0	0	

字段	说明
1 PAIR1INVEN	配对通道 1 反相使能 0: 禁用反相 1: 使能反相
0 PAIR0INVEN	配对通道 0 反相使能 0: 禁用反相 1: 使能反相

10.5.24 通道软件输出控制寄存器(PWM_CHOSWCR)

表 10-32 PWM_CHOSWCR 寄存器

PWM_CHOSWCR				PWM 通道 软件输出控制寄存器								Reset:00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称					CH 3S W CV	CH 2S W CV	CH 1S W CV	CH 0S W CV					CH 3S W EN	CH 2S W EN	CH 1S W EN	CH0SW EN
访问					R W	R W	R W	R W					R W	R W	R W	RW
Reset					0	0	0	0					0	0	0	0

字段	说明
11 CH3SWCV	通道 3 软件输出控制值 0: 软件输出控制将通道输出强制置为 0 1: 软件输出控制将通道输出强制置为 1
10 CH2SWCV	通道 2 软件输出控制值 0: 软件输出控制将通道输出强制置为 0 1: 软件输出控制将通道输出强制置为 1
9 CH1SWCV	通道 1 软件输出控制值 0: 软件输出控制将通道输出强制置为 0 1: 软件输出控制将通道输出强制置为 1
8 CH0SWCV	通道 0 软件输出控制值 0: 软件输出控制将通道输出强制置为 0 1: 软件输出控制将通道输出强制置为 1
3 CH3SWEN	通道 3 软件输出控制使能 0: 通道输出不受软件输出控制的影响 1: 通道输出受软件输出控制的影响
2 CH2SWEN	通道 2 软件输出控制使能 0: 通道输出不受软件输出控制的影响 1: 通道输出受软件输出控制的影响
1 CH1SWEN	通道 1 软件输出控制使能

字段	说明
	0: 通道输出不受软件输出控制的影响 1: 通道输出受软件输出控制的影响
0 CH0SWEN	通道 0 软件输出控制使能 0: 通道输出不受软件输出控制的影响 1: 通道输出受软件输出控制的影响

11 脉冲宽度检测定时器(PWDT)

11.1 简介

脉冲宽度检测定时器（PWDT）被用做测量脉冲宽度的工具或作为 16 位普通定时器使用。该 MCU 设备包含 1 个 PWDT 模块，支持 3 路外部通道和 1 路内部通道输入。

11.2 特性

- 4 个可选脉冲时钟输入
- 支持 2 个功能：脉冲宽度测量功能和定时功能
 - 脉冲宽度测量功能
 - 可编程起始触发沿
 - 4 个可编程测量模式
 - 支持 3 个霍尔传感器的信号输入测量
 - 支持来自模拟比较器的 3 个输入
 - 定时器功能
- 16 位计数器，用做脉冲宽度测量或定时器功能
- 中断
 - OVF：计数器溢出中断
 - RDYF：脉宽数据就绪中断

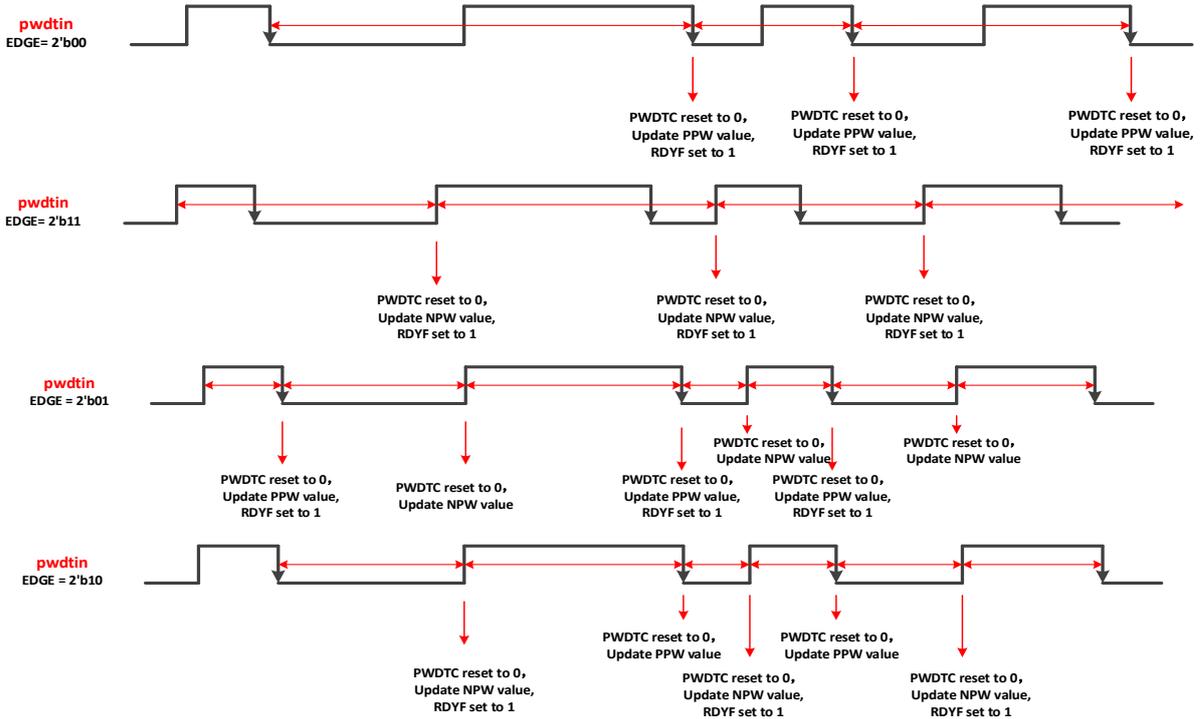


图 11-2 四种基本测量模式(HALLEN=0)

11.4.1.2 霍尔测量模式

使用霍尔测量模式时，应设置 EDGE[1: 0] = 2'b01、HALLEN = 1'b1，当 CMPEN = 1'b0 时，模块测量从 3 个通道(pwdt_in0、pwdt_in1、pwdt_in2)输入信号异或 (XOR) 后得到的脉冲输入；当 CMPEN = 1'b1 时，模块测量从 ACMP HALL A/B/C Output 通道输入信号异或 (XOR) 后得到的脉冲输入。

对于霍尔信号测量，只需选择此模式用于电机速度计算或换相，如图 11-3 所示。与上图中的基本测量模式相比，脉宽数据就绪标志 RDYF 在上升沿和下降沿均设置为 1。

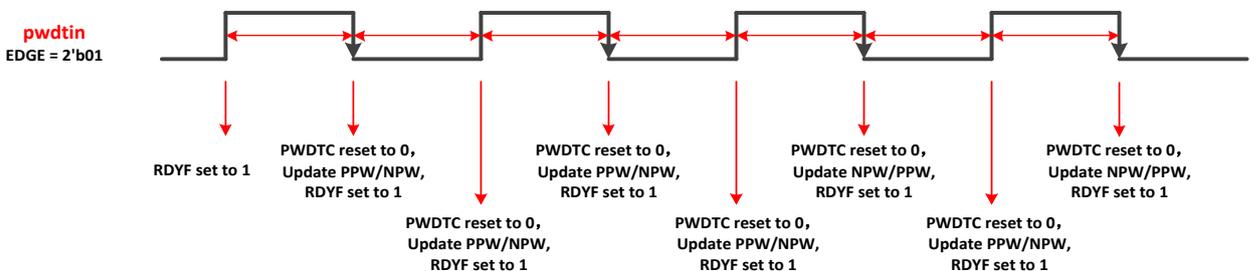


图 11-3 霍尔测量模式(HALLEN=1)

在电动机中，霍尔装置的安装用于检测转子的位置以适当地换相。通常有两种安装方式，如图 11-4 所示。一个是 120 电度间隔，另一个是 60 电度间隔。

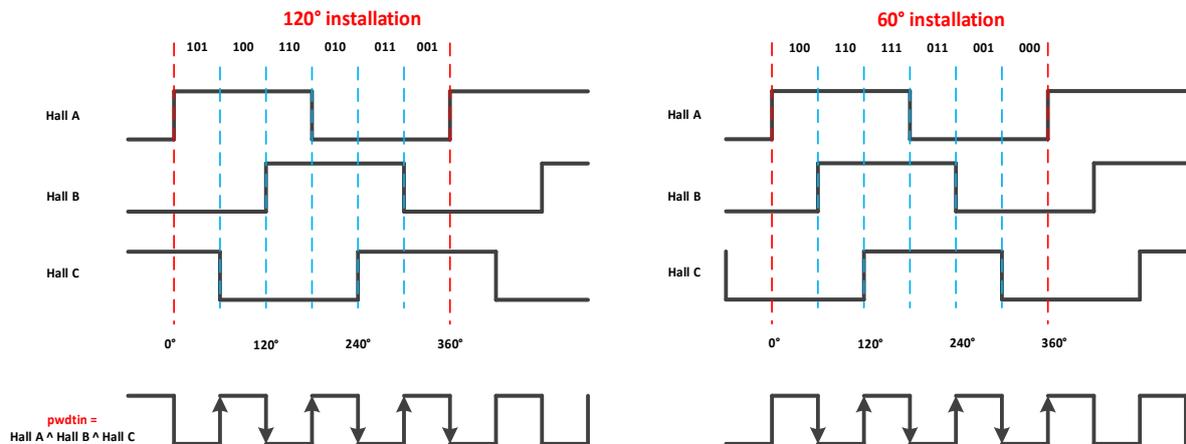


图 11-4 两种常见 Hall 安装方式

11.4.1.3 输入滤波器

输入滤波器用于滤除高/低电平小于特定宽度的噪声信号。FILTPSC[3: 0]和 FILTVAL [3: 0]设置确定最大和最小噪声脉冲宽度。图 11-5 和图 11-6 介绍了噪声滤波原理，当用户配置 FILTVAL = 15 和 FILTPSC = 2 时，滤波器脉冲宽度为 60 bclk，小于 60 bclk 的脉冲被判断为噪声脉冲并将被过滤掉。可滤波的脉冲宽度如表 11-1 所示。

注：使能滤波器功能会导致信号存在延迟（延迟时间为设置的滤波宽度）。因此，如果 PWDT 模块运行在电机应用的霍尔测量模式下，可能需要考虑该延迟导致换相时间偏移的情况，用户可基于滤波延迟做相应的软件补偿。

表 11-1 可滤波脉冲宽度范围

项	时钟	时间
可滤波脉冲宽度范围	2 bclk ~ 15*4096 bclk	0.04μs ~ 1.229ms

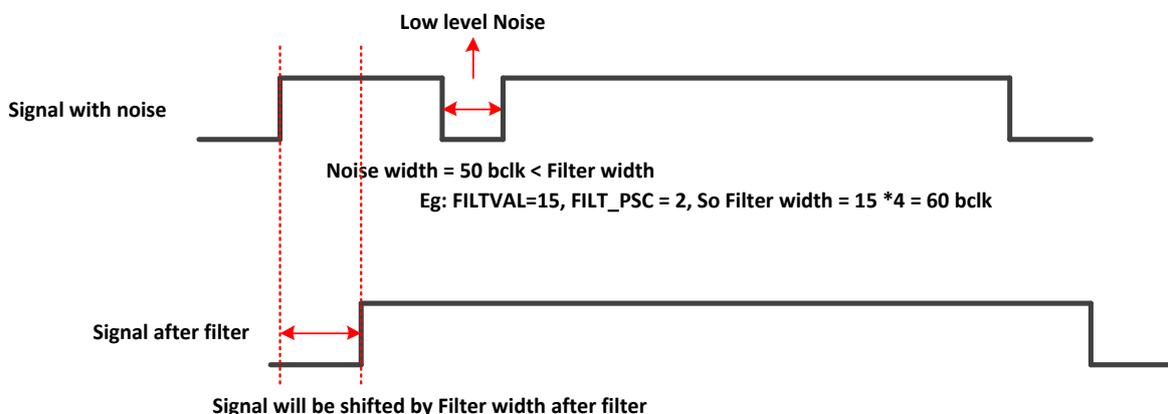


图 11-5 低电平噪音和滤波器示例

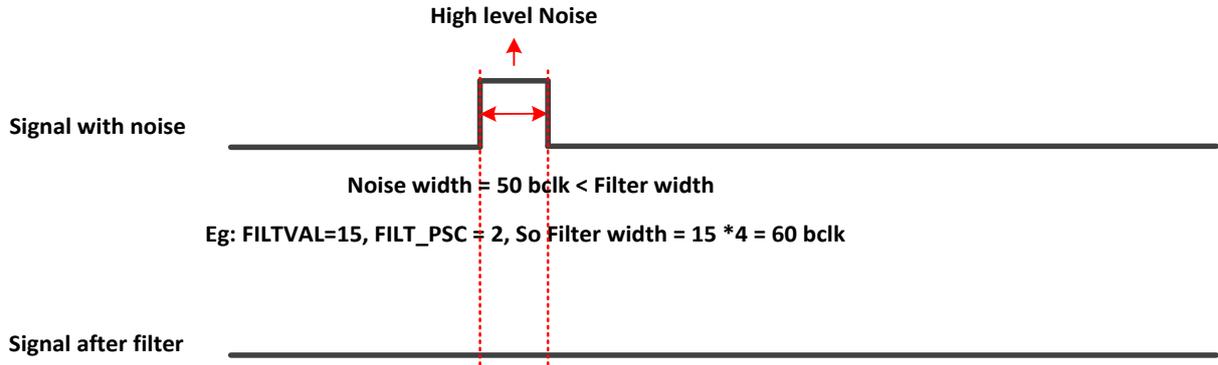


图 11-6 高电平噪音和滤波器示例

11.4.1.4 测量误差

用户应了解脉冲宽度测量的测量精度，配置 PSC 适当的值，以达到更准确的测量值。一个基本原则是，使用较小的 PSC 可以获得更准确的测量值。显然，输入脉冲越窄，相对测量误差越大。图 11-7 描述了脉冲宽度测量功能运行时的误差。在图 11-7 中，当 pwdtin 脉冲从高电平变为低电平或从低电平变为高电平时，PWDC 计数器和 pwdtclk 除数计数器同时复位为 0，并且恰好在这里发生计数错误，该错误源自图 11-7 中所示的最后计数值。实际宽度值小于测量值不足一个 pwdtclk 周期。

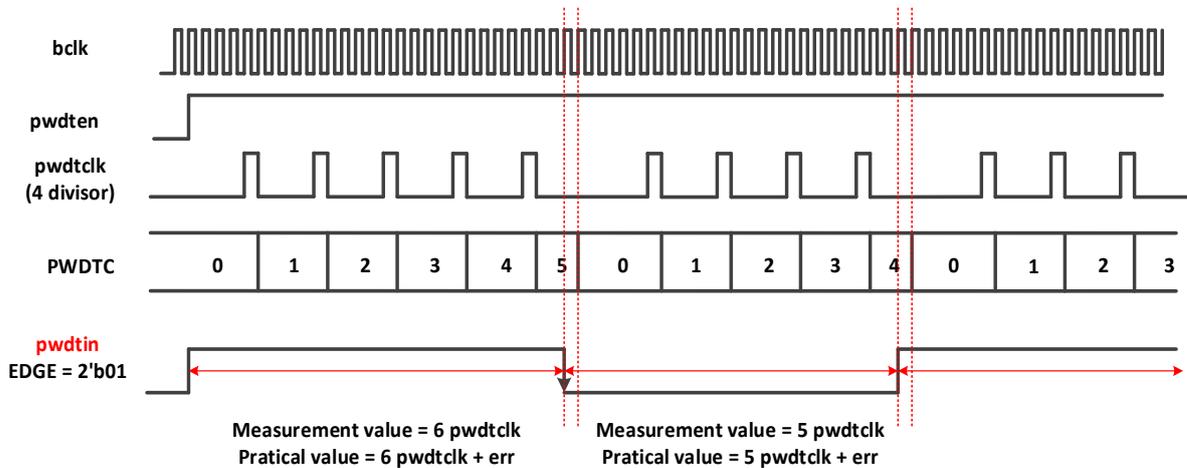


图 11-7 PWDC 计数器和计数错误

11.4.2 定时器功能

对于定时器功能，只有 OVF 状态有效，在 PWDC 计数器溢出时置位。计数值 TIMLDVAL[15: 0]可以一直修改。但是，在不同的时间点修改计数器负载值会导致 MCU 执行不同的操作，如图 11-8 和图

11-9 所示。

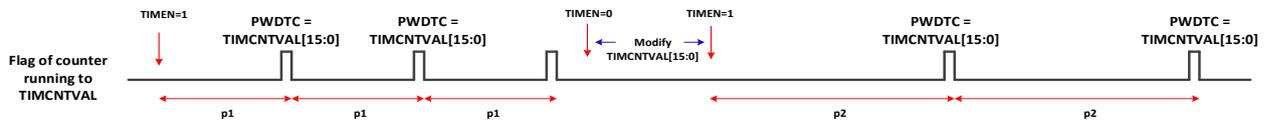


图 11-8 在 TIMEN=0 期间修改 TIMLDVAL

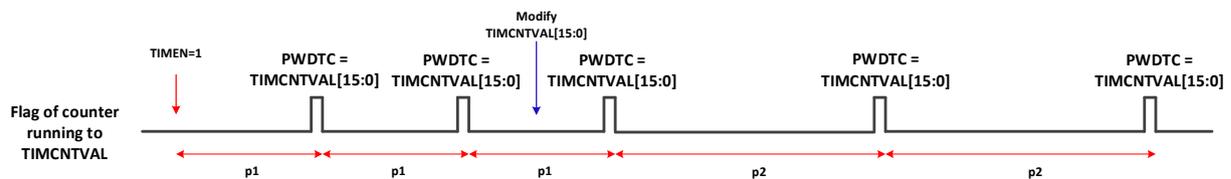


图 11-9 TIMEN=1 期间修改 TIMLDVAL

11.4.3 中断请求

表 11-2 PWDT 中断汇总

中断源	标志	本地使能	全局使能
PWDT 中断	OVF/RDYF	OVIE/PRDYIE	IE

11.4.4 低功耗模式

表 11-3 PWDT 模块低功耗模式

模式	唤醒源	备注
休眠模式	中断	模块正常工作，可通过中断唤醒 MCU
停止模式	---	模块被关闭

11.5 应用说明

11.5.1 脉冲宽度测量功能编程指南

用户必须牢记，PWDTEN 应该在所有其他控制位之后配置为 1。否则，可能会出现异常情况。对于内部 3 个比较器输入，HALLEN 和 CMPEN 应配置为 1。

11.5.2 定时器功能编程指南

只需配置 TIMLDVAL, PRESCALE 和 TIMEN 等，即可轻松使用定时器功能。用户应将 TIMEN 设置为 1，并且不能将 PWDTEN 设置为 1，因为脉冲宽度测量功能优先于定时器功能。

11.6 寄存器定义

表 11-4 PWDT 寄存器映射

PWDT0 基地址 = 0x40017000

地址	名称	宽度	描述
PWDT0 基地址+0x00	PWDT_INIT0	32	通用控制和状态位，及正脉宽内容
PWDT0 基地址+0x04	PWDT_NPW	32	负脉宽内容及 16 位自由运行计数器
PWDT0 基地址+0x08	PWDT_INIT1	32	霍尔功能控制和定时器功能控制

11.6.1 初始化寄存器 0(PWDT_INIT0)

表 11-5 PWDT_INIT0 寄存器

PWDT_INIT0		PWDT 初始化寄存器 0										Reset:00000000	
位	31~16	15~14	13~12	11~10	9~7	6	5	4	3	2	1	0	
名称	PPW	PSC1	PINSEL	EDGE	PSC0	PWDTEN	IE	PRDYIE	OVIE		RDYF	OVF	
访问	RO	RW	RW	RW	RW	RW	RW	RW	RW		R/WC	R/WOC	
Reset	0	0	0	0	0	0	0	0	0		0	0	

字段	说明
31: 16 PPW	正脉宽 正脉宽值
15: 14 PSC1	PWDT 计数器前级预分频 00 ~ 11: 分别代表 1/2/4/预留
13: 12 PINSEL	引脚选择 00/01/10/11: 分别选择 pwdt_in0/ pwdt_in1/ pwdt_in2/ ACMP_OUT.
11: 10 EDGE	选择输入边沿触发类型 00: 第一个下降沿开始，在所有之后的下降沿触发要捕获的脉宽 01: 第一个上升沿开始，在所有之后的上升沿和下降沿触发要捕获的脉宽 10: 第一个下降沿开始，在所有之后的上升沿和下降沿触发要捕获的脉宽 11: 第一个上升沿开始，在所有之后的上升沿触发要捕获的脉宽
9: 7 PSC0	PWDT 计数器后级预分频 000 ~ 111: 分别代表 1/2/4/8/.../128
6 PWDTEN	PWDT 脉宽测量模式使能 0: 禁用

字段	说明
	1: 使能
	注意: PWDTEN 使能(pwdt 脉宽测量模式)优先于 TIMEN 使能(定时器模式), 因此当要使能定时器模式时, 必须禁用 PWDTEN。
5 IE	PWDT 模块中断使能 0: 禁用 1: 使能
4 PRDYIE	PWDT 脉宽数据就绪中断使能 0: 禁用 1: 使能
3 OVIE	PWDT 计数器溢出中断使能 0: 禁用 1: 使能
1 RDYF	PWDT 脉宽数据就绪 0: pwdt 脉宽寄存器未更新 1: pwdt 脉宽寄存器已更新, 写 0 清除
0 OVF	PWDT 计数器溢出 0: 无溢出 1: pwdt 计数器溢出, 写 0 清除

11.6.2 脉宽计数寄存器(PWDT_NPW)

表 11-6 PWDT_NPW 寄存器

PWDT_NPW		PWDT NPW 计数值	Reset: 00000000
位	31~16	15~0	
名称	PWDC	NPW	
访问	RO	RO	
Reset	0	0	

字段	说明
31: 16 PWDC	脉宽计数器 用于脉宽测量或定时器计数
15: 0 NPW	负脉宽 负脉宽值

11.6.3 初始化寄存器 1(PWDT_INIT1)

表 11-7 PWDT_INIT1 寄存器

PWDT_INIT1		PWDT 初始化寄存器 1						Reset:00000000			
位	31	30	29	28	27~12	11	10	9	8	7~4	3~0
名称		HA LL A	HA LL B	HA LL C	TIMLDVAL	CMPE N	TIMEN	HALLEN	FILT EN	FILTPSC	FILTVAL
访问		RO	RO	RO	RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0	0	0	0

字段	说明
30: 28 HALLA/B/C	HALLA/HALLB/HALLC 状态值 如果 3 个霍尔传感器安装间距为 60 电度： 100 → 110 → 111 → 011 → 001 → 000 否则，3 个霍尔传感器安装间距为 120 电度： 101 → 100 → 110 → 010 → 011 → 001
27: 12 TIMLDVAL	定时器装载值 定时器从 0x0000 运行 TIMLDVAL。
11 CMPEN	比较器输入使能 0：使能来自 pad PWDT_IN0 ~ PWDT_IN2 外部的 pwdt_in0 ~ pwdt_in2 1：使能来自 acmp0_0 ~ acmp0_2 内部的 pwdt_in0 ~ pwdt_in2 注意： 当 CMPEN=1 时，pwdt_in0 ~ pwdt_in2 来自于 acmp0_0 ~ acmp0_2 内部。然后 HALLEN=1，可以通过 HALL 传感器的行为来测量 acmp0_0 ~ acmp0_2 信号，否则将基于 PINSEL 测量其中一个信号。
10 TIMEN	定时器使能 0：禁用 1：使能定时器功能 注意： PWDTEN 使能（pwdt 脉宽测量模式）优先于 TIMEN 使能（定时器模式），因此当要使能定时器模式时，必须禁用 PWDTEN。
9 HALLEN	霍尔传感器信号检测使能 0：禁用霍尔传感器信号检测功能 1：使能霍尔传感器信号检测功能
8 FILTEN	输入滤波器使能 0：禁用。 1：使能滤波器功能

字段	说明
7 ~ 4 FILTPSC	滤波器预分频器 1 ~ 12: 分别表示 2/4/8.../4096 分频 0, 13 ~ 15: 不分频滤波器时钟
3 ~ 0 FILTVAL	滤波器值 0: 禁止滤波器 1 ~ 15: 滤波值

12 周期性中断定时器 (TIMER)

12.1 简介

TIMER 模块是用于定时发起中断和触发的定时器。

12.2 特性

- 定时器能够生成中断
- 定时器能够生成触发脉冲
- 每个定时器都具有独立的超时周期
- 支持 4 个 32bit 定时器
- 支持链 (Link) 模式

12.3 结构框图

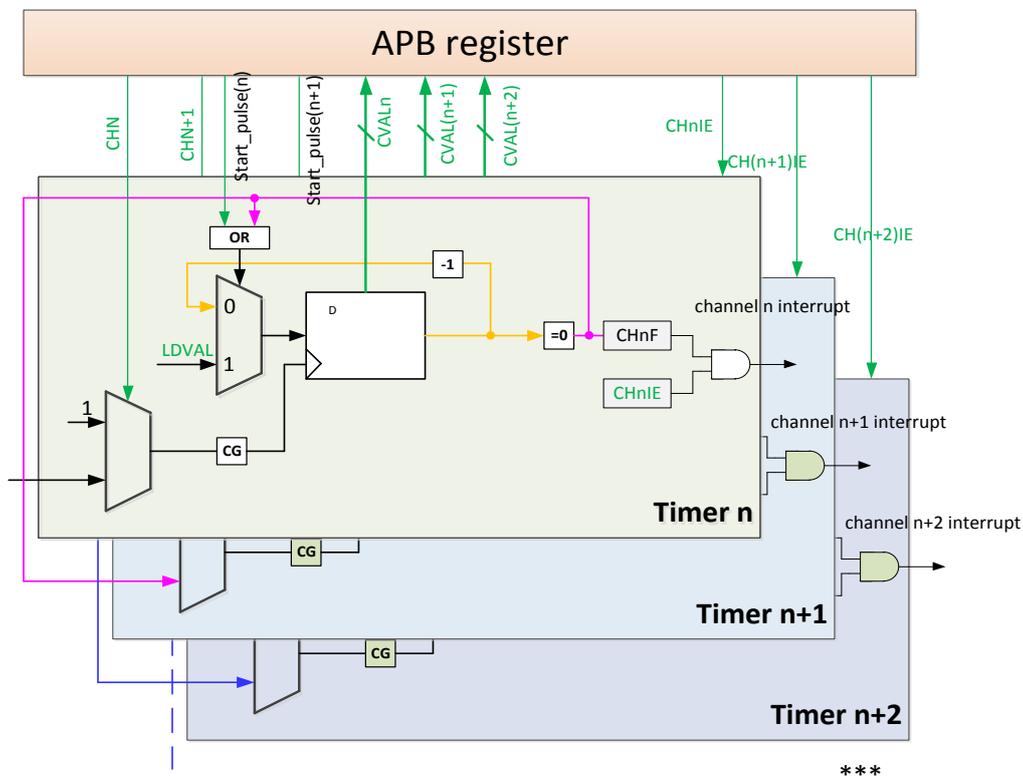


图 12-1 TIMER 结构框图

12.4 功能描述

12.4.1 普通模式

定时器使能时自动加载 `TIMER_LDVAL` 寄存器指定起始值，倒计数至 0 时，将生成一个触发脉冲并设置超时标志，然后再次装载起始值重新计数。

如有需要，可以通过 `TIMER_CVAL` 寄存器读取定时器的当前计数值。通过 `TIMER_INIT[TEN]` 先禁用再使能，可以重新启动计数周期。

12.4.2 链接模式

当某个定时器的链接模式处于使能状态，那么只有在上一个定时器溢出后，它才会开始计时。因此，如果定时器 `n-1` 已倒数至 0，定时器 `n` 的值将递减 1。这样就能将某些定时器连接起来形成更长的定时器。
注：第一个定时器 (timer 0) 不能链接至其他定时器。

12.4.3 中断

定时器中断可通过置位 `TIE` 来使能。当相关定时器发生超时，`TFLG` 超时标志置位为 1，写入 1 清零。在使用链接功能时，一般只使能定时器 `n` 中断，与 `n` 链接的定时器中断都是关闭状态。

12.5 寄存器定义

表 12-1 定时器寄存器映射

TIMER 基地址 = 0x40011000

TIMER_CH0 基地址=0x40011100

TIMER_CH1 基地址=0x40011110

TIMER_CH2 基地址=0x40011120

TIMER_CH3 基地址=0x40011130

地址	名称	宽度	描述
TIMER 基地址+0x00	<code>TIMER_MCR</code>	32	模块控制器寄存器
TIMER_CHx 基地址+0x00	<code>TIMER_LDVAL</code>	32	初始值寄存器
TIMER_CHx 基地址+0x04	<code>TIMER_CVAL</code>	32	当前值寄存器
TIMER_CHx 基地址+0x08	<code>TIMER_INIT</code>	32	初始化寄存器
TIMER_CHx 基地址+0x0C	<code>TIMER_TF</code>	32	标志寄存器

【说明】 上表中，`x=0~3`。

12.5.1 定时器模块控制寄存器(TIMER_MCR)

表 12-2 TIMER_MCR 寄存器

TIMER_MCR 定时器模块控制寄存器 Reset: 0x00000002

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

字段	说明
1	模块禁用 - (TIMER 部分)
MDIS	0: 使能定时器模块 1: 禁用定时器模块 禁用定时器模块，必须在执行任何其他设置前使能该字段。 注意: MDIS 可以实现 4 路定时器同时暂停/再次开始计数

12.5.2 定时器装载值寄存器(TIMER_LDVAL)

表 12-3 TIMER_LDVAL 寄存器

TIMER_LDVAL 定时器装载值寄存器 Reset: 0x00000000

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

字段	说明
31: 0	装载值寄存器
LDVAL	定时器起始值。定时器将倒计时至 0，然后生成一个中断并再次装载该寄存器的值。将新值写入该寄存器不会重启定时器。相反定时器过期后，会装载新值。要中止当前周期并用新值开始一个定时器周期，必须先禁用该定时器然后再将其使能。

字段	说明
	TVAL 与时间的换算公式如下： the timing period(Unit: second) = (TVAL + 1) / timing clock frequency(Unit: Hz)

12.5.3 定时器当前值寄存器(TIMER_CVAL)

表 12-4 TIMER_CVAL 寄存器

TIMER_CVAL		定时器当前值寄存器														Reset: 0x00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段		CVAL[31: 16]															
访问		RO															
Reset		0															
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段		CVAL[15: 0]															
访问		RO															
Reset		0															

字段	说明
31: 0	当前定时器值
CVAL	代表实时计数值。向下计数为 0 后自动重新装载起始值。

12.5.4 定时器初始寄存器 (TIMER_INIT)

表 12-5 TIMER_INIT 寄存器

TIMER_INIT		定时器初始寄存器														Reset: 0x00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段																	
访问																	
Reset																	
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段													LINKEN	TIE	TEN		
访问													RW	RW	RW		
Reset													0	0	0		

字段	说明
2	链接模式
LINKEN	0: 定时器不链接 1: 定时器链接至前一定时器。例如，对于定时器 2，若该字段置位，则定时器 2 链接至定时 1。 激活时，定时器 n-1 需先到期，定时器 n 才能递减 1。不能链接定时器 0。

字段	说明
1 TIE	<p>定时器中断使能</p> <p>0：禁用定时器中断请求 1：一旦置位 TFLG，请求中断</p> <p>当某个中断挂起或 TFLG 置位时，使能该中断将立即引起中断事件。要避免这种情况，必须先清零相关的 TFLG。</p>
0 TEN	<p>定时器使能</p> <p>0：禁用定时器 n 1：使能定时器 n</p> <p>使能或禁用定时器 n</p>

12.5.5 定时器标志寄存器(TIMER_TF)

表 12-6 TIMER_TF 寄存器

TIMER_TF		定时器标志寄存器																Reset: 0x00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
字段																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
字段																		TFLG
访问																		R/W1C
Reset																		0

字段	说明
0 TFLG	<p>定时器超时标志</p> <p>在定时器周期结束时置 1。将 1 写入该标志可将其清零，写入 0 则无效。若使能 TIE = 1，TFLG 将引发中断请求。</p> <p>0：尚未发生超时 1：超时已经发生</p>

13 采集传输终端 (CTU)

13.1 简介

CTU 模块可用于模块间的互连，片上不同模块之间传递信号。

13.2 特性

- ACMP 输出捕获
- RTC 输出捕获
- ADC 硬件触发
- PWM 软件同步

13.3 结构框图

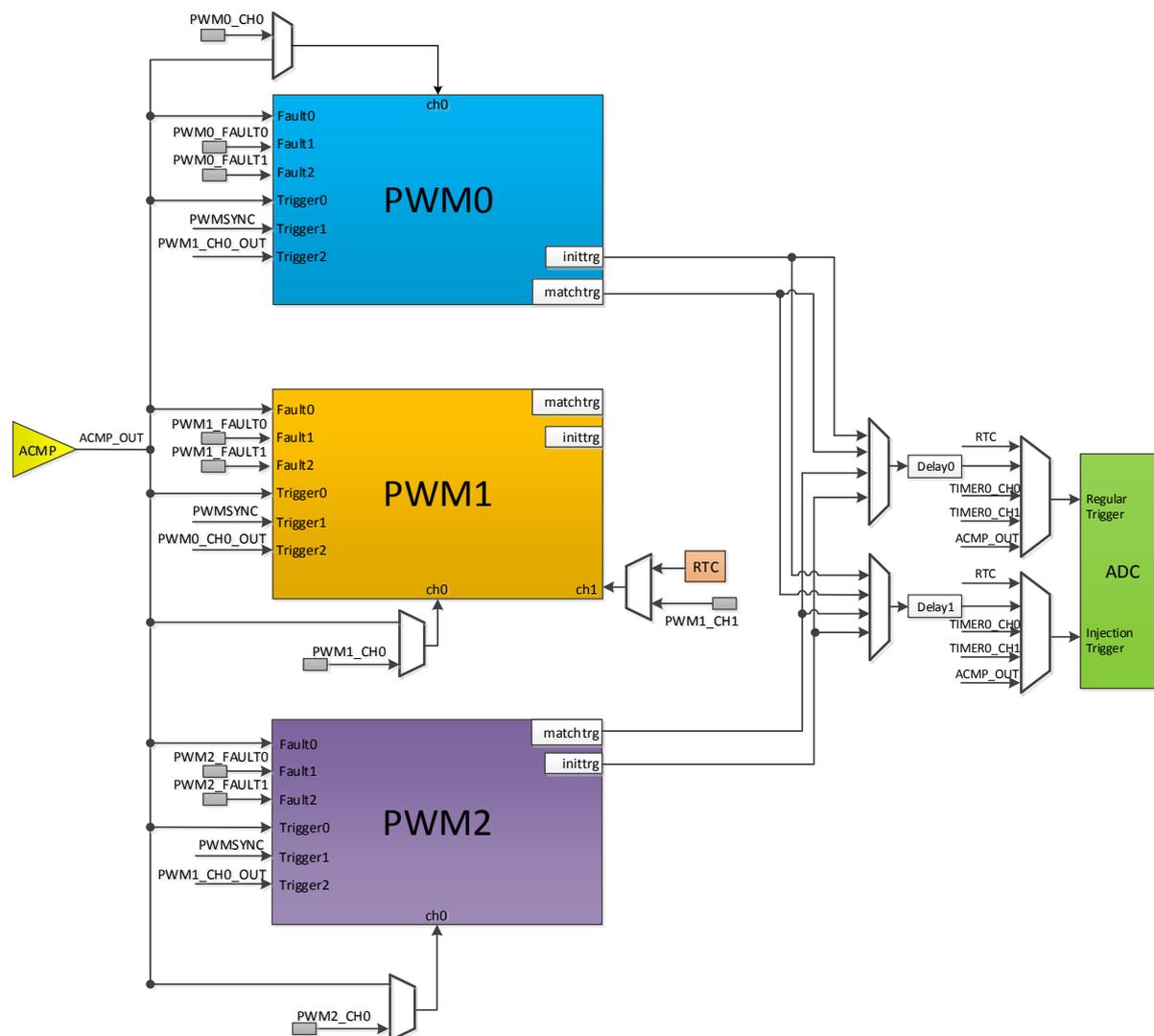


图 13-1 CTU 结构框图

13.4 功能描述

13.4.1 ACMP 输出捕获

CTU_CONFIG0[PWMx_ACIC]位使能 ACMP0 的输出连接到 PWMx_CH0，PWMx_CH0 对应的外部引脚可复用为其他功能，可参考表 14-2 GPIO 复用功能描述，其中 x = 0/1/2。ACMP0 输出内部已连接到 PWDT 输入（用于 Brushless Direct Current Motor 使用），或可用作 PWM 触发/故障输入和 ADC 硬件触发。

13.4.2 RTC 捕获

RTC 溢出信号可通过设置 **CTU_CONFIG0** [RTCC]位由 PWM1_CH1 捕获。该字段置位后，RTC 溢出连接到 PWM1_CH1 以便进行捕获，而 PWM1_CH1 对应的外部引脚可复用为其他功能，可参考表 14-2 GPIO 复用功能描述。

13.4.3 ADC 硬件触发

ADC 模块可以通过硬件触发器来启动转换。通过 **CTU_CONFIG0** [ADHWT0]字段设置规则组硬件触发源，**CTU_CONFIG1**[ADHWT1] 字段设置注入组硬件触发源。当 ADC 硬件触发器选择 PWM 触发器输出时，将使能一个 8 位延迟模块。该逻辑使用 8 位计数器延迟 PWM 的任何触发，计数器的值由 DELAY 指定。该模块的参考时钟是具有 **CTU_CONFIG0** [PSC]指定的可选预分频器的总线时钟。

13.4.4 PWM 软件同步

PWM 包含三个同步输入触发器，CTU 提供其中一个触发器，通过 **CTU_CONFIG0** [PWMSYNC]写 1 来触发软件，将 0 写入该字段不起任何作用，该字段始终读到的是 0。

13.4.5 低功耗模式

表 13-1 CTU 模块低功耗模式

模式	唤醒源	备注
休眠模式	---	模块正常工作
停止模式	---	模块被关闭

13.5 寄存器定义

表 13-2 CTU 寄存器映射

CTU 基地址 = 0x40016000

地址	名称	宽度	描述
CTU 基地址+0x00	CTU_CONFIG0	32	配置 0 寄存器
CTU 基地址+0x04	CTU_CONFIG1	32	配置 1 寄存器

13.5.1 配置寄存器 0(CTU_CONFIG0)

表 13-3 CTU_CONFIG0 寄存器

CTU_CONFIG0

CTU 配置 0 寄存器

Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	DELAY0								DLYACT0	ADHWT0					PSC		
访问	RW								RO	RW					RW		
Reset	0								0	0					0		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称		P W M S Y N C		PW M0_ ACI C	P W M 1_ AC IC	PW M2_ ACI C	R T C C										
访问		R W		RW	R W	RW	R W										
Reset		0		0	0	0	0										

字段	说明
31: 24 DELAY0	<p>规则触发延时单元</p> <p>PWM 初始或匹配触发到 ADC 硬件规则触发的延迟。8 位模数值允许从 0 到 255 的延迟，计数频率由 PSC 决定。这是一个单次计数器，当触发到达时开始计数，当计数器值达到定义的模数值时停止计数。</p>
23 DLYACT0	<p>DELAY0 延迟有效</p> <p>0：延迟无效 1：延迟有效</p> <p>该只读字段指定 PWM 初始或匹配延迟有效时的状态。当 PWM 规则触发到达且延迟计数器正在计数时，该字段置位。否则，该字段会被清除。</p>
22: 20 ADHWT0	<p>ADC 规则组硬件触发源</p> <p>000：RTC 溢出作为 ADC 硬件触发源 001：PWM0 初始化触发，具有 8 位可编程计数器延迟 010：PWM0 匹配触发，具有 8 位可编程计数器延迟 011：PWM2 初始化触发，具有 8 位可编程计数器延迟 100：PWM2 匹配触发，具有 8 位可编程计数器延迟 101：TIMER 通道 0 溢出作为 ADC 硬件触发 110：TIMER 通道 1 溢出作为 ADC 硬件触发 111：ACMP0 输出作为 ADC 硬件触发</p> <p>选择 ADC 硬件规则组触发源，所有触发源都在上升沿启动 ADC 转换。</p>
18: 16 PSC	<p>总线时钟预分频</p> <p>000：总线 1 分频 001：总线 2 分频 010：总线 4 分频</p>

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	DELAY1									DLYACT1						
访问	RW									RW						
Reset	0									0						
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称										ADHWT1						
访问										RW						
Reset										0						

字段	说明
30:23 DELAY1	<p>注入触发延时单元</p> <p>PWM 初始或匹配触发到 ADC 硬件注入触发的延迟。8 位模数值允许从 0 到 255 的延迟，计数频率由 PSC 决定。这是一个单次计数器，当触发到达时开始计数，当计数器值达到定义的模数值时停止计数。</p>
22 DLYACT1	<p>DELAY1 延迟有效</p> <p>0：延迟无效 1：延迟有效</p> <p>该只读字段指定 PWM 初始或匹配延迟有效时的状态。当 PWM 注入触发到达且延迟计数器正在计数时，该字段置位。否则，该字段会被清除。</p>
8: 6 ADHWT1	<p>ADC 注入组硬件触发源</p> <p>000：RTC 溢出作为 ADC 硬件触发源 001：PWM0 初始化触发，具有 8 位可编程计数器延迟 010：PWM0 匹配触发，具有 8 位可编程计数器延迟 011：PWM2 初始化触发，具有 8 位可编程计数器延迟 100：PWM2 匹配触发，具有 8 位可编程计数器延迟 101：TIMER 通道 0 溢出作为 ADC 硬件触发 110：TIMER 通道 1 溢出作为 ADC 硬件触发 111：ACMP0 输出作为 ADC 硬件触发</p> <p>选择 ADC 注入硬件触发源。所有触发源在上升沿开始转换。</p>

14 通用输入/输出 (GPIO)

14.1 简介

通用输入输出 (General-Purpose Input/Output, GPIO) 模块通过 AHB 访问, 以实现最高的引脚性能。

当引脚配置为 GPIO 功能时, 端口配置寄存器 `GPIO_CR` 控制每个引脚的方向。端口输出数据寄存器 `GPIO_ODR` 控制每个引脚输出数据, 也可以通过端口置位/复位寄存器 `GPIO_BSRR`, 端口复位寄存器 `GPIO_BRR` 置位, 控制 GPIO 输出的高低电平。

当引脚配置用于输入功能时, GPIO 输入数据寄存器显示每个引脚上的高低电平 (1 代表高电平, 0 代表低电平)。

MCU I/O 引脚通过多路复用器连接到外设/模块, 多路复用器一次只允许一个外设的复用功能连接到 I/O 引脚。这样, 共享同一个 I/O 引脚的外设之间不会发生冲突。每个 I/O 引脚都有一个多路复用器, 可通过 `GPIO_PINMUX` 寄存器进行配置。当某一个外设/模块的功能需要从当前 IO 转移到另一个 IO 时, 除了新的 IO 需要将复用功能连接到该外设/模块, 原 IO 的复用配置也需要关闭, 否则会导致外设/模块在新 IO 管脚工作异常。

14.2 特性

GPIO 引脚支持如下模式:

- 最多支持 27 个 I/O
- 输出状态: 推挽或开漏 (与 I2C 有关)
- 输出数据来自输出寄存器 `GPIO_ODR` 或 外设 (可选功能输出)
- 每个 I/O 的驱动能力选择
- 输入状态: 浮空, 上拉/下拉, 模拟 (和 ADC/ACMP 有关)
- 输入数据至输入数据寄存器 `GPIO_IDR` 或 外设 (可选功能输入)
- 位置位和复位寄存器 `GPIO_BSRR` 用于按位写入访问 `GPIO_ODR`
- 高灵活度的引脚复用, 允许将 I/O 引脚用作 GPIO 或作为多种外设功能之一
- 可配置上升沿/下降沿/双边沿中断
- 低功耗模式唤醒中断

14.3 结构框图

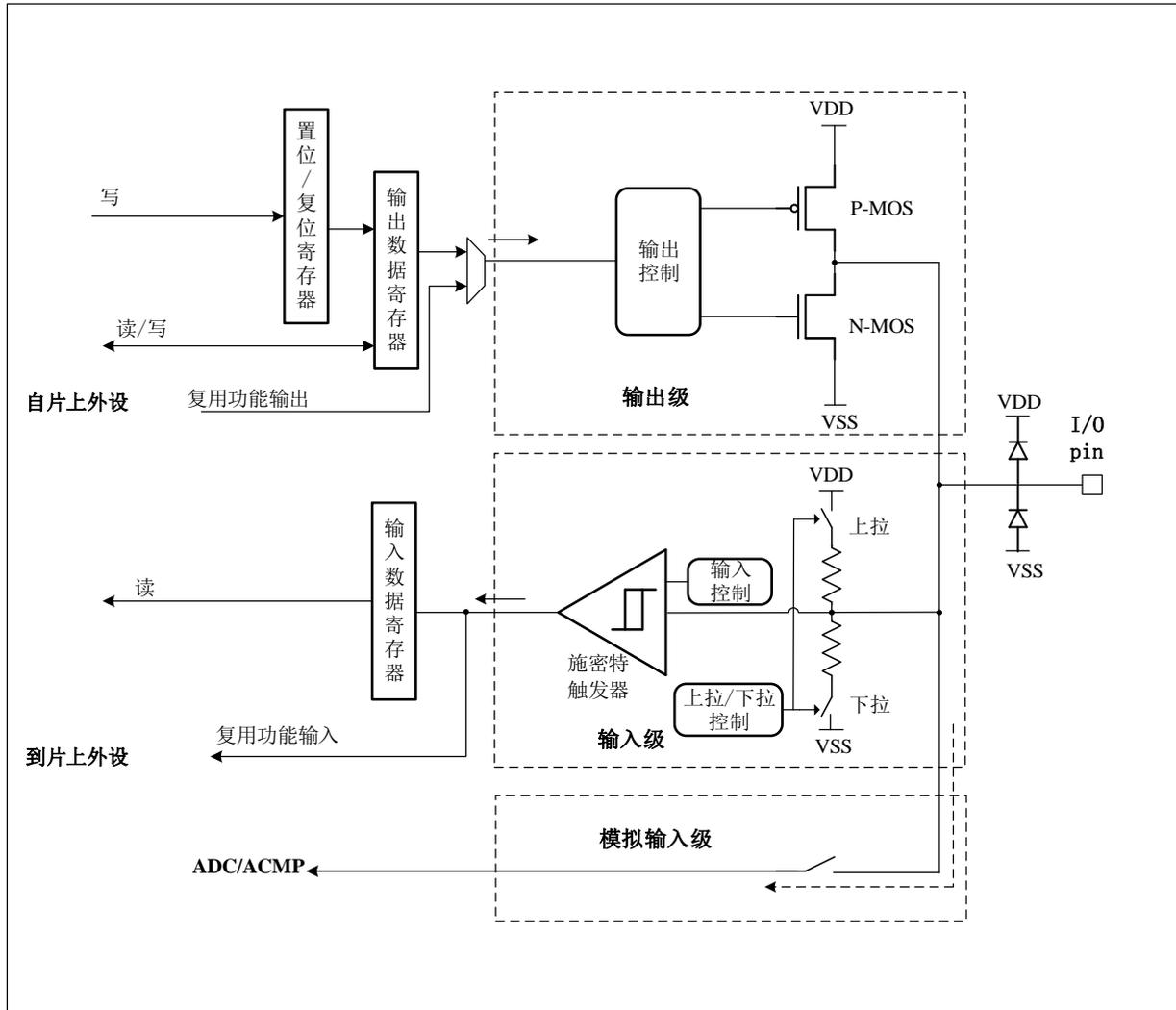


图 14-1 GPIO 结构框图

14.4 功能描述

14.4.1 外部中断

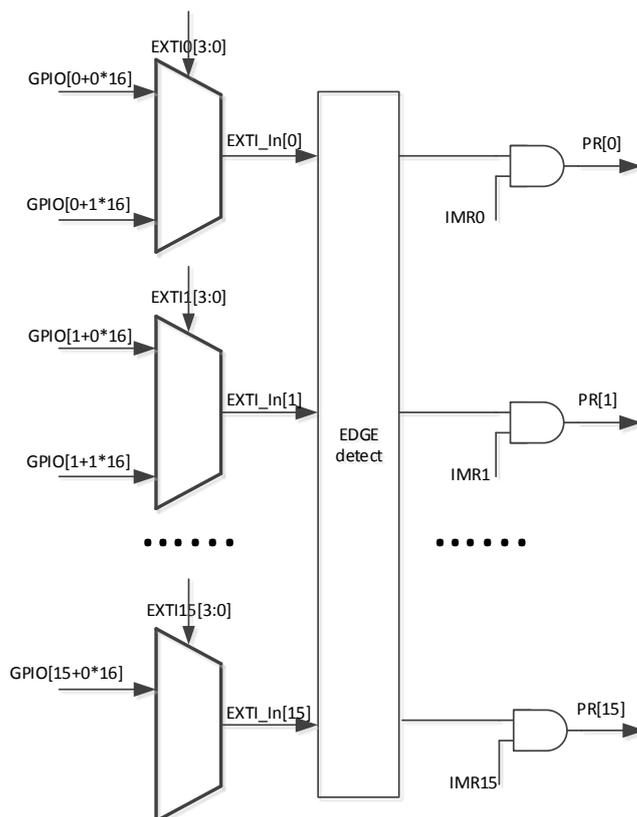


图 14-2 GPIO 外部中断

GPIO 以组的形式划分，每 16 个 IO 构成一组。以图中 $GPIO[y+x*16]$ 为例，‘y’表示某一组 IO 中的第 y 个 PIN 脚， $x*16$ 表示第 x 组 GPIO。如 $GPIO[1+0*16]$ 表示 GPIOA_PIN_1， $GPIO[1+1*16]$ 表示 GPIOB_PIN_1。

外部中断线与中断向量的对应关系：

- 当 $m \leq 2$ 时， $EXTI_In[m]$ 对应着中断向量 $EXTIm_IRQn$
- 当 $3 \leq m \leq 8$ 时， $EXTI_In[m]$ 对应着中断向量 $EXTI3_8_IRQn$
- 当 $9 \leq m \leq 15$ 时， $EXTI_In[m]$ 对应着中断向量 $EXTI9_15_IRQn$

GPIO 外部中断与中断处理函数对应关系如下表示。

表 14-1 GPIO 外部中断和中断处理函数对应关系

GPIO 引脚	中断标志位	中断处理函数
PA0~PB0	EXTI0	EXTI0_IRQHandler
PA1~PB1	EXTI1	EXTI1_IRQHandler
PA2~PB2	EXTI2	EXTI2_IRQHandler
PA3~PB3	EXTI3	EXTI3_8_IRQHandler
PA4~PB4	EXTI4	
PA5~PB5	EXTI5	
PA6~PB6	EXTI6	
PA7~PB7	EXTI7	
PA8~PB8	EXTI8	
PA9~PB9	EXTI9	EXTI9_15_IRQHandler
PA10~PB10	EXTI10	
PA11	EXTI11	
PA12	EXTI12	
PA13	EXTI13	
PA14	EXTI14	
PA15	EXTI15	

14.4.2 复用功能

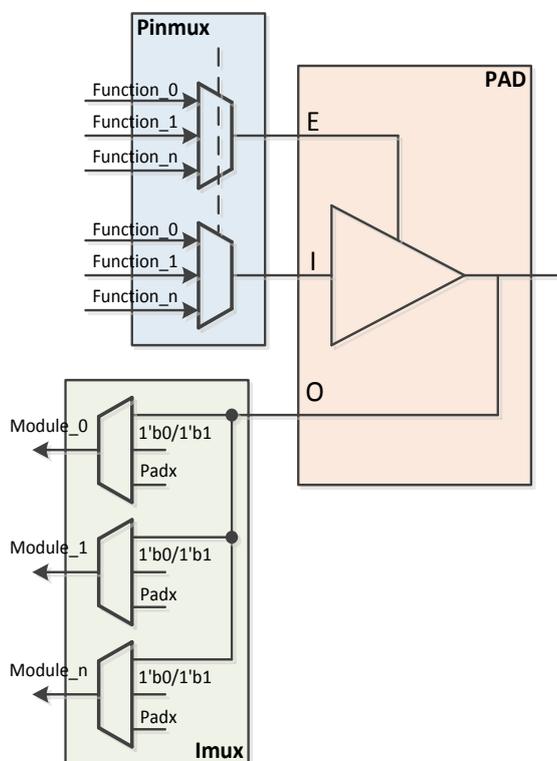


图 14-3 GPIO 复用功能

每个 IO 都有复用功能，如果要开始外设通信，应该先配置 GPIO 复用功能。每个 GPIO 的相应复用功能描述如下。

表 14-2 GPIO 复用功能描述

32 Pin	20 Pin	引脚名称	功能 0	功能 1	功能 2	功能 3	复用功能配置	GPIO (序号)
1		PB0	GPIO	ADC_IN18	PWM1_CH1	PWM2_FLT1	PINMUX1[20:18]	16
2		PB1	GPIO		PWM1_CH0	DAC_OUT	PINMUX1[23:21]	17
3	4	VDD1	VDD1					
4	5	VSS1	VSS1					
5	6	PA12	GPIO	I2C0_SCL	OSC_OUT ¹	PWM0_FLT0	PINMUX1[8:6]	12
6	7	PA15	GPIO	I2C0_SDA	OSC_IN ¹	PWDT0_IN0	PINMUX1[17:15]	15
7	8	PA0	GPIO	PWM2_CH3	VREF-/ADC_IN10	I2C0_SCL	PINMUX0[2:0]	0
8	9	PA1	GPIO	PWM2_CH2	VREF+/ADC_IN9	I2C0_SDA	PINMUX0[5:3]	1
9		PB3	GPIO	PWM2_CH0	ADC_IN13	SPI0_MOSI	PINMUX1[29:27]	19
10	10	PA2	GPIO	PWM2_CH1	ADC_IN8	SPI0_MISO	PINMUX0[8:6]	2
11	11	PA3	GPIO	PWM2_CH0	ADC_IN7	SPI0_SCK	PINMUX0[11:9]	3
12	12	PA4	GPIO	PWM0_CH1	ADC_IN6/ACMP_IN6	UART1_TX	PINMUX0[14:12]	4

32 Pin	20 Pin	引脚名称	功能 0	功能 1	功能 2	功能 3	复用功能配置	GPIO (序号)
13	13	PA5	GPIO	PWM0_CH0	ADC_IN5/ACMP_IN5	UART1_RX	PINMUX0[17:15]	5
14	14	PA6	GPIO	BOOT ¹	GPIO	SPI0_NSS	PINMUX0[20:18]	6
15		PB4	GPIO	PWM2_CH1	ADC_IN12	SPI0_MISO	PINMUX2[2:0]	20
16		PB5	GPIO	PWM0_CH0	ADC_IN11	SPI0_SCK	PINMUX2[5:3]	21
17	15	PA7	GPIO	UART0_TX	ADC_IN4/ACMP_IN4	SPI0_MOSI	PINMUX0[23:21]	7
18	16	PA8	GPIO	UART0_RX	ADC_IN3/ACMP_IN3	SPI0_NSS	PINMUX0[26:24]	8
19	17	PA9	GPIO	PWM2_FLT0	ADC_IN2/ACMP_IN2	RTC_CLKIN	PINMUX0[29:27]	9
20		VSS2	VSS2					
21	18	VDD2	VDD2					
22		PB6	GPIO	ADC_IN16	PWM1_FLT0	PWM0_FLT1	PINMUX2[8:6]	22
23		PB7	GPIO	ADC_IN15	ACMP_IN7	I2C0_SCL	PINMUX2[11:9]	23
24		PB8	GPIO	ADC_IN14	PWDT0_IN2	I2C0_SDA	PINMUX2[14:12]	24
25	19	PA10	GPIO	PWM1_CH1	ADC_IN1/ACMP_IN1	PWDT0_IN2	PINMUX1[2:0]	10
26	20	PA11	GPIO	PWM1_CH0	ADC_IN0/ACMP_IN0	PWDT0_IN1	PINMUX1[5:3]	11
27		PB9	GPIO	PWM2_CH3	I2C0_SCL	UART0_TX/UARTLIN_TX	PINMUX2[17:15]	25
28		PB10	GPIO	PWM2_CH2	I2C0_SDA	UART0_RX/UARTLIN_RX	PINMUX2[20:18]	26
29		PB2	GPIO	NMI_B	PWM1_FLT1	PWDT0_IN0	PINMUX1[26:24]	18
30	1	PA13	GPIO	SWD_CLK ¹		RTC_CLKOUT	PINMUX1[11:9]	13
31	2	RESET_B	RESET_B					
32	3	PA14	GPIO	SWD_DIO ¹	ACMP_OUT	PWM0_CH1	PINMUX1[14:12]	14


注意

1. 除了一些专用引脚外(如标注¹)，所有引脚在第一次上电时默认为**功能 0**。

GPIO 控制寄存器中，除 **GPIO_PINMUX** 外，其他如 **GPIO_CR**，**GPIO_IDR** 等， $x=0,1$ ，分别代表 PA,PB，每个寄存器的低 16 位分别代表 $Px0\sim Px15$ 。

2. 输入/输出配置举例：设置 PB1 为输出模式 $GPIO1_CR[1] = 1$ 。
3. 复用功能配置以 32Pin 举例：如果想要将 PB0 配置为 PWM1_CH1，应该设置 $PMUX1[20:18]=2$ 。
4. NMI_B, RESET_B 的后缀_B 代表的是低电平有效。

14.4.3 低功耗模式

MCU 在 Stop 模式下，可以通过任意一个 IO 产生中断并唤醒 MCU。

14.5 应用说明

14.5.1 外部输入

外部中断/事件控制器由最多 16 个边沿检测器组成，用于生成事件/中断请求。每个输入线可以独立配置，以选择类型（事件或中断）和相应的触发事件（上升沿，下降沿或两者），每条线也可以独立屏蔽。相应中断输入线上的中断请求都记录在外部中断标志暂停寄存器 [GPIO_PR](#) 中。

14.5.2 复用功能

为了缩小封装，引脚通过信号多路复用提供多种可用的功能。复用功能选择寄存器 [GPIO_PINMUX](#) 控制外部引脚上的信号。请参考复用功能选择寄存器 [GPIO_PINMUX](#) 以及 [14.4.2 复用功能](#) 章节。

14.5.3 开漏输出

GPIO PIN 没有单独配置为开漏输出的寄存器，但是可以通过寄存器的组合配置以达到开漏输出的功能。

模拟开漏输出高（需要外接上拉电阻）：

1. 配置该 PIN 的 [GPIO_CR](#) 寄存器对应 BIT 为 0，输入模式；
2. 输出数据寄存器 [GPIO_ODR](#) 对应 BIT 设置为 0；
3. 将 [GPIO_PU](#), [GPIO_PD](#) 寄存器对应的 BIT 配置为 0，不带上拉，也不带下拉。

模拟开漏输出低：

1. 配置该 PIN 的 [GPIO_CR](#) 寄存器对应 BIT 为 1，输出模式；
2. 输出数据寄存器 [GPIO_ODR](#) 对应 BIT 设置为 0；
3. 将 [GPIO_PU](#), [GPIO_PD](#) 寄存器对应的 BIT 配置为 0，不带上拉，也不带下拉。

14.5.4 HIGH_Z 模式

GPIO PIN 配置为 HIGH_Z 模式流程如下：

1. 将 [GPIO_PU](#), [GPIO_PD](#) 寄存器对应的 BIT 配置为 0，不带上拉，也不带下拉；
2. 配置该 PIN 的 [GPIO_CR](#) 寄存器对应 BIT 为 0，关闭该 PIN 的输出功能；
3. 配置该 PIN 的 [GPIO_IES](#) 寄存器对应 BIT 为 0，关闭该 PIN 输入功能；

14.5.5 GPIO 功能

在复位期间和刚刚复位后，GPIO 功能处于活动状态，I/O 端口配置为不带上下拉输入模式，RST 和 ARM 调试接口相关引脚除外。当进入 Stop 模式前，如果 IO 在不带上下拉输入的状态下，需要外部给一个高或低的稳定电平，避免 IO 电平不稳定导致漏电。

用户可以对 GPIO_PINMUX 寄存器进行编程，将 I/O 端口从其他功能更改为 GPIO 功能。

当引脚配置为输出时，写入输出数据寄存器的值将输出到 I/O 引脚。可以在推挽模式或开漏模式（当输出为 0 时，仅激活 N-MOS）下使用输出驱动器。输入数据寄存器 GPIO_IDR 在每个 AHB 时钟周期捕获 I/O 引脚上的数据。

所有 GPIO 引脚都具有弱内部上拉和下拉电阻，可以激活或不激活，具体取决于 GPIO_PU 和 GPIO_PD 寄存器中的值。

14.5.6 编程指南

首先，在复位后，除 RST 和 ARM 调试接口外，所有 I/O 端口都处于 GPIO 输入模式。

其次，软件可以编程复用功能选择寄存器 GPIO_PINMUX 寄存器来映射/重映射 I/O 功能。

作为 GPIO 功能的 I/O，软件可以编程外部中断。当 MCU 处于低功耗模式时，外部中断使用内部 32K 振荡器产生的时钟。

14.6 寄存器定义

表 14-3 GPIO 寄存器映射

GPIOA 基地址：0x20080000

GPIOB 基地址：0x20080030

地址	名称	宽度	描述
GPIOx 基地址 + 0x00	GPIO_CR	32	端口配置寄存器
GPIOx 基地址 + 0x04	GPIO_IDR	32	端口输入数据寄存器
GPIOx 基地址 + 0x08	GPIO_ODR	32	端口输出数据寄存器
GPIOx 基地址 + 0x0C	GPIO_BSRR	32	端口置位/复位寄存器
GPIOx 基地址 + 0x10	GPIO_BRR	32	端口复位寄存器
GPIOx 基地址 + 0x18	GPIO_PD	32	下拉使能寄存器
GPIOx 基地址 + 0x1C	GPIO_PU	32	上拉使能寄存器
GPIOx 基地址 + 0x20	GPIO_E4_E2	32	驱动能力选择寄存器
GPIOx 基地址 + 0x24	GPIO_IES	32	输入使能寄存器
GPIOA 基地址 + 0x140	GPIO_PINMUX	32	复用功能选择寄存器：共有 3 个寄存器
GPIOA 基地址 + 0x144			
GPIOA 基地址 + 0x148			

地址	名称	宽度	描述
GPIOA 基地址 + 0x160	GPIO_PR	32	外部中断标志暂停寄存器
GPIOA 基地址 + 0x164	GPIO_IMR	32	中断掩码寄存器
GPIOA 基地址 + 0x168	GPIO_RTSTR	32	上升沿触发事件配置寄存器
GPIOA 基地址 + 0x16C	GPIO_FTSTR	32	下降沿触发事件配置寄存器
GPIOA 基地址 + 0x170 GPIOA 基地址 + 0x174 GPIOA 基地址 + 0x178 GPIOA 基地址 + 0x17C	GPIO_EXTICR	32	外部中断寄存器：共有 4 个寄存器

【说明】上表中，x=A、B。

14.6.1 端口配置寄存器(GPIO_CR)

表 14-4 GPIO_CR 寄存器

GPIO_CR		端口配置寄存器																Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称	MODE(16*x+15)	MODE(16*x+14)	MODE(16*x+13)	MODE(16*x+12)	MODE(16*x+11)	MODE(16*x+10)	MODE(16*x+9)	MODE(16*x+8)	MODE(16*x+7)	MODE(16*x+6)	MODE(16*x+5)	MODE(16*x+4)	MODE(16*x+3)	MODE(16*x+2)	MODE(16*x+1)	MODE(16*x+0)		
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

字段	说明
[15: 0] MODE	<p>Mode(y): 端口 y 配置位</p> <p>0: 输入(复位默认状态)</p> <p>1: 输出模式</p> <p>这些位由软件写入，以配置 I/O 方向模式。</p>

14.6.2 端口输入数据寄存器 (GPIO_IDR)

表 14-5 GPIO_IDR 寄存器

GPIO_IDR		端口输入数据寄存器																Reset: 0x00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称		IDR(16*x+5)	IDR(16*x+4)	IDR(16*x+3)	IDR(16*x+2)	IDR(16*x+1)	IDR(16*x+0)	IDR(16*x+9)	IDR(16*x+8)	IDR(16*x+7)	IDR(16*x+6)	IDR(16*x+5)	IDR(16*x+4)	IDR(16*x+3)	IDR(16*x+2)	IDR(16*x+1)	IDR(16*x+0)	
访问		RO																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

字段	说明
[15: 0] IDR	IDR(y): 端口 y 输入数据 这些位只读，包含相应 I/O 端口的输入值。

14.6.3 端口输出数据寄存器(GPIO_ODR)

表 14-6 GPIO_ODR 寄存器

GPIO_ODR		端口输出数据寄存器																Reset: 0x00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称		ODR(16*x+15)	ODR(16*x+14)	ODR(16*x+13)	ODR(16*x+12)	ODR(16*x+11)	ODR(16*x+10)	ODR(16*x+9)	ODR(16*x+8)	ODR(16*x+7)	ODR(16*x+6)	ODR(16*x+5)	ODR(16*x+4)	ODR(16*x+3)	ODR(16*x+2)	ODR(16*x+1)	ODR(16*x+0)	
访问		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

字段	说明
[15: 0] ODR	ODR(y): 端口(y) 输出数据 这些位可以通过软件读写。 注意: 对于 PIN 置位/复位, 可以通过写入 GPIOx_BSRR 寄存器 (x = A, B) 和 GPIOx_BRR 寄存器 (x = A, B) 来单独设置和复位 ODR 位。

14.6.5 端口复位寄存器(GPIO_BRR)

表 14-8 GPIO_BRR 寄存器

GPIO_BRR 端口复位寄存器 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	BR(16* $x+15$)	BR(16* $x+14$)	BR(16* $x+13$)	BR(16* $x+12$)	BR(16* $x+11$)	BR(16* $x+10$)	BR(16* $x+9$)	BR(16* $x+8$)	BR(16* $x+7$)	BR(16* $x+6$)	BR(16* $x+5$)	BR(16* $x+4$)	BR(16* $x+3$)	BR(16* $x+2$)	BR(16* $x+1$)	BR(16* $x+0$)
访问	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[15: 0] BR	BR(y): Port(y) 复位位 y 0: 对应的 ODRy 位无操作 1: 复位对应的 ODRy 位
这些位是只写的，对这些位的读取返回 0x0000。	

14.6.6 下拉使能寄存器(GPIO_PD)

表 14-9 GPIO_PD 寄存器

GPIO_PD 下拉使能寄存器 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	PD(16* $x+15$)	PD(16* $x+14$)	PD(16* $x+13$)	PD(16* $x+12$)	PD(16* $x+11$)	PD(16* $x+10$)	PD(16* $x+9$)	PD(16* $x+8$)	PD(16* $x+7$)	PD(16* $x+6$)	PD(16* $x+5$)	PD(16* $x+4$)	PD(16* $x+3$)	PD(16* $x+2$)	PD(16* $x+1$)	PD(16* $x+0$)
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[15: 0] PD	PD (y): 下拉使能 0: 禁用下拉 1: 使能下拉

字段	说明
----	----

这些位可以通过软件进行读写。

注意：上拉和下拉不支持同时使能。

14.6.7 上拉使能寄存器(GPIO_PU)

表 14-10 GPIO_PU 寄存器

GPIO_PU		上拉使能寄存器														Reset: 0x00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																	
访问																	
Reset																	
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		PU(16* x+15)	PU(16* x+14)	PU(16* x+13)	PU(16* x+12)	PU(16* x+11)	PU(16* x+10)	PU(16* x+9)	PU(16* x+8)	PU(16* x+7)	PU(16* x+6)	PU(16* x+5)	PU(16* x+4)	PU(16* x+3)	PU(16* x+2)	PU(16* x+1)	PU(16* x+0)
访问		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
----	----

[15: 0] **PU (y): 上拉使能**

PU

0: 禁用上拉

1: 使能上拉

这些位可以通过软件进行读写。

注意：上拉和下拉不支持同时使能。

14.6.8 驱动能力选择寄存器 (GPIO_E4_E2)

表 14-11 GPIO_E4_E2 寄存器

GPIO_E4_E2		驱动能力选择寄存器														Reset: 0x00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称		E4_E2(16*x+ 15)		E4_E2(16*x+ 14)		E4_E2(16*x+ 13)		E4_E2(16*x+ 12)		E4_E2(16*x+ 11)		E4_E2(16*x+ 10)		E4_E2(16*x+ 9)		E4_E2(16*x+ 8)	
访问		RW		RW		RW		RW		RW		RW		RW		RW	
Reset		0		0		0		0		0		0		0		0	
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		E4_E2(16*x+ 7)		E4_E2(16*x+ 6)		E4_E2(16*x+ 5)		E4_E2(16*x+ 4)		E4_E2(16*x+ 3)		E4_E2(16*x+ 2)		E4_E2(16*x+ 1)		E4_E2(16*x+ 0)	
访问		RW		RW		RW		RW		RW		RW		RW		RW	
Reset		0		0		0		0		0		0		0		0	

字段	说明
[1: 0]	E4_E2 (y): 驱动能力选择
[3: 2]	
[5: 4]	00: 4mA
.....	01: 8mA
E4_E2	10: 12mA
	11: 16mA

这些位可以通过软件进行读写。

14.6.9 输入使能寄存器(GPIO_IES)

表 14-12 GPIO_IES 寄存器

GPIO_IES		输入使能寄存器														Reset: 0x0000FFFF
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	IES(16*x+15)	IES(16*x+14)	IES(16*x+13)	IES(16*x+12)	IES(16*x+11)	IES(16*x+10)	IES(16*x+9)	IES(16*x+8)	IES(16*x+7)	IES(16*x+6)	IES(16*x+5)	IES(16*x+4)	IES(16*x+3)	IES(16*x+2)	IES(16*x+1)	IES(16*x+0)
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

字段	说明
[15: 0]	IES (y): 输入使能
IES	0: 禁用输入功能 1: 使能输入功能

这些位可以通过软件进行读写。

注意: GPIOB IES 寄存器 bit11 需要保持为“1”，清零会导致系统异常。

14.6.10 复用功能选择寄存器(GPIO_PINMUX)

表 14-13 GPIO_PINMUX 寄存器

GPIO_PINMUX		复用功能选择寄存器										Reset: 0x00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称			PINMUXx[29: 27]			PINMUXx[26: 24]			PINMUXx[23: 21]			PINMUXx[20: 18]			PINMUXx[17: 15]	
访问			RW			RW			RW			RW			RW	
Reset			0			0			0			0			0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称			PINMUXx[14: 12]		PINMUXx[11: 9]			PINMUXx[8: 6]			PINMUXx[5: 3]			PINMUXx[2: 0]		
访问			RW		RW			RW			RW			RW		
Reset			0		0			0			0			0		

字段	说明
[2: 0]	PINMUXx (y): 复用功能
[5: 3]	
[8: 6]	000: 功能 0
.....	001: 功能 1
PINMUXx	010: 功能 2
	011: 功能 3

这些位由软件写入，以配置复用功能 I/O。

注: GPIO_PINMUX0 寄存器默认值为 0x00040000;

GPIO_PINMUX1 寄存器默认值为 0x00011280;

GPIO_PINMUX2 寄存器默认值为 0x00000000

14.6.11 外部中断标志暂停寄存器(GPIO_PR)

表 14-14 GPIO_PR 寄存器

GPIO_PR		外部中断标志暂停寄存器															Reset: 0x00000000	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称	PR(15)	PR(14)	PR(13)	PR(12)	PR(11)	PR(10)	PR(9)	PR(8)	PR(7)	PR(6)	PR(5)	PR(4)	PR(3)	PR(2)	PR(1)	PR(0)		
访问	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C	R/W1 C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

字段	说明
[15: 0] PR	PR (y): 外部中断标志暂停位 0: 没有发生触发请求 1: 发生所选的触发请求
当所选边沿事件达到外部中断线时, 该位置 1。通过向该位写 1 来清除该位。	

14.6.12 中断掩码寄存器(GPIO_IMR)

表 14-15 GPIO_IMR 寄存器

GPIO_IMR		中断掩码寄存器														Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	IMR(15)	IMR(14)	IMR(13)	IMR(12)	IMR(11)	IMR(10)	IMR(9)	IMR(8)	IMR(7)	IMR(6)	IMR(5)	IMR(4)	IMR(3)	IMR(2)	IMR(1)	IMR(0)
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[15: 0] IMR	IMR (y): y 输入线的中断掩码 0: 来自 y 输入线的中断请求被屏蔽 1: 来自 y 输入线的中断请求未被屏蔽

14.6.13 上升沿触发事件配置寄存器(GPIO_RTSTR)

表 14-16 GPIO_RTSTR 寄存器

GPIO_RTSTR		上升沿触发事件配置寄存器														Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	RTSR(15)	RTSR(14)	RTSR(13)	RTSR(12)	RTSR(11)	RTSR(10)	RTSR(9)	RTSR(8)	RTSR(7)	RTSR(6)	RTSR(5)	RTSR(4)	RTSR(3)	RTSR(2)	RTSR(1)	RTSR(0)
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

字段	说明
[15: 0]	RTSR (y): y 输入线的上升沿触发事件配置
RTSR	0: 输入线 y 上升沿触发禁用 (对于事件和中断) 1: 输入线 y 上升沿触发使能 (对于事件和中断)

14.6.14 下降沿触发事件配置寄存器(GPIO_FTISR)

表 14-17 GPIO_FTISR 寄存器

GPIO_FTISR		下降沿触发事件配置寄存器																Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称	FTSR(15)	FTSR(14)	FTSR(13)	FTSR(12)	FTSR(11)	FTSR(10)	FTSR(9)	FTSR(8)	FTSR(7)	FTSR(6)	FTSR(5)	FTSR(4)	FTSR(3)	FTSR(2)	FTSR(1)	FTSR(0)		
访问	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

字段	说明
[15: 0]	FTSR (y): y 输入线的下降沿触发事件配置
FTSR	0: 输入线 y 下降沿触发禁用 (对于事件和中断) 1: 输入线 y 下降沿触发使能 (对于事件和中断)

14.6.15 外部中断寄存器(GPIO_EXTICR)

表 14-18 GPIO_EXTICR 寄存器

GPIO_EXTICR		外部中断寄存器																Reset: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称	EXTI(4*x+3)				EXTI(4*x+2)				EXTI(4*x+1)				EXTI(4*x+0)					
访问	RW				RW				RW				RW					
Reset	0				0				0				0					

字段	说明
[3: 0]	EXTI (y): EXTI y 配置
[7: 4]	

字段	说明
[11: 8]	0000: PA[x] 引脚
.....	0001: PB[x] 引脚
EXTI	
这些位由软件写入，用于选择 EXTI(y)外部中断源输入。	

15 I2C 总线模块 (I2C)

15.1 简介

I2C 总线是一种简单、双向二线制同步串行总线，通过时钟线（SCL）与数据线（SDA）进行数据传输。SCL 是由主机驱动产生的时钟信号，SDA 是双向数据信号，既可由主设备产生，也可由从设备产生。

15.2 特性

- 主从一体模式
- 标准 100kHz、快速 400kHz、快速+ 1MHz 模式
- 支持 7bit 范围地址
- 支持 10bit 扩展地址
- 支持从机 Stretch
- 支持从机低功耗模式唤醒
- 支持从机监测功能
- 支持多主机仲裁
- 主机仲裁丢失切换为从机
- 可编程毛刺滤波器
- 总线开始/停止信号检测
- 软件控制应答信号
- 地址匹配中断
- 字节传输中断
- 无应答中断
- 发送/接收溢出中断
- SDA/SCL 低超时中断
- 支持时钟同步
- 支持 SMBus 2.0 – 警告地址
- 支持 SDA/SCL 低超时检测

15.3 结构框图

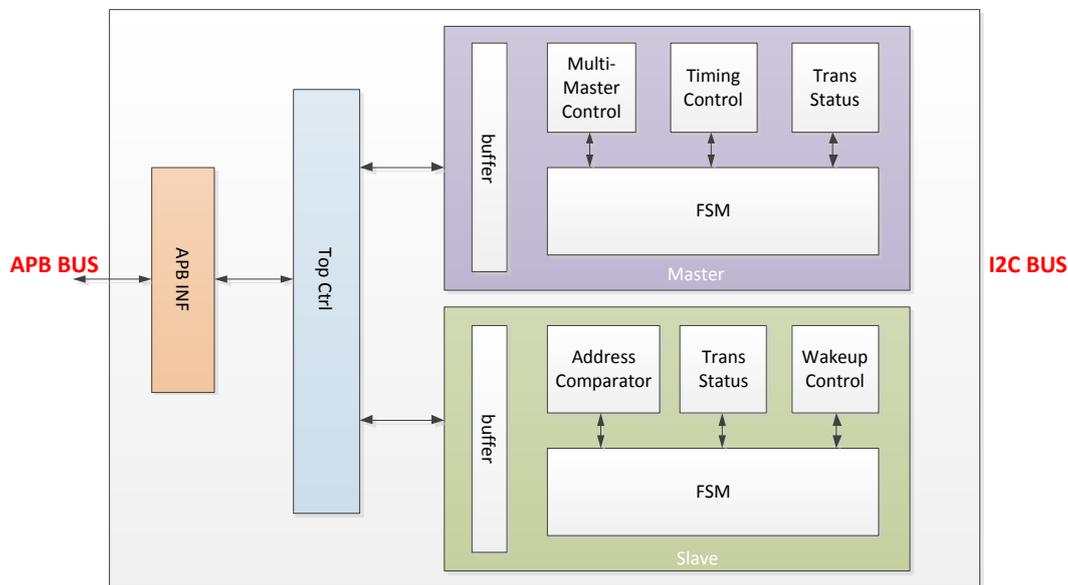


图 15-1 I2C 结构框图

15.3.1 I2C 信号组成

I2C 通讯以起始信号开始，以停止信号结束。当 SCL 为高电平时，SDA 线上高电平到低电平的跳变定义了 START 条件。当 SCL 为高电平时，SDA 线上低电平到高电平的跳变定义了 STOP 条件。

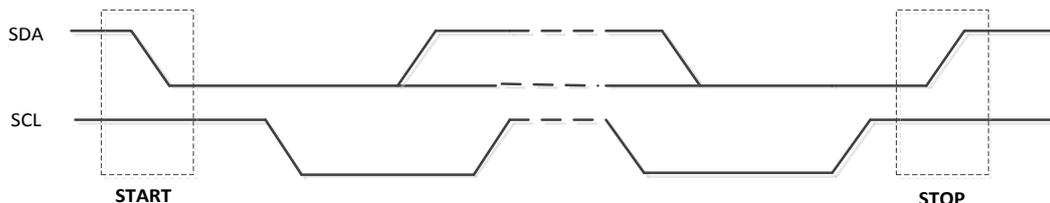


图 15-2 START 和 STOP 条件

每帧数据由 9 位组成，8 位数据（MSB 在前）和 1 位应答信号，传输次数不受限制。

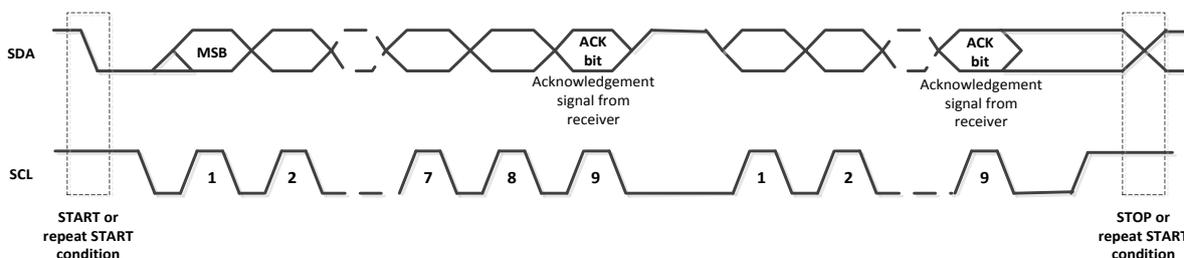


图 15-3 数据传输格式

起始信号后的第一个数据是地址字节，地址字节后的继续传输的数据都是数据字节。

7bit 地址模式下，I2C 协议规定地址字节的高 7 位是从机的地址，最低位为 0，表示写入从机，最低位为 1，表示读取从机。

10bit 地址模式下，地址由两个字节组成。

写从机时，I2C 协议规定：第一个字节发送 11110XX0，高 5 位固定为 11110，bit2，bit1 是 10bit 地址中的高 2 位，bit0 是方向位，值为 0，表示方向为写；第二个字节是 10bit 地址的低 8 位。

读从机时，I2C 规定：先发写的地址（两个字节），再发读的地址（读的地址只需要发送一个字节）。具体流程如下：先发写的地址，第一个字节发送 11110XX0，高 5 位固定为 11110，bit2、bit1 是 10bit 地址的高 2 位，bit0 是方向位，值为 0，表示方向位写，第二个字节是 10bit 地址的低 8 位；再发读的地址，发送 11110XX1，bit2、bit1 是 10bit 地址中的高 2 位，bit0 是方向位，值为 1，表示方向位读。

15.3.2 波特率组成

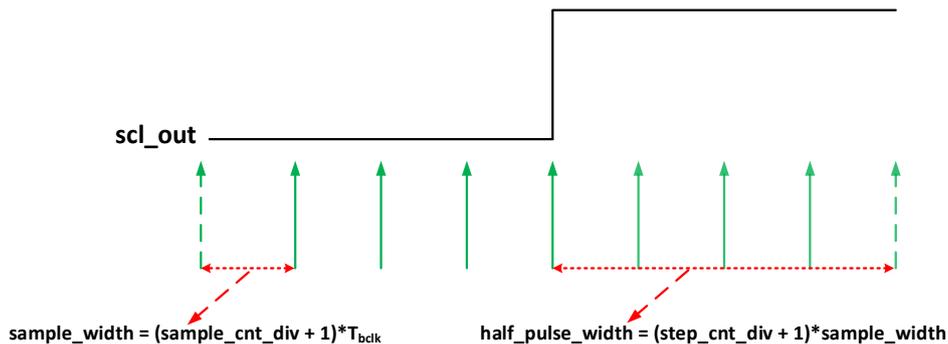


图 15-4 波特率生成

波特率： $f_{SCL} = f_{bclk} / (((SAMPLE_CNT_DIV + 1) * (STEP_CNT_DIV + 1)) * 2)$

f_{bclk} 是 APB 总线时钟频率。

15.3.3 数据流程

作为发送器，数据被写入一个 8 位的发送缓冲区。然后由波特率控制器加载到发送移位寄存器。MSB 在前，发送器会在每个字节的第 9 个 SCL 采样应答位。

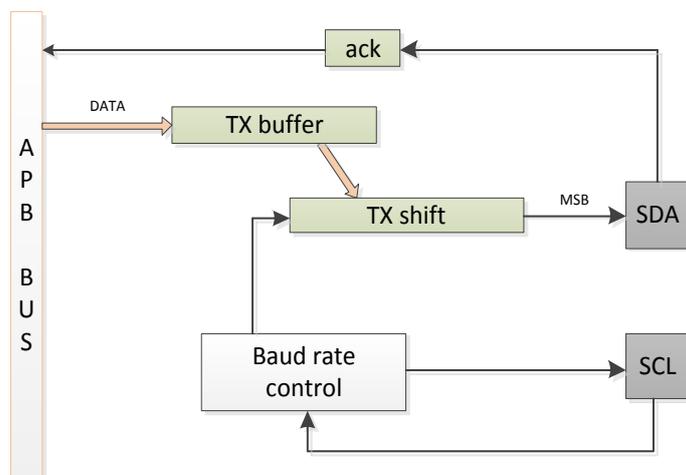


图 15-5 发送器数据流程

对于接收器，接收移位寄存接收每个比特位并依次传入移位寄存器，数据在每第 8 个 SCL 之后将被存储到 8 位接收缓冲器中，MSB 在前。应答位在第 9 个 SCL 驱动到 SDA 线上。

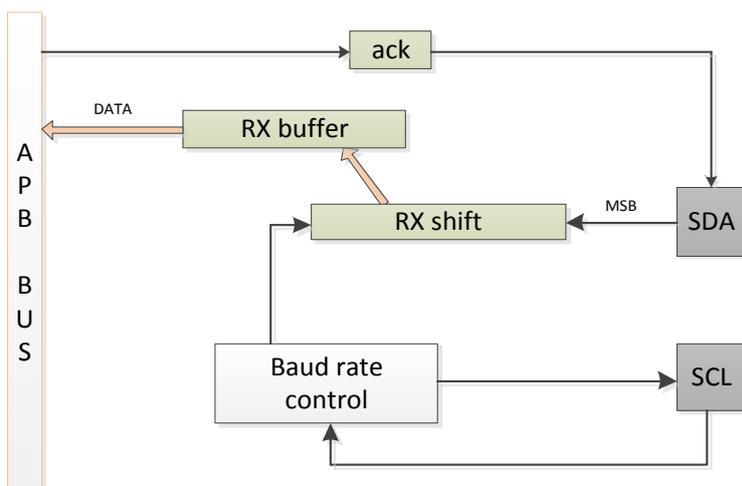


图 15-6 接收器数据流程

15.4 功能描述

I2C 模块支持标准、快速与快速+ 通信模式，I2C 外设同一时刻只能当做主机或从机模式使用。一旦配置为从机模式，除非模块复位，则会一直处于从机模式下。

15.4.1 主机模式

在主机模式下，主机发送起始信号后，每发送完一字节数据，BND 标志位都会置位。

主机可发送重复起始信号，在不发送 STOP 信号前提下，重新对从机寻址并写入或读取数据，如图 15-7 所示。

主机支持时钟同步，在多主从机通讯或者有从机 stretch 总线时，能与其它主从机的时钟进行同步：当主机 SCL 输出高时，不会开始它的高电平时间计数，而是等待总线的 SCL 被所有主机拉高后开始计数，这样实现了多主机之间的时钟同步。此时实际的 SCL 时钟频率会变慢，因为 SCL 拉高是通过外部上拉电阻充电拉高，此时的频率取决于外部总线电容和上拉电阻的取值。

支持主机仲裁，在多主机通讯情况下，I2C 模块对总线上传输的数据进行逐 bit 比较。当主机发送 1 而检测到总线数据为 0 时，仲裁丢失，置位仲裁丢失标志，可产生中断请求，并将自己切换为从机模式，因为此时其它主机可能在寻址自己。该功能的前提条件是时钟同步功能先打开。

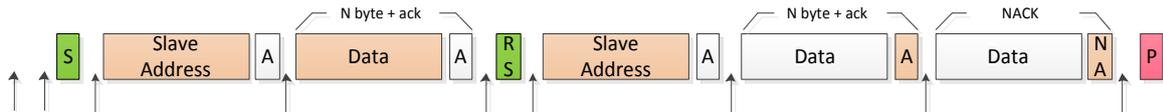


图 15-7 主机组合模式

15.4.2 从机模式

在从机模式下，从机支持 7bit 地址、10bit 地址、7bit 范围地址、广播地址四种地址匹配模式。

从机在以上四种模式下地址得到匹配后，SAMF 标志位置位，SRW 标志位根据主机的写/读信号置 0/1。地址匹配成功后，主机在地址字节后传输的每一个字节数据，从机的 BND 标志位都会置位，当主机在读取从机时，发送了 NACK 信号，后续若主机继续传输，从机不会产生 BND 标志，需要重新被寻址，才能与从机通讯。

从机具备 stretch 功能，使能后，从机收到一字节数据而软件没有读取，或者主机读取时从机还没有将数据准备好，从机将主动拉低 SCL 总线，防止主机发出下一帧数据的时钟信号，当然主机需要使能时钟同步功能，以等待从机释放 SCL 总线。

从机具有四种状态标志：

- **发送缓存区空标志 TXEF**，当从机发送寄存器未载入发送数据或一字节发送完成时，此标志置位，可以产生中断请求。
- **发送缓存区溢出标志 TXUF**，当从机发送完一字节数据，但没有写入下一字节数据，而此时主机再次发完一字节时钟信号后，此标志置位，可以产生中断请求。
- **接收缓存区满标志 RXFF**，当从机数据寄存器收到一字节数据后，此标志置位，可以产生中断请求。
- **接收缓存区溢出标志 RXOF**，当从机接收完一字节数据后，若软件没有及时读取数据，而此时主机再次发完一字节时钟信号后，此标志置位，可以产生中断请求。

15.4.3 SMBus

I2C 模块支持系统管理总线（SMBus）2.0 规范。SMBus 是一种简单的单端两线总线，通常用于低带宽通信。

15.4.3.1 SMBus – SCL/SDA 低超时检测

I2C 用作主机或从机时，支持 SCL/SDA 低超时检测。超时时间可软件配置。默认检测 SCL，可选择是否检测 SDA。若 I2C_CTRL2[PLTIE] = 1，PLTF 置位时将产生中断请求。

15.4.3.2 SMBus –警告地址响应

I2C 作为从机可响应 SMBus 的警告地址 0001100X。从机在地址得到匹配后，I2C_STATUS0[SAMF] 标志置位时 I2C_STATUS1[SARF]标志同时置位。

15.4.4 中断请求

I2C 模块共有 11 个中断。

表 15-1 I2C 中断汇总

中断源	标志	本地使能	全局使能
完成 1 个字节的传输	BND		IICIE
从机地址匹配	SAMF		IICIE
仲裁丢失	ARBLOST		IICIE
总线起始信号检测	START	SSIE	IICIE
总线停止信号检测	STOP	SSIE	IICIE
RX 缓冲区溢出	RXOF	RXOFIE	IICIE
TX 缓冲区溢出	TXUF	TXUFIE	IICIE
RX 缓冲区满	RXFF	RXFIE	IICIE
TX 缓冲区空	TXEF	TXEIE	IICIE
获得应答	RACK	NACKIE	IICIE
SDA/SCL 低超时	PLTF	PLTIE	IICIE

15.4.5 从机低功耗唤醒

MCU 在低功耗模式下，若使能唤醒功能，则地址匹配时，将在第 9 个 SCL 处产生 ACK 低电平信号，并产生唤醒信号唤醒 MCU，紧跟在地址后的数据 I2C 都不回 ACK，I2C 重新初始化或重新收到起始信号后可以正常接收数据。

15.5 应用说明

15.5.1 数据传输

写 STARTSTOP [START]为 '1' 发送 START 信号到 I2C 总线。传输结束写 STARTSTOP [STOP]为 '1' 发送 STOP 信号到 I2C 总线。

主机发送时，TX 控制置位 1，对数据寄存器写入后，波特率控制器自动开始向 I2C 总线传输数据，一字节传输完成后，BND 标志置位，表明可以再次写入数据，写入数据或向 BND 位写 1 可清除 BND 标志。

主机接收时，TX 控制位置 0，对数据寄存器的读操作，会触发波特率控制器自动向 I2C 总线发出时钟信号，并根据 RACK 控制应答，收到的数据载入数据寄存器。BND 标志位置位，对数据寄存器进行读取或向 BND 位写 1 可清除 BND 标志。

从机发送时，先将待发送数据写入数据寄存器，当收到主机读取的完整时钟信号后，数据被传出，TXEF 标志位置位，BND 标志位置位，对数据寄存器的写入操作可清除 TXEF 和 BND 标志位，或向 BND 位写 1 也可清除 BND 标志。

从机接收时，主机发送一字节完整数据后，数据被载入从机数据寄存器，RXFF 标志位置位，BND 标志位置位，读取数据寄存器可清除 RXFF 和 BND 标志位，或向 BND 位写 1 也可清除 BND 标志。

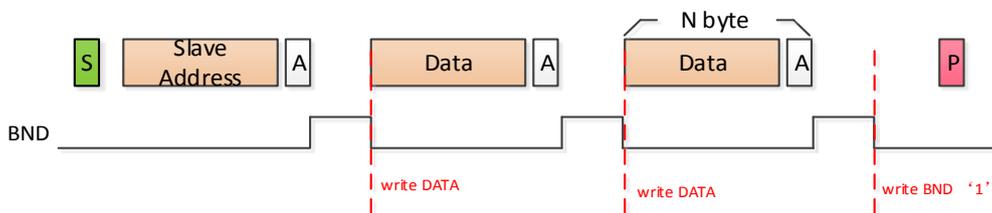


图 15-8 主机写从机模式的 BND 序列

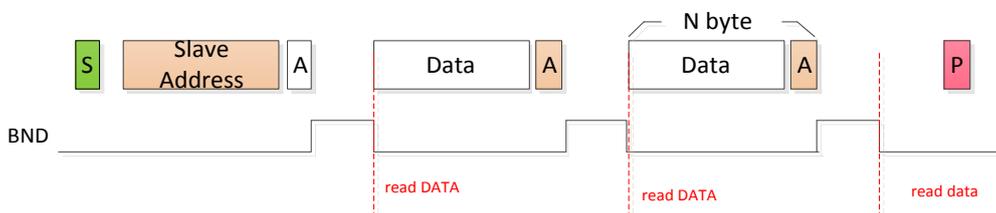


图 15-9 主机读从机模式的 BND 序列

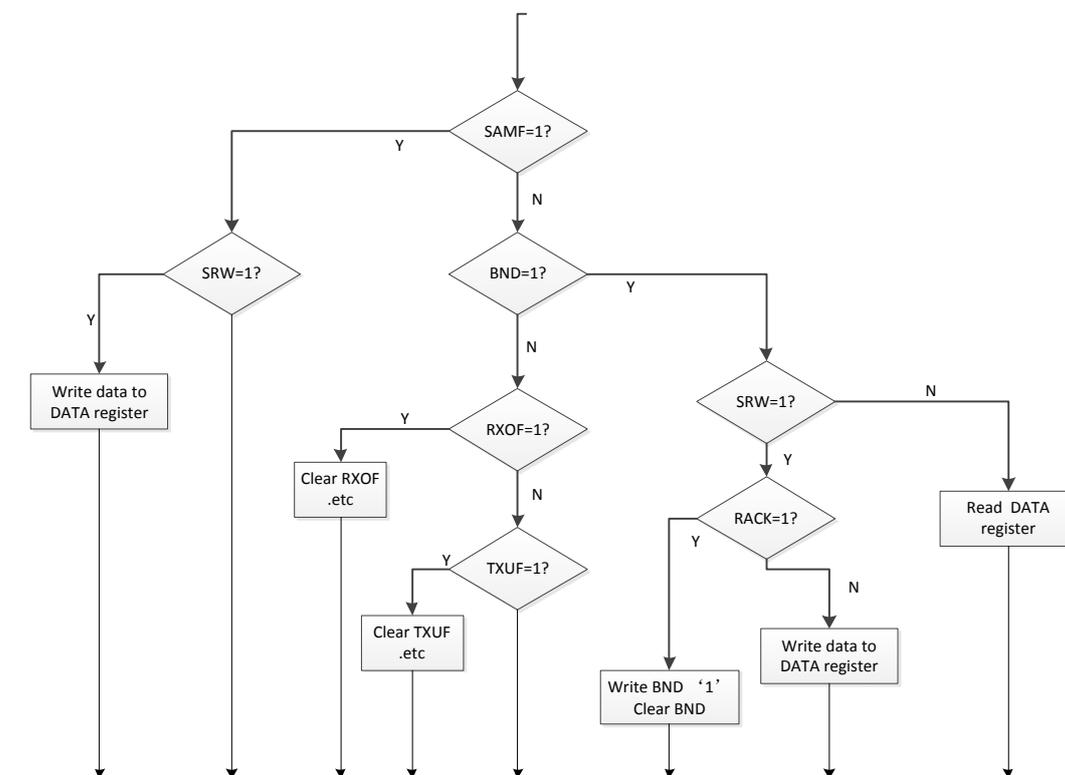


图 15-10 典型 I2C 从机中断程序

15.5.2 应答控制

I2C 模块通过软件方式控制应答信号。

在主机接收数据时，TACK 位的值决定下一字节读取完后，在第 9 个时钟信号上的应答值。

在从机接收数据时，TACK 位的值决定下一字节读取完后，在第 9 个时钟信号上的应答值。

15.6 寄存器定义

表 15-2 I2C 寄存器映射

I2C0 基地址: 0x4000E000

地址	名称	宽度	描述
I2Cx 基地址+0x00	I2C_ADDR0	32	地址寄存器 0
I2Cx 基地址+0x04	I2C_ADDR1	32	地址寄存器 1
I2Cx 基地址+0x08	I2C_SAMPLE_CNT	32	波特率配置寄存器 0
I2Cx 基地址+0x0C	I2C_STEP_CNT	32	波特率配置寄存器 1
I2Cx 基地址+0x10	I2C_CTRL0	32	控制寄存器 0
I2Cx 基地址+0x14	I2C_CTRL1	32	控制寄存器 1

I2Cx 基地址+0x18	I2C_CTRL2	32	控制寄存器 2
I2Cx 基地址+0x1C	I2C_CTRL3	32	控制寄存器 3
I2Cx 基地址+0x20	I2C_STATUS0	32	状态寄存器 0
I2Cx 基地址+0x24	I2C_STATUS1	32	状态寄存器 1
I2Cx 基地址+0x28	I2C_DGLCFG	32	毛刺滤波配置寄存器
I2Cx 基地址+0x2C	I2C_DATA	32	数据寄存器
I2Cx 基地址+0x30	I2C_STARTSTOP	32	起始与停止信号控制寄存器

【说明】上表中，x=0。

15.6.1 地址寄存器 0(I2C_ADDR0)

表 15-3 I2C_ADDR0 寄存器

I2C_ADDR0		地址寄存器 0								Reset: 0x00000FE
位	31: 8	7	6	5	4	3	2	1	0	
名称	AD [6: 0]									
访问	RW									
Reset	0x7F									

字段	说明
7: 1	I2C 从机 7Bit 地址
AD[6: 0]	在从机模式下设置 7 位地址或 10 位地址中的低 7 位。

15.6.2 地址寄存器 1(I2C_ADDR1)

表 15-4 I2C_ADDR1 寄存器

I2C_ADDR1		地址寄存器 1											Reset: 0x00007F7		
位	31: 13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称		RMEN		RAD										AD[9: 7]	
访问		RW		RW										RW	
Reset		0		0x7F										0x07	

字段	说明
12	从机模式时，范围地址使能位
RMEN	1: 使能 0: 禁用
	范围地址使能控制位。

字段	说明
10: 4 RAD	从机模式时，范围地址值 从机 7bit 范围地址值，范围地址使能时，从机接收到的地址大于 AD[6: 0]且小于等于 RAD[6:0]时，将得到匹配。
2: 0 AD [9: 7]	I2C 从机 10Bit 地址 在从机模式下设置 10 位地址的高 3 位。

15.6.3 波特率配置寄存器 0(I2C_SAMPLE_CNT)

表 15-5 I2C_SAMPLE_CNT 寄存器

I2C_SAMPLE_CNT		波特率配置寄存器 0								Reset: 0x00000004
位	31: 8	7	6	5	4	3	2	1	0	
名称		SAMPLE_CNT								
访问		RW								
Reset		0x04								

字段	说明
7: 0 SAMPLE_CNT	调整每个采样点的宽度 $\text{采样周期} = (\text{SAMPLE_CNT} + 1) * \text{总线时钟周期}$

15.6.4 波特率寄存器 1(I2C_STEP_CNT)

表 15-6 I2C_STEP_CNT 寄存器

I2C_STEP_CNT		波特率配置寄存器 1								Reset: 0x00000004
位	31: 8	7	6	5	4	3	2	1	0	
名称		STEP_CNT								
访问		RW								
Reset		0x04								

字段	说明
7: 0 STEP_CNT	调整半脉冲宽度 $\text{半波特率周期} = (\text{STEP_CNT} + 1) * \text{采样宽度}$ 注意：STEP_CNT 最小为 3。

15.6.5 控制寄存器 0(I2C_CTRL0)

表 15-7 I2C_CTRL0 寄存器

I2C_CTRL0 控制寄存器 0 Reset: 0x00000000

位	31: 8	7	6	5	4	3	2	1	0
名称		IICEN	IICIE	MSTR	TX	TACK	WUEN		
访问		RW	RW	RW	RW	RW	RW		
Reset		0	0	0	0	0	0		

字段	说明
7 IICEN	I2C 模块使能 1: 使能 0: 禁用
6 IICIE	I2C 全局中断使能 1: 使能 I2C 模块全局中断 0: 禁用
5 MSTR	I2C 模式选择 1: 主机模式 0: 从机模式 注意：如果 ARB_LOST 置位时，该位会自动清零。
4 TX	传输方向选择 1: 发送(TX) 0: 接收(RX) 主要作用在主机模式下的读取：TX=0，读取 DATA 寄存器会触发读时钟；TX=1，读取 DATA 寄存器不会触发读时钟。
3 TACK	应答控制 1: 向接下来的接收字节上的总线发送 NACK 信号 0: 向接下来的接收字节上的总线发送 ACK 信号 指定在主机或从机的数据应答周期过程中驱动到 SDA 的值。
2 WUEN	唤醒功能使能 1: 在低功耗模式下使能唤醒功能 0: 禁用唤醒功能 当出现从机地址匹配时，I2C 模块可将 MCU 从低功耗模式唤醒。

字段	说明
	注意：当 I2C 模块处于从机模式并进入低功耗模式时，发生 7bit、10bit (ADEXT=1)，或通用广播地址 (GCAEN=1) 匹配，或范围地址匹配 (RMEN=1) 时，I2C 模块将唤醒 MCU。

15.6.6 控制寄存器 1(I2C_CTRL1)

表 15-8 I2C_CTRL1 寄存器

I2C_CTRL1		控制寄存器 1						Reset: 0x00000000	
位	31: 8	7	6	5	4	3	2	1	0
名称		GCAEN	ADEXT		SYNCEN	ARBEN		SAEN	STREN
访问		RW	RW		RW	RW		RW	RW
Reset		0	0		0	0		0	0

字段	说明
7 GCAEN	<p>从机通用广播地址使能</p> <p>1: 使能 0: 禁用</p>
6 ADEXT	<p>从机地址扩展使能</p> <p>1: 10 位地址模式 0: 7 位地址模式</p>
4 SYNCEN	<p>主机 SCL 同步使能</p> <p>1: 使能 0: 禁用</p> <p>使能主机的 SCL 同步功能。</p>
3 ARBEN	<p>主机仲裁使能</p> <p>1: 使能 0: 禁用</p> <p>使能主机仲裁功能。 注意：如果在多主机系统中使用 I2C，则多主机功能必须同时设置 SYNCEN 和 ARBEN。 如果 I2C 在单主机系统中使用，且从机具有 SCL 拉伸功能，则可以只设置 SYNCEN。</p>
1 SAEN	<p>从机 SMBus 警告地址使能</p> <p>1: 使能 0: 禁用</p>
0	<p>从机 SCL 拉伸使能</p>

字段	说明
----	----

STREN

1: 使能

0: 禁用

使能此位，从机将在每个字节第 9 个 SCL 下降沿后将 SCL 拉低。

注意：

在 SAMF=1 后，如果 SRW=1，TXEF=1 将产生 scl stretch，如果 SRW=0，RXFF=1 将产生 scl stretch。因此当使能 STREN 且从机为发送器时(TX)，在每个字节（包括地址字节）的第 9 个 SCL 下降沿后，从机会拉低 SCL，直到写 DATA 寄存器。类似地，当从机为接收器时(RX),在每个字节的第 9 个 SCL 下降沿后，从机会拉低 SCL，直到读取数据寄存器。

在从机模式下，当使能 GCAEN 或 MNTEN 时，从机不会拉低 SCL。

15.6.7 控制寄存器 2(I2C_CTRL2)

表 15-9 I2C_CTRL2 寄存器

I2C_CTRL2		控制寄存器 2					Reset: 0x00000000		
位	31: 8	7	6	5	4	3	2	1	0
名称		RXOFIE	TXUFIE	RXFIE	TXEMIE		PLTIE	NACKIE	MNTEN
访问		RW	RW	RW	RW		RW	RW	RW
Reset		0	0	0	0		0	0	0

字段	说明
----	----

7 从机接收缓存区溢出中断使能

RXOFIE

1: 使能

0: 禁用

6 从机发送缓存溢出中断使能

TXUFIE

1: 使能

0: 禁用

5 从机接收缓存满中断使能

RXFIE

1: 使能

0: 禁用

4 从机发送缓存空中断使能

TXEMIE

1: 使能

0: 禁用

2 SDA/SCL 低超时中断使能

PLTIE

字段	说明
	1: 使能 0: 禁用
1 NACKIE	NACK 中断使能
	1: 使能 0: 禁用
0 MNTEN	从机监测功能使能
	1: 使能 0: 禁用

15.6.8 控制寄存器 3(I2C_CTRL3)

表 15-10 I2C_CTRL3 寄存器

I2C_CTRL3		控制寄存器 3			Reset: 0x00000000	
位	31: 16	15: 3	2	1	0	
名称		PINLOW	TIMECFG			
访问		RW	RW			
Reset		0	0			

字段	说明
15:3 PINLOW	SDA/SCL 低超时时间 设置 SDA 或 SCL 低电平检测超时时间。当 SCL 或 SDA 低电平时间超过 PINLOW * 256 个时钟周期，会置位 PLTF。当 PINLOW 值为 0 时，则禁用 SDA/SCL 低超时检测功能。
2 TIMECFG	超时检测类型 1: SDA/SCL 0: SCL

15.6.9 状态寄存器 0(I2C_STATUS0)

表 15-11 I2C_STATUS0 寄存器

I2C_STATUS0		状态 寄存器 0							Reset: 0x00000008	
位	31: 8	7	6	5	4	3	2	1	0	
名称		BND	SAMF	BUSY	ARBLOST	READY	SRW		RACK	
访问		R/W1C	R/W1C	RO	R/W1C	RO	RO		R/W1C	
Reset		0	0	0	0	1	0		0	

字段	说明
7 BND	<p>字节结束标志</p> <p>1: 一个字节传输完成(包括 ACK 位, 共 9 个 SCL) 0: 传输进行中, 一个字节传输未结束</p> <p>复位后, <i>BND</i> 为 '0'。<i>BND</i> 仅在总线上 START 和 STOP 信号之间的数据传输期间设置。<i>BND</i> 会在每 9 个 SCL 下降沿之后设置。</p> <p>在 Master 模式下, 发送数据时, 软件写 DATA 寄存器会清零此位, 并发出 DATA; 读取数据时, 一字节传输完成, 软件读取 DATA 寄存器会清零此位, 并发出下一字节数据时钟。</p> <p>在 Slave 模式下, 地址匹配后置位 SAMF, 此时 BND 标志不会置位, BND 在地址匹配后的每次数据传输置位。具体为: 地址匹配后, 主机写数据, 从机置位 BND, 当从机主动 NACK 后, 主机若继续写数据, 从机不再置位 BND; 主机读取数据, 从机置位 BND, 若收到主机的 NACK, 主机继续读数据, 从机不再置位 BND。</p> <p>注意: 当 I2C 为从机模式且 MNTEN=1, 当 BND 为 '1' 时, 读取 DATA 寄存器将清零此位。写 '1' 也可以清零此位。</p>
6 SAMF	<p>从机地址匹配标志</p> <p>1: 地址匹配 0: 地址不匹配</p> <p>注意: 该位在满足如下条件之一时置位:</p> <ol style="list-style-type: none"> 1. 7 位地址匹配 2. 10 位地址, 第 1st 字节和第 2nd 字节匹配。并且在第 2nd 字节匹配后置位。 3. 通用广播地址匹配。 4. 范围地址匹配 5. 警告地址匹配 <p>写 '1' 清零此位。</p>
5 BUSY	<p>总线忙</p> <p>1: 总线忙 0: 总线空闲</p> <p>指示总线的状态, 对从机模式和主机模式都有效。当硬件在总线上检测到 START 信号时, 该位置位。检测到 STOP 信号时, 该位清零。</p>
4 ARBLOST	<p>仲裁丢失标志</p> <p>1: 仲裁丢失 0: 仲裁未丢失</p> <p>注意:</p>

字段	说明
	写 '1' 清零此位。 当仲裁丢失时，I2C 主机将切换至从机模式，MSTR 位由硬件清零。
3 READY	准备好接收新的命令 1: 内部硬件准备好接收软件的新命令 0: 内部硬件未准备好 该位指示内部硬件状态，仅对主机模式有效。 注意： 此位对主机模式有效，在主机产生 START/STOP 信号时使用。当 I2C 模块处于主机模式时，写 STARTSTOP 寄存器产生 START/STOP 信号后必须等待 READY 位为 1。
2 SRW	从机读取/写入方向 1: 从机发送(TX)，主机从从机读取 0: 从机接收(RX)，主机向从机写入
0 RACK	接收应答 该字段只在发送方向有效，主机发送或从机发送。 1: 未检测到应答信号 0: 在总线上完成一个字节的传送后，接收到应答信号 注意： 若 NACKIE=1, RACK=1 将会产生中断请求，写 1 清零此位。

15.6.10 状态寄存器 1(I2C_STATUS1)

表 15-12 I2C_STATUS1 寄存器

I2C_STATUS1		状态寄存器 1								Reset: 0x00000081
位	31: 8	7	6	5	4	3	2	1	0	
名称		IDLE	SARF	PLTF	GCMF	RXOF	TXUF	RXFF	TXEF	
访问		RO	R/W1C							
Reset		1	0	0	0	0	0	0	1	

字段	说明
7 IDLE	I2C 硬件状态 1: 空闲 0: 非空闲
6 SARF	从机 SMBus 警告地址匹配标志 1: 使能警告地址并匹配 0: 禁用警告地址或未匹配

字段	说明
5 PLTF	<p>写“1”清零此位。</p> <p>SDA/SCL 低超时标志</p> <p>1: SDA/SCL 低超时 0: 未超时</p> <p>注意: 当 SDA/SCL 低电平维持时间超过设置的 PINLOW 将置位该标志。 只要 SDA/SCL 一直为低电平该标志不能被清除。 在 I2C 发送起始信号前，必须清除该标志。 写“1”清零此位。</p>
4 GCMF	<p>通用广播地址匹配标志</p> <p>1: 发生通用广播地址匹配 0: 未发生通用广播地址匹配</p> <p>注意: I2C 的 START 或 STOP 信号会自动清除该标志 写“1”清零此位。</p>
3 RXOF	<p>从机 RX 缓冲区溢出标志</p> <p>1: RX 缓冲区溢出 0: 未溢出</p> <p>注意: 当 RX 缓冲区溢出时，新接收的数据不会存储在 RX 缓冲区中。 写‘1’清零此位。</p>
2 TXUF	<p>从机 TX 缓冲区溢出标志</p> <p>1: TX 缓冲区溢出 0: 未溢出</p> <p>注意: 当 TX 缓冲区溢出时，再次发送时发送缓冲区的最后一个 DATA。 写‘1’清零此位。</p>
1 RXFF	<p>从机 RX 缓冲区满标志</p> <p>1: RX 缓冲区满 0: 未满</p> <p>注意: 读取 DATA 寄存器将清零此位 写“1”清零此位。</p>
0 TXEF	<p>从机 TX 缓冲区空标志</p>

字段	说明
	1: TX 缓冲区为空 0: 非空
	注意: 写 DATA 寄存器会清零此位 写“1” 清零此位。

15.6.11 毛刺滤波配置寄存器(I2C_DGLCFG)

表 15-13 I2C_DGLCFG 寄存器

I2C_DGLCFG		毛刺滤波配置寄存器					Reset: 0x00000000		
位	31: 8	7	6	5	4	3	2	1	0
名称		DGLEN	STOPF	SSIE	STARTF	DGL_CNT			
访问		RW	R/W1C	RW	R/W1C	RW			
Reset		0	0	0	0	0			

字段	说明
7 DGLEN	毛刺滤波器使能 1: 使能 0: 禁用
6 STOPF	总线 STOP 标志 1: 在 I2C 总线上检测到 STOP 信号 0: 在 I2C 总线上未检测到 STOP 信号 当 I2C 总线上检测到 STOP 信号时, 由硬件置位此位。 注意: 写“1” 清零此位。
5 SSIE	总线 STOP 或 START 中断使能 1: 使能 STRAT 或 STOP 检测中断 0: 禁用 注意: 写‘1’ 清除 STARTF 或 STOPF 标志。
4 STARTF	总线 START 标志 1: 在 I2C 总线上检测到 START 0: 未检测到 START 当 I2C 总线上检测到 START 信号时, 由硬件置位此位。 注意: 写‘1’ 清零此位
3: 0 DGL_CNT	毛刺滤波器计数器。 0h: 无滤波器 / 旁路

字段	说明
	1-Fh: 对宽度最多为 1-15 个总线时钟周期的毛刺进行过滤
	选择毛刺滤波宽度。对于毛刺宽度小于或等于所设置的滤波宽度，滤波器不会允许该毛刺通过。

15.6.12 数据寄存器(I2C_DATA)

表 15-14 I2C_DATA 寄存器

I2C_DATA		数据寄存器								Reset: 0x000001FF
位	31: 9	8	7	6	5	4	3	2	1	0
名称		MAK	DATA							
访问		RO	RW							
Reset		1	0xFF							

字段	说明
8 MAK	<p>从机监控功能 ACK 位</p> <p>对于从机监测器，该字段为 I2C 总线的 ACK 位。</p> <p>注意：MAK='1', NACK MAK='0', ACK 对于监测器，DATA 是 I2C 总线上传送的数据，第 8 位为 ACK 位。</p>
7: 0 DATA	<p>数据</p> <p>在主机传送(TX)模式下，当向该寄存器中写入数据时，会启动数据传输。首先发送最高有效位。在主机接收(RX)模式下，读该寄存器会触发主机发出读时钟。</p> <p>注意：当从主机接收模式转换时，在读取 DATA 寄存器之前需要先切换 I2C 模式，以防意外启动主机数据接收传输流程。</p>

15.6.13 起始与停止信号控制寄存器(I2C_STARTSTOP)

表 15-15 I2C_STARTSTOP 寄存器

I2C_STARTSTOP		起始停止信号控制寄存器		Reset: 0x00000000
位	31: 2	1	0	
名称		STOP	START	
访问		RW	RW	
Reset		0	0	

字段	说明
1	主机发送停止信号

字段	说明
STOP	写“1”，主机会发送 STOP 信号 读取此位会始终返回“0”
0	主机发送起始信号
START	写“1”，主机将会发送 START 或 RESTART 信号 读取此位会始终返回“0”

16 串行外设接口 (SPI)

16.1 简介

串行外设接口 SPI (Serial Peripheral Interface) 总线系统是一种同步串行外设接口, 支持串行、同步、全双工协议。SPI 模块包含主机和从机并以 4 线方式进行通信。

图 16-1 给出了 SPI 主机和 SPI 从机之间的连接示例, 如下图所示。

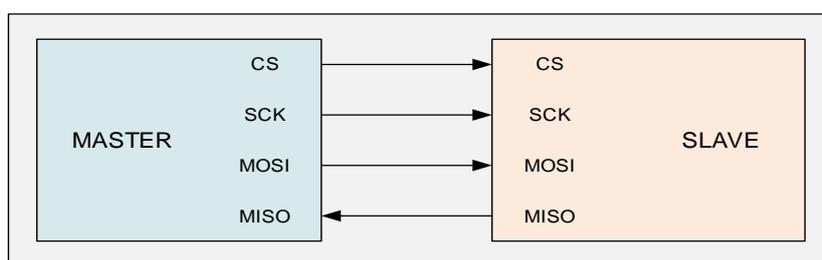


图 16-1 SPI 系统连接

16.2 特性

- 主机模式或从机模式操作
 - 作为主机, 波特率最高支持 8 M
 - 作为从机, 波特率最高支持 8 M
- 全双工模式
- 主机可编程波特率
- 串行时钟相位和极性选择
- 可配置连续或不连续 CS (从机选择) 输出
- 带 CPU 中断功能的模式错误标志位
- 可供选择的最高有效位 (MSB) 优先或 最低有效位 (LSB) 优先移位
- 可配置的 CS 建立时间, 保持时间和空闲时间
- 可配置的 SCK 高和低周期
- 4-16 位传输帧格式选择
- 带中断功能的 TX 缓冲区下溢及 RX 缓冲区溢出标志位
- 从机支持停止 (Stop) 模式唤醒功能

16.3 结构框图

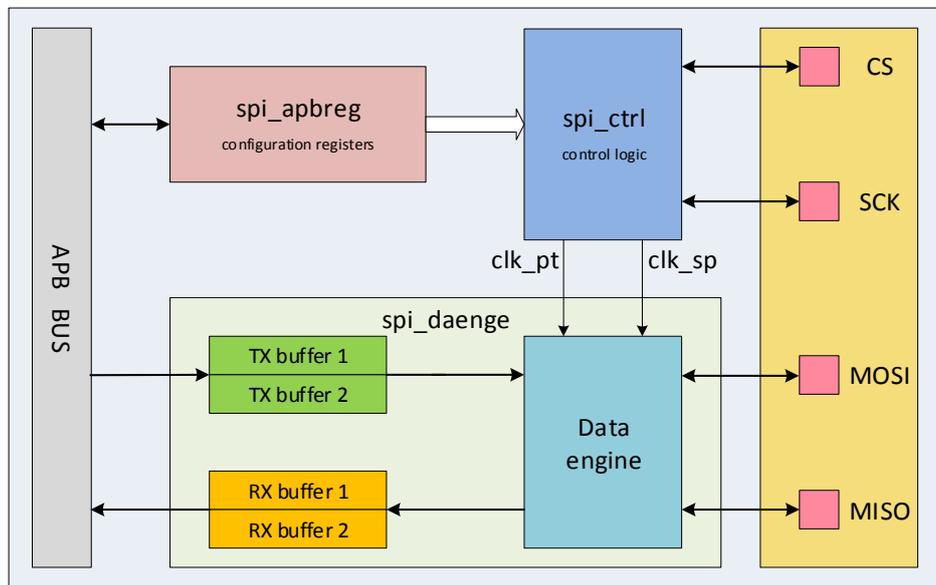


图 16-2 SPI 结构框图

16.4 功能描述

16.4.1 数据流 & 算法

对于主机模式，数据被写入一个 16 位 TX 缓冲区，然后参考波特率控制逻辑，加载到移位寄存器。由 TMSBF 控制输出数据是高位先发还是低位先发。在 FRMSIZE 指定的 SCK 周期数之后，移位寄存器从 MISO 引脚移入数据。接收的数据存储在 16 位 RX 缓冲区中。

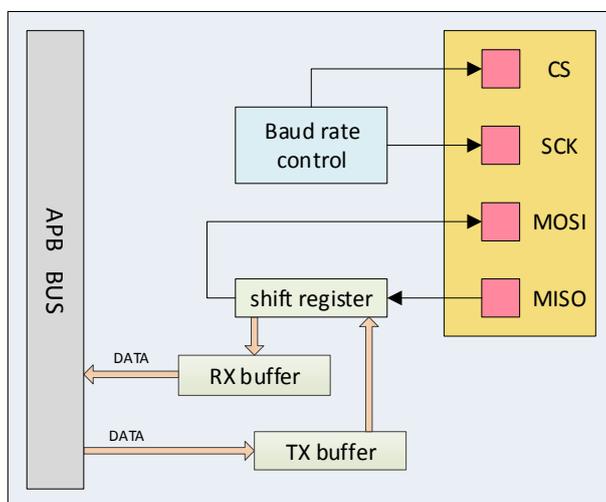


图 16-3 主机数据流

对于从机模式，数据流类似于主机模式。但 CS 引脚是从机选择输入，SCK 是来自主机的 SPI 时钟输入。在发生数据传输之前，从机 SPI 的 CS 引脚必须为低电平。MOSI 是从机数据输入引脚，MISO 是数据输出引脚。

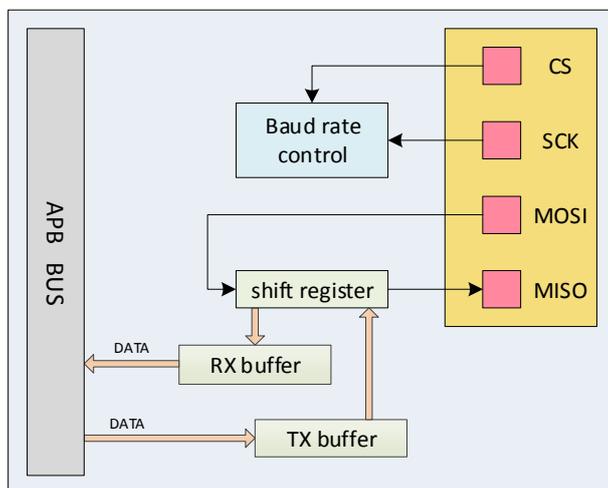


图 16-4 从机数据流

16.4.2 输入输出时序

16.4.2.1 CPHA = 0 传输格式

SCK 线上的第一个边沿用于将从机的第一个数据位计时到主机，将主机的第一个数据位计时到从机。在某些外设中，只要选择了从机，从机的数据输出引脚就会提供从机数据的第一位。在这种格式中，在 CS 变低之后，第一个 SCK 边沿发出半个周期。

半个 SCK 周期后，第二个边沿出现在 SCK 线上。当第二个边沿出现时，先前从串行数据输入引脚锁存的值被移入移位寄存器。

在第二个边沿之后，SPI 主机的下一位从主机的串行数据输出引脚传输到从机的串行输入引脚。该过程在 SCK 线上总共持续 16 个边沿，数据被锁存在奇数边沿上并在偶数边缘上移位。

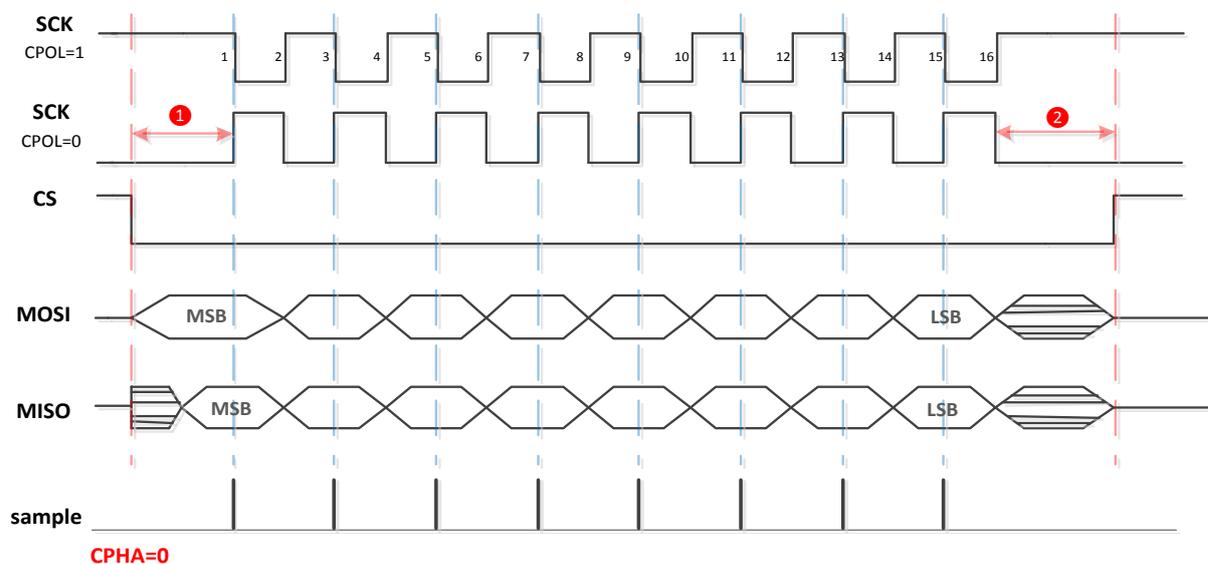


图 16-5 CPHA=0 传输格式

16.4.2.2 CPHA = 1 传输格式

某些外设的第一个数据位在数据输出引脚变得可用之前需要第一个 SCK 边沿，第二个边沿将数据计入系统。

SCK 的第一个边沿在半个 SCK 时钟周期同步延迟之后立即发生。第一个边沿命令从机将其第一个数据位传输到主机的串行数据输入引脚。半个 SCK 周期后，第二个边沿出现在 SCK 引脚上。这是主机和从机的锁存边沿。

当第三个边沿出现时，先前从串行数据输入引脚锁存的值被移入移位寄存器。在此边沿之后，主机数据的下一位从主机的串行数据输出引脚耦合到从机上的串行输入引脚。

该过程在 SCK 线上总共持续 16 个边沿，数据被锁存在偶数边沿上并且移位发生在奇数边沿上。

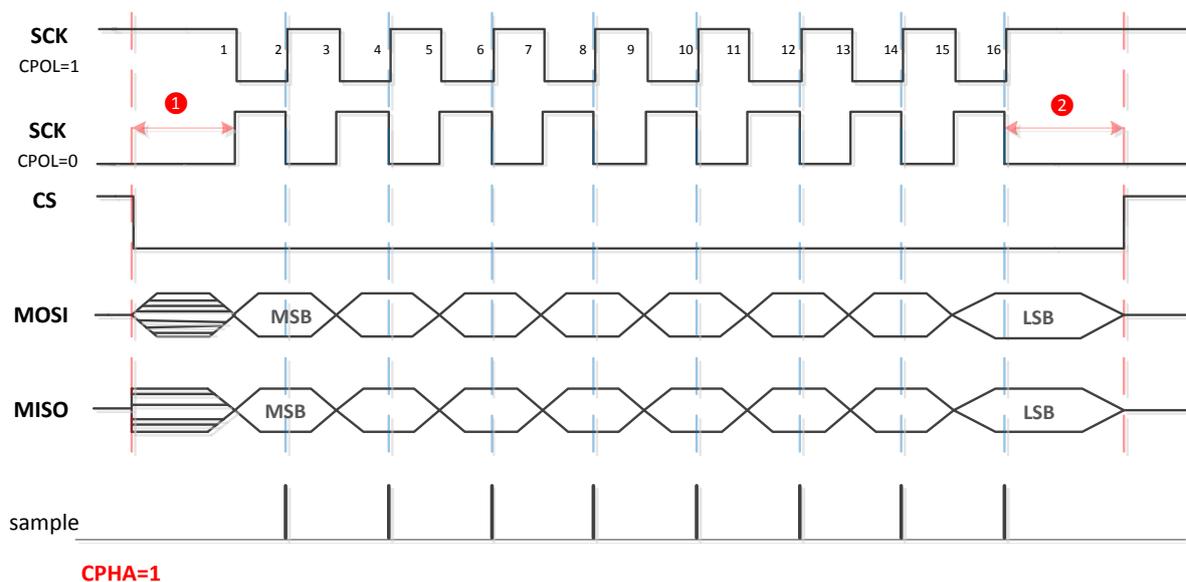


图 16-6 CPHA=1 传输格式

16.4.3 主机 SCK 输出时序设置

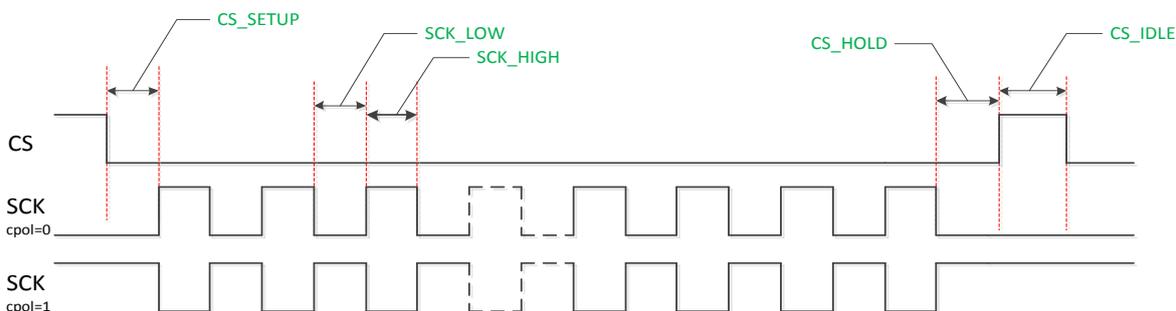


图 16-7 波特率生成

波特率： $f_{SCL} = f_{blk} / (SCK_LOW + 1 + SCK_HIGH + 1)$, f_{blk} 是 APB 总线时钟频率。

16.4.4 主机模式故障检测

如果在 SPI 主机初始化传输之前 CS 引脚已被驱动为低电平，则置位 MODEF。主模式故障检测功能仅在 MSTR=1, MODFEN=1, CSOE=1 时有效。

图 16-8 的红色部分为主机检测故障的时间段。

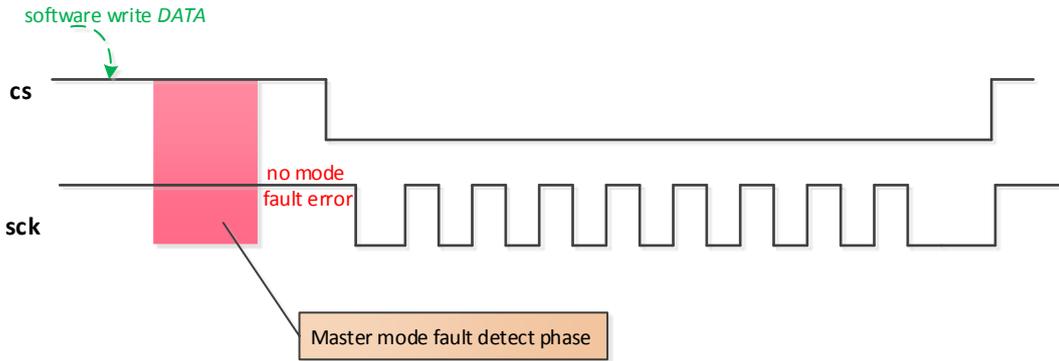


图 16-8 在模式故障检测使能时的 SCK 输出时序

在某些特殊情况下，主机 1 的主机模式故障检测功能可能无法检测到模式故障。如果在图 16-9 中的 (1) 周期内，主机 2 驱动 CS 为低电平，则主机 1 的控制器不会设置 MODEF。只有主机 2 在 (1) 周期之前驱动 CS 为低电平，主机 1 才可以正常设置 MODEF。

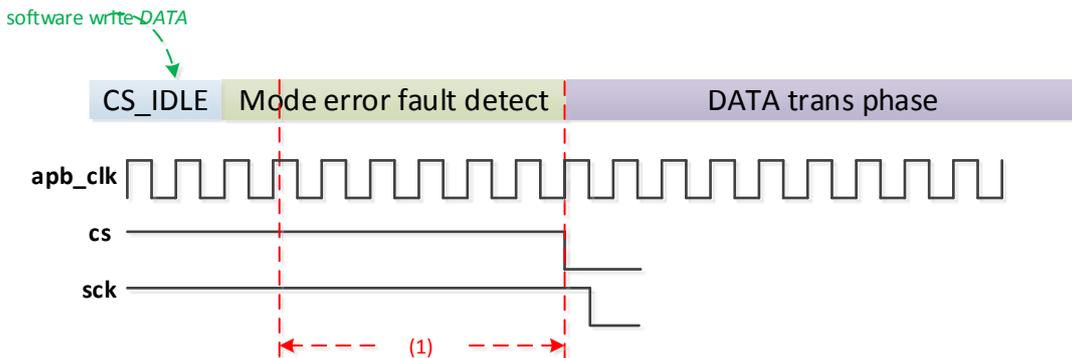


图 16-9 模式故障检测限制

16.4.5 从机低功耗唤醒

处于停止模式的 SPI 从机可以产生异步中断，以在接收到数据时将 CPU 从低功耗模式唤醒。为确保 SPI 模块低功耗唤醒功能正确，系统必须遵循一些规定。在 CPU 进入低功耗模式之前，系统必须确认 SPI 模块处于空闲状态。软件可以检查状态寄存器 SPI_STATUS[8] IDLEF 位的状态。对于主模式，tx 缓冲区为空，rx 缓冲区为空，内部硬件空闲，IDLEF 为“1”。对于从机模式，tx 缓冲区为空，rx 缓冲区为空，CS 为无效（CS 为高电平），IDLEF 为“1”。如果当 SPI 模块忙时，CPU 进入低功耗模式，则 SPI 模块无法确保数据有效或唤醒功能正确。

从机仅在如下条件全部得到满足时生成异步唤醒中断：

- a. SPI 模块处于从机模式；
- b. SPI 从机处于空闲状态；
- c. WUEN 位为‘1’；

d. 单字节唤醒序列传输结束。

CS 从高到低会初始化唤醒阶段，从机在 FRMSIZE 指定的 SCK 周期后生成异步唤醒请求。SPI 从机可以接收唤醒相位字节的数据。芯片唤醒结束（时钟恢复）后，RXFF 标志位会置位，读取数据寄存器将返回主机发送的唤醒相位字节的数据。唤醒阶段只能传输一个字节。在唤醒阶段，来自主机的连续传输会破坏接收到的数据，并可能导致无法正确设置 RXFF 标志。

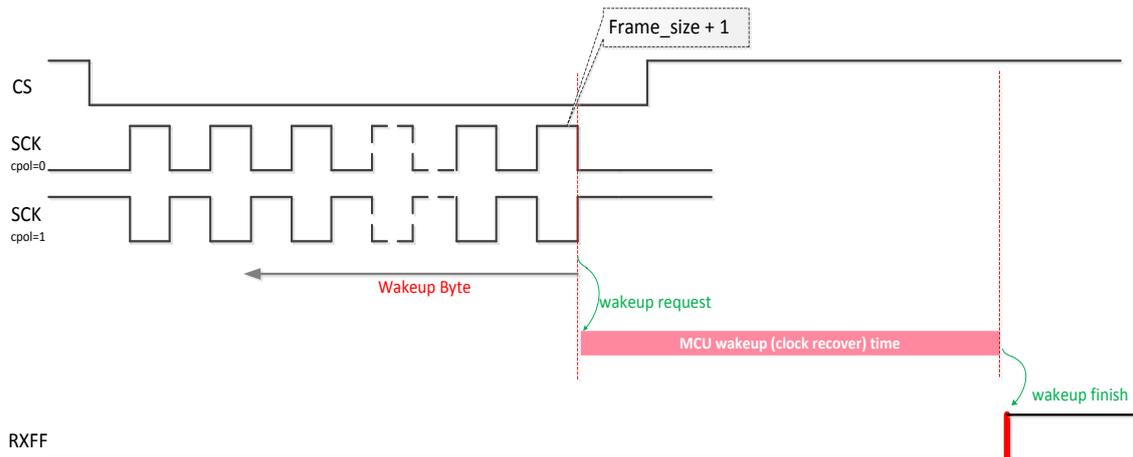


图 16-10 唤醒序列

16.4.6 中断

SPI 共有 5 个中断。

表 16-1 中断汇总

状态标志位	中断使能位
发送缓冲区空标志 (TXEF)	TXEIE
接收缓冲区非空标志 (RXFF)	RXFIE
发送缓冲区下溢 (TXUF)	TXUIE
接收缓冲区溢出 (RXOF)	RXOIE
主模式失效事件 (MODEF)	MODFIE

16.5 应用说明

16.5.1 主机 CS 连续模式

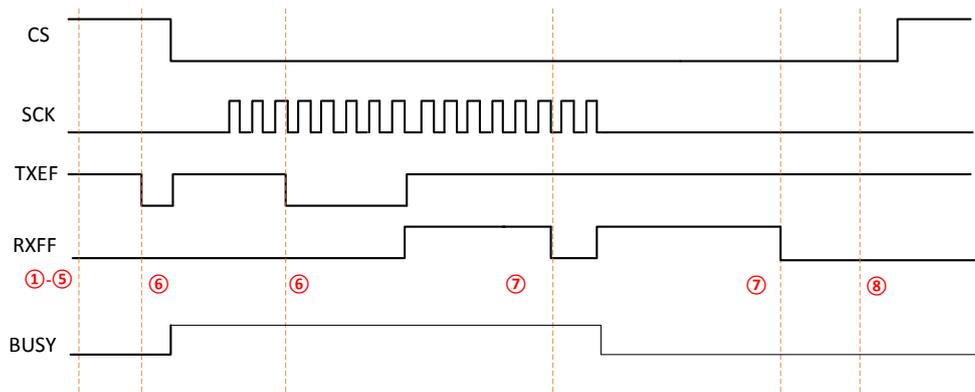


图 16-11 CS 连续模式

CS 连续输出

1. 配置寄存器 `SPI_CFG0`: `CS_SETUP`, `CS_HOLD`, `SCK_LOW`, `SCK_HIGH`;
2. 配置寄存器 `SPI_CFG1`: `CS_IDLE`;
3. 配置 `FRMSIZE`, `CPHA`, `CPOLRMSBF`, `TMSBF` 等;
4. 配置 `CSOE`, `CONT_CS`, `MSTR`;
5. `SPIEN=1`;
6. `TXEF=1`, 写数据至 `DATA`;
7. `RXFF=1`, 从 `DATA` 中读数据;
8. 写 `CSRLS'1'`, 释放 `CS`, 然后硬件进入空闲状态。即 `CSOE = 1`, `CONT_CS = 1` 时, `CS` 由硬件自动拉低, 但是, 数据发送完成后, 需要用户软件写 `CSRLS'1'`, 拉高 `CS`。

16.5.2 主机 CS 非连续输出

1. 配置寄存器 `SPI_CFG0`: `CS_SETUP`, `CS_HOLD`, `SCK_LOW`, `SCK_HIGH`;
2. 配置寄存器 `SPI_CFG1`: `CS_IDLE`;
3. 配置 `FRMSIZE`, `CPHA`, `CPOL`, `RMSBF`, `TMSBF` 等;
4. 配置 `CSOE`, `CONT_CS`, `MSTR`;
5. `SPIEN=1`;
6. `TXEF=1`, 写数据至 `DATA`;

7. RXFF=1, 从 DATA 读取数据。

当 CS 处于不连续模式时, CS 通过硬件变为低或高电平, 所以软件不需要配置 CSRLS。

16.5.3 从机模式

1. 配置 FRMSIZE,CPHA,CPOL,RMSBF,TMSBF 等;
2. 配置 MSTR;
3. SPIEN=1;
4. TXEF=1, 写数据至 DATA;
5. RXFF=1, 从 DATA 读取数据。

16.6 寄存器定义

表 16-2 SPI 寄存器映射

SPI0 基地址: 0x4000c000

地址	名称	宽度	描述
SPIx 基地址+0x00	SPI_CFG0	32	SPI 配置寄存器 0
SPIx 基地址+0x04	SPI_CFG1	32	SPI 配置寄存器 1
SPIx 基地址+0x08	SPI_CMD	32	SPI 命令寄存器
SPIx 基地址+0x0c	SPI_STATUS	32	SPI 状态寄存器
SPIx 基地址+0x10	SPI_DATA	32	SPI 数据寄存器
SPIx 基地址+0x14	SPI_CFG2	32	SPI 配置寄存器 2

【说明】上表中, x=0。

16.6.1 配置寄存器 0(SPI_CFG0)

表 16-3 SPI_CFG0 寄存器

SPI_CFG0 配置寄存器 0 Reset: 0x05050505

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	CS_SETUP								CS_HOLD							
访问	RW								RW							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	SCK_LOW								SCK_HIGH							
访问	RW								RW							
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

字段	说明
31: 24 CS_SETUP	<p>CS 建立时间配置</p> <p>片选建立时间 = (CS_SETUP + 1)*CLK_PERIOD, 其中 CLK_PERIOD 是 SPI 引擎采用的时钟周期时间</p>
23: 16 CS_HOLD	<p>CS 保持时间配置</p> <p>片选保持时间 = (CS_HOLD + 1)*CLK_PERIOD.</p>
15:8 SCK_LOW	<p>SCK 低电平时间配置</p> <p>注意: CPOL 为 0 时, 该位配置的是 SCK_LOW 的时间。CPOL 为 1 时, 配置的是 SCK_HIGH 的时间。</p> <p>SCK 时钟低电平时间 = (SCK_LOW + 1)*CLK_PERIOD.</p>
7: 0 SCK_HIGH	<p>SCK 高电平时间配置</p> <p>注意: CPOL 为 0 时, 该位配置的是 SCK_HIGH 的时间。CPOL 为 1 时, 配置的是 SCK_LOW 的时间。</p> <p>SCK 时钟高电平时间 = (SCK_HIGH + 1)*CLK_PERIOD.</p>

16.6.2 配置寄存器 1(SPI_CFG1)

表 16-4 SPI_CFG1 寄存器

SPI_CFG1		配置寄存器 1								Reset: 0x027C0005							
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称		WKEN		CONTS		MODFEN	CSOE		FRMSIZE				RMSBF	MSBF	CPHA	CPOL	
访问		RW		RW		RW	RW		RW				RW	RW	RW	RW	
Reset		0		0		0	1		0	1	1	1	1	1	0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称			MODFE	MSTR	RXOIE	TXUIE	RXFIE	TXEIE	CS_IDLE								
访问			RW	RW	RW	RW	RW	RW	RW								
Reset			0	0	0	0	0	0	0	0	0	0	0	1	0	1	

字段	说明
30 WKUEN	<p>从机唤醒功能使能</p> <p>0: 从机唤醒功能禁用</p>

字段	说明
	1: 从机唤醒功能使能
28 CONT_CS	CS 连续输出使能 0: CS 输出不连续 1: CS 输出连续
26 MODFEN	主机模式故障检测使能 0: 禁用主机模式故障检测功能 1: 使能主机模式故障检测功能
25 CSOE	CS 硬件输出使能 0: 禁用 CS 硬件输出 1: 使能 CS 硬件输出
23: 20 FRMSIZE	帧大小 0000: 4bit 0001: 4bit 0010: 4bit 0011: 4bit 0100: 5bit ... 1110: 15bit 1111: 16bit
19 RMSBF	RX 最高有效位优先 0: 移位器移位的第一位是输入数据的 LSB 1: 移位器移位的第一位是输入数据的 MSB
18 MSBF	TX 最高有效位优先 0: TX LSB 优先(LSB 位首先移出) 1: TX MSB 优先(MSB 位首先移出)
17 CPHA	时钟相位 0: 第一个 SCK 过渡边沿为数据移出边沿 1: 第一个 SCK 过渡边沿为数据捕获边沿
16 CPOL	时钟极性 0: 空闲时, SCK 为 0 1: 空闲时, SCK 为 1
13 MODFIE	模式故障中断使能 0: 禁用 1: 使能
12	主机或从机模式选择

字段	说明
MSTR	0: 从机模式 1: 主机模式
11 RXOIE	RX 缓冲区溢出中断使能 0: 禁用 1: 使能
10 TXUIE	TX 缓冲区下溢中断使能 0: 禁用 1: 使能
9 RXFIE	RX 缓冲区满中断使能 0: 禁用 1: 使能 注意: RX 缓冲区非空即可以产生中断
8 TXEIE	TX 缓冲区空中断使能 0: 禁用 1: 使能 注意: TX 缓冲区非满即可以产生中断
7: 0 CS_IDLE	CS 空闲时间 CS 空闲时间 = (CS_IDLEA_COUNT+1)*CLK_PERIOD

16.6.3 命令寄存器 (SPI_CMD)

表 16-5 SPI_CMD 寄存器

SPI_CMD		命令寄存器																Reset: 0x00000000		
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
名称																				
访问																				
Reset																				
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
名称										RO TRIG	CS RLS	SW RST					SPI EN			
访问										RW	RW	RW					RW			
Reset										0	0	0					0			

字段	说明
6 ROTRIG	主机只读模式触发 注意: 当 CFG2 中的 ROEN=1 时, 将此位写入 '1' 会触发读序列。读取此位将始终返回 '0'。

字段	说明
5 CSRLS	<p>CS 释放</p> <p>0: 不起作用 1: 释放 CS</p> <p>软件将此位写入‘1’，然后 CS 将变为高电平。读取此位始终返回“0”，该位对 CS 连续输出有效(CONT_CS=1, CSOE=1)。</p>
4 SWRST	<p>软件复位</p> <p>0: 不起作用 1: 复位</p> <p>注意：本复位位只能复位主机引擎/缓冲区/标志位逻辑，从机缓冲区 /标志位逻辑。CFG0/CFG1/CFG2/CMD 控制位不会复位。</p>
0 SPIEN	<p>SPI 使能</p> <p>0: 禁用 1: 使能</p>

16.6.4 状态寄存器(SPI_STATUS)

表 16-6 SPI_STATUS 寄存器

SPI_STATUS																状态寄存器				Reset: 0x0000101			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
名称																							
访问																							
Reset																							
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
名称								IDLEF	MEBY				MOD EF	RX OF	TX UF	RX FF	TX EF						
访问								RO	RO				RO	R/W1C	R/W1C	RO	RO						
Reset								1	0				0	0	0	0	1						

字段	说明
8 IDLEF	<p>SPI 空闲标志</p> <p>0: SPI 模块硬件非空闲 1: SPI 模块硬件空闲</p> <p>注意：对于主机，TX 缓冲区为空，RX 缓冲区为空，内部硬件空闲，该位可以为‘1’。 对于从机，TX 缓冲区为空，RX 缓冲区为空，CS 无效(没有选中)，该位可以为‘1’。</p>
7 MEBY	<p>SPI 主机引擎忙标志</p>

字段	说明
	0: SPI 主机硬件一个数据传输完成 1: SPI 主机硬件一个数据传输未完成
4 MODEF	主机模式故障检测标志 0: 未检测到主机模式故障 1: 检测到主机模式故障 当 SPI 配置为主机时，它会在驱动 CS 低电平之前检测 CS 线路状态。如果 CS 已经为低电平，表明另一个主机已经发起一个传输，置起模式故障检测错误标志。 注：发生该标志置时，需要配置 SWRST=1，SPI 进行软复位恢复。
3 RXOF	RX 缓冲区溢出标志 0: 没有溢出 1: RX 缓冲区溢出 接收缓冲区有 2 个 FIFO，如果接收到 2 帧数据后，RX 缓冲区数据都没有被用户及时读取，下一帧接收的数据过来，会产生 RX 溢出。 注意：写“1”清零此位。产生溢出时，需要及时清除，以免影响 RXFF 标志。 如果溢出，则 DATA 寄存器中仅能读出一帧有效数据。
2 TXUF	TX 缓冲区下溢标志 0: 没有下溢 1: TX 缓冲区下溢 在 slave 模式下，如果没有往 TX 数据寄存器写数据，master 端就开始通讯，会产生 TX 下溢。 注意：写“1”清零此位。产生下溢时，需要及时清除。
1 RXFF	RX 缓冲区满标志 0: 不满 1: 满 注意：只要 rx 缓冲区中有有效的接收数据 (1 帧或 2 帧)，此位将为‘1’。因此该位 ‘1’ 不表示在 rx 缓冲区中接收到 2 帧的数据。读取 DATA 寄存器将会自动清零该位。 此位称为 RX 缓冲区非空更为合理。
0 TXEF	TX 缓冲区空标志 0: 非空 1: 空 注意：只要 tx 缓冲区不满(2 帧数据)，该位将为‘1’，写 DATA 寄存器将自动清零此硬件位。 此位称为 TX 缓冲区非满更为合理。

16.6.5 数据寄存器(SPI_DATA)

表 16-7 SPI_DATA 寄存器

SPI_DATA		数据寄存器																Reset: 0x00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																		
访问																		
Reset																		
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	DATA																	
访问	RW																	
Reset	0																	

字段	说明
15:0 DATA	SPI 数据端口寄存器 读：读操作将返回接收的 DATA 值 写：写入要发送的 DATA

16.6.6 配置寄存器 2(SPI_CFG2)

表 16-8 SPI_CFG2 寄存器

SPI_CFG2		配置寄存器 2																Reset: 0x00000000
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	CS_CHECK																	
访问	RW																	
Reset	0																	
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称													ROEN	TOEN	MNDC			
访问													RW	RW	RW			
Reset													0	0	0			

字段	说明
31:24 CS_CHECK	CS 检测记数周期 0: 1 个功能时钟检测周期 1: 2 个功能时钟检测周期 ... 255:256 个功能时钟检测周期

3 仅 RX 模式使能

ROEN

0: 禁用

1: 使能

注意： 仅 RX 模式，TXEF 将保持 0。即使 TXEIE=1 也不触发 TXEF IRQ。
 主机使用仅 RX 模式时，建议使用 poll 和中断方式传输。

2 仅 TX 模式使能

TOEN

0: 禁用

1: 使能

注意： 仅 TX 模式，在一帧数据传输完成后，RXFF 不置位。即使 RXFIE=1，也不触发 RXFF IRQ。

1 主机非溢出模式

MNOV

0: 禁用

1: 使能

如果 rxbuff 为满，则写入 txbuff 不会触发新的发送操作。

17 看门狗模块 (WDG)

17.1 简介

看门狗 WDG (Watchdog) 是一个独立定时器，一般用来检测系统软件程序是否按预期运行。如果看门狗模块没有被按时刷新，看门狗会产生系统复位，比如程序中存在死循环，但因某些原因没有跳出，或者使用多任务操作系统，喂狗任务没有按时执行，都会由于看门狗模块没有被刷新而产生系统复位。一般用于高安全性场合。

17.2 特性

- 4 种时钟源
- 可选固定 256 时钟预分频器
- 可编程 32 bit 超时时间
- 窗口模式
- 超时中断

17.3 结构框图

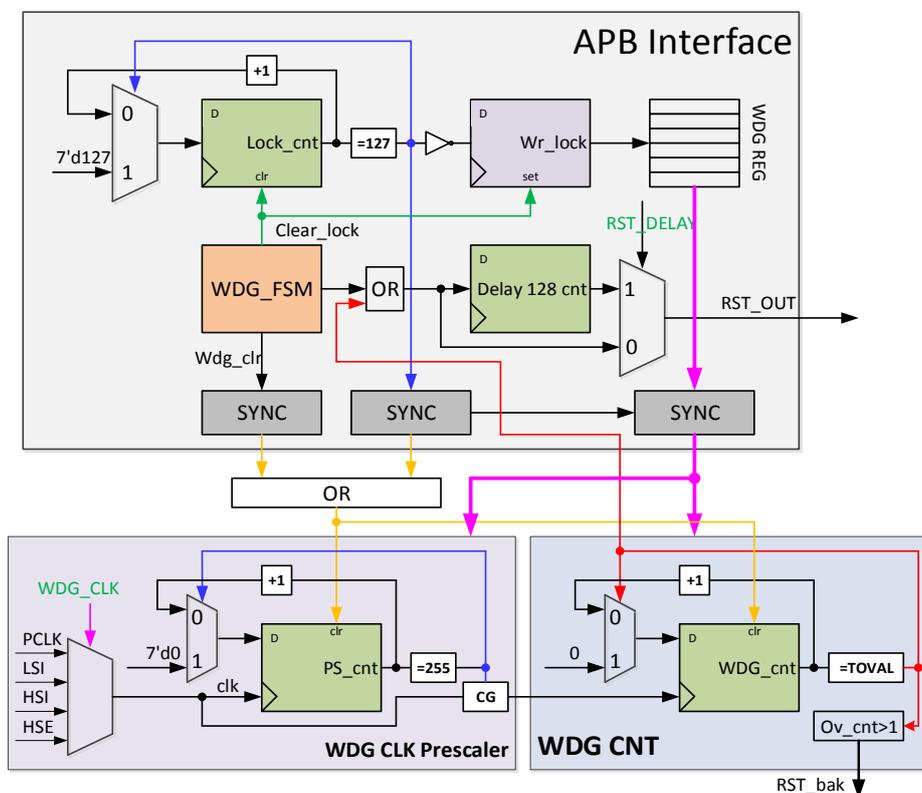


图 17-17-1 WDG 结构框图

17.4 功能描述

17.4.1 基本看门狗

看门狗具有四种时钟源：总线时钟、内部 32kHz 时钟、内部 32MHz 时钟、外部 XOSC 时钟。看门狗定时器使用一个 32 位的可编程向上计数器以及一个可选的固定 256 分频器。

看门狗使能后，开始计数，若计数到 TOVAL 值，会产生系统复位，在计数达到 TOVAL 值前，刷新看门狗会使计数器复位并重新开始计数。

17.4.2 看门狗默认超时行为

在 MCU 上电后，看门狗计数器默认使用总线时钟（16MHz）进行计数，并在 0x271000 个时钟周期后超时。这将导致 MCU 在上电约 160ms 后发生复位，为了避免这种情况的产生，用户需确保在 MCU 上电约 160ms 内配置或刷新看门狗。

17.4.3 窗口看门狗

看门狗具有窗口模式，在该模式下，在计数器达到窗口值 WIN 前刷新看门狗或者计数器达到 TOVAL 值前未刷新看门狗都会导致复系统复位，在计数器大于等于 WIN 值，小于 TOVAL 值之间刷新看门狗会使计数器复位并重新开始计数。

17.4.4 低功耗行为

MCU 在 Stop 模式下，看门狗可以保持运行，但需要以内部 32kHz 时钟作为时钟源。

看门狗在 Stop 模式下需要两次超时才能产生复位。

17.4.5 调试模式

在调试模式下，看门狗不能使能。

17.5 应用说明

17.5.1 配置看门狗

对看门狗的所有寄存器进行配置的条件是更新位 WDG_CS0[UPDATE]为 1 且看门狗解锁。解锁后，只能在 128 个总线时钟内对看门狗的任意寄存器进行配置，然后所有寄存器自动上锁，再次配置需要再次满足上述两个条件。

一旦用户将 `WDG_CS0` [UPDATE]配置为 0 后，除非强制复位，否则无法修改看门狗寄存器配置，在 `WDG_CS0`[UPDATE]为 0 的情况下解锁会引起系统复位。

解锁序列：向 `WDG_CNT` 寄存器先后写入 `0xE064D987`、`0x868A8478`，写入的值不正确或顺序颠倒会引起系统复位。

17.5.2 刷新看门狗

在基本看门狗与窗口看门狗模式下，为了保证看门狗不复位系统，需要软件在规定时间内刷新看门狗。看门狗刷新后，看门狗计数器重新从 0 开始计数，软件需要再次刷新看门狗。这种机制使得软件必须定时刷新看门狗，在一定程度上反映了程序的正常运行，当程序意外跑飞后，看门狗会复位系统。

刷新序列：向 `WDG_CNT` 寄存器先后写入 `0x7908AD15`、`0x5AD5A879`，写入的值不对或顺序颠倒会引起系统复位。

17.5.3 看门狗中断

看门狗具有中断功能，中断使能后，若看门狗计数器超时，看门狗不会立即复位系统，而是延迟 128 个总线时钟后复位系统。这 128 个总线时钟是留给中断服务程序响应的的时间，用户软件可以在中断程序中进行简单的程序处理，但不建议在此时刷新看门狗。

17.6 寄存器定义

表 17-1 WDG 寄存器映射

WDG 基地址：0x4000B000

地址	名称	宽度	描述
WDG 基地址+0x00	<code>WDG_CS0</code>	32	配置寄存器 0
WDG 基地址+0x04	<code>WDG_CS1</code>	32	配置寄存器 1
WDG 基地址+0x08	<code>WDG_CNT</code>	32	计数器寄存器
WDG 基地址+0x0C	<code>WDG_TOVAL</code>	32	超时值寄存器
WDG 基地址+0x10	<code>WDG_WIN</code>	32	窗口值寄存器

17.6.1 配置寄存器 0(WDG_CS0)

表 17-2 WDG_CS0 寄存器

 WDG_CS0 配置寄存器 0 Reset: 0x000000A0

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

字段	说明
7 EN	看门狗使能 0：看门狗禁用 1：看门狗使能 使能看门狗计数器开始计数。
6 INT	看门狗中断 0：看门狗中断禁用，看门狗复位系统无延迟 1：看门狗中断使能，看门狗复位系统延迟 128 个总线时钟 在看门狗超时或非法写入时，若使能了看门狗中断，会触发看门狗中断，并在中断响应 128 个总线时钟后再发生系统复位。
5 UPDATE	看门狗配置更新位 0：不允许更新。完成初始配置后，除非强制复位，否则无法修改看门狗寄存器配置 1：允许更新。执行解锁操作后，软件可以在 128 个总线时钟内修改看门狗配置寄存器

17.6.2 配置寄存器 1(WDG_CS1)

表 17-3 WDG_CS1 寄存器

WDG_CS1 配置寄存器 1 Reset: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称									WIN	FLG				PRES			CLK
访问									RW	R/W1C				RW			RW
Reset									0	0				0			0

字段	说明
7 WIN	看门狗窗口 0: 窗口模式禁用 1: 窗口模式使能 使能看门狗窗口模式
6 FLG	看门狗中断标志 0: 未发生中断 1: 发生中断 该位指示中断产生, 写入 1 将其清零
4 PRES	看门狗预分频器 0: 256 预分频器禁用 1: 256 预分频器使能 使能看门狗计数器 256 倍预分频
1:0 CLK	看门狗时钟 00: 总线时钟 01: 32kHz 内部 RC 振荡器 10: 32MHz 内部 RC 振荡器 11: 外部 XOSC 选择看门狗功能时钟源。功能时钟频率不能高于总线时钟频率, 通过配置 WDG_CS1[PRES]避免该情况。

17.6.3 计数器寄存器(WDG_CNT)

表 17-4 WDG_CNT 寄存器

WDG_CNT																计数器寄存器																Reset: 0x00000000															
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																															
名称	CNT[31: 16]																																														
访问	RO																																														
Reset	0																																														
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
名称	CNT[15: 0]																																														
访问	RO																																														
Reset	0																																														

字段	说明
31: 0 CNT	看门狗计数值 看门狗计数器寄存器指示当前看门狗计数器的数值，软件可在任意时刻对计数器寄存器进行读操作。软件无法直接更改计数器的值。 但对这个寄存器的写操作具有特殊功能：刷新序列和解锁序列。

17.6.4 超时值寄存器(WDG_TOVAL)

表 17-5 WDG_TOVAL 寄存器

WDG_TOVAL																超时值寄存器																Reset: 0x00271000															
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																															
名称	TOVAL[31: 16]																																														
访问	RW																																														
Reset	0x27																																														
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
名称	TOVAL[15: 0]																																														
访问	RW																																														
Reset	0x1000																																														

字段	说明
31: 0 TOVAL	看门狗超时值 看门狗计数器与超时数值连续作比较。若计数器达到超时值，则看门狗复位系统。计时长度等于 WDG_TOVAL+1，默认值为 0x271000。

17.6.5 窗口值寄存器(WDG_WIN)

表 17-6 WDG_WIN 寄存器

WDG_WIN 窗口值寄存器 Reset: 0x00000000

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

字段	说明
31: 0 WIN	看门狗窗口值 窗口模式使能后（WDG_CS1[WIN]置位），WIN 值决定看门狗允许刷新的最早时间，早于窗口值刷新会引起系统复位。

18 实时计数器模块 (RTC)

18.1 简介

实时计数器模块 RTC (Real Time Clock)，主要功能是实时计数。在 Stop 低功耗模式下，RTC 具有保持运行并唤醒 MCU 的功能。

18.2 特性

- 32 位向上计数器
- 可编程 20 位预分频器
- 计数溢出翻转 GPIO

18.3 结构框图

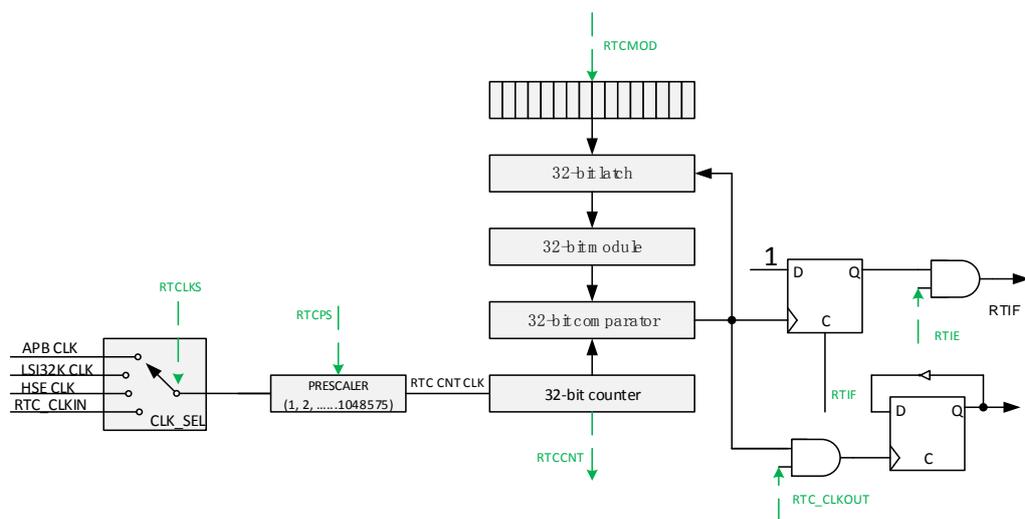


图 18-1 RTC 结构框图

18.4 功能描述

18.4.1 时钟源选择

RTC 模块有四种时钟源可选择使用：总线时钟、内部 32KHz RC 振荡器时钟、HSE 时钟以及外部 RTCIN 引脚输入时钟。

18.4.2 计时特性

RTC 是一个向上增加的计数器，计数达到预设的模值后，产生 RTC 溢出标志。如果使能了 RTC 溢出中断，则产生一次 RTC 中断请求，溢出后 RTC 计数器将从 0 开始新的计数。RTC 同时内置一个向上计数的预分频器，当达到预设的预分频值后，产生预分频器溢出标志，如果使能了预分频器中断，则产生一次预分频器中断请求，预分频器计数溢出后将从 0 开始新的计数。

18.4.3 RTC 计时信号输出

PA13 可以复用为 RTC_CLKOUT 功能，复用后，RTC 计数溢出能翻转 PA13。

18.4.4 低功耗唤醒

MCU 在低功耗模式 Stop 下，RTC 可作为唤醒源唤醒 MCU，需要在进入低功耗模式前选择内部 32K 作为时钟源，并启动模块。Stop 模式下，RTC 溢出后产生 RTC 中断，唤醒 MCU。预分频中断无法唤醒 MCU。从低功耗模式唤醒后，RTC 计数器会继续保持计数。

18.5 应用说明

18.5.1 RTC 基本使用

使用前，配置 RTC_SC 寄存器，初始化 RTC 模块的时钟、模值、中断等，在最后配置预分频器为非 0 值时，RTC 计数器与预分频计数器开始计数。

RTC 预分频计数器溢出时，标志位 RPIF 置位，写 1 清除标志。RTC 计数器溢出时，标志位 RTIF 置位，写 1 清除标志。在计数器运行时，对 RTC_CLK 或 RTC_PS 的更改将清 0 计数器。

RTCO 位为 1，使能 RTC 计数器溢出翻转指定 GPIO，GPIO 同时需要复用为 RTC_CLKOUT 功能。

18.5.2 RTC 低功耗唤醒

使用 RTC 唤醒低功耗状态 Stop 下的 MCU，需要先在 SPM 模块中使能 RTC 唤醒，同时选择内部 32K 时钟或 RTC_CLKIN 时钟作为 RTC 时钟源并启动计时。

18.6 寄存器定义

表 18-1 RTC 寄存器映射

RTC 基地址: 0x40008400

地址	名称	宽度	描述
RTC 基地址+0x00	RTC_SC	32	状态与控制寄存器
RTC 基地址+0x04	RTC_MOD	32	模值寄存器
RTC 基地址+0x08	RTC_CNT	32	计数器寄存器
RTC 基地址+0x0C	RTC_PS	32	预分频器寄存器
RTC 基地址+0x10	RTC_PSCNT	32	预分频器计数器寄存器

18.6.1 控制与状态寄存器(RTC_SC)

表 18-2 RTC_SC 寄存器

RTC_SC 控制与状态寄存器 Reset: : 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称															RPIF	RPIE	
访问															R/W1C	RW	
Reset															0	0	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	RTCCLKS								RTIF	RTIE		RTCO					
访问	RW								R/W1C	RW		RW					
Reset	0								0	0		0					

字段	说明
17 RPIF	<p>实时预分频器中断标志</p> <p>0: RTC 预分频器计数器没有达到 RTC 预分频器寄存器的值 1: RTC 预分频器计数器达到 RTC 预分频器寄存器的值</p> <p>该状态位表示 RTC 预分频器计数器达到了 RTC 预分频器寄存器中的值。写入逻辑 0 无效。写入逻辑 1 会清除此位和实时中断请求。复位将 RPIF 清零。</p>
16 RPIE	<p>实时预分频器中断使能</p> <p>0: 禁用实时预分频器中断请求 1: 使能实时预分频器中断请求</p> <p>使能实时预分频器中断。如果 RPIE 置位, 当 RPIF 置位时产生中断请求。</p>
15: 14 RTCLKS	<p>实时时钟源选择</p> <p>00: 总线时钟 01: 内部 32KHz RC 振荡器</p>

字段	说明
	10: HSE 11: 外部 RTCIN 引脚输入时钟
	RTC 模块时钟源选择位。切换时钟源会清零预分频器与 RTC 计数器。 复位清除 RTCLKS 为 0。RTC 溢出时间为 $((MOD + 1) * (RTCPS + 1)) / RTCLKS$
7 RTIF	实时中断标志 0: RTC 计数器没有达到 RTC 模数寄存器的值 1: RTC 计数器达到 RTC 模数寄存器的值 该状态位表示 RTC 计数器达到了 RTC 模数寄存器中的值。写入逻辑 0 无效。写入逻辑 1 会清除该位和实时中断请求。
6 RTIE	实时中断使能 0: 禁用实时中断请求 1: 使能实时中断请求 使能实时中断。如果 RTIE 置位, 则在 RTIF 置位时会产生中断请求。
4 RTCO	实时计数器输出 0: 实时计数器输出禁用。 1: 实时计数器输出使能 使能在 RTC 计数溢出时指定引脚输出电平信号翻转。如果该位置 1, RTC 计数器溢出时将 RTC_CLKOUT 引脚电平翻转。

18.6.2 模值寄存器(RTC_MOD)

表 18-3 RTC_MOD 寄存器

RTC_MOD		模值寄存器														Reset: : 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称	MOD[31: 16]															
访问	RW															
Reset	0															
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	MOD[15: 0]															
访问	RW															
Reset	0															

字段	说明
31: 0 MOD	RTC 模值 用于与当前计数值(RTC_CNT)进行比较的模数值，计数值等于模数时计数将重置为 0x0，置位 SC[RTIF]。

18.6.3 计数器寄存器(RTC_CNT)

表 18-4 RTC_CNT 寄存器

RTC_CNT		计数器寄存器														Reset: 0x00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称		CNT[31: 16]															
访问		RO															
Reset		0															
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		CNT[15: 0]															
访问		RO															
Reset		0															

字段	说明
31: 0 CNT	RTC 计数器 指示 RTC 计数器当前计数值。对此寄存器写无效，将不同的值写入 SC [RTCLKS]或 PS[RTCPS]重新开始计数

18.6.4 预分频器寄存器(RTC_PS)

表 18-5 RTC_PS 寄存器

RTC_PS		预分频器														Reset: 0x00000000	
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																RTCPS[19: 16]	
访问																RW	
Reset																0	
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		RTCPS[15: 0]															
访问		RW															
Reset		0															

字段	说明
19: 0 RTCPS	RTC 预分频器 更改预分频器值会清除预分频器计数器和 RTC CNT 计数器。 RTCPS 为 0 时，RTC 计数器停止计数，RTCPS 为非 0 时，开始计数。

18.6.5 预分频器计数寄存器(RTC_PSCNT)

表 18-6 RTC_PSCNT 寄存器

RTC_PSCNT		预分频器计数器												Reset: 0x00000000			
位		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称														PSCNT[19: 16]			
访问														RW			
Reset														0			
位		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称		PSCNT[15: 0]															
访问		RW															
Reset		0															

字段	说明
19: 0 PSCNT	RTC 预分频器计数器 指示当前预分频计数器的值。写入对此寄存器无效，复位或将不同的值写入 SC [RTCLKS]和 RTCPS 将计数清除为0x0。

19 片内 Flash (Embedded Flash)

19.1 简介

片内 Flash 控制器是 Cortex™-M0+和片内 Flash 之间的桥梁，在实际应用中，用户代码存放于片内 Flash 中，片内 Flash 启动作为主要启动模式。

片内 Flash 以下简称 eFlash。包含程序存储器 (P-flash)、数据存储器 (D-flash)、信息区。

19.2 特性

- P-flash 存储器大小：32KB，页容量 = 512B
- D-flash 存储器大小：2KB，页容量 = 8B，块容量 = 16B
- 信息区大小：512B，页容量 = 128B
- 数据保持：>100years @25°C，>10years @ 125°C
- 寿命：P-Flash > 10K, D-Flash > 100K
- P-flash 存储器支持下面操作：
 - 读，页编程，整片擦除，页擦除，页擦除验证，整片擦除验证
- D-flash 存储器支持下面操作：
 - 读，页编程，整片擦除，页擦除，块擦除，整片擦除验证
- 信息区支持下面操作
 - 读，页编程，页擦除
- 支持读写保护
 - 写保护避免存储数据的意外编程或擦除
 - 读保护避免非法获取数据
- 支持 AEC-Q100 grade 1

19.3 结构框图

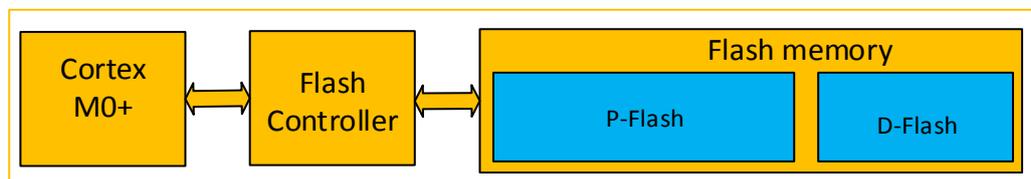


图 19-1 eflash 和 eflash 控制器结构框图

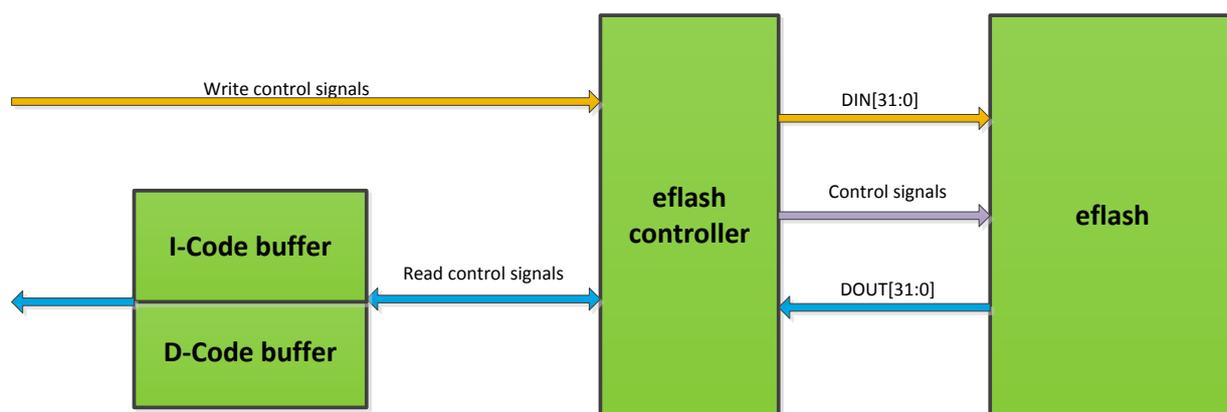


图 19-2 eflash 和 eflash 控制器数据流

19.4 功能描述

19.4.1 片内 Flash 组织

在介绍命令之前，先了解表 19-1 中的片内 Flash 组织。整个片内 Flash 由三部分组成：P-Flash 存储器、D-Flash 存储器、信息区。

- P-Flash 存储器用于存储用户代码和数据，用户代码可以对 P-flash 存储器进行擦除、编程、验证和读取命令操作。
- D-Flash 存储器用于存储用户数据，用户代码可以对 D-flash 存储器进行擦除、编程和读取命令操作。
- 信息区也叫选项字节区（Option byte），用于存放整个 P-flash 和 D-flash 的读写保护信息，第 2 页前 8 个字节用于存放用户特殊的数据（即 19.4.2.2 节的用户数据）。对于选项字节区，用户可以擦除、编程和读取。

表 19-1 片内 Flash 存储器组织

Flash 存储器	名称	地址	尺寸 (字节)	用户权限 (说明)
P-flash (32KB)	Page 0	0x0800 0000 ~ 0x0800 01FF	512	
	Page 1	0x0800 0200 ~ 0x0800 03FF	512	
	Page 2	0x0800 0400 ~ 0x0800 05FF	512	

	Page 3	0x0800 0600 ~ 0x0800 07FF	512	页擦除/整片擦除
	页编程/
	Page 63	0x0800 7E00 ~ 0x0800 7FFF	512	读取/ 页擦除验证/整片擦除验证
D-flash (2KB)	Page 0	0x0802 0000 ~ 0x0802 0007	8	页擦除/块擦除/整片擦除
	Page 1	0x0802 0008 ~ 0x0802 000F	8	页编程/
	Page 2	0x0802 0010 ~ 0x0802 0017	8	读取/
	Page 3	0x0802 0018 ~ 0x0802 001F	8	整片擦除验证
	
	Page 254	0x0802 07F0 ~ 0x0802 07F7	8	
	Page 255	0x0802 07F8 ~ 0x0802 07FF	8	
选项字节区 (512B)	Page 0	0x0804 0000 ~ 0x0804 007F	128	选项字节擦除
	Page 1	0x0804 0080 ~ 0x0804 00FF	128	选项字节编程
	Page 2	0x0804 0100 ~ 0x0804 017F	128	读取
	Page 3	0x0804 0180 ~ 0x0804 01FF	128	

19.4.2 片内 Flash 保护

在选项字节页中保存的主要内容有读保护，写保护等。为了避免非法访问 eflash，控制器对 P-Flash 存储器和 D-Flash 存储器的写入和读取具有保护功能。相关信息存储在以下选项字节中，其中写保护信息也被加载到 FLASH_PWPR0 ~ FLASH_PWPR1 和 FLASH_DWPR 寄存器。修改选项字节中内容后，需要复位或重新上电后才生效，其中补码部分(如 nRDP / nP_WPRT_EN / nD_WPRT_EN / nDATAx)由硬件自动实现。

当写保护生效后，不支持对有写保护页的 eflash 区域执行擦除以及编程操作，但不影响正常读取 eflash 数据。

当失能写保护生效后，对应的 eflash 区域就可以正确擦除和编程。

当读保护生效后，无法通过 JTAG/SWD 方式正确读取 eflash 主存储区数据（读取时显示 0x0），当失能读保护生效后 eflash 主存储区的数据将被擦除。



使能和失能读写保护都是通过编程选项字节地址且都是需要复位后生效，详细设置参考下文所述。

表 19-2 选项字节内容列表

页	地址	[31: 24]	[23: 16]	[15: 8]	[7: 0]	默认值	注释
Page0	0x0804 0000	0xFF	nRDP	0xFF	RDP	0xFF53FFAC	P-flash 和 D-flash 读保护
Page1	0x0804 0084	nP_WPRT_EN[15: 0]		P_WPRT_EN[15: 0]		0xFFFFFFFF	P-flash 写保护信息
	0x0804 0088	nP_WPRT_EN[31: 16]		P_WPRT_EN[31: 16]		0xFFFFFFFF	
	0x0804 008C	nP_WPRT_EN[47: 32]		P_WPRT_EN[47: 32]		0xFFFFFFFF	

	0x0804 0090	nP_WPRT_EN[63: 48]	P_WPRT_EN[63: 48]	0xFFFFFFFF	
Page2	0x0804 0100	nDATA0	DATA0	0xFFFFFFFF	用户数据
	0x0804 0104	nDATA1	DATA1	0xFFFFFFFF	
Page3	0x0804 0184	nD_WPRT_EN[15: 0]	D_WPRT_EN[15: 0]	0xFFFFFFFF	D-flash 写保护信息
	0x0804 0188	nD_WPRT_EN[31: 16]	D_WPRT_EN[31: 16]	0xFFFFFFFF	

19.4.2.1 读写保护

参考表 19-2 可知：P-Flash 存储器和 D-Flash 存储器读保护所在的选项字节地址为 0x0804 0000 ~ 0x0804 0003，P-flash 存储器写保护所在的选项字节地址为 0x0804 0084 ~ 0x0804 0093，D-Flash 存储器写保护所在的选项字节地址为 0x0804 0184 ~ 0x0804 018B。读写保护设置如表 19-3 和表 19-4 所示。

表 19-3 读保护设置

条件	RDP	nRDP	读保护状态
Case1	0xFF	0xFF	受保护
Case2	0xAC	0x53	不受保护（默认）
其他 case	除了 case1 和 case2 以外任意值		受保护

表 19-4 写保护设置

条件	P_WPRT_EN[x]/ D_WPRT_EN[x]	nP_WPRT_EN[x]/ nD_WPRT_EN[x]	写保护状态
Case1	1	0	受保护
其他 case	除了 case1 以外任意值		不受保护（默认）



在 P-flash 写保护值中，一个比特位对应一页；在 D-flash 写保护值中，一个比特位对应 8 页即 64 字节。

19.4.2.2 用户数据

参考表 19-2 可知：用户数据 DATAx/nDATAx 选项字节地址为 0x0804 0100 ~ 0x0804 0107。其中低 16 位 DATAx 用于用户自由存储数据，高 16 位 nDATAx 为用户数据补码，由硬件自动计算。

19.4.3 Flash 命令 ID

表 19-5 Flash 命令 ID

操作类型	命令	命令 ID	描述
擦除	页擦除	0x1	按页擦除 P-Flash 或者 D-Flash
	选项字节擦除	0x2	按页擦除选项字节区域
	整片擦除	0x3	整个擦除 P-Flash 或 D-Flash 的主存储区
	块擦除	0x4	按块擦除 D-Flash 的主存储区
编程	页编程	0x5	按照 32-bit 对齐方式编程 P-Flash 或者 D-Flash
	选项字节编程	0x6	按照 32-bit 对齐方式编程选项字节区域
验证	页擦除验证	0x7	按照 32-bit 对齐方式验证 P-Flash 主存储区是否被擦除
	整片擦除验证	0x8	验证整个 P-Flash 或 D-Flash 主存储区是否被擦除

19.5 应用说明

在本节中，将通过流程图介绍页擦除，整片擦除，页编程，选项字节擦除，选项字节编程，页擦除验证和整片擦除验证。所有命令操作主要涉及以下寄存器：EFLASH_GCR，EFLASH_CCR，EFLASH_CAR，EFLASH_CDR，EFLASH_CSR。对于读操作，用户可以按照 8 bit/16 bit/32 bit 访问方式直接读取 eflash 存储所需地址，因此本文档中将省略相关描述。

以下流程图仅说明了单个命令操作。在多个命令操作之前只需要进行一次解锁。对于片内 Flash 存储器，共有 64 页 P-Flash 存储器、256 页 D-Flash 存储器、4 个选项字节页组成。

以下重点介绍页擦除，页擦除验证和页编程流程，其他命令操作可以参考这些流程。

19.5.1 页擦除

页擦除操作仅作用于 eflash 中的 P-Flash 存储器和 D-Flash 存储器，页擦除操作不能用于选项字节区。以下详细描述页擦除流程，其他命令操作可参考。

1. 在配置 EFLASH_GCR 和 EFLASH_CCR 之前，必须通过读取 LOCK 的状态来检查控制器是否被锁定。如果控制器处于锁定状态，必须顺序写入 0xac7802 和 0x01234567 至 EFLASH_UKR，然后才能解锁。如果控制器不处于锁定状态，可以直接进入下一步；
2. 解锁后，先通过读出 CMDBSY 的状态来检查是否有正在进行的命令操作。CMDBSY 等于 1 表示某些命令操作未完成，必须等到 CMDBSY 等于 0。事实上，解锁过程可以在检查 CMDBSY 之前或之后完成，下面的图表只是说明了“之前”的情况；
3. 当 CMDBSY 变为 0 时，可以配置如下两个寄存器：EFLASH_GCR 和 EFLASH_CCR。对于所有擦除和编程命令操作，必须按比例调整 EFLASH_GCR 中 FREQ 值，调整公式为 $FREQ = \text{eflash 控制器时钟频率} / 1$ ，比如 eflash 控制器时钟频率为 32MHz， $FREQ = 32 / 1 = 32 = 0x20$ ，但是建议配置的这个值最好是比 0x20 大一点，比如 0x21；

4. 在 EFLASH_CAR 中配置页擦除起始地址，该值是所需擦除页的起始地址；通过地址值区分擦除的是 P-Flash 存储器还是 D-Flash 存储器，且地址值必须 4 字节对齐。
5. 通过控制 EFLASH_CCR 来启动命令并触发它。应保证在 EFLASH_CCR 中配置其他位后，START 位必须从 0 变为 1 才能获得有效触发。CMD 配置为 0x1 以确定传入命令操作是页擦除。其他位应为 0，并将用于其他命令操作；
6. 在有效触发之后，命令操作开始。应通过读出 CMDBSY 来检查命令操作是否完成。当 CMDBSY 等于 0 时，表示命令操作已完成；
7. 清除 EFLASH_CCR；
8. 读出状态以检查是否存在某些错误，例如 PGMERR，特别是对于写保护情况，用户无法擦除已经被写保护的页。

页擦除流程如下图所示。其他命令操作的过程类似于页擦除，区别的地方将在后续章节进行描述。

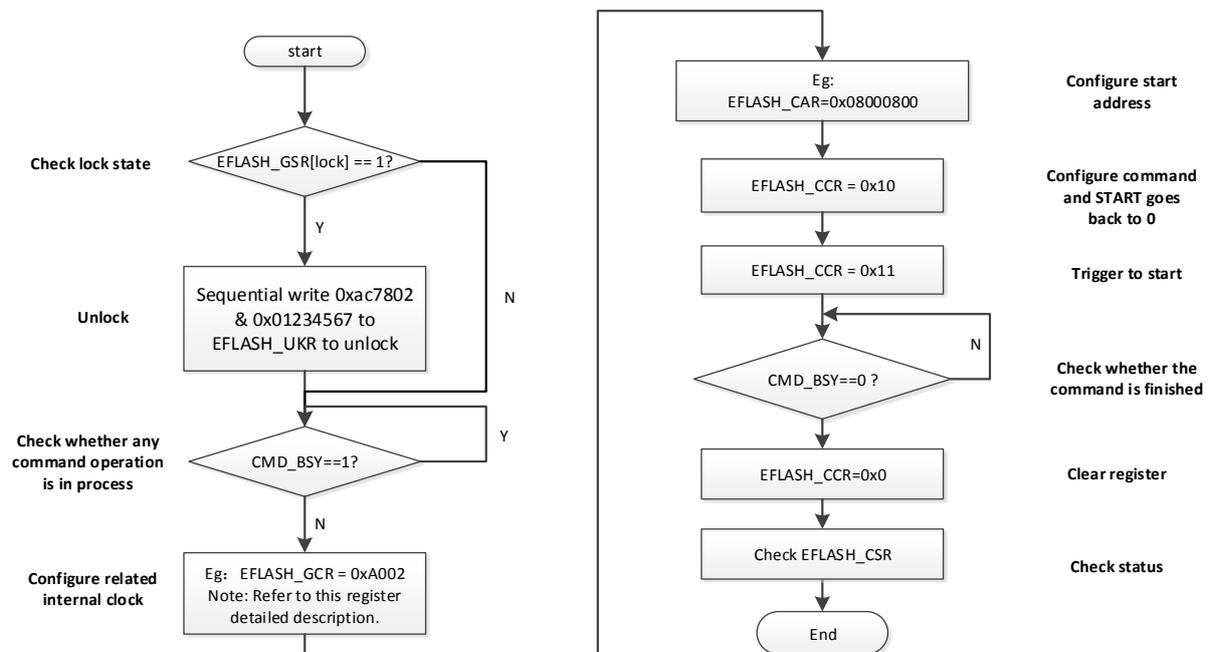


图 19-3 页擦除命令操作流程

19.5.2 块擦除

块擦除可以用于 D-flash 存储器。与页擦除命令流程相比，块擦除流程有两个不同：一是 EFLASH_CCR 寄存器 CMD & SEL 位设置值不同；二是块擦除只能用于 D-flash 存储器。

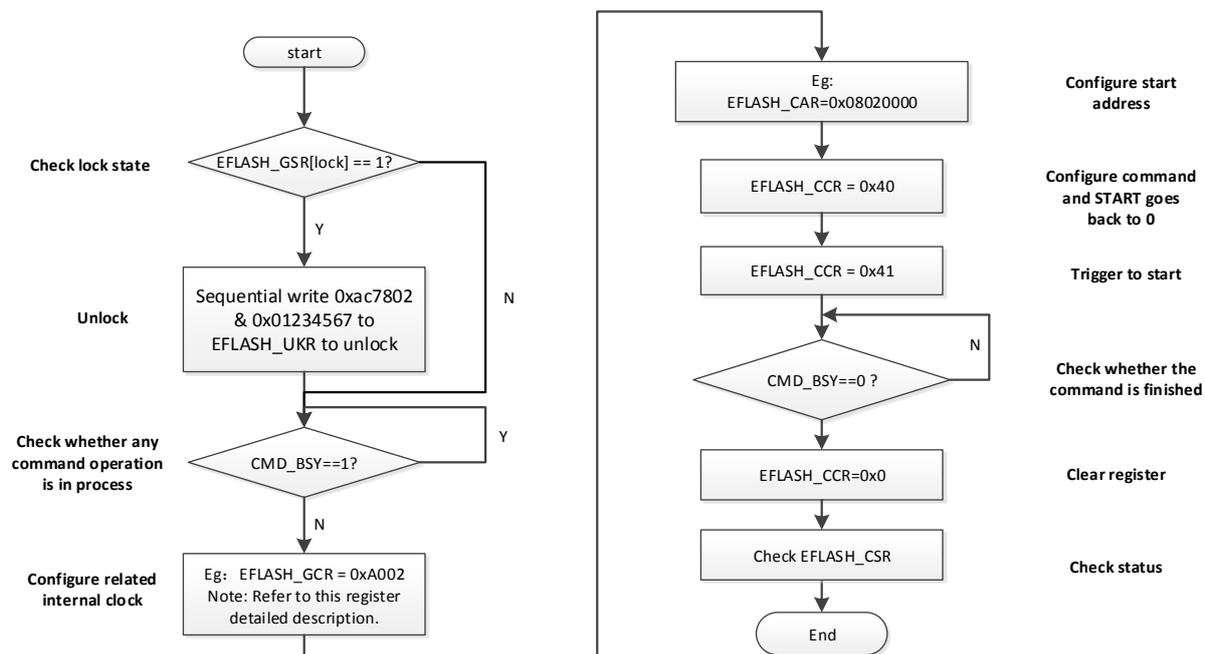


图 19-4 块擦除命令操作流程

19.5.3 整片擦除

整片擦除可以擦除整个 P-Flash 存储器或 D-Flash 存储器，流程如图 19-5 所示。与页擦除命令流程相比，整片擦除流程有两个不同：一是 EFLASH_CCR 寄存器 CMD & SEL 位设置值不同，通过 SEL 区分擦除的是 P-Flash 存储器还是 D-Flash 存储器；二是不需要在 EFLASH_CAR 寄存器中指定擦除地址。

特别地，当主存储器属性从读保护改变为非读保护时，将自动执行整片擦除以保护用户代码不被非法读取。例如，当用户将选项字节中的 RDP 从默认的 0xFF 编程为 0xAC 时，整片擦除会自动执行。

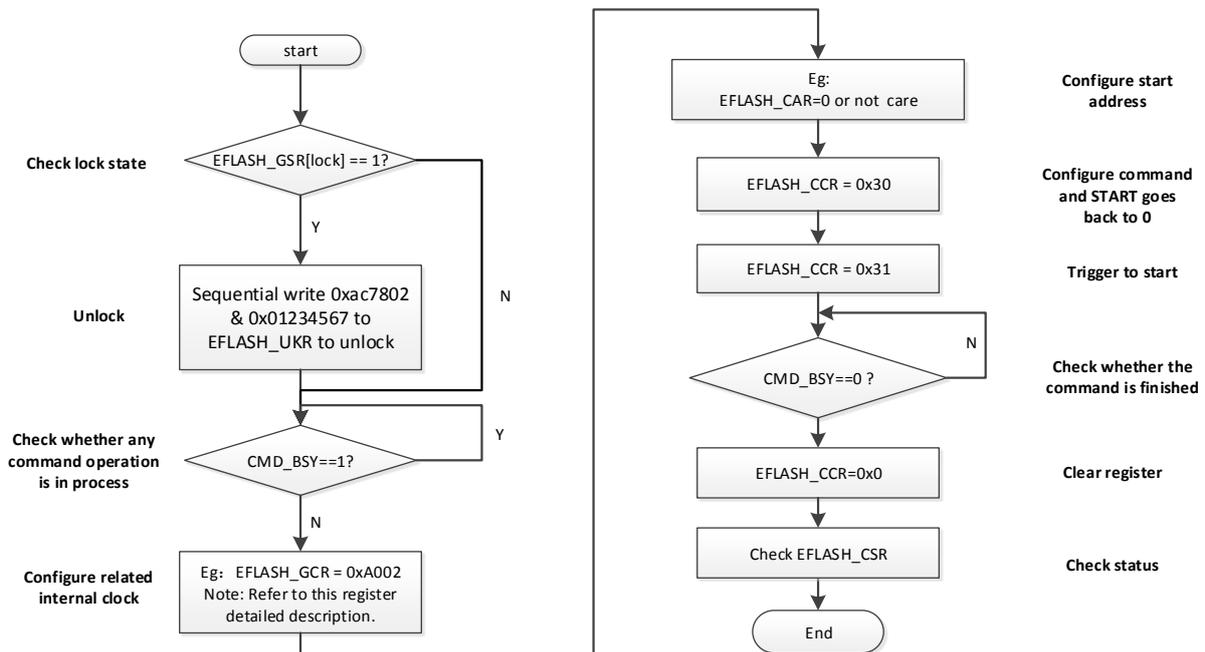


图 19-5 整片擦除命令操作流程

19.5.4 页编程

页编程命令可用在 eflash 中的 P-Flash 存储器和 D-Flash 存储器，流程如图 19-6 所示。页编程命令流程也类似于页擦除，但 CMD 值不同、且需要配置编程长度和编程数据，具体步骤如下：

1. 配置编程地址值到 FLASH_CAR 寄存器；
2. 配置 EFLASH_CCR 寄存器 LEN [15: 0]位来配置页编程命令写入的字长度；
3. 配置 EFLASH_CCR 寄存器 CMD 位为 0x5；
4. 配置 EFLASH_CCR 寄存器 START 位为 1，即启动编程命令；
5. 软件判断 EFLASH_CSR 寄存器 DBUFRDY 等于 1 时候，将要编程的数据写入 EFLASH_CDR 寄存器，直到写完 LEN [15: 0]指定的长度。

注意：

- 1、步骤 1、2、3 顺序可以变化
- 2、LEN [15: 0]位配置为指定值，以字为单位来指定编程长度。例如，如果 LEN [9: 0]配置为 150，则用户应写入不超过 150 个字。如果用户写入超过 150 字，例如 180 字，则最后 30 字实际上不会写入 eflash。同时，如果用户写入少于 150 字，如 110 字，则写操作将会验证正常，但编程过程不会结束，直到数据数目等于长度或使用 abort 命令强制编程终止。

关于页编程命令，有一些自动预检机制可以进行内容保护，如写保护检查，空白内容检查和地址边界检查等。因此，用户应在每个程序命令后仔细检查 EFLASH_CSR 寄存器，以确认写操作是否成功。

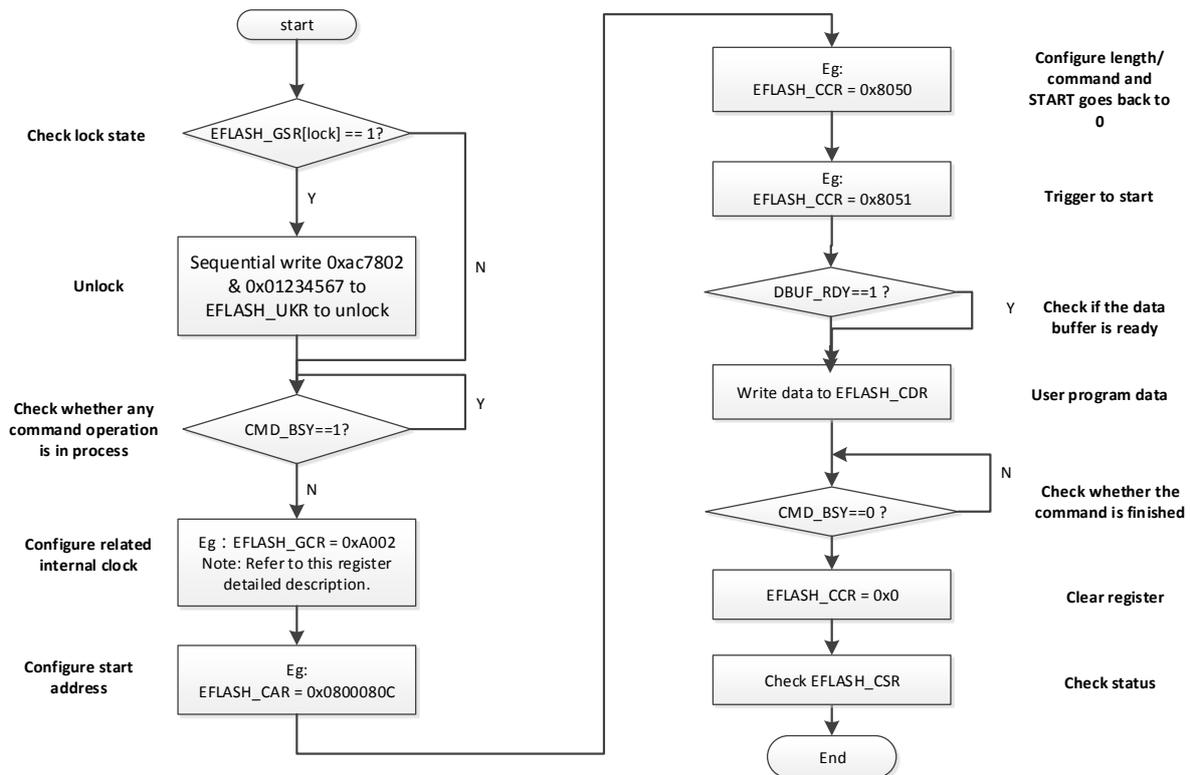


图 19-6 页编程命令操作流程

19.5.5 页擦除验证

页擦除验证命令用于 eflash 中的 P-flash 存储器。该操作通常在擦除操作之后执行，以验证擦除操作是否成功执行，流程如图 19-7 所示。与页擦除命令流程相比，页擦除验证流程只有两个区别：一是 EFLASH_CCR 寄存器 CMD 位设置值不同；二是页擦除可以用于 D-flash 存储器，但页擦除验证不能。

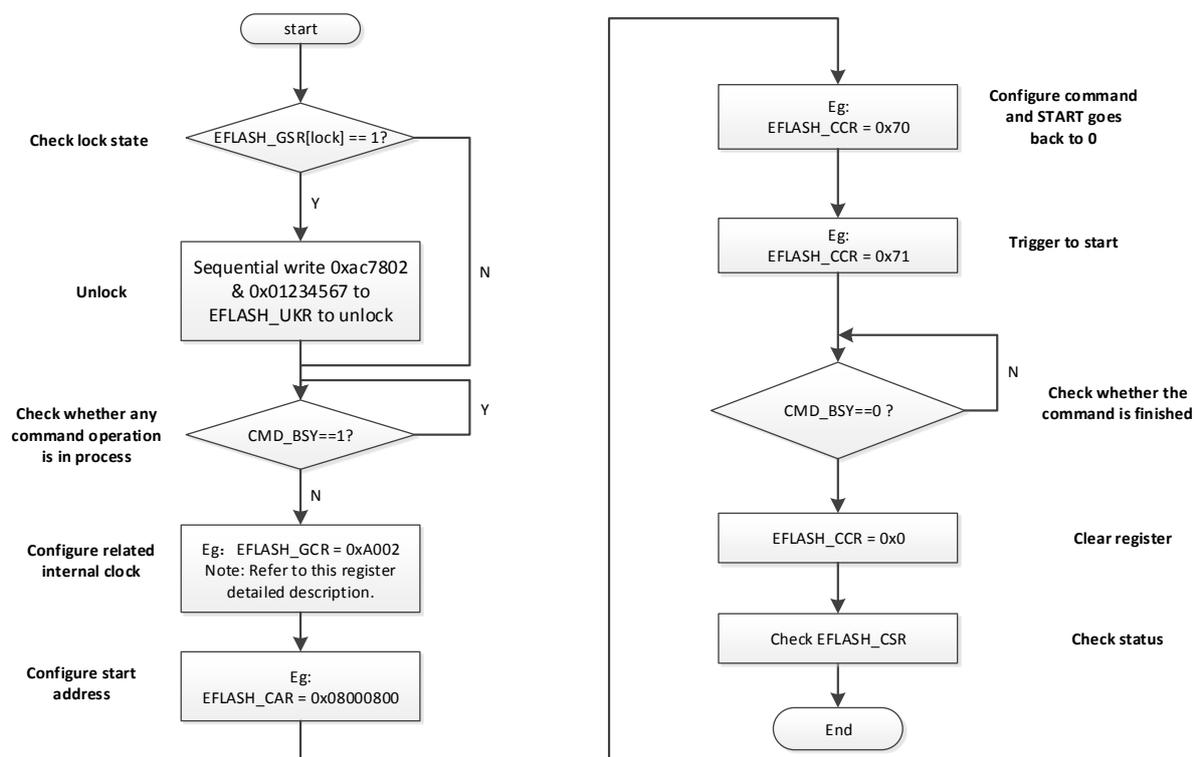


图 19-7 页擦除验证命令操作流程

19.5.6 整片擦除验证

整片擦除验证命令可用于 eflash 中的 P-Flash 存储器 and D-Flash 存储器。该操作通常在整片擦除操作之后执行，以便验证整片擦除操作是否成功完成，流程如图 19-8 所示。与页擦除命令流程相比，整片擦除验证流程有两个不同：一是 EFLASH_CCR 寄存器 CMD 位设置值不同，二是不需在 EFLASH_CAR 寄存器指定擦除地址。由于达到了整个存储器，因此无需为此命令操作指定 EFLASH_CAR。

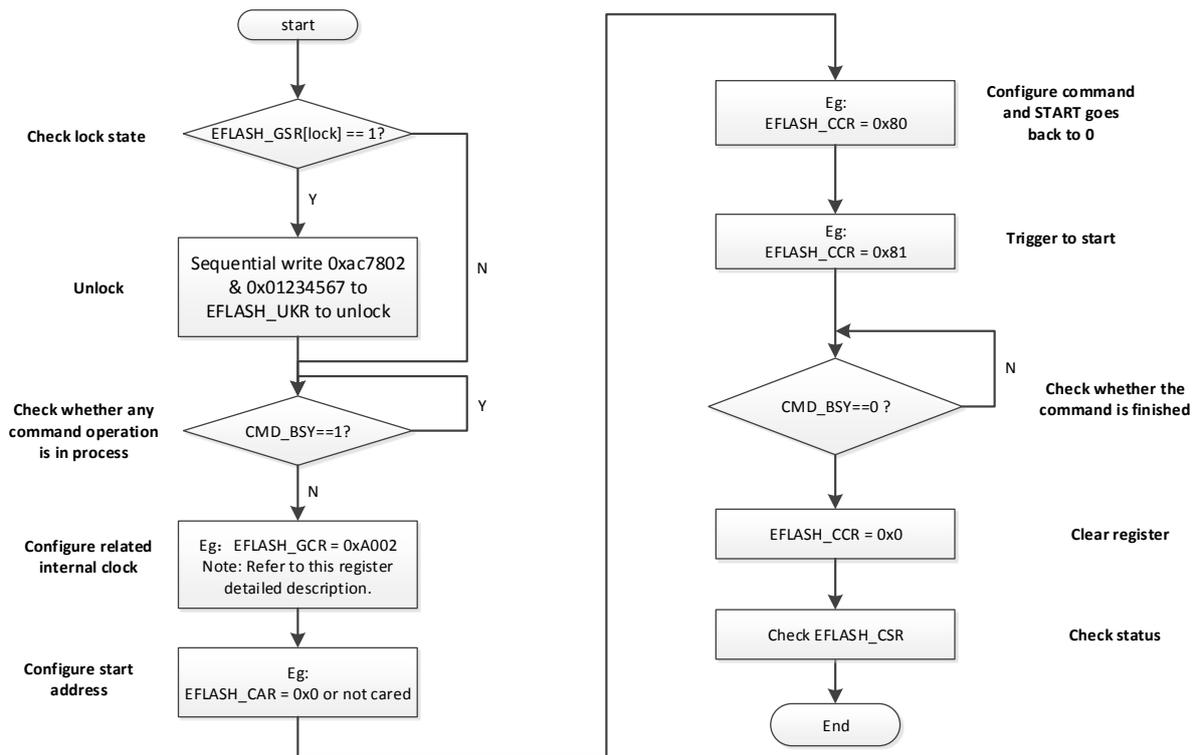


图 19-8 整片擦除验证命令操作流程

19.5.7 选项字节擦除

选项字节擦除命令用于整个选项字节，流程如图 19-9 所示。与页擦除命令流程相比，选项字节擦除流程有两点区别：一是 EFLASH_CCR 寄存器 CMD & OPTEN 位设置值不同，二是在 EFLASH_CAR 寄存器中指定地址为选项字节地址。

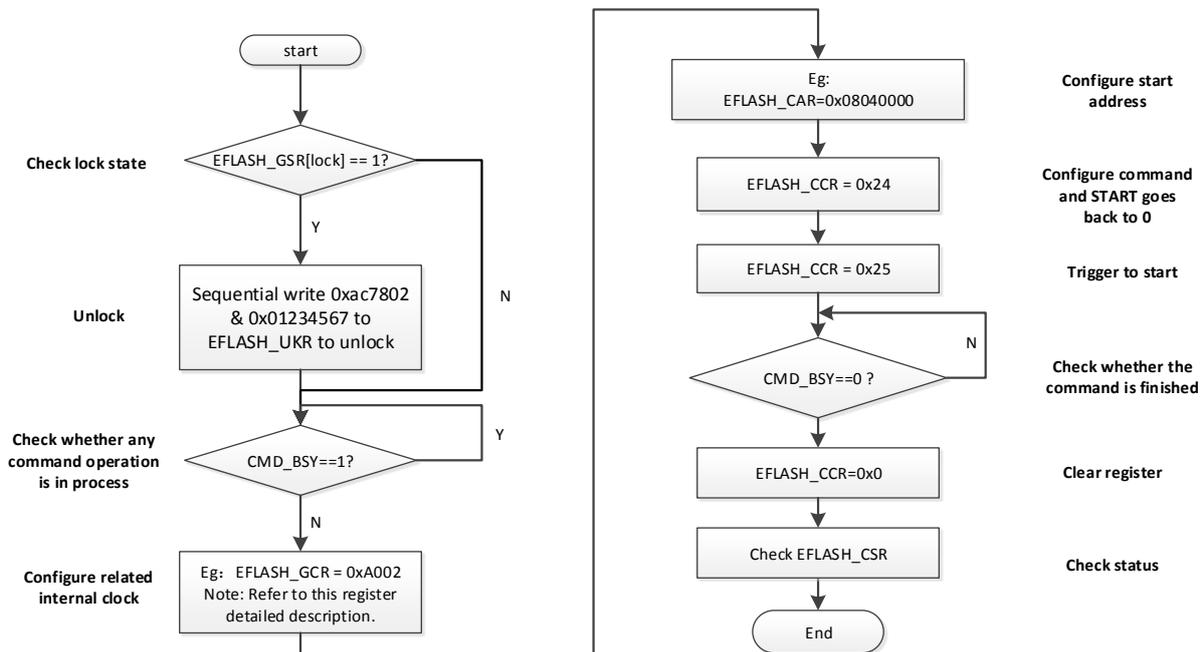


图 19-9 选项字节擦除命令操作流程

19.5.8 选项字节编程

选项字节编程命令用于整个选项字节，流程如图 19-10 所示。与页编程命令流程相比，选项字节编程流程有两点区别：一是 EFLASH_CCR 寄存器 CMD 位设置值不同，二是在 EFLASH_CAR 寄存器中指定地址为选项字节地址。

在进行选项字节编程时，用户应高度重视读取保护字符的更改，如果将读取保护字符从保护更改为不保护，eflash 控制器将自动执行整片擦除命令，此时将会擦除整个用户页的内容。

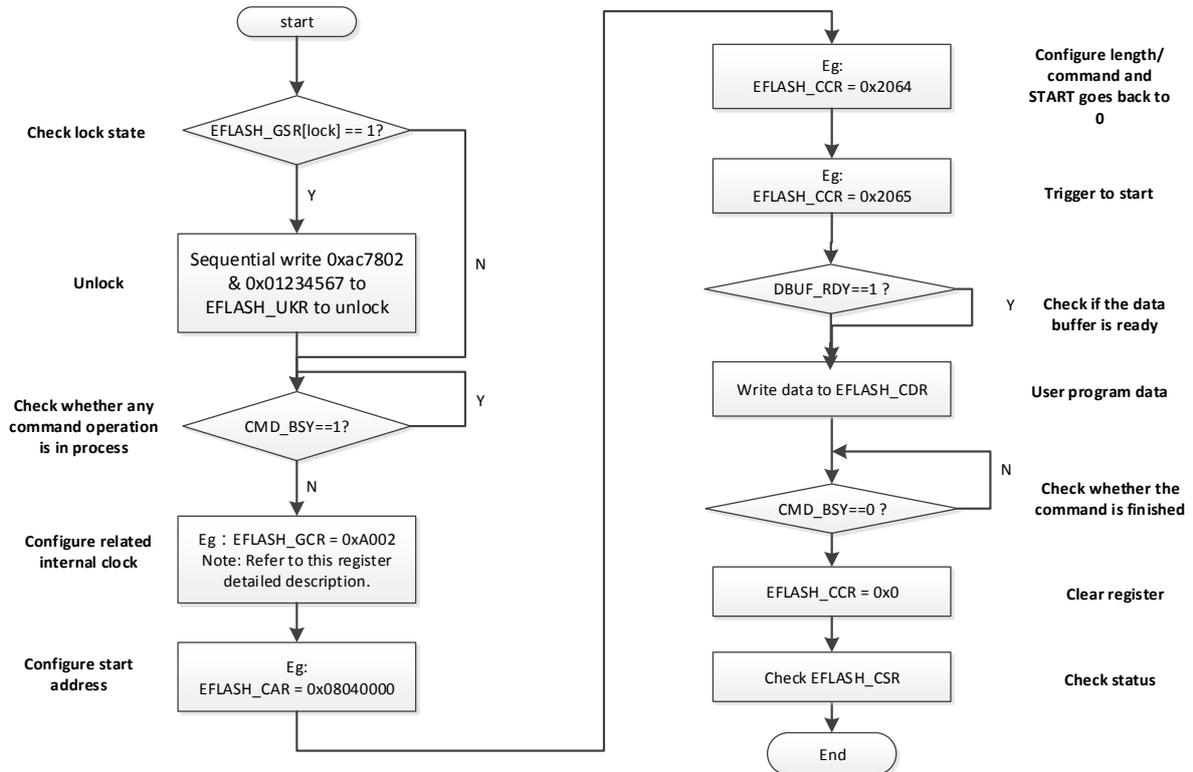


图 19-10 选项字节编程命令操作流程

19.6 寄存器定义

表 19-6 片内 Flash 寄存器映射

EFLASH 基地址: 0x40002000

地址	名称	宽度	描述
EFLASH 基地址+0x0	EFLASH_UKR	32	解锁序列寄存器
EFLASH 基地址+0x4	EFLASH_GSR	32	全局信息寄存器
EFLASH 基地址+0x8	EFLASH_GCR	32	全局控制寄存器
EFLASH 基地址+0xC	EFLASH_CSR	32	命令状态寄存器
EFLASH 基地址+0x10	EFLASH_CCR	32	命令控制寄存器
EFLASH 基地址+0x14	EFLASH_CAR	32	命令地址寄存器
EFLASH 基地址+0x18	EFLASH_CDR	32	命令数据寄存器
EFLASH 基地址+0x40	EFLASH_PWPR0	32	P-flash 写保护寄存器 0
EFLASH 基地址+0x44	EFLASH_PWPR1	32	P-flash 写保护寄存器 1
EFLASH 基地址+0x48	EFLASH_PWPR1	32	D-flash 写保护寄存器

19.6.1 解锁 序列寄存器(EFLASH_UKR)

表 19-7 EFLASH_UKR 寄存器

EFLASH_UKR

解锁 序列寄存器

RESET: 0x00000000

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

字段	说明
31: 0	解锁 eflash 控制寄存器的锁定
KEY	当 EFLASH_GSR[LOCK]位为 1 时, 表示片内 Flash 控制器寄存器被锁定时, 顺序写入 0xac7802 和 0x01234567 以解锁。

19.6.2 全局信息寄存器(EFLASH_GSR)

表 19-8 EFLASH_GSR 寄存器

EFLASH_GSR 全局信息寄存器 RESET: 0x00000001

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称																	
访问																	
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称								RPRT					BUSY				LOCK
访问								RO					RO				RW
Reset								0					0				1

字段	说明
8 RPRT	P-flash 和 D-flash 读保护状态 0: 读保护失能 1: 读保护使能 注意: 该位复位值取决于当前读保护使能状态。
4 BUSY	表示 EFLASH 控制器是否处于忙状态 0: 没有处于忙状态 1: 处于忙状态
0 LOCK	片内 Flash 控制器锁定寄存器 0: 表示片内 Flash 控制器寄存器未锁定, 此时直接写 1 会锁定片内 Flash 控制器 1: 表示片内 Flash 控制器寄存器锁定, 不能直接写 0 来解锁, 只能通过操作寄存器 EFLASH_KEY 来解锁

19.6.3 全局控制寄存器(EFLASH_GCR)

表 19-9 EFLASH_GCR 寄存器

EFLASH_GCR 全局控制寄存器 RESET: 0x00006002

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称	FRE Q_LO CK	RD_ MOD E	FREQ													HDFEN

访问	RW	RW	RW		RW
Reset	0	0	0x20		0

字段	说明
15 FREQ_LOCK	FREQ 和 RD_MOD 寄存器锁定寄存器 0: 未锁定 1: 锁定 注意: 如果上电后该位写了 1, 那么直到断电再上电才可以重新配置。
14 RD_MOD	读取速度是否除以 2 设置 0: flash 控制器频率不除以 2 1: flash 控制器频率除以 2
13:8 FREQ	时钟分频用以产生 1μs 脉冲 必须根据 eflash 控制器的速度配置合理的值才能在以下操作之前获得 1μs 周期: 页编程, 页擦除, 基于命令操作流程的整片擦除。例如, 如果 eflash 控制器时钟频率为 32MHz, CKDIV= 32/1 = 32= 0x20, 但是建议配置的这个值最好是比 0x20 大一点, 比如 0x21。
0 HDFEN	当触犯读保护规则时候是否产生 hardfault 控制 0: 失能产生 hardfault 1: 使能产生 hardfault

19.6.4 命令状态寄存器(EFLASH_CSR)

表 19-10 EFLASH_CSR 寄存器

EFLASH_CSR				命令状态寄存器								RESET: 0x00000000				
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称				OP TD AT ER	OPT DW ERR		OPT PW ERR	OPT RER R								
访问				RO	RO		RO	RO								
Reset				0x0	0x0		0x0	0x0								
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称							VRF ERR	ERS ERR	PG ME RR	DRV IO	DW VIO	PRV IO	PW VIO	DB UFL ST	DB UFR DY	C M DB US Y
访问							RO	RO	RO	RW	RW	RW	RW	RO	RO	RO
Reset							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

字段	说明
28 OPTDATERR	选项字节 DATAx 验证错误 0: 无错误 1: 有错误
27 OPTDWERR	选项字节 D_WPRT_EN 验证错误 0: 无错误 1: 有错误
25 OPTPWERR	选项字节 P_WPRT_EN 验证错误 0: 无错误 1: 有错误
24 OPTRERR	选项字节 RDP 验证错误 0: 无错误 1: 有错误
9 VRFERR	表示验证命令操作中是否存在错误 0: 没有错误 1: 数据验证存在问题
8 ERSERR	表示擦除命令操作中是否存在错误 0: 没有错误 1: 存在错误, 因为 EFLASH_CAR 中的地址是非法的
7 PGMERR	表示编程命令操作中是否存在错误 0: 没有错误 1: 存在错误, 因为程序地址非法或在程序命令操作不正常之前验证操作
6 DRVIO	表示 D-flash 存储器操作违反了读保护规则 0: 不违反读保护规则 1: 违反读保护规则 写 1 将 DRVIO 清除为 0.
5 DWVIO	表示 D-flash 存储器操作违反了写保护规则 0: 不违反写保护规则 1: 违反写保护规则 写 1 将 DWVIO 清除为 0.
4 PRVIO	表示 P-flash 存储器操作违反了读保护规则 0: 不违反读保护规则

字段	说明
	1: 违反读保护规则
	写 1 将 PRVIO 清除为 0.
3 PWVIO	表示 P-flash 存储器操作违反了写保护规则
	0: 不违反写保护规则 1: 违反写保护规则
	写 1 将 PWVIO 清除为 0.
2 DBUFLST	表示命令操作是否达到数据 buffer 最后
	0: 不是数据 buffer 最后 1: 是数据 buffer 最后
1 DBUFRDY	表示数据 buffer 是否就绪
	0: DAT_BUF 写未就绪 1: DAT_BUF 写就绪
0 CMDBSY	表示是否正在执行任何命令操作
	0: 所有操作都没有进行中 1: 至少有一个操作在进行中

19.6.5 命令控制寄存器(EFLASH_CCR)

表 19-11 EFLASH_CCR 寄存器

EFLASH_CCR									命令控制寄存器								RESET: 0x00000000			
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
名称	LEN								LEN											
访问	RW								RW								RW			
Reset	0x0								0x0								0x0			
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
名称	LEN								CMD				SE L	OP TE N	AB OR T	ST AR T				
访问	RW								RW				RW	RW	RW	RW				
Reset	0x0								0x0				0x0	0x0	0x0	0x0				

字段	说明
23: 8 LEN	编程操作长度 页编程、选项字节编程、页擦除验证命令的编程长度。 注意：单位是字(4 字节)，且最大不能超出存储器边界

字段	说明
7:4 CMD	<p>FLASH 命令类型选择</p> <p>0x0: 空闲 0x1: 页擦除 0x2: 选项字节擦除 0x3: 整片擦除 0x4: 块擦除 0x5: 页编程 0x6: 选项字节编程 0x7: 页擦除验证 0x8: 整片擦除验证 其他值: 保留</p> <p>注意: CMD 取值的有效性需结合 OPTEN 的配置。</p>
3 SEL	<p>命令操作区域选择</p> <p>0: 选择 P-flash 存储器 1: 选择 D-flash 存储器</p> <p>注意: SEL 只对整片擦除和整片擦除验证有效。</p>
2 OPTEN	<p>使能选项字节区域相关的命令操作</p> <p>0: 寄存器域 CMD 取值 0x6 或 0x2 无效, 其余取值有效 1: 寄存器域 CMD 取值 0x6 或 0x2 有效, 其余取值无效</p> <p>注意: 参考 CMD。</p>
1 ABORT	<p>终止当前命令控制</p> <p>0: 不终止当前命令 1: 终止当前命令</p> <p>注意: 写 1 终止当前命令, 硬件自动清零。</p>
0 START	<p>控制起始命令操作</p> <p>写 1 以触发起始命令且会清除 EFLASH_CSR 寄存器的错误状态位[9: 7]。</p>

19.6.6 命令地址寄存器(EFLASH_CAR)

表 19-12 EFLASH_CAR 寄存器

EFLASH_CAR	命令地址寄存器																RESET: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	ADDR																
访问	RW																
Reset	0																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	ADDR																
访问	RW																
Reset	0																

字段	说明
31: 0 ADDR	<p>起始地址命令</p> <p>(1) 页擦除 ADDR [31: 0]是 eflash 地址绝对地址，就在这个页中。例如，如果用户想要擦除从 0x0800 7E00 到 0x0800 7FFF 的第 63 页，那么，用户可以将 ADDR [31: 0]配置为从 0x0800 7E00 到 0x0800 7FFF 的任何值。</p> <p>(2) 编程 ADDR [31: 0]为编程起始地址。ADDR [31: 0]和 LEN[15: 0]共同确定可编程地址范围。</p> <p>(3) 页验证 ADDR [31: 0] 是该页第一个 eflash 地址。</p> <p>(4) 整片擦除/验证 ADDR[31: 0] 未关注。</p> <p>注意：在使用擦除/编程/验证命令时，必须配置起始地址。</p>

19.6.7 命令数据寄存器(EFLASH_CDR)

表 19-13 EFLASH_CDR 寄存器

EFLASH_CDR	命令数据寄存器																RESET: 0x00000000
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	DATA																
访问	RW																
Reset	0																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	DATA																
访问	RW																
Reset	0																

字段	说明
31: 0 DATA	编程命令的数据

19.6.8 P-flash 写保护寄存器 0 (EFLASH_PWPR0)

表 19-14 EFLASH_PWPR0 寄存器

EFLASH_PWPR0		P-flash 写保护寄存器 0														RESET:	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	PWPR0																
访问	RO																
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	PWPR0																
访问	RO																
Reset																	

字段	说明
31: 0 PWPR0	Bit31:0 分别表示 P-flash 第 31~0 页是否存在写保护 0: 没有写保护 1: 存在写保护 注意: 上表中 Reset 值会受选项字节配置而发生变化

19.6.9 P-flash 写保护寄存器 1 (EFLASH_PWPR1)

表 19-15 EFLASH_PWPR1 寄存器

EFLASH_PWPR1		P-flash 写保护寄存器 1														RESET:	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	PWPR1																
访问	RO																
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	PWPR1																
访问	RO																
Reset																	

字段	说明
31: 0 PWPR1	Bit31:0 分别表示 P-flash 第 63~32 页是否存在写保护 0: 没有写保护 1: 存在写保护 注意: 上表中 Reset 值会受选项字节配置而发生变化

19.6.10 D-flash 写保护寄存器 (EFLASH_DWPR)

表 19-16 EFLASH_DWPR 寄存器

EFLASH_DWPR		D-flash 写保护使能寄存器														RESET:	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
名称	DWPR																
访问	RO																
Reset																	
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
名称	DWPR																
访问	RO																
Reset																	

字段	说明
31: 0 DWPR	<p>Bit31:0 分别表示 D-flash 第 32~0 region 是否存在写保护，其中一个 region 含 8 页即 64 字节</p> <p>0: 没有写保护 1: 存在写保护</p> <p>注意：上表中 Reset 值会受选项字节配置而发生变化 该寄存器把 D-flash 划分为 32 个区域，D-flash 是 2KB，则每个区域大小为 64B 即 8 页，每一位代表一个区域的写保护状态。</p>

20 SRAM 错误检测纠正 (ECC_SRAM)

20.1 简介

ECC_SRAM 的全称是 SRAM Error Correcting Code, 是用于 SRAM 的差错检测和修正。SRAM 出错的时候一般不会造成整个 SRAM 不能读取或是全部出错, 而是整个 SRAM 中只有一个或几个 bits 出错。ECC_SRAM 采用汉明码 ECC 单 bit 纠错、两 bits 检测算法, 计算速度很快, 对 1 bit 以上的错误无法纠正, 对 2 bits 以上的错误不保证能检测。

20.2 特性

- 支持 1 bit 和 2 bits 检错
 - 可以检出 1 bit 和 2 bits 错误状态
 - 软件可以通过寄存器读取出错的地址以及错误的状态
- 支持 1 bit 错误状态纠错
 - 当检测出 1 bit 错误状态时, 硬件自动纠错, 不产生中断和复位
 - 2 bits 错误状态硬件不可以纠错
- 支持 2bit 错误状态中断
 - 软件可以配置当 ECC_SRAM 检测到 2 bits 错误时产生中断
 - 软件可以配置在 ECC_SRAM 检测到 2 bits 错误时是否使能系统复位
- 2 bits 以上错误状态不能保证检测

20.3 功能描述

1. ECC_SRAM 默认使能且不可禁用。
2. 当使能 ERR2_IRQEN=1 且 ECC_SRAM 检测到 2 bits 错误时将会产生中断。当发生中断时,
 - 中断标志寄存器 ERR2_STATUS=1
 - ECC_SRAM 状态寄存器 ERR_STATUS=1
 - ECC_SRAM 错误地址寄存器 ERR2_ADDR 会自动填充当前产生错误的地址
 - 通过向中断标志寄存器 ERR2_STATUS 写 1 清除该标记位
3. 软件可以查询寄存器 ERR_STATUS 的值来判断当前 ECC_SRAM 错误的状态。
 - 如果 ERR_STATUS=2'b00, 表明 ECC_SRAM 没有检测到错误。

- 如果 ERR_STATUS=2'b01, 表明 ECC_SRAM 检测到 2 bits 错误, 此时软件通过读取寄存器 ERR2_ADDR 获取当前发生 2 bits 错误的地址。
- 如果 ERR_STATUS=2'b10 或者 2b'11, 表明 ECC_SRAM 检测到 1 bit 错误, 此时软件通过读取寄存器 ERR1_ADDR 获取当前发生 1 bit 错误的地址。
- ECC_SRAM 错误状态寄存器 ERR_STATUS 和错误地址寄存器 ERR_ADDRx(x=1,2)在系统不复位情况下会保留最近一次 ECC_SRAM 检查到的错误状态以及错误地址, 软件通过向 ERR_STATUS 写入 2'b11 清除该寄存器以及错误地址寄存器 ERR_ADDRx(x=1,2)。
- 当 RESET_CRTL(0x4000000C)寄存器的 bit 23(ECC2_RST_EN)被编程为 1 且 ECC_SRAM 检测到 2 bits 错误时系统将会产生复位。

20.4 寄存器定义

表 20-1 ECC_SRAM 寄存器映射

ECC_SRAM 基地址: 0x40000020

地址	名称	宽度	描述
ECC_SRAM 基地址+0x0	ECC_SRAM_CTRL	32	ECC 控制以及状态寄存器
ECC_SRAM 基地址+0x4	ECC_SRAM_ERR1_ADDR	32	ECC 1 bit 错误地址寄存器
ECC_SRAM 基地址+0x8	ECC_SRAM_ERR2_ADDR	32	ECC 2 bits 错误地址寄存器

20.4.1 控制以及状态寄存器(ECC_SRAM_CTRL)

表 20-2 ECC_SRAM_CTRL 寄存器

ECC_SRAM_CTRL													ECC_SRAM 控制以及状态寄存器				RESET: 0x00000001	
位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
名称																		
访问																		
Reset																		
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
名称											ERR_STATUS		ERR2_STATUS		ERR2_IRQEN			
访问											R/W1C		R/W1C		RW			
Reset											0		0		0			

字段	说明
5: 4 ERR_STATUS	ECC_SRAM 错误状态 00: 没有错误 01: 2 bits 错误且不能纠正 10/11: 1 bit 错误且可以纠正

字段	说明
	注意：向这两位写入 2'b11 可以清除这两位以及 ERR _x _ADDR(x=1,2)
2 ERR2_STATUS	ECC_SRAM 2 bits 错误状态 注意： 向该位写 1 可以清除该位 如果使能 ERR2_IRQEN 且产生了 2 bits 错误，此时必须写 1 清除该位，否则该模块会一直产生 2 bits 错误中断
1 ERR2_IRQEN	ECC 2 bits 错误中断使能 1: 使能 0: 禁止

20.4.2 1 bit 错误地址寄存器(ECC_SRAM_ERR1_ADDR)

表 20-3 ECC_SRAM_ERR1_ADDR 寄存器

ECC_SRAM_ERR1_ADDR ECC_SRAM 1 bit 错误地址寄存器 RESET: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				ERR1_ADDR [12: 0]												
访问				RO												
Reset				0												

字段	说明
12: 0 ERR1_ADDR	1 bit 错误地址 注意：使用 ERR1_ADDR 的值乘以 4 加上 0x20000000 得到发生 1 bit 错误的 SRAM 地址。

20.4.3 2 bits 错误地址寄存器(ECC_SRAM_ERR2_ADDR)

表 20-4 ECC_SRAM_ERR2_ADDR 寄存器

ECC_SRAM_ERR2_ADDR ECC_SRAM 2bit 错误地址寄存器 RESET: 0x00000000

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
名称																
访问																
Reset																
位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
名称				ERR2_ADDR [12: 0]												
访问				RO												
Reset				0												

字段	说明
12: 0 ERR2_ADDR	2 bits 错误地址 注意：使用 ERR2_ADDR 的值乘以 4 加上 0x20000000 得到发生 2 bits 错误的 SRAM 地址。

21 调试

21.1 简介

该器件基于 ARM CoreSight 架构进行调试。外部调试器通过调试接口访问寄存器和存储器，控制程序运行/停止/复位等操作。该器件仅支持一个调试器接口，即串行线调试(SWD)。

21.2 特性

- 4 个硬件断点
- 2 个数据观察点
- SWD 接口访问