
智能直流马达专用芯片，内部集成了 32-bit 处理器

- 工作电压范围：2.5V~5.5V
- 工作温度范围：-40℃~85℃
- FLASH
 - 擦写次数：100,000 次 @85℃
 - 数据保持时间：10 年 @85℃
- 32-bit CPU Core
 - 最高 42MHz 工作频率
 - 单周期乘法
- 存储器
 - 32KB Flash
 - 4K SRAM
- 电机专用协处理器
 - PI/DIV/SQRT/SVPWM/CORDIC
- 时钟、复位和电源管理
 - 2~5.5V 供电和 I/O 引脚
 - 上电/掉电复位 (POR/LBOR)
 - 内建出厂校准的 21MHz 的 HRC 振荡器
 - 内建出厂校准的 32KHz 的 RC 振荡器
 - 产生 CPU 和系统时钟的 PLL
 - 4-16Mhz 晶体振荡器
 - 独立看门狗
- 低功耗
 - Sleep 模式
 - Hold 模式
- 12 bit SAR ADC
 - 1uS 转换时间
 - ADC 输入范围：0~AVCC
 - 支持外部 12 路 ADC 输入
 - DC offset cancel
- DMA
 - 3 个独立可配置信道
 - 支持的外设：UART、SPI、I2C、Timer、SRAM、Soft request、SAR-ADC
- GPIO
 - 默认输入上拉，出厂为高阻态
 - 支持 TTL 电平
 - 具有输入负脉冲滤波功能
- 调试模式
 - 串行单线调试 (SWD)
- 多达 6 个定时器
 - 1 个 16 位 4 通道高级控制定时器，有 4 通道 PWM 输出，死区生成和紧急停止功能
 - 系统时间定时器：24 位自减型计数器
 - 4 个通用定时器 16 位 Timer0-3
- 多达 5 个通信接口
 - I2C 接口数：1
 - UART 接口数：2
 - SPI 接口数：1
- 内置 CRC 模块
- 4 个差分输入 OPA
- 5 个比较器
- 采用绿色封装：LQFP48; QFN32; TSSOP28

版本更新说明

版本号	修改内容
V1.0	初版
V1.01	修正目录与页数跳号问题
V1.02	- Flash 控制选项字节说明更新 - 新增寄存器保护机制删除 - HRC 22M 高频方案说明修正 - CMU 寄存器列表修正 - HRCDIV, SYSCLKDIV 寄存器默认值修正 - LFCLKCFG 寄存器说明更新 - CMU_NEWREG_EN 寄存器说明更新 - 删除 PMU 的 PDTFLT 寄存器 - WAKEIF 寄存器删除掉 RCCaliIF - 中断向量表更新 - ME 修正 BEMFA, BEMFB 偏移地址
V1.03	修正 cover page
V1.05	更新 FLASH 规格
V1.06	- CMP CRV_SRC default 1 - TIM8 add DOE 说明 - SADCTRGLEN 的 CH4_EN 和 CH5_EN 位置颠倒
V1.07	- ADC 精度由 12 位改为 11 位（伪差分） - 删除参考电压选择位 SADCREFSSEL[1:0] - 扩展 SADCDAT 和 SDACxDAT 到 20 位，新增具体描述 - 扩展 SADC_CALDATx 和 SADC_TPSOffset 为 16 位有符号数，新增具体描述 - 新增自校准 CAL 操作方法。 - 更新寄存器列表
V1.08	修正 RTC 规格
V1.09	- 更正表 75 关闭状态 OCxN_EN=0 - 新增寄存器标题目录
V1.10	- 更正页脚页码 - 增加 ADC 公式
V1.11	- TIM8_BDTR 寄存器更新 28pin 引脚排列说明更新 17.7.13 后 ADC 寄存器添加名称 - 删除与霍尔传感器接口章节 - 更换 Cover page 位置 - 更新从机选择寄存器（SPISSN） - 更新 CALDATx 校准值寄存器

目录

1	概述.....	12
1.1	简介.....	12
1.2	框图.....	13
1.3	引脚排列.....	14
1.4	引脚定义.....	16
1.5	缩略语.....	19
2	存储器模块.....	20
2.1	概述.....	20
2.2	存储器映射图.....	21
2.3	FLASH 操作	22
2.3.1	Code Flash 的操作说明.....	22
2.4	FLASH 控制功能	23
2.5	写保护寄存器说明.....	25
2.6	特殊功能寄存器列表.....	26
2.7	特殊功能寄存器说明.....	26
2.7.1	写保护寄存器 (WPREG)	26
2.7.2	Flash 控制寄存器 (FLASHCON)	26
2.7.3	Flash 锁定寄存器 (FLASHLOCK)	27
3	时钟单元.....	28
3.1	时钟分类.....	28
3.2	时钟框图.....	28
3.3	时钟停振检测框图.....	30
3.3.1	PLL 分档可调.....	30
3.3.2	停振检测算法.....	31
3.3.3	HRC 22M 高频方案.....	31
3.4	时钟说明.....	32
3.4.1	内部低频 RC 时钟 (Flrc)	32
3.4.2	内部高频 RC 时钟 (Fhrc)	32
3.4.3	外部高频晶振时钟 (Fhse)	32
3.4.4	内部 PLL 时钟 (Fpll)	32
3.4.5	时钟安全机制.....	32
3.4.6	时钟源的起振时间.....	33
3.4.7	时钟异常状态处理.....	33
3.5	特殊功能寄存器列表.....	34
3.6	特殊功能寄存器说明.....	34
3.6.1	写保护寄存器(WPREG).....	34
3.6.2	系统时钟配置寄存器(SYSCLKCFG).....	35
3.6.3	芯片状态指示寄存器(JTAGSTA).....	35
3.6.4	LRC 时钟调整寄存器 (LRCADJ)	36
3.6.5	HRC 时钟调整寄存器 (HRCADJ)	36
3.6.6	HRC 时钟分频寄存器 (HRCDIV)	37
3.6.7	时钟状态寄存器 (CLKSTA)	37
3.6.8	系统时钟分频寄存器 (SYSCLKDIV)	38
3.6.9	CLKOUT 时钟选择寄存器 (CLKOUTSEL)	38
3.6.10	CLKOUT 时钟分频寄存器 (CLKOUTDIV)	39
3.6.11	内部模块使能控制寄存器 0 (CLKCTRL0)	39
3.6.12	内部模块使能控制寄存器 1 (CLKCTRL1)	41
3.6.13	指令预取使能控制寄存器 (PREFETCH)	42
3.6.14	芯片版本寄存器 (CHIPID)	43
3.6.15	PLL 锁定条件设置寄存器 (PLLLOCK_PRE)	43
3.6.16	多功能配置寄存器 (MULTFUNCFG)	44

3.6.17	低频时钟源配置寄存器 (LFCLKCFG)	45
3.6.18	PLL 时钟频率产生控制寄存器 (C_PLL)	45
3.6.19	时钟滤波控制寄存器 (FLTCTR)	46
3.7	应用场景	48
3.7.1	Flrc 应用	48
3.7.2	Fhrc 应用	48
3.7.3	Fp11 应用	48
3.7.4	时钟切换	49
3.7.5	停振检测判断	49
3.7.6	时钟异常及 NMI 处理	49
4	电源单元	50
4.1	概述	50
4.2	修改寄存器概述	50
4.3	框图	50
4.4	电源单元详细功能说明	51
4.4.1	电源	51
4.4.2	电源实时监测	51
4.4.3	内建 1.5V 电源	51
4.4.4	BOR 检测功能 (BOR_DET)	51
4.4.5	系统电源检测功能 (VCC_DET)	53
4.4.6	VCC_DET, BOR_DET 分时检测时序	54
4.5	特殊功能寄存器列表	55
4.6	特殊功能寄存器说明	55
4.6.1	PMU 配置寄存器 (PMUCON)	55
4.6.2	电源检测阈值配置寄存器 (VDETCFG)	55
4.6.3	电源检测周期配置寄存器 (VDETPCFG)	56
4.6.4	PMU 中断使能寄存器 (PMUIE)	57
4.6.5	PMU 中断标志寄存器 (PMUIF)	58
4.6.6	PMU 状态寄存器 (PMUSTA)	58
5	调试支持	60
5.1	概况	60
5.2	SW 引脚分布	60
5.3	SW 口使用说明	60
5.4	测试模式定义	61
5.5	应用场景	61
6	工作模式	62
6.1	工作模式	62
6.2	深眠模式 (SLEEP)	62
6.2.1	Sleep 模式下各模块开关	62
6.2.2	Sleep 模式下的唤醒	63
6.2.3	从 Sleep 模式唤醒后的唤醒方式确认	63
6.2.4	进入 Sleep 模式	63
6.3	待机模式 (HOLD)	63
6.3.1	进入 Hold 模式	64
6.4	模式转换图	64
6.5	特殊功能寄存器列表	64
6.6	特殊功能寄存器说明	65
6.6.1	系统控制寄存器 (SCR)	65
6.6.2	唤醒标志寄存器 (WAKEIF)	65
7	GPIO 模块	66

7.1	概述.....	66
7.2	引脚排列.....	67
7.3	引脚定义.....	69
7.4	芯片引脚结构说明.....	71
7.5	关于高阻状态的说明.....	71
7.6	I/O 端口基地址列表.....	72
7.7	特殊功能寄存器说明.....	73
7.7.1	端口功能配置寄存器 1(IOCFG).....	73
7.7.2	端口复用功能配置寄存器 1(AFCFG1).....	73
7.7.3	端口复用功能配置寄存器 2(AFCFG2).....	73
7.7.4	端口方向配置寄存器(PTDIR).....	74
7.7.5	端口上拉配置寄存器(PTUP).....	74
7.7.6	端口数据寄存器(PTDAT).....	75
7.7.7	端口设置寄存器(PTSET).....	75
7.7.8	端口复位寄存器(PTCLR).....	76
7.7.9	端口翻转寄存器(PTTOG).....	76
7.7.10	端口开漏配置寄存器(PTOD).....	77
7.7.11	模拟功能输入滤波控制寄存器(FILT).....	77
7.7.12	端口高阻控制寄存器(HIIPM).....	77
7.7.13	新增复用功能生效控制寄存器 (NEWAFEN).....	78
7.8	应用场景.....	78
8	中断模块.....	80
8.1	中断向量说明.....	80
8.2	EXTI 中断说明.....	81
8.3	特殊功能寄存器列表.....	81
8.4	特殊功能寄存器说明.....	82
8.4.1	外部中断边沿配置寄存器(EXTIE).....	82
8.4.2	外部中断标志寄存器(EXTIF).....	82
8.4.3	引脚数字滤波使能寄存器(PINFLT).....	83
8.4.4	外部中断边沿配置寄存器 2(EXTIE2).....	83
8.4.5	外部中断标志寄存器 2(EXTIF2).....	84
8.4.6	引脚数字滤波使能寄存器 2(PINFLT2).....	84
9	复位模块.....	85
9.1	复位优先级.....	85
9.2	复位说明.....	85
9.2.1	上电复位.....	86
9.2.2	低电压检测复位.....	86
9.2.3	外部引脚复位.....	86
9.2.4	掉电复位.....	87
9.2.5	看门狗复位.....	87
9.2.6	软复位.....	88
9.2.7	调试复位.....	88
9.2.8	唤醒复位.....	89
9.3	特殊功能寄存器列表.....	89
9.4	特殊功能寄存器说明.....	89
9.4.1	复位标志寄存器(RSTSTA).....	89
10	UART 通讯模块.....	91
10.1	功能说明.....	91
10.2	波特率计算.....	91
10.3	串口通讯模式说明.....	91
10.3.1	方式 1.....	91

10.3.2	方式 2.....	92
10.3.3	方式 3.....	93
10.3.4	方式 4.....	93
10.4	特殊功能寄存器列表.....	94
10.5	特殊功能寄存器说明.....	94
10.5.1	串口功能选择寄存器(MODESEL).....	94
10.5.2	UART 功能配置寄存器(UARTCON).....	94
10.5.3	串口波特率发生寄存器(SREL).....	96
10.5.4	串口数据缓冲寄存器(SBUF).....	96
10.5.5	UART 状态寄存器(UARTSTA).....	96
11	WDT 模块.....	98
11.1	概述.....	98
11.2	工作模式.....	98
11.3	特殊功能寄存器列表.....	98
11.4	特殊功能寄存器说明.....	99
11.4.1	WDT 喂狗与时间配置寄存器(WDTCCLR).....	99
11.4.2	WDT 计数寄存器(WDTCNT).....	99
12	TIMER 通用定时器模块.....	100
12.1	定时器单元概述.....	100
12.2	周期定时功能.....	100
12.3	PWM 功能.....	101
12.4	捕获功能.....	103
12.5	事件计数功能.....	103
12.6	中断功能.....	103
12.6.1	周期定时中断.....	103
12.6.2	捕获中断.....	104
12.6.3	比较中断.....	104
12.6.4	事件计数中断.....	104
12.7	特殊功能寄存器列表.....	105
12.8	特殊功能寄存器说明.....	105
12.8.1	定时器控制寄存器(TMRCON).....	105
12.8.2	预分频寄存器(TMRDIV).....	106
12.8.3	周期寄存器(TMRPRD).....	106
12.8.4	捕获数据寄存器(TMRCAP).....	107
12.8.5	计数寄存器(TMRCNT).....	107
12.8.6	比较寄存器(TMRCMP).....	107
12.8.7	定时器中断使能寄存器(TMRIE).....	108
12.8.8	定时器中断标志寄存器(TMRIF).....	108
13	TIM8 高级定时器模块.....	110
13.1	TIM8 简介.....	110
13.2	TIM8 主要特性.....	110
13.3	TIM8 功能描述.....	111
13.3.1	时基单元.....	111
13.3.2	计数器模式.....	113
13.3.3	重复计数器.....	122
13.3.4	时钟选择.....	123
13.3.5	捕获/比较通道.....	126
13.3.6	输入捕获模式.....	128
13.3.7	PWM 输入模式.....	129
13.3.8	强置输出模式.....	130
13.3.9	输出比较模式.....	130

13.3.10	PWM 模式	131
13.3.11	互补输出和死区插入	134
13.3.12	使用刹车功能	136
13.3.13	在外部事件时清除 OCxREF 信号	139
13.3.14	产生六步 PWM 输出	139
13.3.15	单脉冲模式	140
13.3.16	编码器接口模式	142
13.3.17	定时器输入异或功能	144
13.3.18	TIMx 定时器和外部触发的同步	145
13.4	TIM8 寄存器说明	148
13.4.1	控制寄存器(TIM8_CR1)	148
13.4.2	控制寄存器 2(TIM8_CR2)	149
13.4.3	控制寄存器(TIM8_SMCR)	151
13.4.4	控制寄存器(TIM8_DIER)	153
13.4.5	状态寄存器(TIM8_SR)	154
13.4.6	控制寄存器(TIM8_EGR)	157
13.4.7	控制寄存器(TIM8_CCMR1)(输出比较模式)	159
13.4.8	控制寄存器(TIM8_CCMR1)(输入捕获模式)	160
13.4.9	控制寄存器(TIM8_CCMR2)(输出比较模式)	161
13.4.10	控制寄存器(TIM8_CCMR2)(输入捕获模式)	162
13.4.11	捕获/比较使能寄存器(TIM8_CCER)	163
13.4.12	计数器(TIM8_CNT)	165
13.4.13	预分频器(TIM8_PSC)	166
13.4.14	自动重载寄存器(TIM8_ARR)	166
13.4.15	重复计数寄存器(TIM8_RCR)	166
13.4.16	捕获/比较寄存器 1(TIM8_CCR1)	168
13.4.17	捕获/比较寄存器 2(TIM8_CCR2)	168
13.4.18	捕获/比较寄存器 3(TIM8_CCR3)	168
13.4.19	捕获/比较寄存器 4(TIM8_CCR4)	169
13.4.20	刹车和死区寄存器(TIM8_BDTR)	169
13.4.21	DMA 控制寄存器(TIM8_DCR)	171
13.4.22	连续模式的 DMA 地址(TIM8_DMAR)	172
13.4.23	捕获/比较寄存器 5(TIM8_CCR5)	172
13.4.24	捕获/比较模式寄存器 3(TIM8_CCMR3)(输出比较模式)	173
13.4.25	控制寄存器(TIM8_CCMR3)(输入捕获模式)	173
14	SPI 模块	175
14.1	概述	175
14.2	详细功能说明	175
14.2.1	SPI 主要特征	175
14.2.2	SPI 模块框图	176
14.2.3	SPI 接口传输格式	176
14.2.4	主机模式传输格式	176
14.2.5	从机模式传输格式	178
14.2.6	中断功能	179
14.3	特殊功能寄存器列表	179
14.3.1	控制寄存器(SPICON)	179
14.3.2	状态寄存器(SPISTA)	180
14.3.3	数据寄存器(SPIDAT)	181
14.3.4	从机选择寄存器(SPISSN)	181
14.3.5	SPI cs/delay 控制寄存器(CSDLY)	182
15	I2C 模块	183
15.1	概述	183

15.2	框图.....	183
15.3	功能描述.....	184
15.3.1	操作模式.....	184
15.3.2	串行时钟生成.....	184
15.3.3	中断生成.....	184
15.3.4	传输模式.....	184
15.4	特殊功能寄存器列表.....	192
15.5	特殊功能寄存器说明.....	192
15.5.1	I2C 数据寄存器 (I2CDAT)	192
15.5.2	地址寄存器 (I2CADR)	192
15.5.3	控制寄存器 (I2CCON)	193
15.5.4	状态寄存器 (I2CSTA)	193
16	RTC 模块.....	195
16.1	概述.....	195
16.2	功能描述.....	195
16.3	时间和万年历.....	195
16.4	中断功能.....	195
16.5	RTC 指示寄存器读写流程.....	196
16.5.1	读取 RTC 指示寄存器流程.....	196
16.5.2	写入 RTC 指示寄存器流程.....	196
16.6	寄存器复位的相关说明.....	197
16.7	特殊功能寄存器列表.....	198
16.8	特殊功能寄存器说明.....	198
16.8.1	RTC 控制寄存器 (RTCCON)	198
16.8.2	RTC 中断使能寄存器 (RTCIE)	199
16.8.3	RTC 中断标志寄存器 (RTCIF)	199
16.8.4	闹钟寄存器 (ALMR)	200
16.8.5	RTC 定时器 1 寄存器 (RTCTMR1)	201
16.8.6	RTC 定时器 2 寄存器 (RTCTMR2)	201
16.8.7	RTC 秒寄存器 (SECR)	202
16.8.8	RTC 分寄存器 (MINR)	202
16.8.9	RTC 时寄存器 (HOURR)	202
16.8.10	RTC 日寄存器 (DAYR)	203
16.8.11	RTC 月寄存器 (MONTHR)	203
16.8.12	RTC 年寄存器 (YEARR)	203
16.8.13	RTC 周寄存器 (WEEKR)	204
16.8.14	RTC 读控制寄存器 (RTC RD)	204
16.8.15	RTC 写控制寄存器 (RTC WR)	205
17	SAR ADC.....	206
17.1	概述.....	206
17.1.1	特性.....	206
17.1.2	ADC 公式.....	207
17.1.3	模块结构图.....	207
17.2	功能描述.....	208
17.2.1	ADC 模块开关.....	208
17.2.2	ADC 时钟.....	208
17.2.3	ADC 触发方式.....	208
17.2.4	ADC 模拟输入与通道号.....	209
17.2.5	差分输入功能.....	209
17.2.6	ADC 工作模式.....	210
17.2.7	ADC 采样转换时间.....	212
17.2.8	ADC 参考源.....	213

17.2.9	温度传感器	213
17.3	特殊功能寄存器列表	213
17.4	特殊功能寄存器说明	214
17.4.1	SADC 控制寄存器 (SADCCON)	214
17.4.2	SARADC 时钟配置寄存器 (SADCCLKCFG)	215
17.4.3	差分输入配置寄存器 (SADCDFCFG)	216
17.4.4	SADC 序列转换控制寄存器 (SADCSTR)	217
17.4.5	SADC 中断控制寄存器 (SADCIE)	217
17.4.6	SADC 中断标志寄存器 (SADCIF)	218
17.4.7	SADC 采样时间配置寄存器 (SADCSAMP)	218
17.4.8	SADC 采样时间配置寄存器 (SADCSAMP2)	219
17.4.9	SADC 触发源及扫描序列长度寄存器 (SADCTRLEN)	220
17.4.10	SADC 序列信道配置寄存器 (SADCSEQCFG)	220
17.4.11	SADC 通用数据寄存器 (SADC DAT)	221
17.4.12	SADCx 序列数据寄存器 (SADCx DAT) (x = 0~7)	222
17.4.13	SADC_CALCH (SADC 校准通道选择寄存器)	222
17.4.14	SADC_CALDATx (x=0-4) (CALDATx 校准值寄存器)	223
17.4.15	SADC_TPSOffset (TPS 校准值寄存器)	224
17.4.16	SADCCRV (SADC 电流控制寄存器)	224
17.4.17	SADCTAD (SADC 温度滤波值寄存器)	224
17.5	应用说明	225
17.5.1	滑动滤波说明	225
17.5.2	ADC 工作模式切换应用说明	225
17.5.3	SAR ADC 使用注意说明	225
18	CMP 比较器	226
18.1	CMP 简介	226
18.2	比较器功能描述	226
18.2.1	简介	226
18.2.2	比较器开关控制	227
18.2.3	比较器输入和输出	227
18.2.4	比较器用法	227
18.2.5	比较器锁定机制	227
18.2.6	迟滞现象	227
18.3	比较器寄存器说明	228
18.3.1	控制寄存器 (CMPx_CR1)	228
18.3.2	控制寄存器 (CMPx_CR2)	229
18.3.3	CMPx_CAL	230
19	OPAMP 运算放大器	232
19.1	运算放大器简介	232
19.2	运算放大器主要特征	232
19.3	运算放大器功能说明	232
19.3.1	控制寄存器 (OPA_CR)	232
19.3.2	控制寄存器 (OPA_CR2)	233
20	CRC 模块	235
20.1	CRC 概述	235
20.2	特殊功能寄存器列表	236
20.3	特殊功能寄存器说明	236
20.3.1	控制寄存器 (CRCCON)	236
20.3.2	数据寄存器 (CRC DAT)	237
20.3.3	CRC 初始化种子寄存器 (CRCINIT)	237

21	DMA 功能.....	238
21.1	概述.....	238
21.2	功能描述.....	238
21.3	DMA 通道请求列表.....	239
21.4	DMA 数据传输说明.....	240
21.5	特殊功能寄存器列表.....	241
21.6	特殊功能寄存器说明.....	241
21.6.1	DMA 中断使能寄存器(DMAIE).....	241
21.6.2	DMA 中断标志寄存器(DMAIF).....	242
21.6.3	DMA 状态寄存器(CHNSTA).....	242
21.6.4	DMA 通道控制寄存器(CHNxCTL).....	243
21.6.5	DMA 信道源地址寄存器(CHNxSRC).....	244
21.6.6	DMA 通道目的地址寄存器(CHNxTAR).....	244
21.6.7	DMA 通道传输数量寄存器 (CHNxCNT).....	244
21.6.8	DMA 通道已传输数据个数 (CHNxTCCNT).....	244
21.6.9	DMA 通道块传输设置寄存器 (CHNxBULKNUM).....	245
22	CORTEX-M0 内核简要说明.....	246
22.1	概述.....	246
22.2	系统定时器 SysTick.....	246
22.3	中断优先级说明.....	246
22.4	CMSIS 函数说明.....	247
23	电机专用协同处理器.....	248
23.1	DIV 模块.....	248
23.1.1	概述.....	248
23.1.2	除法器的使用方法.....	248
23.1.3	DIV 寄存器.....	249
23.2	SQRT 模块.....	251
23.2.1	概述.....	251
23.2.2	SQRT 使用方法.....	251
23.2.3	SQRT 寄存器.....	251
23.3	PID 模块.....	253
23.3.1	概述.....	253
23.3.2	PID 使用方法.....	253
23.3.3	PID 寄存器.....	253
23.4	SIN COS TABLE.....	257
23.4.1	使用方法.....	257
23.5	坐标转换.....	257
23.5.1	使用方法.....	257
23.6	IPD.....	258
23.6.1	使用方法.....	258
23.7	SMO.....	258
23.7.1	使用方法.....	258
23.8	SVPWM.....	258
23.8.1	使用方法.....	258
23.9	中断功能.....	259
23.10	ME 寄存器.....	260
24	电气规格 (具体参数需芯片测试).....	272
24.1	极限参数.....	272
24.2	功耗参数.....	272
24.2.1	模块功耗.....	272

24.2.2	运行功耗.....	273
24.3	DC/AC 参数.....	273
24.3.1	存储器.....	273
24.3.2	GPIO.....	274
24.3.3	CMU.....	275
24.3.4	PMU.....	275
24.3.5	中断&RESET.....	277
24.3.6	UART/SPI/I2C.....	278
24.3.7	WDT.....	278
24.3.8	TIMER.....	278
24.3.9	CRC.....	279
24.3.10	TPS&ADC.....	279
24.3.11	RTC.....	280
24.3.12	HRC/LRC 自校正.....	281
24.3.13	M0 内核.....	281
25	封装.....	283
25.1	耐焊性.....	283
25.2	封装图.....	283

1 概述

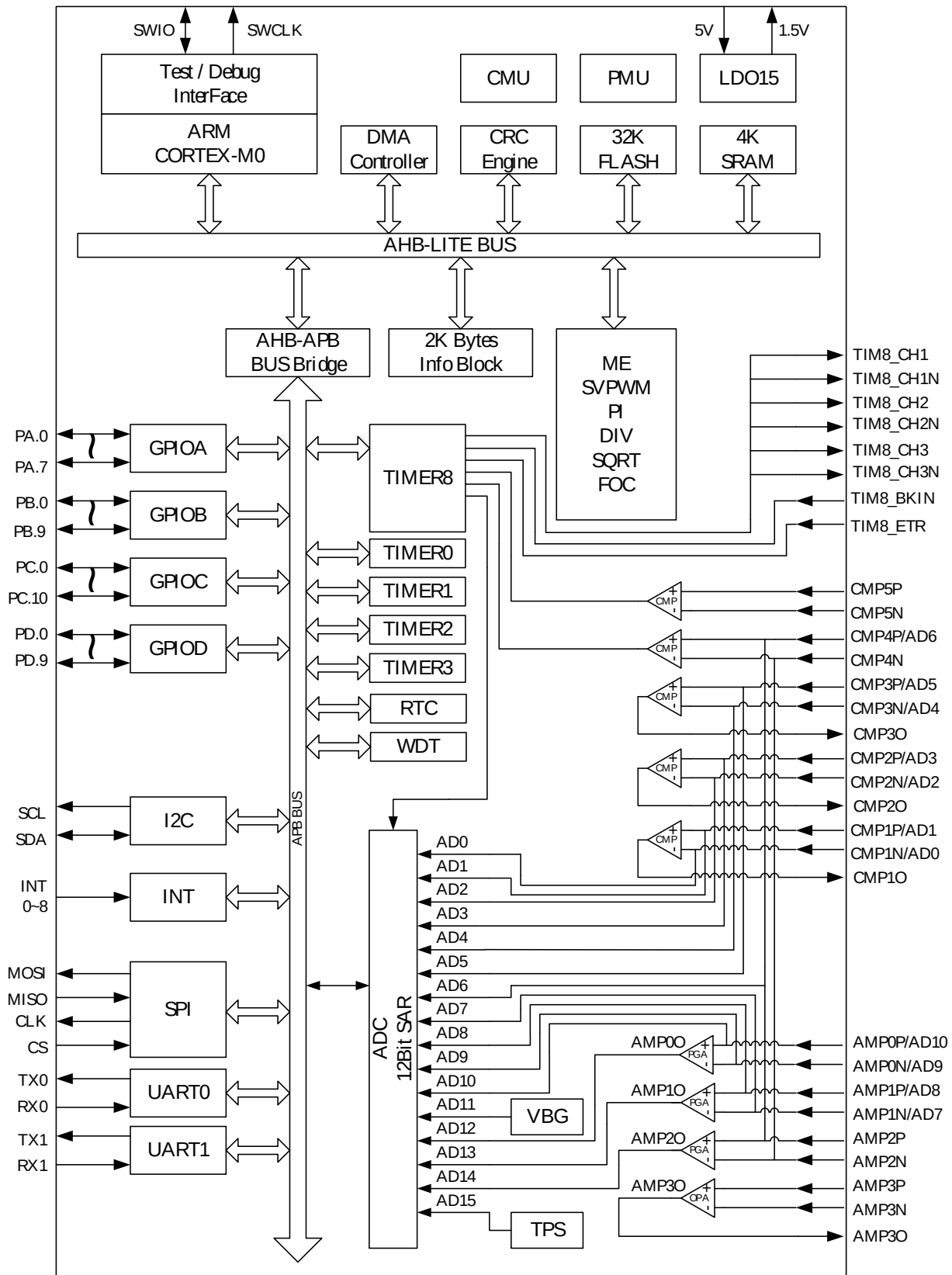
1.1 简介

RX32S10 系列是高性能、低功耗、多功能马达专用 32 位的 MCU 芯片，内部集成了 32-bit 处理器，特别是集成电机专用协同处理器，内置马达专用硬件算法，比纯粹软件运算提供更快速的运算效能，让马达运转的更顺畅更有效率。

芯片更整合了 OP-AMP、Comparator，在使用上能更节省布局空间，更节省成本。功能上还包含时钟管理、电源管理、PLL、高频 RC、低频 RC 等单元，以及 NVIC 和 DEBUG 调试功能。

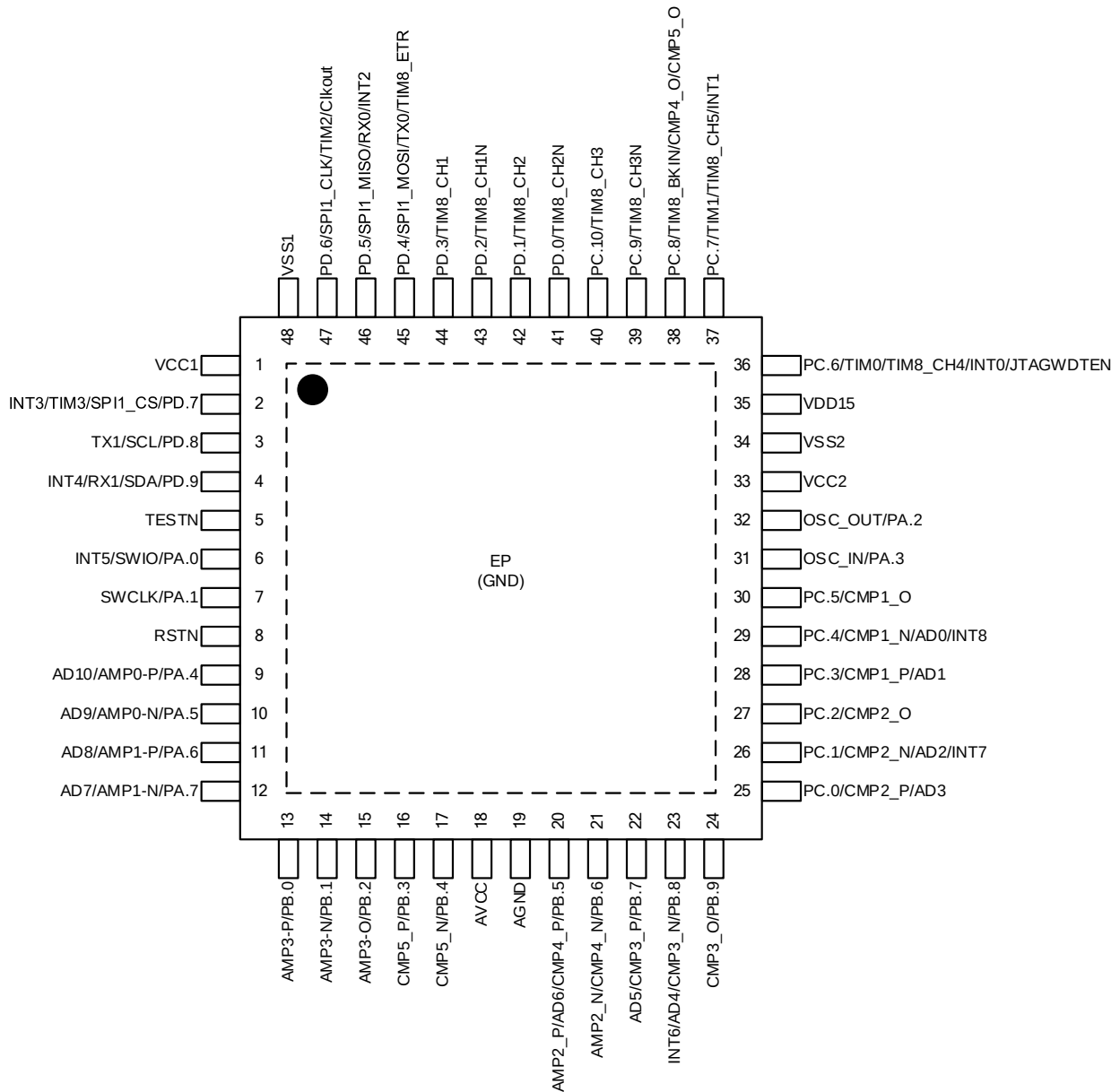
- 工作电压范围：2.5V~5.5V
- 工作温度范围：-40℃~85℃
- FLASH：擦写次数：100,000 次 @85℃、数据保持时间：10 年 @85℃
- 采用 32-bit CPU Core、32K Flash、4K SRAM
- 高速度：CPU 最高工作频率达到 42M（需使能指令预取功能）
- 低功耗：Hold 模式、Sleep 模式
- 高精度温度传感器：-40 度 ~ +125 度温度范围内，温度传感器一致性优于正负 5 度
- 正常模式下，WDT 模块不可关闭，保证系统可靠运行。在 Sleep/hold 模式下，可软件关闭 WDT 模块
- 采用绿色封装：LQFP48、QFN32、TSSOP28

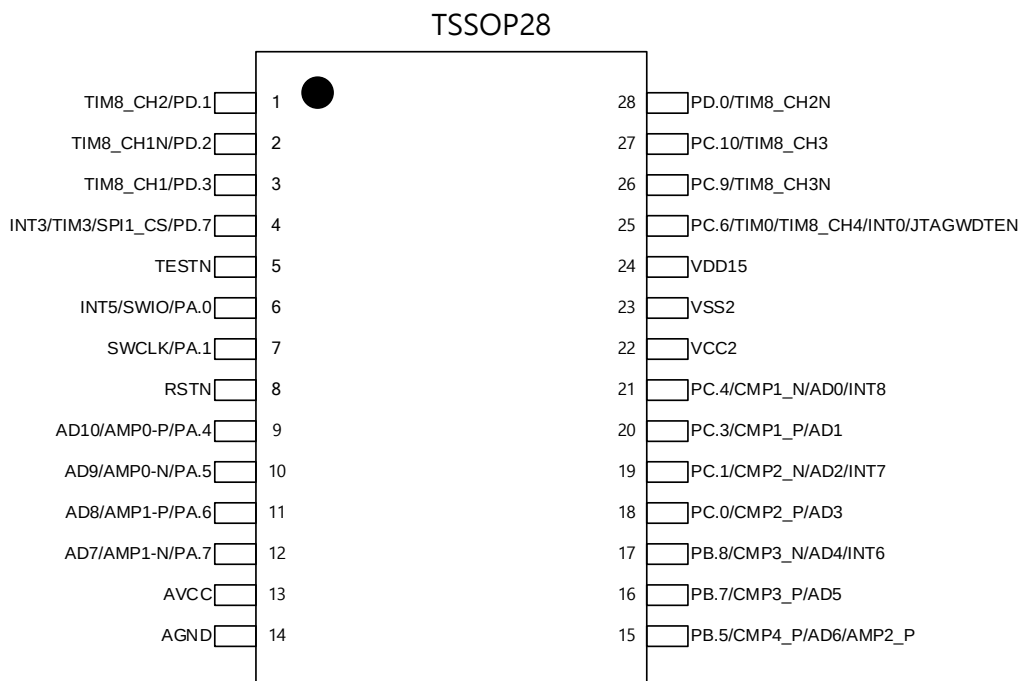
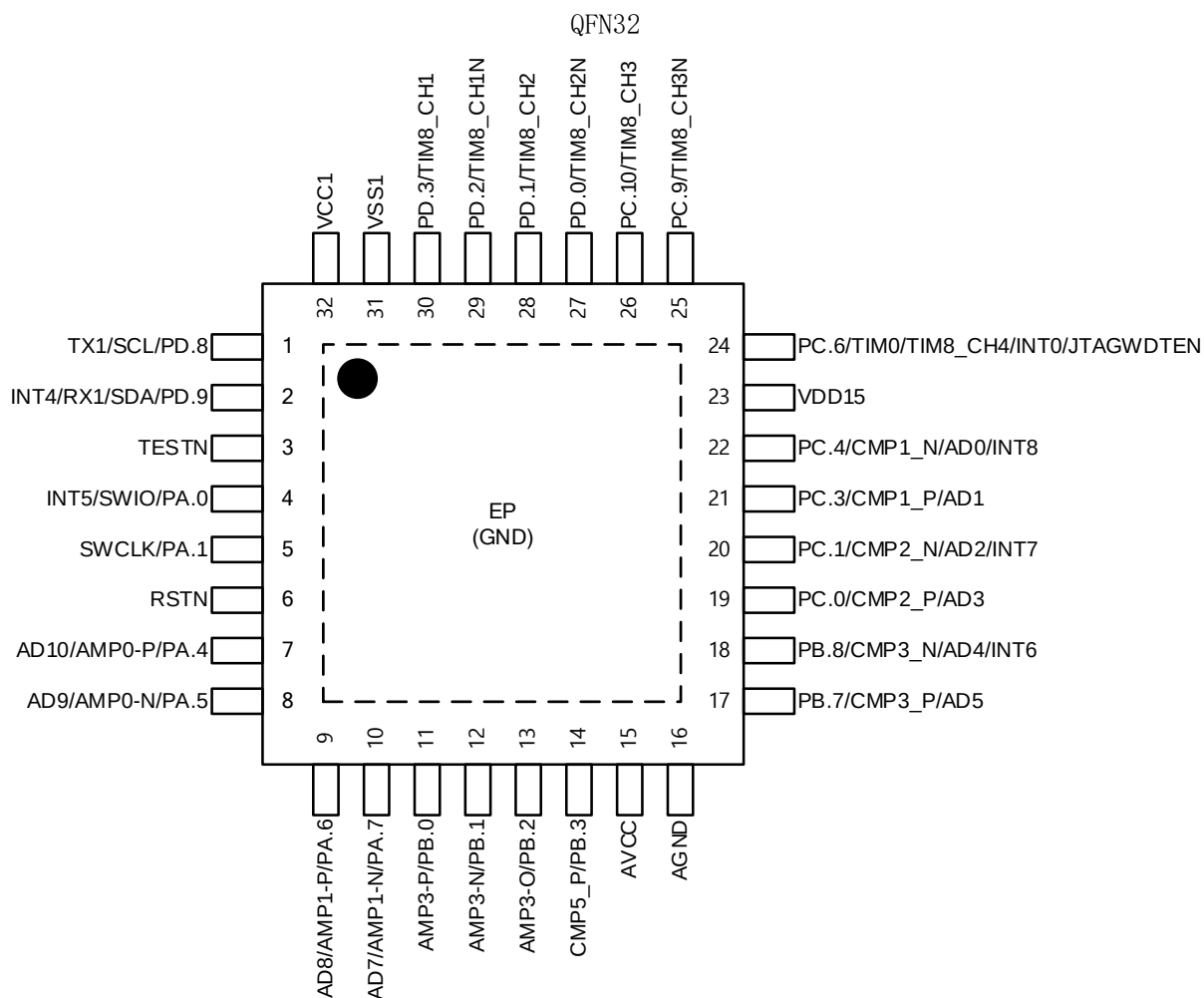
1.2 框图



1.3 引脚排列

LQFP48





1.4 引脚定义

48 PIN	标识	引脚类型	滤波	第一复用功能	第二复用功能	第三复用功能	引脚说明
1	VCC1	P					芯片数字电源
2	PD.7	I/O	2us	SPI1_CS	TIM3	INT3	GPIO\SPI1_CS\Timer输出输入\外部中断, 滤波2us
3	PD.8	I/O		SCL	TX1		GPIO\SCL\TX
4	PD.9	I/O	2us	SDA	RX1	INT4	GPIO\SDA\RX\外部中断, 滤波2us
5	TESTN	I	2us				测试引脚(低电平有效, 内部上拉), 输入滤波2us
6	PA.0	I/O	2us	SWIO		INT5	GPIO\SWIO\外部中断, 滤波2us
7	PA.1	I/O		SWCLK			GPIO\SWCLK
8	RSTN	I	2us				复位信号(低电平有效, 内部上拉), 滤波2us
9	PA.4	I/O		AMP0_P\AD10			GPIO\OP-AMP正端输入\ADC信号输入
10	PA.5	I/O		AMP0_N\AD9			GPIO\OP-AMP负端输入\ADC信号输入
11	PA.6	I/O		AMP1_P\AD8			GPIO\OP-AMP正端输入\ADC信号输入
12	PA.7	I/O		AMP1_N\AD7			GPIO\OP-AMP负端输入\ADC信号输入
13	PB.0	I/O		AMP3_P			GPIO\OP-AMP正端输入
14	PB.1	I/O		AMP3_N			GPIO\OP-AMP负端输入
15	PB.2	I/O		AMP3_O			GPIO\OP-AMP输出
16	PB.3	I/O		CMP5_P			GPIO\比较器正端输入
17	PB.4	I/O		CMP5_N			GPIO\比较器负端输入
18	AVCC	P					芯片模拟电源
19	AGND	G					芯片模拟地
20	PB.5	I/O		CMP4_P\AD6\AMP2_P			GPIO\比较器正端输入\ADC信号输入\OP-AMP正端输入
21	PB.6	I/O		CMP4_N\AMP2_N			GPIO\比较器负端输入\OP-AMP负端输入
22	PB.7	I/O		CMP3_P\AD5			GPIO\比较器正端输入\ADC信号输入
23	PB.8	I/O	2us	CMP3_N\AD4		INT6	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
24	PB.9	I/O		CMP3_O			GPIO\比较器输出
25	PC.0	I/O		CMP2_P\AD3			GPIO\比较器正端输入\ADC信号输入
26	PC.1	I/O	2us	CMP2_N\AD2		INT7	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
27	PC.2	I/O		CMP2_O			GPIO\比较器输出
28	PC.3	I/O		CMP1_P\AD1			GPIO\比较器正端输入\ADC信号输入
29	PC.4	I/O	2us	CMP1_N\AD0		INT8	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
30	PC.5	I/O		CMP1_O			GPIO\比较器输出
31	PA.3	I/O		OSC_In			GPIO\高频晶振时钟输入
32	PA.2	I/O		OSC_Out			GPIO\高频晶振时钟输出
33	VCC2	P					芯片数字电源
34	VSS2	G					芯片数字地
35	VDD15	P					内部1.5V输出, 需外接0.1uF滤波电容
36	PC.6	I/O	2us	TIM0	TIM8_CH4	INT0	GPIO\Timer输出输入\Timer8_CH4\外部中断, 滤波2us\TEST=0该引脚功能为JTAGWDTEN功能
37	PC.7	I/O	2us	TIM1	TIM8_CH5	INT1	GPIO\Timer输出输入\Timer8_CH5\外部中断, 滤波2us
38	PC.8	I/O		TIM8_BKIN	CMP4_O	CMP5_O	GPIO\Timer8_BKIN\比较器4输出\比较器5输出
39	PC.9	I/O		TIM8_CH3N			GPIO\Timer8_CH3N
40	PC.10	I/O		TIM8_CH3			GPIO\Timer8_CH3
41	PD.0	I/O		TIM8_CH2N			GPIO\Timer8_CH2N
42	PD.1	I/O		TIM8_CH2			GPIO\Timer8_CH2
43	PD.2	I/O		TIM8_CH1N			GPIO\Timer8_CH1N
44	PD.3	I/O		TIM8_CH1			GPIO\Timer8_CH1
45	PD.4	I/O		SPI1_MOSI	TX0	TIM8_ETR	GPIO\SPI1_MOSI\TX\Timer8_ETR
46	PD.5	I/O	2us	SPI1_MISO	RX0	INT2	GPIO\SPI1_MISO\RX\外部中断, 滤波2us
47	PD.6	I/O		SPI1_CLK	TIM2	Clkout	GPIO\SPI1_CLK\Timer输出输入\Clkout
48	VSS1	G					芯片数字地



32 PIN	标识	引脚 类型	滤波	第一复用 功能	第二复用 功能	第三复用 功能	引脚说明
1	PD. 8	I/O		SCL	TX1		GPIO\SCL\TX
2	PD. 9	I/O	2us	SDA	RX1	INT4	GPIO\SDA\RX\外部中断, 滤波2us
3	TESTN	I	2us				测试引脚 (低电平有效, 内部上拉), 输入滤波2us
4	PA. 0	I/O	2us	SWIO		INT5	GPIO\SWIO\外部中断, 滤波2us
5	PA. 1	I/O		SWCLK			GPIO\SWCLK
6	RSTN	I	2us				复位信号 (低电平有效, 内部上拉), 滤波2us
7	PA. 4	I/O		AMP0_P\AD10			GPIO\OP-AMP正端输入\ADC信号输入
8	PA. 5	I/O		AMP0_N\AD9			GPIO\OP-AMP负端输入\ADC信号输入
9	PA. 6	I/O		AMP1_P\AD8			GPIO\OP-AMP正端输入\ADC信号输入
10	PA. 7	I/O		AMP1_N\AD7			GPIO\OP-AMP负端输入\ADC信号输入
11	PB. 0	I/O		AMP3_P			GPIO\OP-AMP正端输入
12	PB. 1	I/O		AMP3_N			GPIO\OP-AMP负端输入
13	PB. 2	I/O		AMP3_O			GPIO\OP-AMP输出
14	PB. 3	I/O		CMP5_P			GPIO\比较器正端输入
15	AVCC	P					芯片模拟电源
16	AGND	G					芯片模拟地
17	PB. 7	I/O		CMP3_P\AD5			GPIO\比较器正端输入\ADC信号输入
18	PB. 8	I/O	2us	CMP3_N\AD4		INT6	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
19	PC. 0	I/O		CMP2_P\AD3			GPIO\比较器正端输入\ADC信号输入
20	PC. 1	I/O	2us	CMP2_N\AD2		INT7	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
21	PC. 3	I/O		CMP1_P\AD1			GPIO\比较器正端输入\ADC信号输入
22	PC. 4	I/O	2us	CMP1_N\AD0		INT8	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
23	VDD15	P					内部1.5V输出, 需外接0.1uF滤波电容
24	PC. 6	I/O	2us	TIM0	TIM8_CH4	INT0	GPIO\Timer输出输入\Timer8_CH4\外部中断, 滤波2us\ TEST=0该引脚功能为JTAGWDIEN功能
25	PC. 9	I/O		TIM8_CH3N			GPIO\ Timer8_CH3N
26	PC. 10	I/O		TIM8_CH3			GPIO\ Timer8_CH3
27	PD. 0	I/O		TIM8_CH2N			GPIO\ Timer8_CH2N
28	PD. 1	I/O		TIM8_CH2			GPIO\ Timer8_CH2
29	PD. 2	I/O		TIM8_CH1N			GPIO\ Timer8_CH1N
30	PD. 3	I/O		TIM8_CH1			GPIO\ Timer8_CH1
31	VSS1	G					芯片数字地
32	VCC1	P					芯片数字电源



28 PIN	标识	引脚 类型	滤波	第一复用 功能	第二复用 功能	第三复用 功能	引脚说明
1	PD. 1	I/O		TIM8_CH2			GPIO\ Timer8_CH2
2	PD. 2	I/O		TIM8_CH1N			GPIO\ Timer8_CH1N
3	PD. 3	I/O		TIM8_CH1			GPIO\ Timer8_CH1
4	PD. 7	I/O	2us		TIM3	INT3	GPIO\Timer 输出输入\外部中断, 滤波 2us
5	TESTN	I	2us				测试引脚 (低电平有效, 内部上拉), 输入滤波 2us
6	PA. 0	I/O	2us	SWIO		INT5	GPIO\SWIO\外部中断, 滤波 2us
7	PA. 1	I/O		SWCLK			GPIO\SWCLK
8	RSTN	I	2us				复位信号 (低电平有效, 内部上拉), 滤波 2us
9	PA. 4	I/O		AMP0 P\AD10			GPIO\OP-AMP 正端输入\ADC 信号输入
10	PA. 5	I/O		AMP0 N\AD9			GPIO\OP-AMP 负端输入\ADC 信号输入
11	PA. 6	I/O		AMP1 P\AD8			GPIO\OP-AMP 正端输入\ADC 信号输入
12	PA. 7	I/O		AMP1 N\AD7			GPIO\OP-AMP 负端输入\ADC 信号输入
13	AVCC	P					芯片模拟电源
14	AGND	G					芯片模拟地
15	PB. 5	I/O		CMP4 P\AD6\AMP2 P			GPIO\比较器正端输入\ADC 信号输入\OP-AMP 正端输入
16	PB. 7	I/O		CMP3 P\AD5			GPIO\比较器正端输入\ADC 信号输入
17	PB. 8	I/O	2us	CMP3 N\AD4		INT6	GPIO\比较器负端输入\ADC 信号输入\外部中断, 滤波 2us
18	PC. 0	I/O		CMP2 P\AD3			GPIO\比较器正端输入\ADC 信号输入
19	PC. 1	I/O	2us	CMP2 N\AD2		INT7	GPIO\比较器负端输入\ADC 信号输入\外部中断, 滤波 2us
20	PC. 3	I/O		CMP1 P\AD1			GPIO\比较器正端输入\ADC 信号输入
21	PC. 4	I/O	2us	CMP1 N\AD0		INT8	GPIO\比较器负端输入\ADC 信号输入\外部中断, 滤波 2us
22	VCC2	P					芯片数字电源
23	VSS2	G					芯片数字地
24	VDD15	P					内部 1.5V 输出, 需外接 0.1uF 滤波电容
25	PC. 6	I/O	2us	TIM0	TIM8_CH4	INT0	GPIO\Timer 输出输入\Timer8_CH4\外部中断, 滤波 2us\ TEST=0 该引脚功能为 JTAGWDTEN 功能
26	PC. 9	I/O		TIM8_CH3N			GPIO\ Timer8_CH3N
27	PC. 10	I/O		TIM8_CH3			GPIO\ Timer8_CH3
28	PD. 0	I/O		TIM8_CH2N			GPIO\ Timer8_CH2N

注：1. I=输入；O=输出；P=电源；G=地。

2. 芯片引脚选择 GPIO 功能：

若方向寄存器配置为输出，开漏 OD 功能配置控制有效，上拉控制无效；

若方向寄存器配置为输入，开漏 OD 功能控制无效，上拉控制有效；

3. 芯片引脚选择复用功能：

复用功能的数字输出引脚都可配开漏功能 (Open Drain)，上拉功能配置无效。

复用功能的数字输入引脚 (JTAGWDTEN，这个引脚内部恒定上拉) 都可配上拉功能，开漏功能无效；

若配置为模拟输入 OD 控制和上拉控制都无效；

4. 端口数据寄存器 PTDAT 说明及数据读取

1) 芯片引脚选择 GPIO 功能或复用数字功能

若方向寄存器配置为输出，PTDAT 读取值为寄存器设置值，不随外部 PIN 脚电平变化而变化；

若方向寄存器配置为输入，PTDAT 读取值为 PIN 脚状态值，反映外部 PIN 脚电平变化；

2) 芯片引脚选择复用模拟功能

PTDAT 相应 bit 位值，固定为 0

5. PC. 6 引脚为一个特殊引脚，当 TESTN=0 时，该引脚为输入 JTAGWDTEN 功能。

1.5 缩略语

缩略语	英文原文	中文含义
WDT	Watch Dog Timer	看门狗
GPIO	General Purpose IO	通用 I/O
LVD	Low Voltage Detect	低电压检测
POR	Power On Reset	上电复位
BOR	Brown Out Reset	掉电复位
WKR	Wakeup Reset	唤醒复位
PMU	Power Management Unit	系统电源管理单元
CMU	Clock Management Unit	系统时钟管理单元
RTC	Real Time Clock	实时时钟

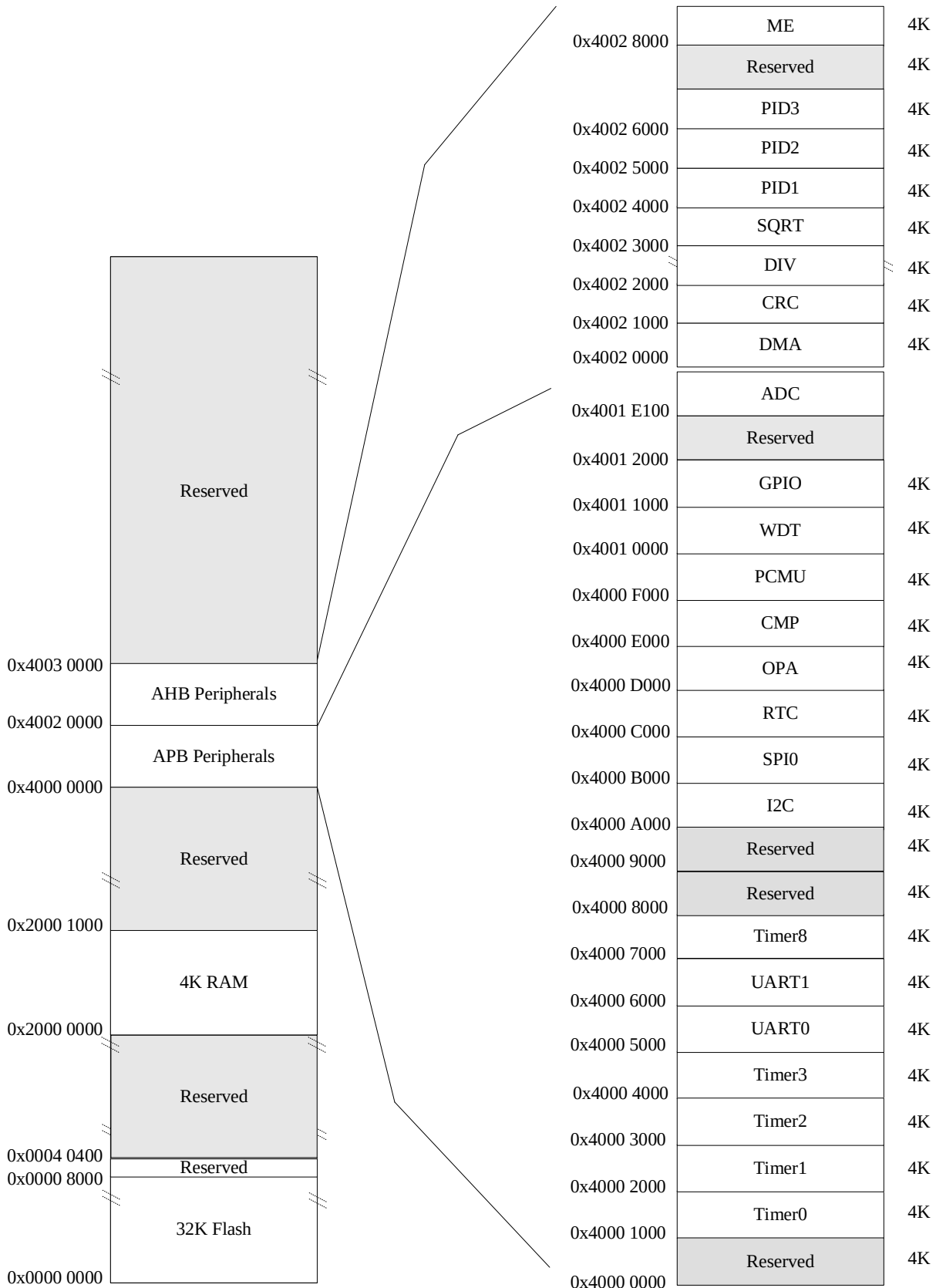
2 存储器模块

2.1 概述

RX32S10 系列内置可编程高可靠 32K Flash 和 4KRAM。其中 Flash 可进行读、写、页擦除和全擦除操作，Flash 的特性如下：

- Flash 字节读取时间：40ns
- Flash 字节写时间：20us (max)
- Flash 页擦除时间：2ms (max)
- Flash 全擦除时间：10ms (max)
- Code Flash 页面大小：1K bytes/page
- 擦写次数：100,000 次
- 数据保持时间：10 年
- 操作温度：-40 度到+85 度

2.2 存储器映射图



2.3 Flash 操作

2.3.1 Code Flash 的操作说明

32K Code Flash 可以执行写/页擦除/全擦除操作，说明如下（伪代码举例，后同）。

1. 推荐首先使用宏定义的方式来实现对 Memory 中的地址写入操作，支持字节操作，半字操作，字操作。

宏定义方式：

```
#define M8(adr)  (*((uint8_t *) (adr)))  
#define M16(adr) (*((uint16_t *) (adr)))  
#define M32(adr) (*((uint32_t *) (adr)))
```

以上宏定义实现对 Flash 地址 addr 的取址

2. 对 32K Code Flash 的字节写操作流程：

```
WPREG = 0xA55A;  
FLASHLOCK = 0x7A68;                //unlock flash memory  
  
FLASHCON = 0x01;                    //program  
M32(prog_address) = prog_data;      //prog_data 为需要编写的数据 (32bit),  
                                     //prog_address 为需要写入的 flash 地址  
//M16(prog_address) = prog_data;    //prog_data 为需要编写的数据 (16bit),  
                                     //prog_address 为需要写入的 flash 地址  
//M8(prog_address) = prog_data;     //prog_data 为需要编写的数据 (8bit),  
                                     //prog_address 为需要写入的 flash 地址  
//当进行字 (32bit) 写入时，prog_address 以 4 为单位递增  
//当进行半字 (16bit) 写入时，prog_address 以 2 为单位递增  
//当进行字节 (8bit) 写入时，prog_address 以 1 为单位递增
```

```
while (FLASHCON.BUSY)                //等待 flash 写操作完成，最长 20us  
;
```

3. 对 32K Code Flash 的页擦除操作流程：

```
WPREG = 0xA55A;  
FLASHLOCK = 0x7A68;                //unlock flash memory  
  
FLASHCON = 0x02;                    //page erase  
M32(prog_address) = prog_data;      //prog_data 可以为任意的数据 (32bit),  
                                     //prog_address 为需要擦除的 Flash 页内的任意一个地址
```

```
while (FLASHCON.BUSY)                //等待 flash 页擦除操作完成，最长 2ms  
;
```

4. 对 32K Code Flash 的全擦除操作流程：

```
WPREG = 0xA55A;  
FLASHLOCK = 0x7A68;                //unlock flash memory  
  
FLASHCON = 0x03;                    //mass erase
```

```
M32(prog_address) = prog_data;           //prog_data 可以为任意的数据 (32bit),
                                           //prog_address 为 32K Flash 的任意地址

while (FLASHCON.BUSY)                     //等待 flash 全擦除操作完成, 最长 10ms
;                                           //全擦除会导致用户执行的代码全部被擦除掉
```

2.4 Flash 控制功能

RX32S10系列微控制器I/O默认态由自动装载字节HIGH_IMPEDANCE[3:0]控制，HIGH_IMPEDANCE[3:0]=1010b或0101b时，I/O默认态为高阻态；其他为默认上拉输入态。

1、复位状态与GPIO配置关系：GPIO模块的寄存器：IOCFG, AFCFG1, AFCFG2, PTDIR, PTUP, PTDAT, PTOD, HIIPM不能被看门狗复位（WatchDog）、调试复位和软复位、唤醒复位（WakeUp Reset）源复位；可以被上电复位（POR）、低电压掉电复位（LBOR）、外部引脚 /RST 复位、掉电复位（BOR）源复位。因此看门狗复位（WatchDog）、调试复位和软复位、唤醒复位（WakeUp Reset）不能装载HIGH_IMPEDANCE[3:0]控制位，HIIPM不能被装载。发生上电复位（POR）、低电压掉电复位（LBOR）、外部引脚 /RST 复位、掉电复位（BOR），HIIPM可以被装载。

2、复位完成后，自加载配置GPIO正确。GPIO状态被AutoLoad配置完成到复位结束，GPIO配置正确。

3、Auto-Load 时序修改为读取 Flash 三次，根据三次读取的结果选择实际加载的值：

- 1) 三次读取的值均不同时，加载第三次读取的值；
- 2) 前两次一致，第三次不同，则加载第二次读取的值；除此之外的情况（含三次读取的值均不同，即情况1）加载第三次读取的值。

当Auto-Load的bit[27:24]值不等于0x05或0x0A时，IO配置为输入上拉状态，否则维持高阻状态。AutoLoadSTA只有当IO状态被复位时才会更新，用于指示AutoLoad是否被正确加载，只有连续三次读取Flash的值一致才会被判定为正确加载，置起标志位。

注：

- 1) AutoLoadSTA仅在IO状态被复位时（POR/BOR/LBOR/外部RST复位）才会更新，其他复位状态即使Auto-Load重新加载也不会更新AutoLoadSTA；
- 2) AutoLoadSTA置0时，表示三次读取的Flash的值不一致，但仍会按照规则加载Auto-Load的值。

4、如果加载为高阻态，则端口上拉关闭，相对应GPIO的PTUP寄存器的控制位为1，HIIPM寄存器的对应控制位为1。非高阻态，端口上拉开启，则对应GPIO的PTUP寄存器的控制位为0，HIIPM寄存器的对应控制位为0。

RX32S10系列微控制器中Flash存储器的0xFC0~0xFC3区域为选项字节区域。当系统上电或从复位状态重启时，设备自动装载选项字节，设置指定的功能。用户需设置以下几项功能：

- LRC开关功能
- 自动装载功能
- POR/LBOR复位RTC功能
- Flash加密功能
- Sleep/Hold模式下WDT使能功能

Flash选项字控制作用如下：

Flash控制选项字节			地址：0xFC0					
	Bit31	30	29	28	27	26	25	Bit24
Read:					HIGH_IMPEDANCE[3:0]			
Write:	X	X	X	X				
	Bit23	22	21	20	19	18	17	Bit16
Read:					WDT_EN[3:0]			
Write:	sramRdf1h[3]	SramRdf1h[2]	sramRdf1h[1]	sramRdf1h[0]				
	Bit15	14	13	12	11	10	9	Bit8
Read:	FLASH[7:0]							
Write:								
	Bit7	6	5	4	3	2	1	Bit0
Read:	X				RTCRST	AUTORELOAD	X	
Write:								

位	功能描述
HIGH_IMPEDANCE[3:0]	I/O默认态控制位 =1010或0101: I/O默认配置为高阻态 =其他: I/O默认配置为上拉输入态
sramRdf1h[3:0]	sramRdf1h[3:0] =1111 时, 程序在 SRAM 空间运行, 对 Flash main block 读取不加密 sramRdf1h[3:0] !=1111时, 程序在SRAM空间运行, 对Flash main block读取加密, 读取值为固定55555555; 此时Flash擦写状态如下: 1) SRAM加密打开, 此时通过SWD口对0-8K flash不可页擦和写(用户程序可页擦和写0-8K), 但可进行全擦操作, 可以页擦和写9K-31K。 2) 不通过SWD接口, 而通过用户程序, 即使SRAM加密有打开, 0-31K也可以全擦、页擦和写。
WDT_EN[3:0]	Sleep 和 Hold 模式看门狗使能位 =0101: 看门狗在 Sleep 和 Hold 模式下关闭 =其他: 看门狗在 Sleep 和 Hold 模式下开启
FLASH[7:0]	Flash加密位 =0xFF: Flash不加密 =其他: Flash加密
RTCRST	RTC复位控制位 =1: LBOR, POR可以复位RTC计时寄存器 =0: LBOR, POR不能复位RTC计时寄存器
AUTORELOAD	自动装载使能位 =1: 自动装载功能使能 =0: 自动装载功能屏蔽

RTC 寄存器下

AUTORELOADSTA (自加载状态寄存器)			基地址: 0x4000C000 偏移地址: 108H					
	Bit15	14	13	12	11	10	9	Bit8

Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	AUTOLO ADSUCI F
Write:								X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
AUTOLOADSUCIF	自加载成功标志位 0: 失败 1: 成功 注: AUTOLOADSUCIF 置 1, 表示自动装载成功即加载值与配置值一致。

2.5 写保护寄存器说明

被写保护的寄存器分布在 CMU、PMU 和 RTC 模块，列表如下：

CMU模块寄存器基地址：0x4000F000			
偏移地址	名称	复位值	功能描述
0x00	WPREG	0x0000	写保护控制寄存器
0x04	SYSCLKCFG	0x0002	系统时钟配置寄存器（写保护）
0x0C	LRCADJ	0x0009	低频 RC 调整寄存器（写保护）
0x10	HRCADJ	0x0040	高频 RC 调整寄存器（写保护）
0x14	HRCDIV	0x0001	高频 RC 分频寄存器（写保护）
0x1C	SYSCLKDIV	0x0001	系统时钟分频寄存器（写保护）
0x24	CLKOUTSEL	0x0002	CLKOUT 时钟选择寄存器（写保护）
0x28	CLKOUTDIV	0x0000	CLKOUT 时钟分频寄存器（写保护）
0x2C	CLKCTRL0	0x04E0	内部模块使能寄存器 0（写保护）
0x30	CLKCTRL1	0x8000	内部模块使能寄存器 1（写保护）
0x34	FLASHCON	0x0000	Flash 访问控制寄存器（写保护）

RTC 模块寄存器基地址：0x4000C000			
偏移地址	名称	复位值	功能描述
0x18	SECR	0x0000	秒寄存器（写保护）
0x1C	MINR	0x0000	分寄存器（写保护）
0x20	HOURLR	0x0000	时寄存器（写保护）
0x24	DAYR	0x0001	日寄存器（写保护）
0x28	MONTHR	0x0001	月寄存器（写保护）
0x2C	YEARR	0x0000	年寄存器（写保护）
0x30	WEEKR	0x0001	周寄存器（写保护）
0x200	SECR2	0x0000	第二套秒寄存器（写保护）
0x204	MINR2	0x0000	第二套分钟寄存器（写保护）
0x208	HOURLR2	0x0000	第二套小时寄存器（写保护）
0x20C	DAYR2	0x0001	第二套天寄存器（写保护）

0x210	MONTHR2	0x0001	第二套月寄存器（写保护）
0x214	YRR2	0x0000	第二套年寄存器（写保护）
0x218	WEEKR2	0x0001	第二套星期寄存器（写保护）

2.6 特殊功能寄存器列表

CMU模块寄存器基地址：0x4000F000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	WPREG	R/W	0x0000	写保护控制寄存器
0x34	FLASHCON	R/W	0x00	Flash 控制寄存器（写保护）
0x38	FLASHLOCK	R/W	0x0000	Flash 锁定寄存器

2.7 特殊功能寄存器说明

2.7.1 写保护寄存器（WPREG）

WPREG (写保护寄存器)			基地址： 0x4000F000 偏移地址： 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	WPREG[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WPREG[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
WPREG[15:0]	- WPREG 写入 0xA55A，则关闭写保护功能，用户可以写操作被保护的寄存器。 - WPREG 写非 0xA55A，则开启写保护功能，用户禁止写操作被保护的寄存器。 - 读该寄存器： 0x0001：表示写保护关闭，用户可以写操作被保护的寄存器 0x0000：表示写保护开启，用户禁止写操作被保护的寄存器

2.7.2 Flash 控制寄存器（FLASHCON）

FLASHCON（写保护） (Flash 控制寄存器)			基地址： 0x4000F000 偏移地址： 34H					
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	BUSY	FOP[1:0]	
Write:						X		
Reset:	0	0	0	0	0	0	0	0

位	功能描述															
BUSY	FLASH 忙标志位 0：表示 Flash 空闲，可以进行操作。 1：表示 Flash 正在进行写/擦除操作。 只读状态寄存器位，写入无效。															
FOP[1:0]	FLASH 操作模式选择 <table><tr><th>FOP1</th><th>FOP0</th><th>FLASH 操作</th></tr><tr><td>0</td><td>0</td><td>处于 Flash 只读模式</td></tr><tr><td>0</td><td>1</td><td>对 STR/STRH 所指 FLASH 区执行 Flash 写操作</td></tr><tr><td>1</td><td>0</td><td>对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作</td></tr><tr><td>1</td><td>1</td><td>对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作</td></tr></table>	FOP1	FOP0	FLASH 操作	0	0	处于 Flash 只读模式	0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作	1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作	1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作
FOP1	FOP0	FLASH 操作														
0	0	处于 Flash 只读模式														
0	1	对 STR/STRH 所指 FLASH 区执行 Flash 写操作														
1	0	对 STR/STRH 所指 FLASH 区执行 Flash 页擦除操作														
1	1	对 STR/STRH 所指 FLASH 区执行 Flash 全擦除操作														

2.7.3 Flash 锁定寄存器 (FLASHLOCK)

FLASHLOCK (Flash 锁定寄存器)			基地址: 0x4000F000 偏移地址: 38H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	KEY[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	KEY[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KEY[15:0]	Flash 锁定控制位 对该寄存器写入 0x7A68 后, FLASH 被解锁, 用户可以写操作 FLASH。 写入非 0x7A68 数据后, FLASH 被锁定, 用户禁止写操作 FLASH。 默认为锁定状态, Flash 不可执行写/页擦除/全擦除 操作 用户写入的是 0x7A68, 读出值为 1; 写入的是非 0x7A68, 读出值为 0

3 时钟单元

3.1 时钟分类

测试温度范围：-40℃~125℃

名称	频率	精度	功耗		
			MIN	TYP	MAX
内部低频 RC 时钟 (Flrc)	32KHz 注 1	TBD		1uA	
内部高频 RC 时钟 (Fhrc)	21MHz	TBD		300nA	
外部高频 OSC 晶振 (HSE)	4-16MHz			1mA	
内部 PLL (Fp11)	42 MHz			1mA	

注 1：内部低频 RC 时钟的误差范围在：13K~50K；

3.2 时钟框图

时钟符号说明：

Flrc：内部低频 RC 时钟 (32KHz)，也作为看门狗时钟源。

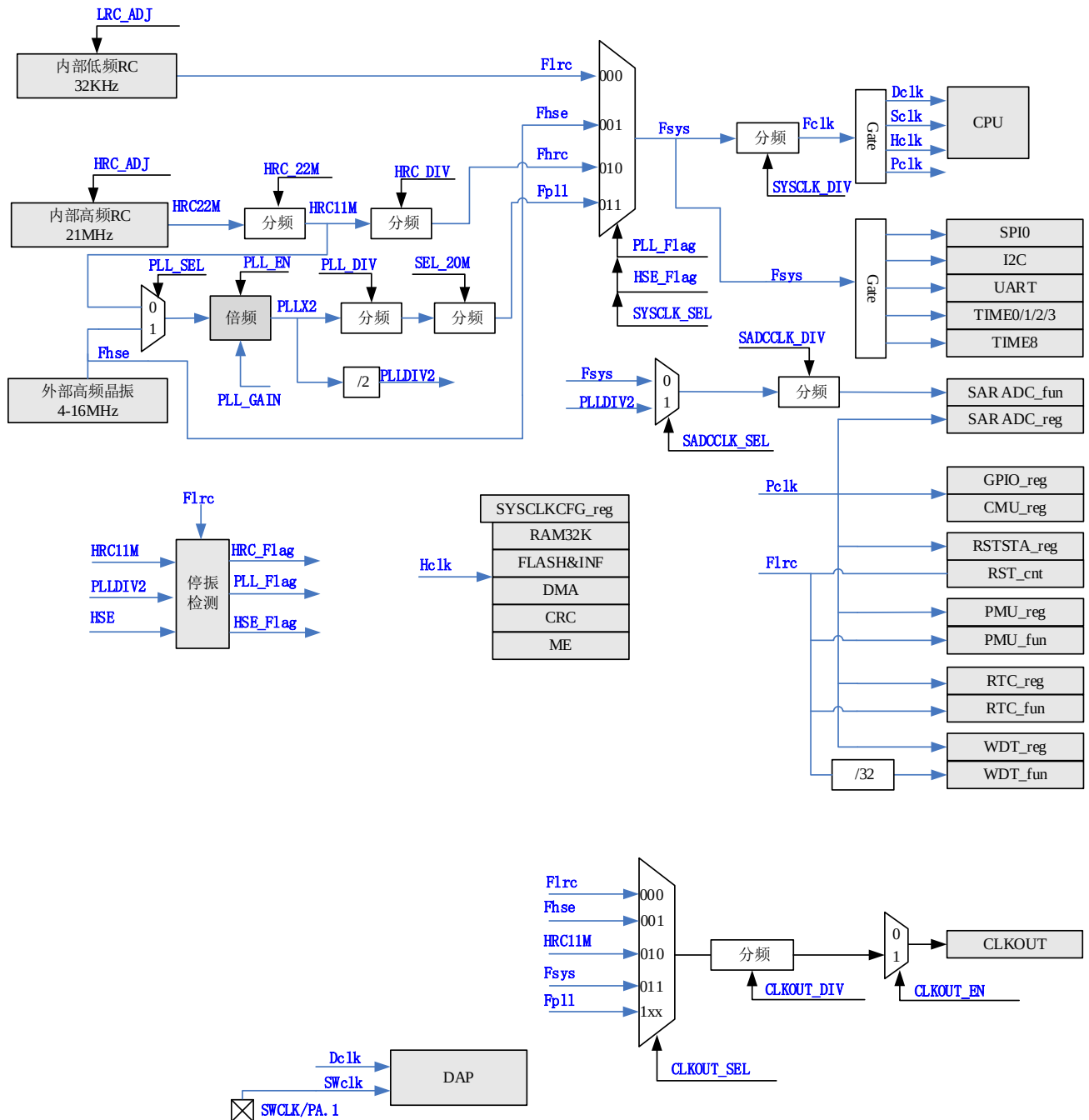
Fhrc：内部高频RC时钟 (21MHz)，系统复位后默认运行在Fhrc。

Fhse：外部高频OSC晶振时钟 (4-16MHz)。

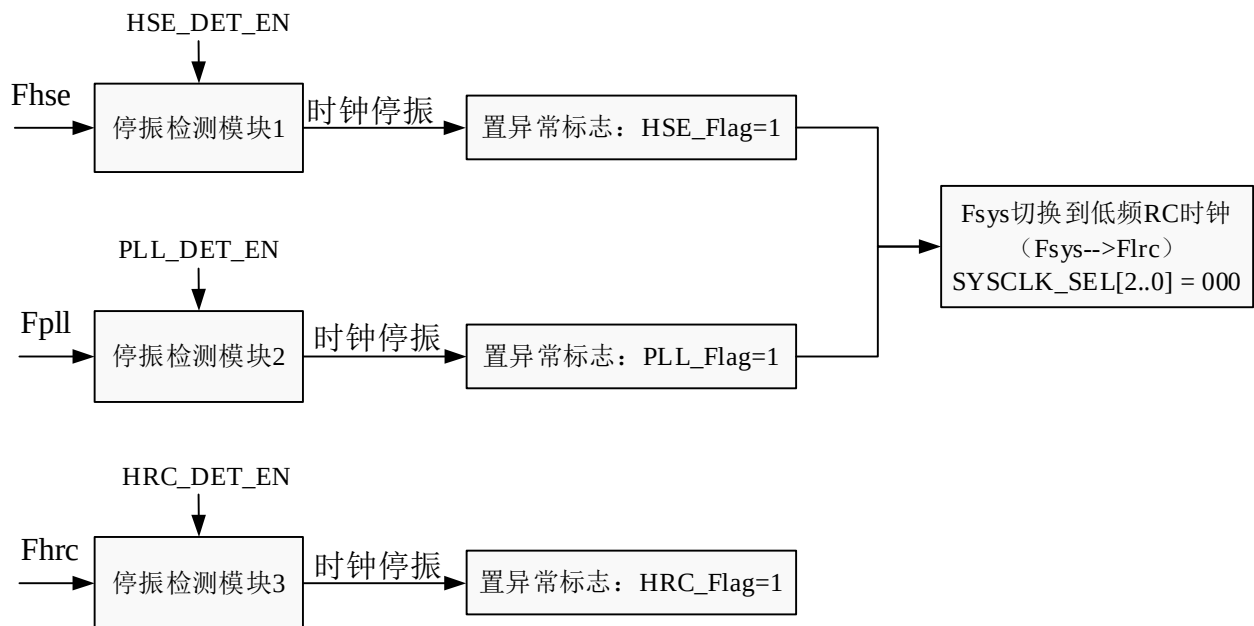
Fp11：内部 PLL 产生的高频时钟 (42MHz)，来源为 Fhse or HRC11M。

Fsys：系统时钟，主要给各个外设提供时钟

Fcpu：给CPU、GPIO、DMA、CRC提供时钟



3.3 时钟停振检测框图



3.3.1 PLL 分档可调

PLL 的时钟来源分别为外部高频晶振(HSE)或者内部高频晶振(HRC11M)，透过 PLL_SEL 选择来源。透过 PLL_GAIN 倍频与 PLL_DIV 除频调整出适当的频率，PLL_GAIN 和 PLL_DIV 各有 16 个档位可以调整。

外部高频晶振(HSE)PLL 输出调整范例

OSC (MHz)	PLL_GAIN	PLL_DIV	PLL_OUT (MHz)
4	8~22	2~32	1~44
6	6~14	2~32	1.125~42
8	4、8、10	2~32	1~40
10	4、6、8	2~32	1.25~40
12	4、6	2~32	1.5~36
14	2、4、6	2~32	0.875~42
16	2、4	2~32	1~32

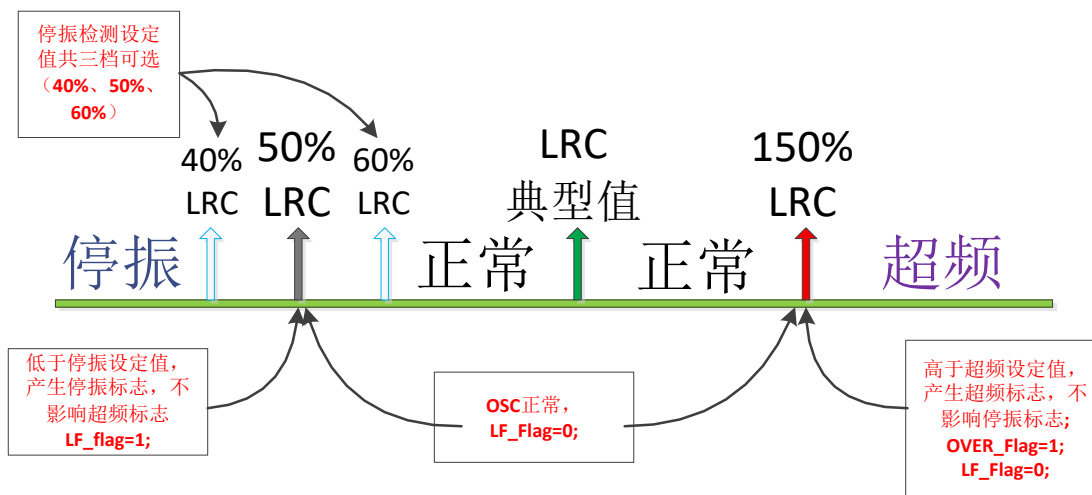
内部高频晶振(HRC11M)PLL 输出调整范例

HRC (MHz)	PLL_GAIN	PLL_DIV	PLL_OUT (MHz)
10.5	2	4	5.25
10.5	2	2	10.5
10.5	4	2	21
10.5	8	2	42

3.3.2 停振检测算法

使用 LRC（内部低频 RC 时钟源）统计 OSC 时钟频率，如果 OSC 频率在设定范围内则 OSC 正常运行且 HSE_flag 清 0，低于设定值设定范围则 OSC 停振且 HSE_flag 置 1，高出设定值范围则 OSC 时钟超频且 OVER_flag 置 1。具体停振检测原理如下：

- 1、首先使用 LRC 分频后的时钟（64 或 128 分频）统计 OSC 时钟个数。
- 2、停振锁定周期数默认值为 16，如果统计 OSC 时钟个数大于最小设定值（根据停振检测范围设定），则 OSC 停振锁定周期数加 1，停振锁定周期数最大值为 LOCKH+1；否则 OSC 停振锁定周期数减 1，停振锁定周期数最小值为 LOCKL-1。
- 3、OSC 停振锁定周期数达到最大值 LOCKH+1 时，停振标志清零，停振锁定周期数达到最小值 LOCKL-1 时，停振标志置 1。
- 4、OSC 时钟个数大于最大设定值（固定为 50%），则超频标志置 1；反之则超频标志清 0。



快速停振检测功能说明：如果 OSC 停振锁定周期个数大于 LOCKH 条件满足，测得 OSC 频率低于 Stop_Limit 设定值时（即 flag_cnt 减 1），则产生停振标志 LF_Flag=1 且 flag_cnt=0（数字内部）。该功能可快速检测到 OSC 停振。

辅助停振标志位（AUXLF_Flag）：OSC 停振锁定周期个数等于 LOCKH+1 后，如果测得 OSC 频率低于 Stop_Limit 设定值（即 flag_cnt 减 1），则该位置 1。该位写 0 清 0。该位不需要快速停振检测功能使能，但需要停振检测模块使能，该模块才工作。

3.3.3 HRC 22M 高频方案

内部高频 RC 时钟频率为 21MHz，可以选择该高频 RC 时钟预分频寄存器控制位（控制位为 HRC_22M 详见时钟框图）作为 10.5Mhz 或 21Mhz 时钟源（兼容 10.5Mhz 方案），其分频输出时钟（分频设置位为 HRC_DIV[1:0]）作为系统时钟（SYSCLK_SEL[2:0]=010）

系统复位后，系统时钟默认选择内部高频 RC 时钟（SYSCLK_SEL[2:0]=010）。HRC_22M 控制位默认为 0，HRC_DIV 默认为 1（即选择 5.25Mhz 作为时钟源）。

当选择内部高频 RC 作为系统时钟时，不能关闭内部高频 RC 时钟，对 HRC_EN 写“0”操作无效。默认 HRC 时钟滤波为 4.5ns。

3.4 时钟说明

3.4.1 内部低频 RC 时钟 (Flrc)

内部低频 RC 时钟振荡频率为 32KHz，提供给看门狗使用，可以选择该低频 RC 时钟作为系统时钟 (SYSCLK_SEL[2:0]=000)。

3.4.2 内部高频 RC 时钟 (Fhrc)

内部高频 RC 时钟频率为 10.5MHz，可以选择该高频 RC 时钟的分频输出时钟（分频设置位为 HRC_DIV[1:0]）作为系统时钟 (SYSCLK_SEL[2:0]=010)。

系统复位后，系统时钟默认选择内部高频 RC 时钟 (SYSCLK_SEL[2:0]=010)。

当选择内部高频 RC 作为系统时钟时，不能关闭内部高频 RC 时钟，对 HRC_EN 写“0”操作无效。

3.4.3 外部高频晶振时钟 (Fhse)

RX32S10 系列芯片外接低功耗晶体振荡器，时钟频率 HSE=4-16MHz，以 HSE 作为系统的内部高频时钟 Fhse，芯片内部集成了其震荡所需的电阻和电容。

3.4.4 内部 PLL 时钟 (Fp11)

内部 PLL 用于对内部高频时钟 Fhse (4-16MHz) 或者 HRC22M(21MHz) 倍频，以对系统提供最高达 42MHz 的高频时钟 Fp11。不过在默认情况下 (SEL_20M=0)，PLL 输出 2 分频后的时钟，即 Fp11 为 21MHz。

如果用户选择 Fcpu 时钟为 42MHz，则必须先使能指令预取功能 (PreFetch_EN=1)。PLL 稳定输出后会给出稳定标志位 PLL_LOCK。

3.4.5 时钟安全机制

芯片内部集成有 3 个独立的时钟停振检测模块，分别对外部高频时钟 Fhse，PLL 输出时钟 Fp11，和内部高频时钟 Fhrc 作检测。3 个停振检测模块均可以由用户软件关闭，控制位分别为 HSE_DET_EN，PLL_DET_EN，HRC_DET_EN。

时钟停振检测模块的时钟源为内部低频 RC 时钟 Flrc。

当对应的的时钟停振检测模块功能开启时，外部高频时钟 Fhse 发生停振，PLL 时钟 Fp11 发生停振，或内部高频 RC 时钟 Fhrc 发生停振，都会产生相应的时钟故障标志 (HSE_FLAG, PLL_FLAG, HRC_FLAG)。

当停振检测模块检测到 Fhse 停振，系统给出时钟停振标志 HSE_FLAG，如系统时钟 Fsys 选择 Fhse 时，系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc，且产生中断 (NMI 中断)，同时将寄存器 SYSCLK_SEL[2:0] 的值置为 000。

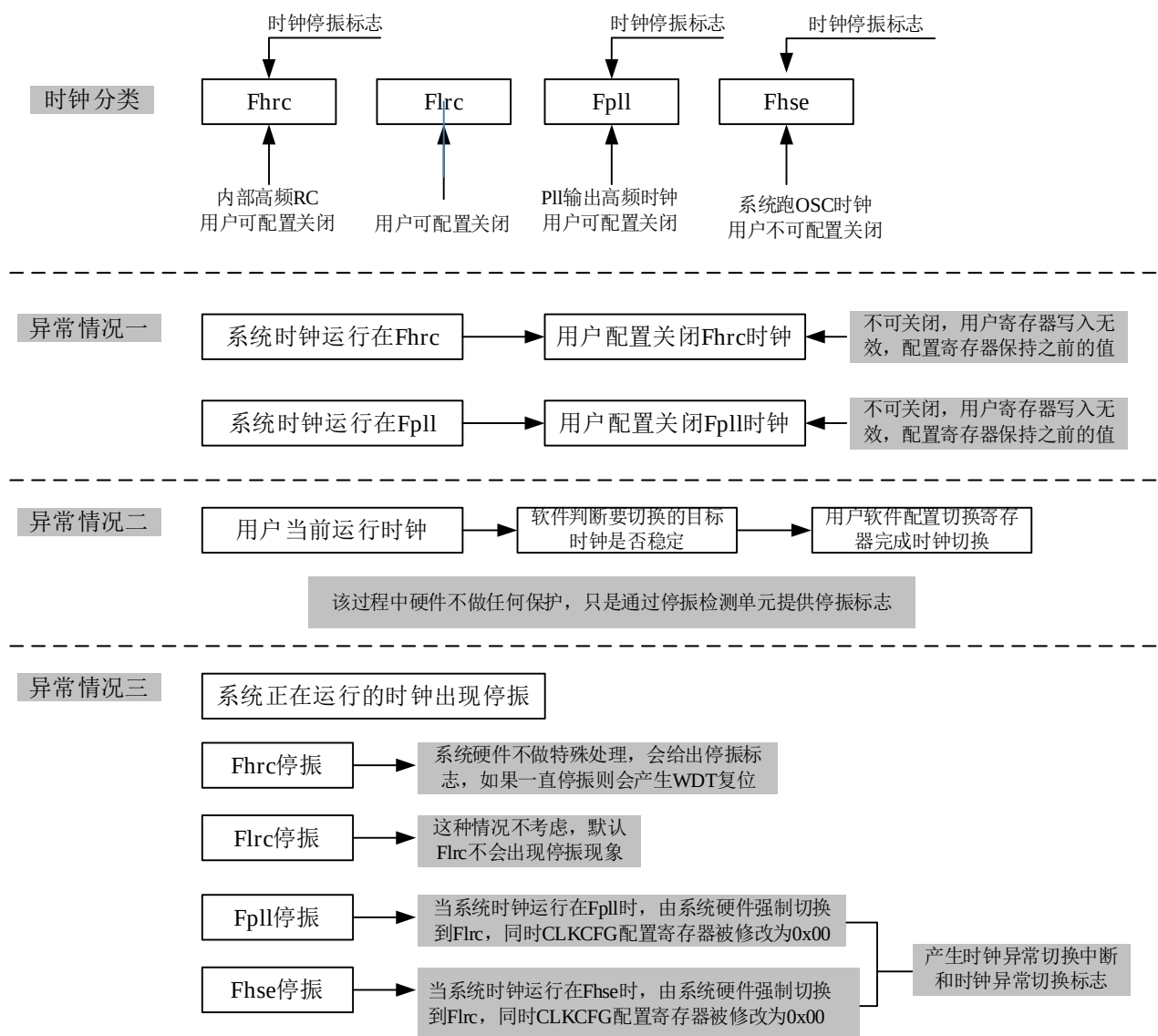
当停振检测模块检测到 Fp11 停振，系统给出时钟停振标志 PLL_FLAG，如系统时钟 Fsys 选择 Fp11 时，系统会由硬件强制将系统时钟 Fsys 切换到内部低频 RC 时钟 Flrc，且产生中断 (NMI 中断)，同时将寄存器 SYSCLK_SEL[2:0] 的值置为 000。

当停振检测模块检测到 Fhrc 停振，系统给出标志位 HRC_FLAG，如系统时钟 Fsys 选择 Fhrc 时，系统不会由硬件强制切换系统时钟，此时系统将停止运行，等待看门狗复位。

3.4.6 时钟源的起振时间

时钟源	起振时间 or 稳定时间
LRC	100us
HRC	10us
PLL	0.1ms

3.4.7 时钟异常状态处理



- 系统运行于低频时钟 Flrc 时，如 PLL 停振，芯片硬件不做任何处理，仅给出停振标志位 PLL_FLAG。
- 系统运行于 PLL 时钟 Fpll 时，如 Fhse 停振或 Fpll 停振，系统时钟由硬件强制切换到 Flrc，同时产生 NMI 中断。

3.5 特殊功能寄存器列表

CMU模块寄存器基地址: 0x4000F000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	WPREG	R/W	0x0000	写保护控制寄存器
0x04	SYCLKCFG	R/W	0x0002	系统时钟配置寄存器 (写保护)
0x08	JTAGSTA	R	0x0001	芯片调试状态指示寄存器
0x0C	LRCADJ	R/W	0x00b5	LRC 时钟调整寄存器 (写保护)
0x10	HRCADJ	R/W	0x0040	HRC 时钟调整寄存器 (写保护)
0x14	HRCDIV	R/W	0x0001	HRC 时钟分频寄存器 (写保护)
0x18	CLKSTA	R	0x0020	时钟状态寄存器 (只读)
0x1C	SYCLKDIV	R/W	0x0001	系统时钟分频寄存器 (写保护)
0x24	CLKOUTSEL	R/W	0x0002	CLKOUT 时钟选择寄存器 (写保护)
0x28	CLKOUTDIV	R/W	0x0000	CLKOUT 时钟分频寄存器 (写保护)
0x2C	CLKCTRL0	R/W	0x24E1	内部模块使能寄存器 0 (写保护)
0x30	CLKCTRL1	R/W	0x8000	内部模块使能寄存器 1 (写保护)
0x3C	PREFETCH	R/W	0x0000	指令预取使能寄存器
0x204	MULTFUNCFG	R/W	0x0000	多功能配置寄存器 (写保护)
0x208	LFCLKCFG	R/W	0x0000	低频时钟源配置寄存器 (写保护)
0xF00	CHIPID	R/W	0x8130	芯片 ID 寄存器
0xF10	PLLLOCK_PRE	R/W		PLL 锁定条件设置寄存器
0xF20	C_PLL	R/W	0x0000	PLL 时钟频率产生控制寄存器
0xF30	FLTCTR	R/W	0x0000	时钟滤波控制寄存器

注: PREFETCH 以实际设计值为准

3.6 特殊功能寄存器说明

3.6.1 写保护寄存器(WPREG)

WPREG (写保护寄存器)			基地址: 0x4000F000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	WPREG[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WPREG[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
位	功能描述							

WPREG[15:0]	4. WPREG 写入 0xA55A, 则关闭写保护功能, 用户可以写操作被保护的寄存器。 5. WPREG 写非 0xA55A, 则开启写保护功能, 用户禁止写操作被保护的寄存器。 6. 读该寄存器: 0x0001: 表示写保护关闭, 用户可以写操作被保护的寄存器 0x0000: 表示写保护开启, 用户禁止写操作被保护的寄存器
-------------	---

3.6.2 系统时钟配置寄存器(SYSCLKCFG)

SYSCLKCFG (写保护) (系统时钟配置寄存器)			基地址: 0x4000F000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	WCLKEN	X	X	X	X	SYSCLK_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述																								
WCLKEN	时钟配置寄存器写保护位 如果用户要更改系统时钟选择，必须同时将 WCLKEN 位置 1，例如： ‘b1xxxxxxx，才可以对系统时钟选择位 SYSCLK_SEL[2:0]进行写操作。																								
SYSCLK_SEL[2:0]	系统时钟选择控制位： <table><tr><th colspan="3">SYSCLK_SEL[2:0]</th><th>系统时钟选择Fsys</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Flrc</td></tr><tr><td>0</td><td>0</td><td>1</td><td>Fhse</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Fhrc(Default)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Fpll</td></tr><tr><td>1</td><td>X</td><td>X</td><td>保留</td></tr></table> 1. 系统时钟可作为芯片内部硬件外设模块的时钟源，经分频后也可作为 CPU 和 GPIO 的时钟源。 2. 复位后，系统时钟默认为内部高频 RC 时钟 Fhrc（SYSCLK_SEL[2:0]=010）。 3. 当选择 PLL 做系统时钟时，必须先打开 PLL 时钟（CLKCTRL0 的 bit4）	SYSCLK_SEL[2:0]			系统时钟选择Fsys	0	0	0	Flrc	0	0	1	Fhse	0	1	0	Fhrc(Default)	0	1	1	Fpll	1	X	X	保留
SYSCLK_SEL[2:0]			系统时钟选择Fsys																						
0	0	0	Flrc																						
0	0	1	Fhse																						
0	1	0	Fhrc(Default)																						
0	1	1	Fpll																						
1	X	X	保留																						

3.6.3 芯片状态指示寄存器(JTAGSTA)

JTAGSTA (芯片状态指示寄存器)			基地址: 0x4000F000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	FLAG	X
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
FLAG	该位用于指示芯片是否处于 JTAG 调试状态 0: 表示芯片处于正常运行状态。 1: 表示芯片处于调试状态。

注: bit0 的值默认为 1, 用户无须更改该位。

3.6.4 LRC 时钟调整寄存器 (LRCADJ)

LRCADJ (写保护) (LRC 时钟调整寄存器)			基地址: 0x4000F000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LRC_ADJ[7:0]							
Write:								
Reset:	1	0	1	1	0	1	0	1

位	功能描述
LRC_ADJ[7:0]	LRC 输出频率调节控制位

3.6.5 HRC 时钟调整寄存器 (HRCADJ)

HRCADJ (写保护) (HRC 时钟调整寄存器)			基地址: 0x4000F000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	HRC_TC[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	reserved	HRC_ADJ[6:0]						
Write:								
Reset:	0	1	0	0	0	0	0	0

位	功能描述
HRC_TC[5:0]	HRC 温度特性调节
HRC_ADJ[5:0]	HRC 输出频率调节控制位 芯片出厂时, HRC 已经过校准, 校准之后频率为 21MHz。HRCADJ 调整值存储于 Info flash 地址 0x00040140。默认配置下, 芯片可正常启动运行。如果加载 Info flash 相应的调整值到 HRCADJ, 可提高 HRC 频率的准确性, 详见 FAQ

3.6.6 HRC 时钟分频寄存器 (HRCDIV)

HRCDIV (写保护) (HRC 时钟分频寄存器)			基地址: 0x4000F000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	HRC_DIV[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述	
HRC_DIV[1:0]	HRC 时钟分频设置:	
	HRC_DIV[1:0]	RC分频后的时钟Fhrc'
	0 0	Fhrc(Default)
	0 1	Fhrc/2
	1 0	Fhrc/4
	1 1	Fhrc/8

3.6.7 时钟状态寄存器 (CLKSTA)

CLKSTA (时钟状态寄存器)			基地址: 0x4000F000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	HOSC_DET_fun
Write:								X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	HSE_LOCK	PLL_LOCK	PLL_FLAG	HRC_FLAG	HSE_FLAG	X	
Write:		X	X	X	X	X		
Reset:	0	1	1	0	0	0	0	0

注: 此寄存器是只读状态寄存器

位	功能描述
HOSC_DET_fu	模拟外部高频停振标志 0: 正常 1: 停振
HSE_LOCK	HSE 时钟锁定状态标志寄存器 0: HSE 时钟锁定异常。 1: HSE 时钟锁定正常。 注: 用来指示芯片内部 PLL 的工作稳定状态, 用户也可以打开 PLL 后等待 4ms 来判断其稳定
PLL_LOCK	PLL 时钟锁定状态标志寄存器

	0: PLL 时钟锁定异常。 1: PLL 时钟锁定正常。 注: 用来指示芯片内部 PLL 的工作稳定状态, 用户也可以打开 PLL 后等待 4ms 来判断其稳定
PLL_FLAG	PLL 时钟 Fp11 停振标志 0: 正常。 1: 停振。
HRC_FLAG	内部高频 RC 时钟 Fhrc 停振标志 0: 正常。 1: 停振。
HSE_FLAG	外部高频 HSE 时钟 Fhse 停振标志 0: 正常。 1: 停振。

3.6.8 系统时钟分频寄存器 (SYSCLKDIV)

SYSCLKDIV (写保护) (系统时钟分频寄存器)			基地址: 0x4000F000 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	SEL_20 M	CPUCLK_DIV[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述																																				
SEL_20M	PLL输出时钟预分频寄存器控制位： =0：PLL 输出时钟 2 分频，即 Fp11=21MHz =1：PLL 输出时钟不分频，即 Fp11=42MHz																																				
CPUCLK_DIV[2:0]	CPU 时钟分频设置： <table><tr><th colspan="3">CPUCLK_DIV [2:0]</th><th>CPU时钟选择(Fcpu)</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Fsys(Default)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>Fsys/2</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Fsys/4</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Fsys/8</td></tr><tr><td>1</td><td>0</td><td>0</td><td>Fsys/16</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Fsys/32</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Fsys/64</td></tr><tr><td>1</td><td>1</td><td>1</td><td>Fsys/128</td></tr></table> 注：如果选择 CPU 时钟为 42M，则必须提前使能指令 预取功能（PreFetch_EN=1）	CPUCLK_DIV [2:0]			CPU时钟选择(Fcpu)	0	0	0	Fsys(Default)	0	0	1	Fsys/2	0	1	0	Fsys/4	0	1	1	Fsys/8	1	0	0	Fsys/16	1	0	1	Fsys/32	1	1	0	Fsys/64	1	1	1	Fsys/128
CPUCLK_DIV [2:0]			CPU时钟选择(Fcpu)																																		
0	0	0	Fsys(Default)																																		
0	0	1	Fsys/2																																		
0	1	0	Fsys/4																																		
0	1	1	Fsys/8																																		
1	0	0	Fsys/16																																		
1	0	1	Fsys/32																																		
1	1	0	Fsys/64																																		
1	1	1	Fsys/128																																		

3.6.9 CLKOUT 时钟选择寄存器 (CLKOUTSEL)

CLKOUTSEL (写保护) (CLKOUT 时钟选择寄存器)	基地址: 0x4000F000 偏移地址: 24H
-------------------------------------	------------------------------

	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	CLKOUT_SEL[2:0]		
Write:	X	X	X	X	X	CLKOUT_SEL[2:0]		
Reset:	0	0	0	0	0	0	0	0

位	功能描述												
CLKOUT_SEL[2:0]	CLKOUT 时钟输出引脚配置 <table border="1"> <thead> <tr> <th>CLKOUT_SEL[2:0]</th><th>CLKOUT时钟选择</th></tr> </thead> <tbody> <tr> <td>0 0 0</td><td>Flrc</td></tr> <tr> <td>0 0 1</td><td>Fhse</td></tr> <tr> <td>0 1 0</td><td>Fhrc</td></tr> <tr> <td>0 1 1</td><td>Fsys</td></tr> <tr> <td>1 x x</td><td>Fpll</td></tr> </tbody> </table> - 用户可将芯片内部时钟源从 CLKOUT 引脚引出，以观测内部时钟。 - 用户可用 CLKOUTDIV 寄存器将内部时钟分频后引出，可作为外部设备的时钟源。	CLKOUT_SEL[2:0]	CLKOUT时钟选择	0 0 0	Flrc	0 0 1	Fhse	0 1 0	Fhrc	0 1 1	Fsys	1 x x	Fpll
CLKOUT_SEL[2:0]	CLKOUT时钟选择												
0 0 0	Flrc												
0 0 1	Fhse												
0 1 0	Fhrc												
0 1 1	Fsys												
1 x x	Fpll												

3.6.10 CLKOUT 时钟分频寄存器（CLKOUTDIV）

CLKOUTDIV（写保护） （CLKOUT 时钟分频寄存器）			基地址： 0x4000F000 偏移地址： 28H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	CLKOUT_DIV[3:0]			
Write:	X	X	X	X	CLKOUT_DIV[3:0]			
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CLKOUT_DIV[3:0]	$\text{CLKOUT 输出频率} = \frac{\text{CLKOUT 选择的时钟源}}{2 \times (\text{CLKOUT_DIV}[3:0] + 1)}$

3.6.11 内部模块使能控制寄存器 0（CLKCTRL0）

CLKCTRL0（写保护） （内部模块使能控制寄存器 0）			基地址： 0x4000F000 偏移地址： 2CH					
	Bit23	22	21	20	19	18	17	Bit16

Read:	OPA_EN	CMP5_EN	CMP4_EN	CMP3_EN	CMP2_EN	CMP1_EN	CMU_NEWREG_EN	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	CRC_EN	X	1P5LBOR_EN	CLKOUT_EN	HSE_EN	HSE_Lock_EN	HRC_DET_EN	PLL_DET_EN
Write:								
Reset:	0	0	1	0	0	1	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	HSE_DET_EN	PLLLOCK_EN	HRC_EN	PLL_EN	I2C_EN	SPI0_EN	X	X
Write:	T_EN							
Reset:	0	1	1	0	0	0		

位	功能描述
OPA_EN	OPA 模块时钟使能 0: 关闭(default) 1: 打开
CMP5_EN	比较器 5 模块时钟使能 0: 关闭(default) 1: 打开
CMP4_EN	比较器 4 模块时钟使能 0: 关闭(default) 1: 打开
CMP3_EN	比较器 3 模块时钟使能 0: 关闭(default) 1: 打开
CMP2_EN	比较器 2 模块时钟使能 0: 关闭(default) 1: 打开
CMP1_EN	比较器 1 模块时钟使能 0: 关闭(default) 1: 打开
CMU_NEWREG_EN	该位 default 为 0 注: 此 bit 受写保护寄存器 WPREG 的控制。
CRC_EN	CRC 模块时钟使能位 0: 关闭(default) 1: 打开
1P5LBOR_EN	LDO 内部 LBOR 使能位 (用户不要修改这个寄存器位) 0: 关闭 1: 打开 (default)
CLKOUT_EN	CLKOUT 使能信号 0: 关闭 (default) 1: 打开
HSE_EN	HSE 使能信号 0: 关闭 (default) 1: 打开
HSE_Lock_EN	HSR 强制锁定使能控制位 0: HSE 不强制锁定 1: HSE 强制锁定 (default)

	注意：当使能该位时，相应的 HSE 锁定标志位 PLL_LOCK 固定为 1，反之锁定标志位 HSE_LOCK 将反应 HSE 的实际工作状态。
HRC_DET_EN	HRC 时钟检测模块控制位 0：关闭 HRC 时钟停振检测单元（default） 1：使能 HRC 时钟停振检测单元
PLL_DET_EN	PLL 时钟检测模块控制位 0：关闭 PLL 时钟停振检测单元（default） 1：使能 PLL 时钟停振检测单元
HSE_DET_EN	HSE 时钟检测模块控制位 0：关闭 HSE 时钟停振检测单元 1：使能 HSE 时钟停振检测单元
PLL_LOCK_EN	PLL 强制锁定使能控制位 0：PLL 不强制锁定 1：PLL 强制锁定（default） 注意：当使能该位时，相应的 PLL 锁定标志位 PLL_LOCK 固定为 1，反之锁定标志位 PLL_LOCK 将反应 PLL 的实际工作状态。
HRC_EN	HF RC 时钟振荡器使能位 0：关闭高频 RC 时钟模块； 1：使能高频 RC 时钟模块；（default） 注意：当用户选择 Fsys 为 Fhrc 时，此时不能关闭 HRC_EN，该寄存器位写入无效 注：此 bit 受写保护寄存器 WPREG 的控制。
PLL_EN	PLL 模块时钟使能位 0：关闭 PLL 模块（default） 1：使能 PLL 模块 注意： 1. 当用户切换至 PLL 时钟作为系统时钟时，必须先打开 PLL_EN。 2. 当用户选择 Fsys 为 Fpll 时，此时不能关闭 PLL_EN，该寄存器位写入无效
I2C_EN	I2C 模块时钟使能位 0：关闭 I2C 模块（default） 1：使能 I2C 模块
SPI0_EN	SPI0 模块时钟使能位 0：关闭 SPI0 模块（default） 1：使能 SPI0 模块

3.6.12 内部模块使能控制寄存器 1（CLKCTRL1）

CLKCTRL1（写保护） （内部模块使能控制寄存器 1）			基地址： 0x4000F000 偏移地址： 30H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	ADC_EN	ME_EN	TIM8_EN	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	SOFTWDT_EN	X	X	X	X	X	X
Write:								

Reset:	1	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	UART1_	UART0_	TMR3_E	TMR2_E	TMR1_E	TMRO_E
Write:			EN	EN	N	N	N	N
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ADC_EN	ADC 模块使能 0:关闭(default) 1:开启
ME_EN	电机专用模块使能 0:关闭(default) 1:开启
TIM8_EN	Timer8 时钟模块使能 0:关闭(default) 1:开启
SOFTWDT_EN	调试模式下看门狗使能位 0: 调试模式下看门狗关闭 (default) 1: 调试模式下看门狗打开
UART1_EN	UART1 时钟使能位 0: 关闭 1: 使能
UART0_EN	UART0 时钟使能位 0: 关闭 1: 使能
TMR3_EN	Timer3 时钟使能位 0: 关闭 1: 使能
TMR2_EN	Timer2 时钟使能位 0: 关闭 1: 使能
TMR1_EN	Timer1 时钟使能位 0: 关闭 1: 使能
TMRO_EN	Timer0 时钟使能位 0: 关闭 1: 使能

注：CRC/IIC/SPI0/UART0-1/TMR0-3 这些模块需先配置对应的时钟使能位，才可以读写模块内寄存器。

3.6.13 指令预取使能控制寄存器（PREFETCH）

PREFETCH (写保护) (指令预取使能控制寄存器)			基地址: 0x4000F000 偏移地址: 3CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	X	X	X	X	X	X	X	PreFet
Write:								ch_EN
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PreFetch_EN	指令预取使能控制位 =0: 禁止 =1: 使能 注: 若 CPU 时钟选择 42M 时钟, 则必须先使能此控制位。

3.6.14 芯片版本寄存器 (CHIPID)

CHIPID 芯片版本寄存器	基地址: 0x4000F000 偏移地址: F00H
	Bit31 ... Bit16
Read:	REVID[15:0]
Write:	
Reset:	0
	Bit15 ... Bit0
Read:	CHIPID[15:0]
Write:	
Reset:	0

位	功能描述
REVID[15...0]	改版编号
CHIPID[15...0]	芯片编号 0x8130

3.6.15 PLL 锁定条件设置寄存器 (PLLLOCK_PRE)

PLLLOCK_PRE (PLL 锁定条件设置寄存器)			基地址: 0x4000F000 偏移地址: F10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	tosc_d ub_clr
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	Sumctr[1:0]		X	X	Tosclenctr[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
tosc_dub_clr	PLL 侦测的周期超应出设定两倍时将锁定周期清为 0 1: enable 0: disable (default)
Sumctr[1:0]	PLL 锁定的 Flrc 周期个数设定
	Sumctr[1:0] Cntsum
	00 16

	01	32	
	10	64	
	11	128	
Tosclen[1:0]	每个Flrc周期内有效的PLL锁定的判断值的设置： PLL时钟为32768*84*16，		
	Tosclen [1:0]	Tosclen-min	Tosclen-max
	00	82	86
	01	80	88
	10	78	90
	11	76	92
此值用来与每个Flrc周期内pll 16分频时钟的个数比对。如果pll 时钟个数一个Flrc周期内累加值位于min,max之间，则认为此Flrc周期内PLL趋于锁定状态。			

3.6.16 多功能配置寄存器（MULTFUNCFG）

MULTFUNCFG （写保护） （多功能配置寄存器）			基地址： 0x4000F000 偏移地址： 204H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	MULTFUNCFG_EN	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	reserved	reserved	reserved	reserved	reserved	reserved	HRC_22M	3T
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
MULTFUNCFG_EN	多功能置寄存器新增功能生效控制位 =0 (default), 寄存器按照芯片默认值运行 =1, 寄存器按照实际配置值运行
HRC_22M	HRC 输出时钟预分频寄存器控制位 =0:HRC 输出时钟 2 分频 (default) =1:HRC 输出时钟不分频
3T	3T 功能选择位 =0: 关闭 (default) =1: 开启 注: 开启 3T 和 PREFETCEH, CPU 运行 3T。关闭 3T 功能, 不影响 CPU 正常运行。

注：唤醒复位、调试和软复位、看门狗复位 BOR、EXRST、LBOR、POR 可以复位该寄存器。

3.6.17 低频时钟源配置寄存器 (LFCLKCFG)

LFCLKCFG (写保护) (低频时钟源配置寄存器)			基地址: 0x4000F000 偏移地址: 208H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	LFCLKC FG_EN	X	X					
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X					MANULF SEL	RTCCLK SEL	X
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述
LFCLKCFG_EN	低频时钟源配置寄存器新增功能生效控制位 =0 (default), 寄存器按照芯片默认值运行 =1, 寄存器按照实际配置值运行
MANULFSEL	设置都为 0
RTCCLKSEL	RTC 模块时钟配置位 =0: LRC =1: LRC (default)

注: 唤醒复位、调试和软复位、看门狗不能复位 LFCLKCFG, BOR、EXRST、LBOR、POR 可以复位该寄存器。

3.6.18 PLL 时钟频率产生控制寄存器 (C_PLL)

C_PLL (PLL 时钟频率产生控制寄存器)			基地址: 0x4000F000 偏移地址: F20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	C_PLL_ EN	X			PLL_GAIN			
Write:								
Reset:	0	0	0	0	0	0	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	X			PLL_SE L	PLL_DIV			
Write:								
Reset:	0	0	0	0	0	0	0	0

注: 唤醒复位、调试和软复位、看门狗复位 BOR、EXRST、LBOR、POR 可以复位该寄存器。

位	功能描述
C_PLL_EN	PLL 时钟频率产生控制寄存器写操作使能位 =0 (default), 关闭写操作使能 =1, 打开写操作使能 注: 仅在 CLKCTRL0.CMU_NEWREG_EN(bit17) = 1 时, 此 bit 操作才有意义。
PLL_GAIN	PLL 放大倍率 0000: x2

	0001: x4 0010: x6 0011: x8 0100: x10 0101: x12 0110: x14 0111: x16 1000: x18 1001: x20 1010: x22 1011: x24 1100: x26 1101: x28 1110: x30 1111: x32
PLL_SEL	1:HSE 0:HRC11M
PLL_DIV	PLL 除频 0000: / 2 0001: / 4 0010: / 6 0011: / 8 0100: / 10 0101: / 12 0110: / 14 0111: / 16 1000: / 18 1001: / 20 1010: / 22 1011: / 24 1100: / 26 1101: / 28 1110: / 30 1111: / 32

3.6.19 时钟滤波控制寄存器 (FLTCTR)

FLTCTR (时钟滤波控制寄存器)			基地址: 0x4000F000 偏移地址: F30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PLL_CTR[1:0]		HRC_CTR[1:0]		LRC_CTR[1:0]		HSE_CTR[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

PLL_CTR[1:0]	PLL 时钟滤波控制位 =00, 0.0ns (Default) =01, 1.5ns =10, 3.0ns =11, 3.0ns
HRC_CTR[1:0]	HRC 时钟滤波控制位 =00, 4.5ns (Default) =01, 3.0ns =10, 1.5ns =11, 0.0ns
LRC_CTR[1:0]	LRC 时钟滤波控制位 =00, 0.0ns (Default) =01, 1.5ns =10, 3.0ns =11, 4.5ns
HSE_CTR[1:0]	HSE 时钟滤波控制位 =00, 0.0ns (Default) =01, 1.5ns =10, 3.0ns =11, 4.5ns

3.7 应用场景

3.7.1 Flrc 应用

1. lrc 作为最可靠时钟源，不可停振；
2. 低功耗需求 (<1uA)，精度差，现有的 lrc 通过测试卡测，只能保证 13k~50k@全温度
3. 可被选择为系统时钟 (Fsys)
normal run 下获得更低功耗应用中，系统时钟会切换到 Flrc，以降低功耗；
进入 Sleep/hold 下之前，系统时钟切换至 Flrc，降低功耗；
4. 固定为 wdt 的时钟源
5. 作为 Fhse/pll 停振检测切换时钟源-----可靠性
6. 支持 Lrc 频率可调 (LRCADJ)，作为 lrc 精度差的调整 option；

3.7.2 Fhrc 应用

1. 作为系统复位默认时钟，可靠性与 lrc 相当；----->hrc 停振检测不作时钟切换处理
2. 频率 10.5MHz，与 PLL 时钟一致性，无缝切换需求；
3. 精度要求，全温度漂移小于 3%@全温度----->主要 UART 通讯限制；
4. 支持 hrc 频率可调 (HRCADJ)
测试厂 FT 测试，通过调整此寄存器，校正 HRC 精度，存储最优调整值至 info 区固定位置；
量产芯片，用户通过读取 info 区校正值，加载至 HRCADJ 寄存器，保证 HRC 精度；此操作一般发生在复位后用户初始化或 normal run 执行修复任务中
5. Normal run 下，作为系统时钟，在常规应用中，复位初始化及上电初始化使用 HRC 时钟，系统时钟切换至 PLL 时钟；进入 Sleep/hold 模式前，系统时钟切到低频，关闭 HRC 和 PLL；
6. Hold 模式下，大功耗 LDO 开启 (pmucon 中控制)，系统时钟可以选择 HRC；若大功耗 LDO 关闭，由于 LDO_LP 驱动能力不足，系统时钟不推荐客户选择高频时钟 HRC/PLL；
Hold 模式下，使用 LDO_LP (大功耗 LDO 关闭)，是否支持开启关闭高频时钟 HRC/PLL，只要取决于 HRC/PLL 工作在 5v 电压域还是 1.5v 电压域下；即是否由 LDO_LP 供电；
7. 支持 enable/disable 控制
HRC 时钟不被使用时，可关闭，控制位在 CLKCTRL0 寄存器，寄存器需要关闭写保护才能操作；
系统时钟选择 HRC，关闭动作，不生效；
8. 所有复位均会复位 HRC_EN 和 SYSCLK_SEL 信号，保证复位后系统运行 HRC 时钟；
9. autoload 自加载的时钟源

3.7.3 Fpll 应用

1. 作为精确的高频时钟源，上电默认关闭；由 osc 或者 hrc 倍频所得；
2. 频率最高 42MHz----->run 操作系统，对高频需求较高；
3. 可选择作为系统时钟
4. Sleep/hold (ldo_lp) 下，选择 PLL 开启并作为系统时钟，回导致死机或异常，现要求客户 Sleep/hold (ldo_lp) 不能开启 PLL 并选择为系统时钟；参考<1.7.2 Fhrc 应用>中第 5 项
5. 支持 enable/disable 控制
PLL 时钟不被使用时，可关闭，控制位在 CLKCTRL0 寄存器，寄存器需要关闭写保护才能操作；
系统时钟选择 PLL，关闭动作，不生效；
- 6.

3.7.4 时钟切换

1. 时钟切换控制，必须有解锁保护，防止误操作或程序干扰，导致误切；
2. 时钟切换，必须保证切换过程不出错；
3. 时钟切换的应用：
上电后初始化：HRC→PLL
停振检测到异常：PLL→LRC（硬件自动），LRC→hrc，起振正常，HRC→PLL
进入低功耗：PLL→Flrc
Sleep 唤醒：Flrc→HRC（硬件自动）
hold 唤醒：Flrc→HRC/PLL，PLL 建立时间较 HRC 长，额外的功耗损失；

3.7.5 停振检测判断

1. LRC 快速起振（约 200us），OSC 慢速起振（约 500ms）；
2. OSC 间歇性起振，即保持一段时间停振，一段时间正常起振，反复循环；
衍生情况：一段时间停振但有噪声干扰，一段时间正常起振，反复循环；
3. 晶振 Floating 状态下，OSC 管脚进入的干扰是否会误判；
4. 工频 50Hz/60Hz 是否会被误判起振；

3.7.6 时钟异常及 NMI 处理

1. 支持 HRC/PLL/HSE 异常检测；但仅 PLL 和 HSE 支持时钟自动切换和 nmi 中断；
2. NMI 中断处理方案：
 - 1) 切换时钟为 HRC，跳出中断服务；若 HRC 建立异常，则 hold 在中断服务程序里等待 WDT 复位
 - 2) 切换时钟为 HRC，跳出中断服务；若 HRC 建立异常，执行系统软复位（NVIC_SysReset()）；
注意：hautfault 或 NMI 中断中，不能使用可以把 DEBUG 模块复位的复位操作，否则芯片一旦处于异常循环后，不可擦写和仿真（不可修复）。

4 电源单元

4.1 概述

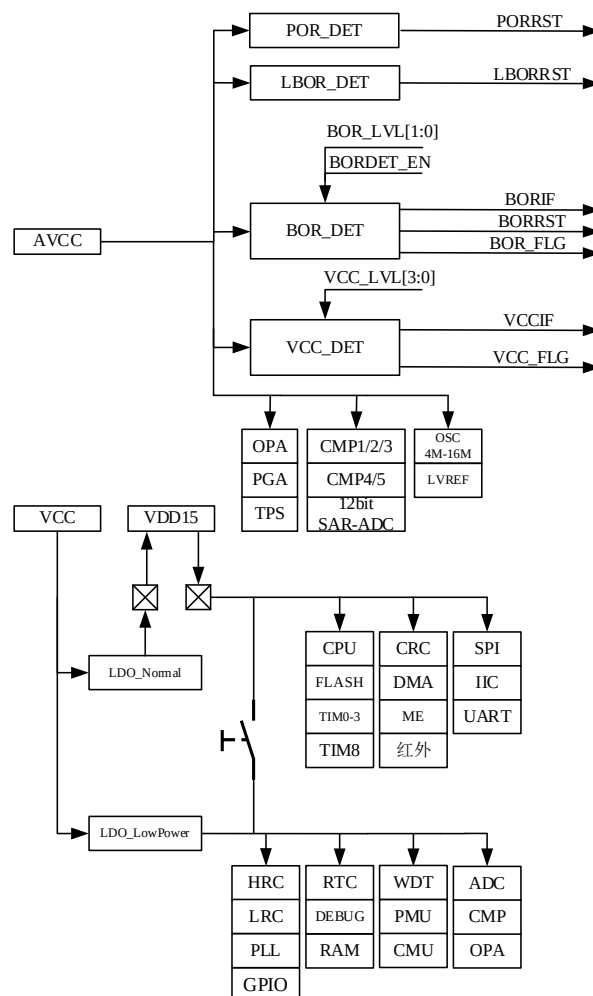
PMU 为芯片的电源管理单元，功能如下：

- 监测系统电源 VCC，可以根据设定阈值产生 BOR、LBOR、POR 复位信号。
- 为芯片内部数字模块提供 1.5V 电源

4.2 修改寄存器概述

配置寄存器必须可读可写状态，涉及寄存器为 PMUCON、VDETCFG、PMUIE、PMUIF、PMUSTA。

4.3 框图



4.4 电源单元详细功能说明

4.4.1 电源

芯片供电电源由芯片外部电路来完成。一般将主电源输入到芯片的VCC1, VCC2, AVCC引脚。

4.4.2 电源实时监测

PMU单元共内置4个电源检测模块，分别实时监测工作电源状态，并将监测结果以二个中断信号和三个复位信号的形式反馈给用户。

- **VCC_DET模块:**

监测系统电源VCC的电压，当电压低于或高于设定阈值时，置位VCCIF标志位；如果使能VCC检测中断（VCCIE），就会产生VCC中断；检测阈值可通过寄存器VDETCFG的位VCC_LVL[3:0]来设置。

- **BOR_DET模块:**

监测系统电源VCC的电压，当电压低于或高于设定阈值时，置位BORIF标志位；如果设置BOR模块产生中断信号（BORRST=0），同时使能BOR检测中断（BORIE），就会产生BOR中断；如果设置BOR模块产生复位信号（BORRST=1），当电压低于设定阈值时，将会立即产生BOR复位。检测阈值可通过寄存器VDETCFG的位BOR_LVL[1:0]来设置。

- **LBOR_DET模块:**

监测系统电源VCC的电压，当电压低于阈值1.9V时，产生LBOR掉电复位。

- **POR_DET模块:**

监测系统电源VCC的电压，当电压上升到阈值0.3V时，产生POR上电复位。

4.4.3 内建 1.5V 电源

芯片内部通过子模块 VREG 将 VCC 电压调制成 1.5V 电压，供芯片内部的 1.5V 工作域使用。

4.4.4 BOR 检测功能(BOR_DET)

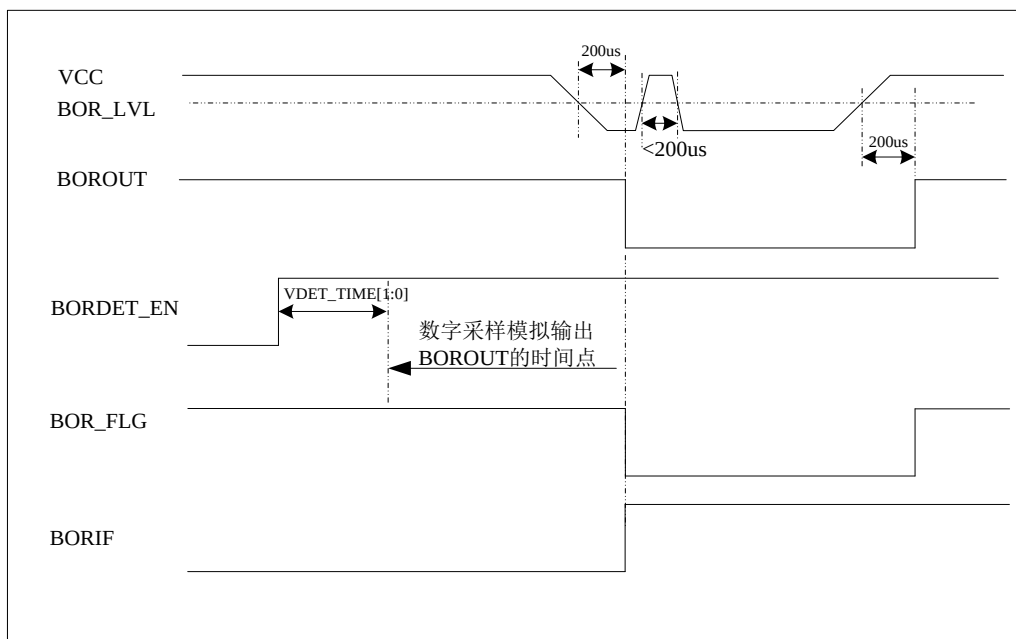
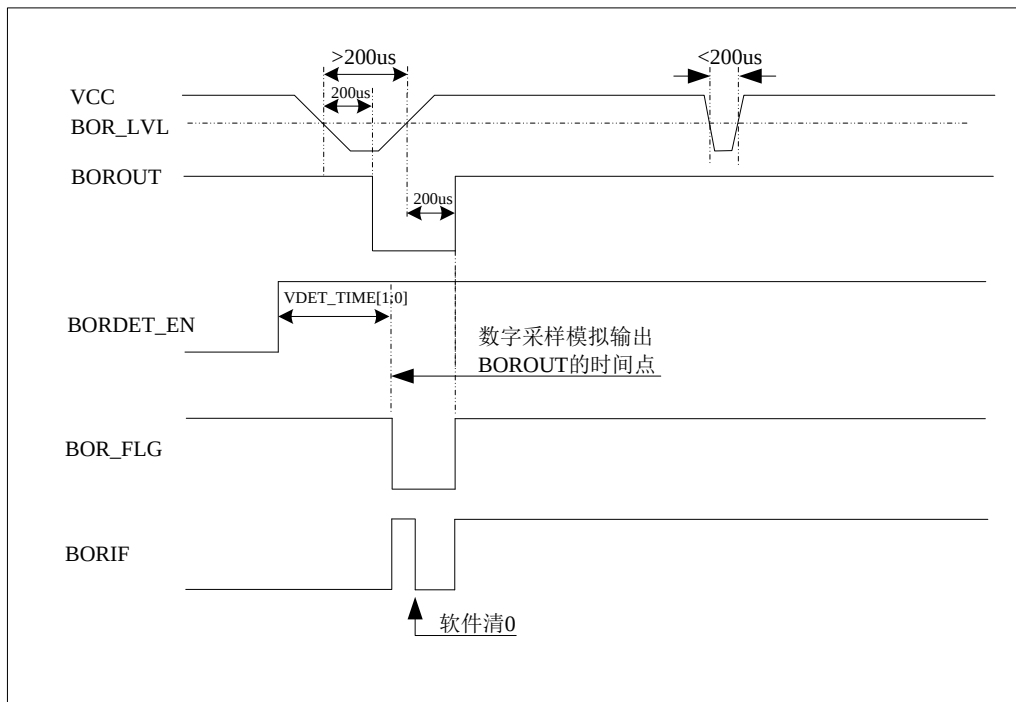
如果用户设置 BOR 模块产生复位信号（BORRST=1），当 BOR 检测模块检测到系统电源 VCC 低于设定电压 Vbor 时，BOR 检测模块内部信号 BOROUT 输出低电平，内部复位信号 IRST 也将变为低电平，复位状态寄存器 RSTSTA 的 BOR 标志位被置为 1。当 BOR 检测模块检测到系统电源 VCC 电压高于设定电压 Vbor 时，BOR 检测模块内部信号 BOROUT 输出高电平，在该高电平持续的 1024 个 Flrc 周期后，内部复位信号 IRST 也变为高电平。

Vbor 具有迟滞特性，迟滞电压为 200mV，BOR 模块的检测阈值 Vbor 可通过 VDETCFG 中的 BOR_LVL[1:0]设置。

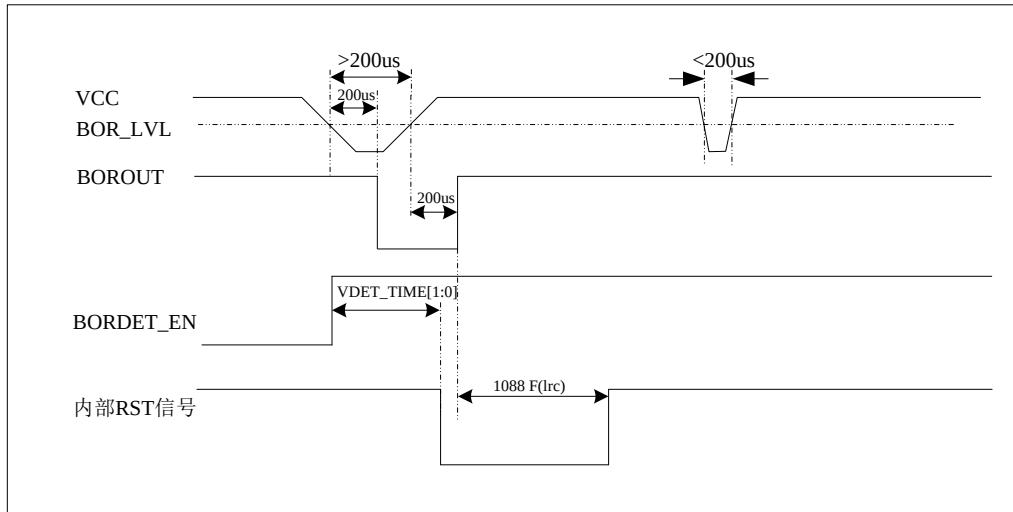
掉电复位 BOR 产生时，下面的事件将会发生：

- 产生一个 BOR 脉冲
- 内部复位信号 IRST 有效
- 计数 1024 个 Flrc
- 复位状态寄存器 RSTSTA 的掉电复位标志位 BOR 被设置为 1。
- CPU 从 0000H 开始执行程序

LBOR_DET 与 BOR_DET 的检测过程基本相同。

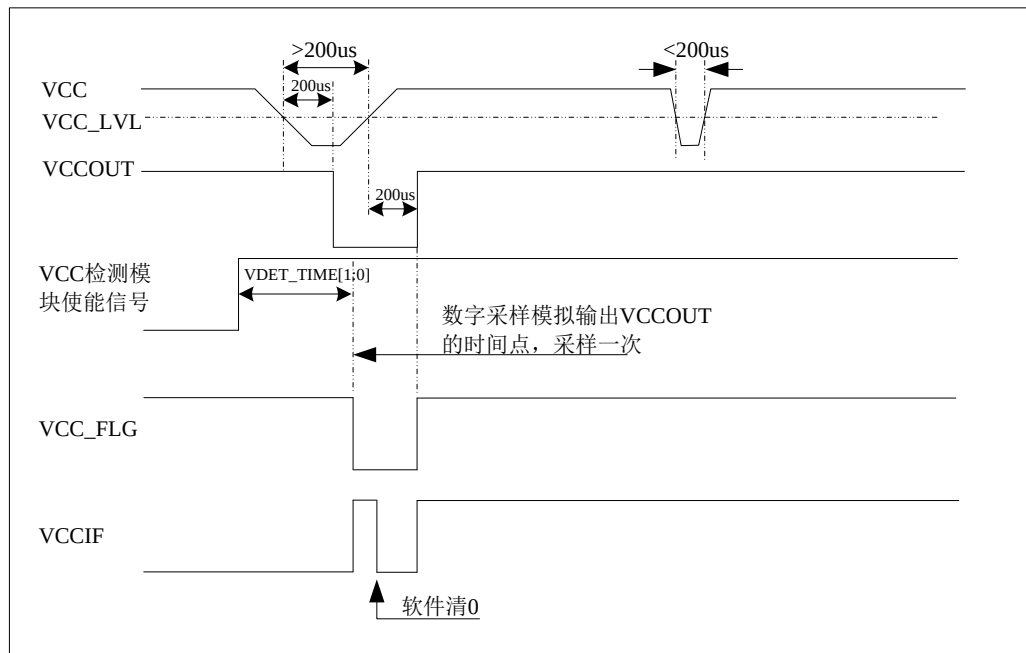


BOR 中断信号产生示意图

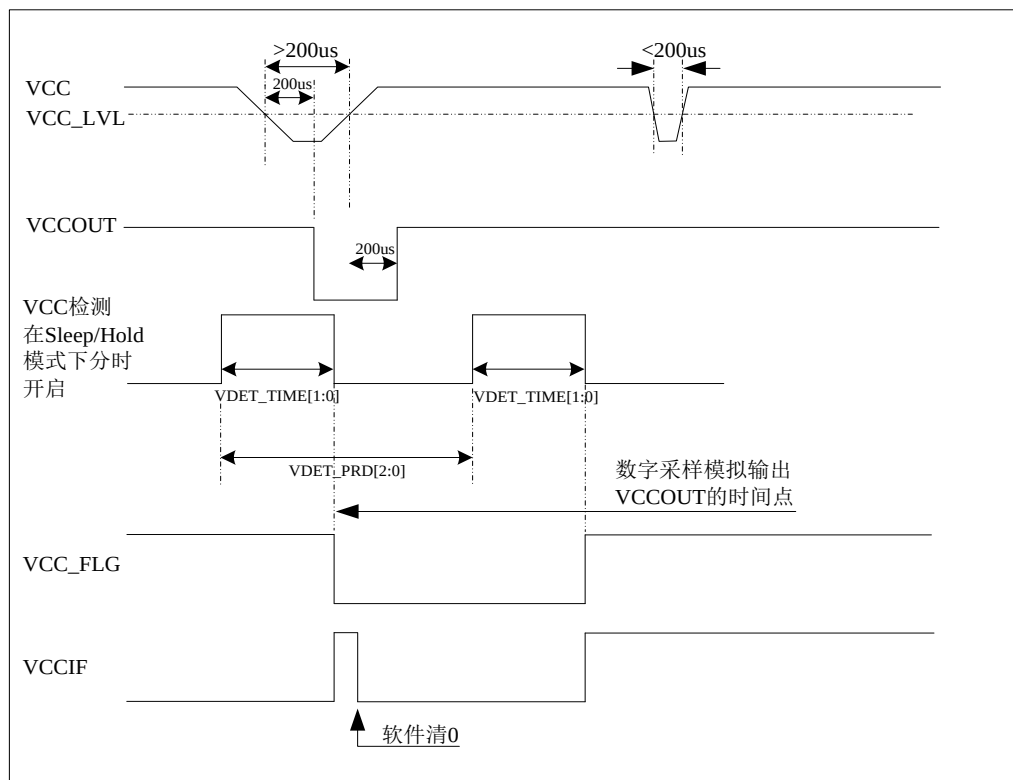


BOR 复位示意图

4.4.5 系统电源检测功能 (VCC_DET)



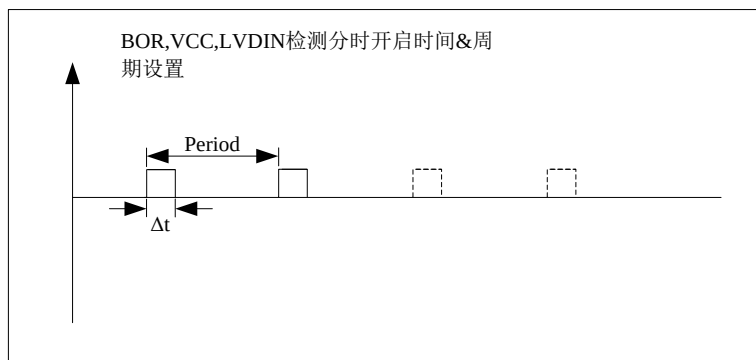
Normal 模式 Vcc 检测



Sleep 模式下 Vcc 分时检测

4.4.6 VCC_DET, BOR_DET 分时检测时序

系统在Hold或Sleep低功耗模式下时，为进一步降低系统功耗，VCC_DET & BOR_DET采用分时开启的方式工作：



其中Period为VCC_DET & BOR_DET检测模块分时开启的周期，可通过VDETPCFG寄存器的VDET_PRD[2:0]位设置。 Δt 是分时检测时每个周期内VCC_DET & BOR_DET工作的时间，可通过VDETPCFG寄存器的VDET_TIME[1:0]位设置。

建议用户在实际应用中：

正常上电运行时，VCC_DET & BOR_DET 模块是连续开启的，即检测电源的前级状态，以使系统能快速的检测到电源异常，并作相应的处理。

当进入低功耗模式时，VCC_DET & BOR_DET 模块自动进入分时开启模式，检测系统电源 VCC 的电压，即检测电源的后级状态，以使系统能进入一个可靠的状态再唤醒。

4.5 特殊功能寄存器列表

PMU 模块寄存器基地址: 0x4000F400				
偏移地址	名称	读写方式	复位值	功能描述
0x00	PMUCON	R/W	0x0013	PMU配置寄存器（写保护）
0x04	VDETCFG	R/W	0x0069	电源检测阈值配置寄存器
0x08	VDETPCFG	R/W	0x0022	电源检测时间周期配置寄存器
0x0C	PMUIE	R/W	0x0000	PMU中断使能寄存器
0x10	PMUIF	R/W	0x0000	PMU中断标志寄存器
0x14	PMUSTA	R	0x0000	PMU状态指示寄存器

4.6 特殊功能寄存器说明

4.6.1 PMU 配置寄存器（PMUCON）

PMUCON（写保护） （PMU 配置寄存器）			基地址: 0x4000F400 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	DISCHA	X	X	X	X	X	X	X
Write:	RGE_EN							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	Hold_L	X	X	BORRST	BORDET
Write:				DO				_EN
Reset:	0	0	0	1	0	0	1	1

位	功能描述
Hold_LDO	在 Hold 模式下选择打开/关闭 大功耗 LDO（默认打开） 0: 关闭大功耗 LDO 1: 打开大功耗 LDO（default） 注：当用户需要在 Hold 模式下达到最低功耗时，可以将该大功耗 LDO 关闭，届时芯片自动切换使用低驱动能力低功耗的 LDO
BORRST	BOR复位/中断选择位 0: VCC电压低/高于设定阈值时产生BOR中断 1: VCC电压低于BOR_LVL[1:0]设定阈值时产生BOR复位（default）
BORDET_EN	BOR_DET 模块使能信号 0: 关闭 BOR_DET 模块 1: 开启 BOR_DET 模块（default）

4.6.2 电源检测阈值配置寄存器（VDETCFG）

VDETCFG （电源检测阈值配置寄存器）	基地址: 0x4000F400 偏移地址: 04H
--------------------------	------------------------------

	Bit15	14	13	12	11	10	9	Bit8
Read:	DISCHARGE_CTRL [1: 0]		X	X	X	X	X	POW_LVL[2]
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X		VCC_LVL[3:0]				BOR_LVL[1:0]	
Write:								
Reset:	0	1	1	0	1	0	0	1

位	功能描述				
VCC_LVL[3:0]	VCC_DET检测阈值控制位				
	VCC_LVL[3:0]				检测电压
	0	0	0	0	2.4V
	0	0	0	1	2.6V
	0	0	1	0	2.8V
	0	0	1	1	3.0V
	0	1	0	0	3.2V
	0	1	0	1	3.4V
	0	1	1	0	3.6V
	0	1	1	1	3.8V
	1	0	0	0	4.0V
	1	0	0	1	4.2V
	1	0	1	0	4.4V (default)
	1	0	1	1	4.6V
	1	1	0	0	4.8V
	1	1	0	1	5V
	1	1	1	X	5V
	BOR_LVL[1:0]	BOR_DET检测阈值控制位			
BOR_LVL[1:0]		检测电压			
0		0	2.2V		
0		1	2.0V (default)		
1		0	2.8V		
1		1	2.6V		

4.6.3 电源检测周期配置寄存器 (VDETPCFG)

VDETPCFG (电源检测周期配置寄存器)			基地址: 0x4000F400 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read	X	X	RESERVED	VDET_TIME[1:0]		VDET_PRD[2:0]		
Write								
Reset:	0	0	1	0	0	0	1	0

位	功能描述																																						
RESERVED	该寄存器位用户不要修改，无意义																																						
VDET_TIME [1:0]	Hold&Sleep 模式下 VCC_DET & BOR_DET 分时检测的时间设定 <table> <tr> <th colspan="2">VDET_Time[1:0]</th> <th>检测时间</th> </tr> <tr> <td>0</td> <td>0</td> <td>300us (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>360us</td> </tr> <tr> <td>1</td> <td>0</td> <td>480us</td> </tr> <tr> <td>1</td> <td>1</td> <td>1120us</td> </tr> </table> 注：内部已经锁定为最大检测时间			VDET_Time[1:0]		检测时间	0	0	300us (default)	0	1	360us	1	0	480us	1	1	1120us																					
VDET_Time[1:0]		检测时间																																					
0	0	300us (default)																																					
0	1	360us																																					
1	0	480us																																					
1	1	1120us																																					
VDET_PRD[2:0]	Hold&Sleep 模式下 VCC_DET & BOR_DET 分时检测的周期设定 <table> <tr> <th colspan="3">VDET_PRD[2:0]</th> <th>检测周期</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>16.5ms</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>33ms</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>67ms (default)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>134ms</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>268ms</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>536ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1072ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2144ms</td> </tr> </table>			VDET_PRD[2:0]			检测周期	0	0	0	16.5ms	0	0	1	33ms	0	1	0	67ms (default)	0	1	1	134ms	1	0	0	268ms	1	0	1	536ms	1	1	0	1072ms	1	1	1	2144ms
VDET_PRD[2:0]			检测周期																																				
0	0	0	16.5ms																																				
0	0	1	33ms																																				
0	1	0	67ms (default)																																				
0	1	1	134ms																																				
1	0	0	268ms																																				
1	0	1	536ms																																				
1	1	0	1072ms																																				
1	1	1	2144ms																																				

4.6.4 PMU 中断使能寄存器（PMUIE）

PMUIE (PMU 中断使能寄存器)			基地址: 0x4000F400 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	X	X	BORIE	VCCIE
Write								
Reset:	0	0	0	0	0	0	0	0

注：需要同时使能 PMUIE 使能的中断才有效。

位	功能描述
BORIE	BOR检测中断使能位

	0: 关闭 1: 允许
VCCIE	VCC检测中断使能位 0: 关闭 1: 允许

4.6.5 PMU 中断标志寄存器 (PMUIF)

PMUIF (PMU 中断标志寄存器)			基地址: 0x4000F400 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	X	X	BORIF	VCCIF
Write								
Reset:	0	0	0	0	0	0	0	0

注: 该寄存器不能被 Wake_UP 唤醒复位。

位	功能描述
BORIF	BOR检测中断标志位 当内部工作电压VCC下降到低于设定阈值或上升到高于设定阈值时, 并且在BORRST=0的情况下, 该位置1。 软件写0清0。
VCCIF	VCC检测中断标志位 当系统电源VCC电压下降到低于设定阈值或上升到高于设定阈值时, 该位置1。 软件写0清0。

4.6.6 PMU 状态寄存器寄存器(PMUSTA)

PMUSTA (PMU 状态寄存器寄存器)			基地址: 0x4000F400 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read	X	X	X	X	X	X	BOR_FL G	VCC_FL G
Write			X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

注: 该寄存器为只读寄存器。

位	功能描述
BOR_FLG	工作电压 VCC 电压状态 0: 表示 VCC 小于设定阈值 (BOR_LVL[3:0]) 1: 表示 VCC 大于设定阈值 (BOR_LVL[3:0])
VCC_FLG	工作电压 VCC 电压状态



	0: 表示 VCC 小于设定阈值 (VCC_LVL[3:0]) 1: 表示 VCC 大于设定阈值 (VCC_LVL[3:0])
--	--

5 调试支持

5.1 概况

RX32S10 系列采用的是 Cortex-M0 内核，该内核含有硬件调试模块。Cortex-M0 处理器支持以下调试特性：

- 程序的暂停、恢复以及单步执行；
- 访问处理器内核寄存器和特殊寄存器；
- 硬件断点（最多 4 个）；
- 软件断点（BKPT 指令）；
- 数据监视点（最多两个）；
- 动态存储器访问；
- 支持串行线（SW）调试协议；

5.2 SW 引脚分布

SW 口调试接口		引脚描述
类型	描述	
输入/输出	SW 数据	PA. 0/SWIO/INT5
输出	SW 时钟	PA. 1/ SWCLK

5.3 SW 口使用说明

工作模式	管脚功能		
	PC. 6/TIM0/TIM8_CH4/INT4/JTAGWDTEN	PA. 0/SWIO/INT5	PA. 1/SWCLK
测试模式	X	X	X
调试模式	JTAGWDTEN	SWIO	SWCLK
正常模式	PC. 6/TIM0/TIM8_CH4/INT4	PA. 0/INT5	PA. 1

注：X 表示不可用

仿真口在上电/服务和启动之后的初始状态：

1、PA. 0\PA. 1 默认 SW 功能(PA. 0 为 SWIO 功能管脚，PA. 1 为 SWCLK 功能管脚)，TESTN 和 JTAG 不拉低，可进入调试模式。

2、PA. 0\PA. 1 管脚高阻态不被自动装载字节 HIGH_IMPEDANCE[3:0]控制。可通过 HIIPM 寄存器配置 PA. 0\PA. 1 管脚的高阻态。

3、TESTN 脚拉低，仿真口便被内部强制配置仿真功能，PA. 1 管脚配置为 SWCLK 固定输入，PA. 0 管脚配置为 SWIO 输出或输入（由执行指令决定，动态变化），PC. 6 管脚配置为 JTAGWDTEN。PA. 0\PA. 1\PC. 6 管脚的 GPIO 寄存器可正常操作，但寄存器与之对应的功能失效（包括开漏和高阻功能），只有调试仿真功能。

4、TESTN 脚拉高，芯片进入正常模式，PA.0\PA.1\PC.6 管脚功能由寄存器配置决定，即寄存器与之相对应的功能生效。

5.4 测试模式定义

TESTN 管脚由低电平变为高电平时，只复位测试模块逻辑。

TESTN 管脚拉低，PA.0\PA.1\PC.6 管脚状态由寄存器控制，不再强制配置为上拉输入。

5.5 应用场景

- 1、现有进入测试模式会影响 PA.0\PA.1\PC.6 所连接的外围器件。
- 2、电表反复上下电 RTC 日历出现异常值，偶尔出现 RTC 复位。
- 3、电表反复上下电，部分 IO 不能被 GPIO 寄存器操作
- 4、反复插拔烧写口，不能正常进入仿真。因误进测试模式，导致不能进入调试仿真。
- 5、PA.0\PA.1\PC.6、TESTN 作为烧写调试口，PA.0\PA.1 作为 SWD 接口。
- 6、LRC 时钟影响 SWD 调试和下载功能，系统时钟使用 LRC 时钟作为系统时钟，LRC 频率过低，造成 SWD 操作命令较慢，影响调试和下载速度。
- 7、MEM_BIST 测试 SRAM 是否异常。
- 8、芯片保持不掉电的情况下从测试模式切换到 Normal 模式，IO 口状态是否有异常。
- 9、缓慢上电至芯片工作电压，芯片是否异常。

6 工作模式

6.1 工作模式

芯片共有五种模式：测试模式，调试模式，正常模式，Sleep 模式，Hold 模式

TESTN	JTAGWDTEN	工作模式
0	1	测试模式
0	0	调试模式 此模式主要用来仿真下载程序
1	X	正常模式 芯片内部所有模块电源正常供电，系统时钟和模块开关配置根据用户软件决定，CPU 正常工作。
		Sleep 模式 在正常模式下，CPU通过执行如下指令进入Sleep模式： SCB->SCR = 0x0004; __WFI();
		Hold 模式 在正常模式下，CPU通过执行如下指令进入Hold模式： SCB->SCR = 0x0000; __WFI();

芯片在正常模式下可以通过软件配置进入两种低功耗模式，分别是 Sleep 模式和 Hold 模式。Sleep 模式和 Hold 模式最主要有以下两点区别：

- Sleep 模式可以获得更低的功耗
- Sleep 模式的唤醒等同复位，而 Hold 模式唤醒则是接着原来运行的代码继续运行

6.2 深眠模式（Sleep）

6.2.1 Sleep 模式下各模块开关

- 数字电源LDO_1P5关闭，其供电的模块相应关闭；
- RTC相关的晶振电路，分频补偿电路一直开启；
- 进入Sleep后，如果用户配置开启BOR和VCC检测功能，则BOR_DET和VCC_DET模块会由硬件分时开启以降低功耗；
- 在Sleep模式下，看门狗可配开关。如果使能看门狗，看门狗计数溢出时，系统会发生WDT复位。Sleep模式下看门狗配置详见2.4Flash控制功能。
- 为降低Sleep模式下的功耗，可以在进入Sleep模式之前，配置GPIO的状态（详见GPIO章节），控制好芯片和外设的状态，防止通过GPIO往外部漏电；
- 如果用户期望在Sleep达到最低功耗：CLKCTRL0和CLKCTRL1寄存器全部清0
- 进入Sleep 模式后，芯片内部会自动关闭LDO_1P5（大功耗）输出，LDO_LowPower（低功耗）保持输出1.5V。

6.2.2 Sleep 模式下的唤醒

CPU从Sleep状态下唤醒等同复位,因此不进入中断向量,不会执行中断服务程序,程序从复位地址0000H开始执行。

在Sleep模式下,Reset复位信号是不可被屏蔽的,包括POR, BOR, LBOR, 外部RESET PIN上产生的外部复位信号,以及内部的WDT复位信号。当系统进入Sleep模式后,如果以上复位信号产生,能够使芯片出现复位动作,程序从复位地址0000H开始执行。

要实现在 Sleep 模式下的唤醒功能,进入 Sleep 模式前需进行以下功能配置,必需先使能总中断:

(1) **外部 INT 中断和串口 RX 引脚唤醒:** 相应的 PIN 要配置为 INT 和 RX 功能(详细见 GPIO 单元),并使能外部中断和 UART 中断。当 INT 上升沿(或下降沿)中断使能相应 PIN 上出现上升沿(或下降沿),RX 中断使能相应 PIN 上出现下降沿,并保持低电平不少于 2 个 Fcpu 的时间,CPU 可从 Sleep 模式下唤醒。注意,这里的 Fcpu 时钟为 CPU 时钟分频寄存器的输出时钟,如果用户在进入 Sleep 之前 CPU 时钟分频寄存器 SYCLKDIV 的分频值过大(如 1/128),则会造成外部唤醒 PIN 上要给出很长时间(超过 2 个 Fcpu 时钟)的低电平信号,才能将芯片从 Sleep 模式下唤醒。

(2) **RTC 中断唤醒:** 配置 RTC 模块中断使能,并且配置对应 RTC 的子中断源(仅配置 RTCIE 相应位中断使能),当 RTC 使能的中断时间到时,或者 RTC 使能的闹钟定时或定时器定时时间到时,可以让 CPU 从 Sleep 模式下唤醒。

(3) **PMU 中断:** 配置 PMU 模块中断使能,并且使能 PMU 对应的子中断源(配置 PMUIE 相应位中断使能),当电源检测超过阈值超过阈值时,可以让 CPU 从 Sleep 模式下唤醒。

(4) **UART 中断唤醒:** 配置需要唤醒的 RX PIN 为 RX 功能,并配置相应管脚 RX 中断使能,打开相应内核中断: NVIC_EnableIRQ(UARTx_IRQn); 当 RX pin 出现有效的下降沿(不少于 2 个 Fcpu 的时间)唤醒信号,CPU 可从 Sleep 模式唤醒。UART 唤醒不用使能 UART 模块。

6.2.3 从 Sleep 模式唤醒后的唤醒方式确认

从Sleep模式唤醒后,可以查询复位标志寄存器(RSTSTA寄存器的WakeupRST位),如果该位为1,则说明确实发生了唤醒复位,然后再通过唤醒标志寄存器WAKEIF确定具体的唤醒源,其中:

- 1) WAKEIF的RTCWKIF位为1,表示CPU是由RTC中断信号引起了唤醒.具体的中断源由RTC的8种中断源确定,可以查询RTCIF对应的位来确认是哪种RTC中断引起的唤醒,详细见RTC单元。
- 2) WAKEIF的PMUWKIF位为1,表示CPU是由PMU中断信号引起了唤醒.具体的中断源由PMU的2种中断源确定,可以查询PMUIF对应的位来确认,详细见PMU单元(电源单元)。

6.2.4 进入 Sleep 模式

Sleep模式通过CortexM0的系统自带指令WFI进入。进入Sleep指令如下:

```
SCB->SCR = 0x0004;
```

```
__WFI();
```

在仿真调试状态下,执行上述指令,cpu停止,但未进Sleep模式。

6.3 待机模式(Hold)

Hold 模式与 Sleep 模式的区别就是在 Hold 模式下,LD0_1P5 是由用户控制开关的(控制位),数字的 LD0_LowPower 供电一直打开,但是由于其低输出驱动能力(20-30uA),导致在这个状态下,很多数字功能模块不能使能,用户可配置开启大功率大输出驱动能力的 LD0_1P5 来适应其应用的需求。

为了降低 Hold 模式下的功耗,BOR_DET, VCC_DET 模块由芯片硬件分时开启。

如果系统在进入 Hold 模式之前配置了中断使能,在进入 Hold 模式后发生相应的中断事件,则会导致芯片从 Hold 模式下唤醒,并进入相应的中断处理程序。

在Hold模式下看门狗可配开关，Hold模式下看门狗配置详见2.4Flash控制功能

6.3.1 进入Hold 模式

Hold模式通过CortexM0的系统自带指令WFI进入。

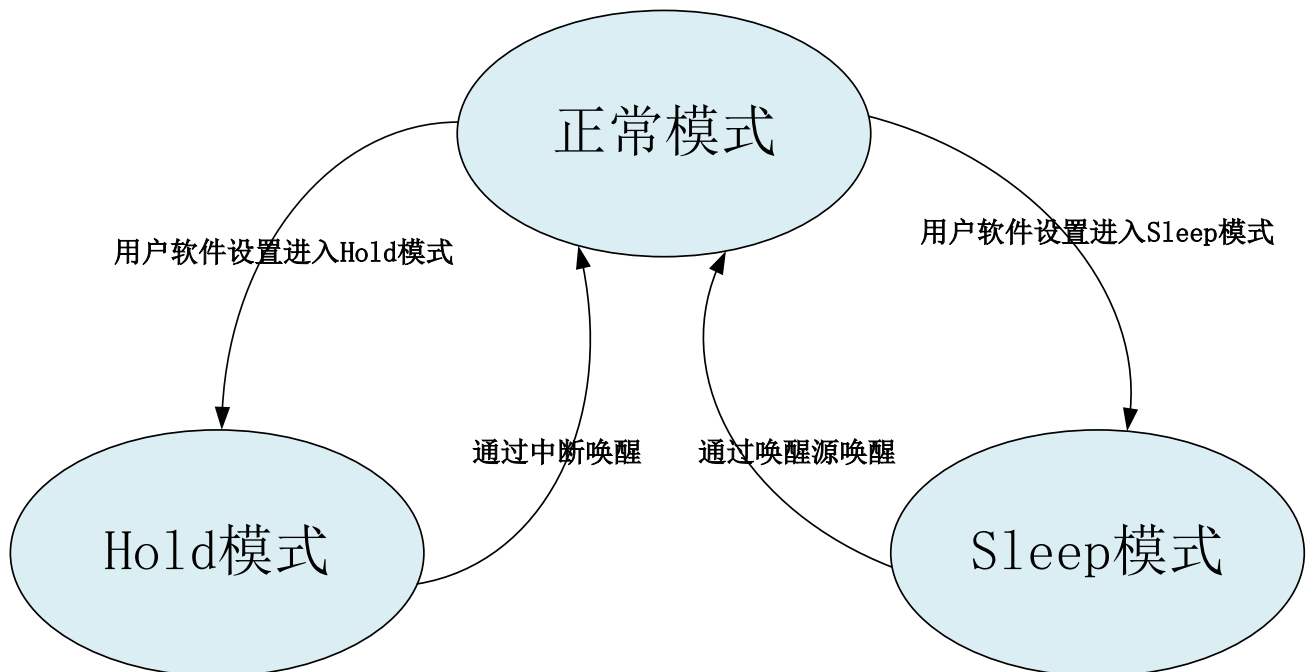
进入Hold指令如下：

```
SCR = 0x00;
```

```
_WFI();
```

在仿真调试状态下，执行上述指令，cpu停止，芯片进入Hold模式。

6.4 模式转换图



6.5 特殊功能寄存器列表

基地址：0xE000ED00				
偏移地址	名称	读写方式	复位值	功能描述
0x10	SCR	R/W	0x0000	系统控制寄存器

基地址：0x4000F400				
偏移地址	名称	读写方式	复位值	功能描述
0x18	WAKEIF	R/W	0x0000	唤醒源标志寄存器

6.6 特殊功能寄存器说明

6.6.1 系统控制寄存器(SCR)

SCR (系统控制寄存器)			基地址: 0xE000ED10 偏移地址: 10H					
	Bit31	30	29	28...11	10	9	8	Bit0
Read:	SCR[31:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:			RESERVE		SLEEPDE		RESERVE	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SLEEPDEEP	=1: 在执行 WFI 指令后, 芯片进入 Sleep 模式 =0: 在执行 WFI 指令后, 芯片进入 Hold 模式

6.6.2 唤醒标志寄存器(WAKEIF)

WAKEIF (唤醒标志寄存器)			基地址: 0x4000F400 偏移地址: 18H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	X	X	INT8WKI	INT7WKI	X	X	X	X
Write:			F	F				
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	RTCWKIF	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	RX1WKIF	RX0WKIF	INT6WKI
Write:								F
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	INT5WKI	INT4WKI	INT3WKI	INT2WKI	INT1WKI	INT0WKI	X	PMUWKIF
Write:	F	F	F	F	F	F		
Reset:	0	0	0	0	0	0	0	0

注: Sleep 唤醒和 Hold 唤醒共享此标志位

该寄存器为只读寄存器, 它永远会保持上一次导致芯片唤醒的唤醒源头, 当一个新的唤醒事件产生时候, 由硬件产生新的唤醒源标志, 同时将之前的唤醒标志清 0

位	功能描述
---	------

INT8WKIF	INT8唤醒标志 INT8唤醒发生时，此位置为1
INT7WKIF	INT7唤醒标志 INT7唤醒发生时，此位置为1
RTCWKIF	RTC唤醒标志 SLEEP/HOLD模式下RTC中断发生时将会产生RTC唤醒，此位置为1。（具体RTC那个唤醒源头需要查看RTCIF寄存器）
RX1WKIF	RX1唤醒标志 RX1唤醒发生时，此位置为1
RX0WKIF	RX0唤醒标志 RX0唤醒发生时，此位置为1
INT6WKIF	INT6唤醒标志 INT6唤醒发生时，此位置为1
INT5WKIF	INT5唤醒标志 INT5唤醒发生时，此位置为1
INT4WKIF	INT4唤醒标志 INT4唤醒发生时，此位置为1
INT3WKIF	INT3唤醒标志 INT3唤醒发生时，此位置为1
INT2WKIF	INT2唤醒标志 INT2唤醒发生时，此位置为1
INT1WKIF	INT1唤醒标志 INT1唤醒发生时，此位置为1
INT0WKIF	INT0唤醒标志 INT0唤醒发生时，此位置为1
PMUWKIF	PMU唤醒标志 Sleep/HOLD模式下PMU事件发生时将会产生PMU唤醒，此位置为1（具体哪个PMU唤醒源需要查看PMUIF寄存器。）

7 GPIO 模块

7.1 概述

RX32S10系列提供的I/O包括：PA.0~PA.7，PB.0~PB.9，PC.0~PC.10，PD.0~PD.9，支持39个双向I/O引脚。

每个I/O口输出都可配为推挽输出或者开漏输出；若配为输入模式，则输入上拉可配置，输入上拉电阻为约90k。

各个I/O都具有5mA的输出驱动能力，参见1.4引脚定义。

PA.0、PA.1默认态为下载模式，非高阻态。

PA.2、PA.3默认为外部高速晶振脚位

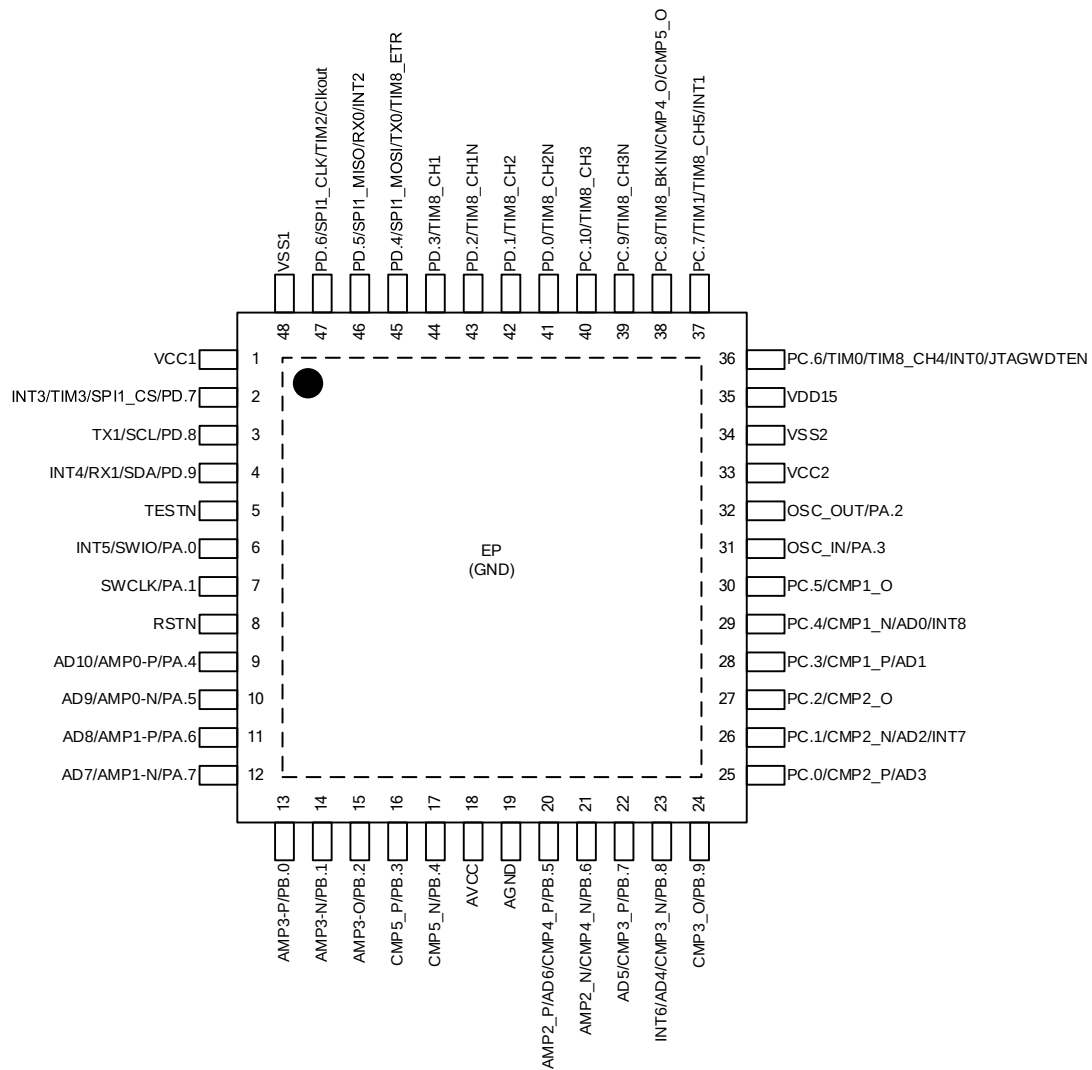
TESTN=0时，PA.0、PA.1为SW下载模式接口，防止SW接口为高阻态，芯片调试和下载不正常。

除PA.0，PA.1以外所有的GPIO上电复位为输入上拉（Autoload未配置的情况下）。

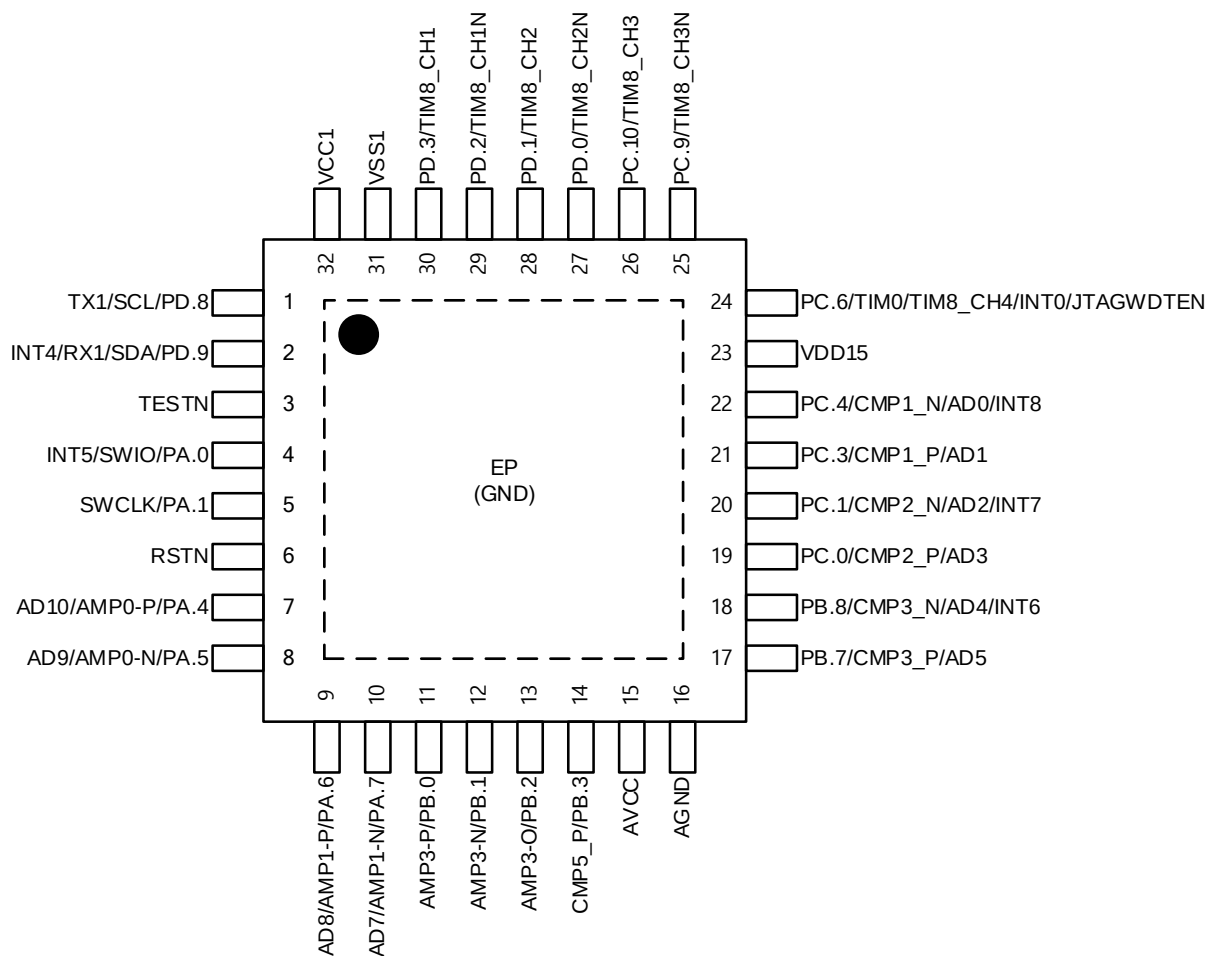
特定GPIO支持，上电初始化状态进行趋向性设计，即电源电压未达到逻辑生效电压（LDO未建立）仍能保持高阻状态

7.2 引脚排列

LQFP48



QFN32



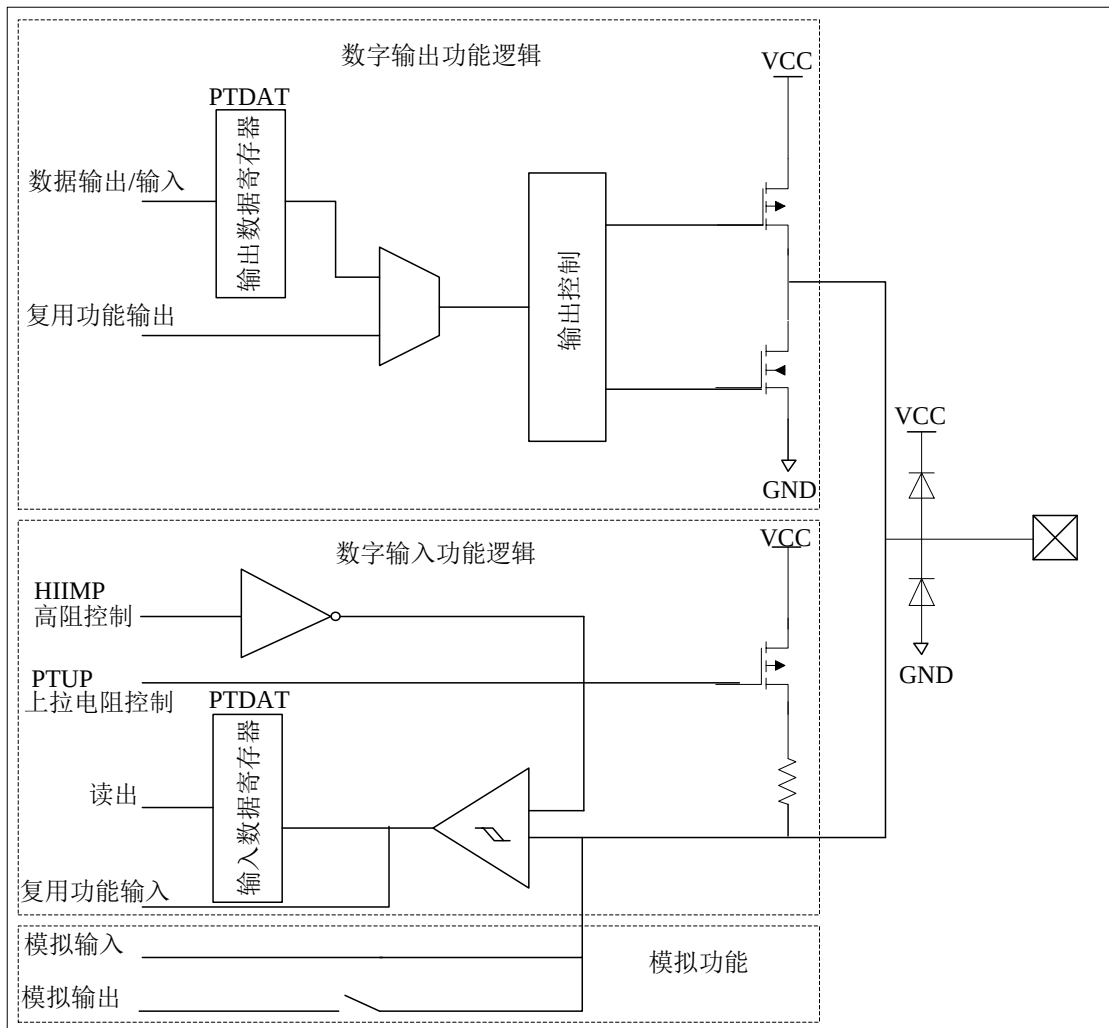
7.3 引脚定义

48 PIN	标识	引脚 类型	滤波	第一复用 功能	第二复用 功能	第三复用 功能	引脚说明
1	VCC1	P					芯片数字电源
2	PD. 7	I/O	2us	SPI1_CS	TIM3	INT3	GPIO\SPI1_CS\Timer输出输入\外部中断, 滤波2us
3	PD. 8	I/O		SCL	TX1		GPIO\SCL\TX
4	PD. 9	I/O	2us	SDA	RX1	INT4	GPIO\SDA\RX\外部中断, 滤波2us
5	TESTN	I	2us				测试引脚(低电平有效, 内部上拉), 输入滤波2us
6	PA. 0	I/O	2us	SWIO		INT5	GPIO\SWIO\外部中断, 滤波2us
7	PA. 1	I/O		SWCLK			GPIO\SWCLK
8	RSTN	I	2us				复位信号(低电平有效, 内部上拉), 滤波2us
9	PA. 4	I/O		AMP0_P\AD10			GPIO\OP-AMP正端输入\ADC信号输入
10	PA. 5	I/O		AMP0_N\AD9			GPIO\OP-AMP负端输入\ADC信号输入
11	PA. 6	I/O		AMP1_P\AD8			GPIO\OP-AMP正端输入\ADC信号输入
12	PA. 7	I/O		AMP1_N\AD7			GPIO\OP-AMP负端输入\ADC信号输入
13	PB. 0	I/O		AMP3_P			GPIO\OP-AMP正端输入
14	PB. 1	I/O		AMP3_N			GPIO\OP-AMP负端输入
15	PB. 2	I/O		AMP3_O			GPIO\OP-AMP输出
16	PB. 3	I/O		CMP5_P			GPIO\比较器正端输入
17	PB. 4	I/O		CMP5_N			GPIO\比较器负端输入
18	AVCC	P					芯片模拟电源
19	AGND	G					芯片模拟地
20	PB. 5	I/O		CMP4_P\AD6\AMP2_P			GPIO\比较器正端输入\ADC信号输入\OP-AMP正端输入
21	PB. 6	I/O		CMP4_N\AMP2_N			GPIO\比较器负端输入\OP-AMP负端输入
22	PB. 7	I/O		CMP3_P\AD5			GPIO\比较器正端输入\ADC信号输入
23	PB. 8	I/O	2us	CMP3_N\AD4		INT6	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
24	PB. 9	I/O		CMP3_O			GPIO\比较器输出
25	PC. 0	I/O		CMP2_P\AD3			GPIO\比较器正端输入\ADC信号输入
26	PC. 1	I/O	2us	CMP2_N\AD2		INT7	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
27	PC. 2	I/O		CMP2_O			GPIO\比较器输出
28	PC. 3	I/O		CMP1_P\AD1			GPIO\比较器正端输入\ADC信号输入
29	PC. 4	I/O	2us	CMP1_N\AD0		INT8	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
30	PC. 5	I/O		CMP1_O			GPIO\比较器输出
31	PA. 3	I/O		OSC_In			GPIO\高频晶振时钟输入
32	PA. 2	I/O		OSC_Out			GPIO\高频晶振时钟输出
33	VCC2	P					芯片数字电源
34	VSS2	G					芯片数字地
35	VDD15	P					内部1.5V输出, 需外接0.1uF滤波电容
36	PC. 6	I/O	2us	TIM0	TIM8_CH4	INT0	GPIO\Timer输出输入\Timer8_CH4\外部中断, 滤波2us\TEST=0该引脚功能为JTAGWDTEN功能
37	PC. 7	I/O	2us	TIM1	TIM8_CH5	INT1	GPIO\Timer输出输入\Timer8_CH5\外部中断, 滤波2us
38	PC. 8	I/O		TIM8_BKIN	CMP4_O	CMP5_O	GPIO\Timer8_BKIN\比较器4输出\比较器5输出
39	PC. 9	I/O		TIM8_CH3N			GPIO\ Timer8_CH3N
40	PC. 10	I/O		TIM8_CH3			GPIO\ Timer8_CH3
41	PD. 0	I/O		TIM8_CH2N			GPIO\ Timer8_CH2N
42	PD. 1	I/O		TIM8_CH2			GPIO\ Timer8_CH2
43	PD. 2	I/O		TIM8_CH1N			GPIO\ Timer8_CH1N
44	PD. 3	I/O		TIM8_CH1			GPIO\ Timer8_CH1
45	PD. 4	I/O		SPI1_MOSI	TX0	TIM8_ETR	GPIO\SPI1_MOSI\TX\Timer8_ETR
46	PD. 5	I/O	2us	SPI1_MISO	RX0	INT2	GPIO\SPI1_MISO\RX\外部中断, 滤波2us
47	PD. 6	I/O		SPI1_CLK	TIM2	Clkout	GPIO\SPI1_CLK\Timer输出输入\Clkout
48	VSS1	G					芯片数字地



32 PIN	标识	引脚 类型	滤波	第一复用 功能	第二复用 功能	第三复用 功能	引脚说明
1	PD. 8	I/O		SCL	TX1		GPIO\SCL\TX
2	PD. 9	I/O	2us	SDA	RX1	INT4	GPIO\SDA\RX\外部中断, 滤波2us
3	TESTN	I	2us				测试引脚 (低电平有效, 内部上拉), 输入滤波2us
4	PA. 0	I/O	2us	SWIO		INT5	GPIO\SWIO\外部中断, 滤波2us
5	PA. 1	I/O		SWCLK			GPIO\SWCLK
6	RSTN	I	2us				复位信号 (低电平有效, 内部上拉), 滤波2us
7	PA. 4	I/O		AMPO_P\AD10			GPIO\OP-AMP正端输入\ADC信号输入
8	PA. 5	I/O		AMPO_N\AD9			GPIO\OP-AMP负端输入\ADC信号输入
9	PA. 6	I/O		AMP1_P\AD8			GPIO\OP-AMP正端输入\ADC信号输入
10	PA. 7	I/O		AMP1_N\AD7			GPIO\OP-AMP负端输入\ADC信号输入
11	PB. 0	I/O		AMP3_P			GPIO\OP-AMP正端输入
12	PB. 1	I/O		AMP3_N			GPIO\OP-AMP负端输入
13	PB. 2	I/O		AMP3_O			GPIO\OP-AMP输出
14	PB. 3	I/O		CMP5_P			GPIO\比较器正端输入
15	AVCC	P					芯片模拟电源
16	AGND	G					芯片模拟地
17	PB. 7	I/O		CMP3_P\AD5			GPIO\比较器正端输入\ADC信号输入
18	PB. 8	I/O	2us	CMP3_N\AD4		INT6	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
19	PC. 0	I/O		CMP2_P\AD3			GPIO\比较器正端输入\ADC信号输入
20	PC. 1	I/O	2us	CMP2_N\AD2		INT7	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
21	PC. 3	I/O		CMP1_P\AD1			GPIO\比较器正端输入\ADC信号输入
22	PC. 4	I/O	2us	CMP1_N\AD0		INT8	GPIO\比较器负端输入\ADC信号输入\外部中断, 滤波2us
23	VDD15	P					内部1.5V输出, 需外接0.1uF滤波电容
24	PC. 6	I/O	2us	TIM0	TIM8_CH4	INT0	GPIO\Timer输出输入\Timer8_CH4\外部中断, 滤波2us\ TEST=0该引脚功能为JTAGWDTEN功能
25	PC. 9	I/O		TIM8_CH3N			GPIO\ Timer8_CH3N
26	PC. 10	I/O		TIM8_CH3			GPIO\ Timer8_CH3
27	PD. 0	I/O		TIM8_CH2N			GPIO\ Timer8_CH2N
28	PD. 1	I/O		TIM8_CH2			GPIO\ Timer8_CH2
29	PD. 2	I/O		TIM8_CH1N			GPIO\ Timer8_CH1N
30	PD. 3	I/O		TIM8_CH1			GPIO\ Timer8_CH1
31	VSS1	G					芯片数字地
32	VCC1	P					芯片数字电源

7.4 芯片引脚结构说明



7.5 关于高阻状态的说明

RX32S10 系列的 GPIO 具有高阻状态，关于高阻状态时，GPIO 具有以下特征：

- 除 PA.0、PA.1 以外的 GPIO 复位后为输入上拉，PA.0、PA.1 复位后为 SWIO, SWCLK
- GPIO 可通过寄存器控制或 Autoload 加载为高阻态：
 - 通过寄存器将 GPIO 配置为高阻的方式为：端口功能为 GPIO，端口方向为输入，关闭上拉电阻，开启高阻控制；
 - 当端口通过 Autoload 加载为高阻态时，端口上拉关闭，此时对应的 HIIPM 寄存器的 GPIO bit 置 1，PTUP 寄存器的 GPIO bit 为默认值 0，但此时此值无意义，不能指示上拉电阻状态；（出厂为此配置）
 - 当端口通过 Autoload 加载为非高阻态时，端口上拉开启，此时对应的 HIIPM 寄存器的 GPIO bit 清 0，PTUP 寄存器的 GPIO bit 清 0。
 - PA.0、PA.1 和无法通过 Autoload 加载为高阻态。

- 当端口配置为高阻时，端口将强制切换为 GPIO 功能，且上拉电阻关闭，除 HIIPM 外的相关寄存器的值将保持配置为高阻前的状态，但不会生效。
- 具有模拟输入复用功能的 IO，其模拟输入功能不会受到 GPIO 模块寄存器配置的影响，即使没有将 IO 复用为相应的功能，IO 仍然可以作为模拟输入 pin 使用。因此为了保证功能正常，在将具有模拟输入功能的 IO 配置为高阻态、或其他不期望模拟输入功能生效的状态时，应同时关闭其模拟输入功能对应的模块使能。

7.6 I/O 端口基地址列表

GPIO 模块寄存器基地址： 0x40011000 (PA 端口)； 0x40011100 (PB 端口)； 0x40011200 (PC 端口)； 0x40011300 (PD 端口)；				
偏移地址	名称	读写方式	复位值	功能描述
0x00	IOCFG	R/W	0x0000	端口功能配置寄存器 1 (写保护)
0x04	AFCFG1	R/W	0x0000	端口复用功能配置寄存器 1 (写保护)
0x08	PTDIR	R/W	0x0000	端口方向配置寄存器
0x0C	PTUP	R/W	0x0000	端口上拉配置寄存器 4*
0x10	PTDAT	R/W	0x0000	端口数据寄存器
0x14	PTSET	W	0x0000	端口设置寄存器 (只写)
0x18	PTCLR	W	0x0000	端口复位寄存器 (只写)
0x1C	PTTOG	W	0x0000	端口翻转寄存器 (只写)
0x20	PTOD	R/W	0xFFFF	端口 Open Drain 功能配置寄存器
0x24	FILT	R/W	0xFFFF	GPIO 输入模拟滤波控制寄存器
0x28	HIIPM	R/W	0xFFFF	端口高阻控制寄存器 4*
0x2C	NEWAFEN	R/W	0x0000	新增复用功能生效控制寄存器
0x30	AFCFG2	R/W	0x0000	端口复用功能配置寄存器 2 (写保护)

注：

- 针对 FilT 寄存器，不同 port 口复位值不同，复位值仅对应的端口有 2us 滤波为 1，其余为 0；
- 端口配置为高阻态方式：端口功能为 GPIO，端口方向为输入，关闭上拉电阻，开启高阻控制；
- 端口复位态为高阻，上拉电阻关闭，此时无法响应外部输入，输入内部锁定为 0；
- 当端口通过 Autoload 加载为高阻态时，端口上拉关闭，此时对应的 HIIPM 寄存器的 GPIO bit 置 1，PTUP 寄存器的 GPIO bit 为默认值 0，但此时此值无意义，不能指示上拉电阻状态；
当端口通过 Autoload 加载为非高阻态时，端口上拉开启，此时对应的 HIIPM 寄存器的 GPIO bit 清 0，PTUP 寄存器的 GPIO bit 清 0。
- PA.1、PA.2 无法通过 Autoload 加载为高阻态。
- 当端口配置为高阻时，端口将强制切换为 GPIO 功能，且上拉电阻关闭，除 HIIPM 外的相关寄存器的值将保持配置为高阻前的状态，但不会生效。

7.7 特殊功能寄存器说明

7.7.1 端口功能配置寄存器 1(IOCFCG)

IOCFCG (写保护) (端口功能配置寄存器 1)			基地址: 0x40011000--0x40011300 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口 IO 功能配置位 0: 对应的端口配置为 GPIO 1: 对应的端口配置为功能 PIN

注:

1. PA. 0、PA. 1 的对应 bit 位默认为 1, 选择第一复用功能 SW;

7.7.2 端口复用功能配置寄存器 1(AFCFG1)

AFCFG1 (写保护) (端口复用功能配置寄存器 1)			基地址: 0x40011000--0x40011300 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	SEL7[1:0]		SEL6[1:0]		SEL5[1:0]		SEL4[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SEL3[1:0]		SEL2[1:0]		SEL1[1:0]		SEL0[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SELx[1:0]	端口复用功能配置位 (此寄存器只在对应端口配置为功能 PIN 时才有效) 0: 复用功能 1 1: 复用功能 2 2: 复用功能 3 3: 预留

7.7.3 端口复用功能配置寄存器 2(AFCFG2)

AFCFG2 (写保护)	基地址: 0x40011000--0x40011300
--------------	-----------------------------

(端口复用功能配置寄存器 2)			偏移地址: 30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	SEL15[1:0]		SEL14[1:0]		SEL13[1:0]		SEL12[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SEL11[1:0]		SEL10[1:0]		SEL9[1:0]		SEL8[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SELx[1:0]	端口复用功能配置位（此寄存器只在对应端口配置为功能 PIN 时才有效） 0: 复用功能 1 1: 复用功能 2 2: 复用功能 3 3: 预留

7.7.4 端口方向配置寄存器(PTDIR)

PTDIR (端口方向配置寄存器)			基地址: 0x40011000--0x40011300 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口方向配置位（此寄存器只在对应端口配置为 GPIO 功能时才有效） 0: 输入 1: 输出

7.7.5 端口上拉配置寄存器(PTUP)

PTUP (端口上拉配置寄存器)			基地址: 0x40011000--0x40011300 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							

Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口上拉配置位（此寄存器只在对应端口配置为数字输入时才有效） 0：使能上拉 1：禁止上拉（浮空）

注：数字输入含配置为输入模式的 GPIO 和具有数字输入属性的复用功能。

7.7.6 端口数据寄存器(PTDAT)

PTDAT (端口数据寄存器)			基地址：0x40011000—0x40011300 偏移地址：10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PT[15:0]	端口数据位（此寄存器只在对应端口配置为 GPIO 功能，或含有数字输入属性的复用功能时才有效） 当端口配置为输入时为读到的 IO 口状态 0：读到的为低电平 1：读到的为高电平 当端口配置为输出时 0：输出低电平 1：输出高电平

注：端口数据寄存器 PTDAT 说明及数据读取规则：

- 1) 芯片引脚选择 GPIO 功能：
若方向寄存器配置为输出，PTDAT 读取值为寄存器设置值，不随外部 PIN 脚电平变化而变化；
若方向寄存器配置为输入，PTDAT 读取值为 pad 状态值，反映外部 PIN 脚电平变化；
- 2) 芯片引脚选择复用数字功能：
若复用为数字输出功能，PTDAT 读取值为切换成复用功能前的 GPIO 的 PTDAT 值，不随外部 PIN 脚电平变化而变化；
若复用为数字输入功能，PTDAT 读取值为 pad 状态值，反映外部 PIN 脚电平变化；
- 3) 芯片引脚选择复用模拟功能，PTDAT 相应 bit 位值，固定为 0。

7.7.7 端口设置寄存器(PTSET)

PTSET (端口设置寄存器)			基地址：0x40011000—0x40011300 偏移地址：14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

注：本寄存器只可写入。

位	功能描述
PT[15:0]	端口设置位（此寄存器只在对应端口配置为 GPIO 且输出时才有效） 0：写 0 无效 1：写 1 将对应的端口输出高电平（同时更新 PTDAT 中对应的值）

7.7.8 端口复位寄存器(PTCLR)

PTCLR (端口复位寄存器)			基地址： 0x40011000—0x40011300 偏移地址： 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

注：本寄存器只可写入。

位	功能描述
PT[15:0]	端口复位位（此寄存器只在对应端口配置为 GPIO 且输出时才有效） 0：写 0 无效 1：写 1 将对应的端口输出低电平（同时更新 PTDAT 中对应的值）

7.7.9 端口翻转寄存器(PTTOG)

PTTOG (端口翻转寄存器)			基地址： 0x40011000—0x40011300 偏移地址： 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:	PT[15:8]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	X
Write:	PT[7:0]							
Reset:	0	0	0	0	0	0	0	0

注：本寄存器只可写入。

位	功能描述
PT[15:0]	端口翻转位（此寄存器只在对应端口配置为 GPIO 且输出时才有效） 0：写 0 无效 1：写 1 将使对应的端口输出电平发生翻转（同时更新 PTDAT 中对应的值）

7.7.10 端口开漏配置寄存器(PTOD)

PTOD (端口开漏配置寄存器)			基地址: 0x40011000—0x40011300 偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT[15:8]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
PT[15:0]	端口开漏配置位（此寄存器只在对应端口配置为数字输出时才有效） 0: 开漏功能使能（开漏输出，输出高为浮空，输出低为低） 1: 开漏功能无效（推挽输出，输出高为高，输出低为低）

注：数字输出含配置为输出模式的 GPIO 和具有数字输出属性的复用功能。

7.7.11 模拟功能输入滤波控制寄存器(FILT)

FILT (模拟功能输入滤波控制寄存器)			基地址: 0x40011000—0x40011300 偏移地址: 24H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PT2u[15:8]							
Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	PT2u[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

注：该寄存器为单独的起始地址

位	功能描述
PT2u[15:0]	端口模拟负脉冲 2us 滤波配置位 0: 模拟滤波有效 1: 模拟滤波无效 (default)

7.7.12 端口高阻控制寄存器(HIIPM)

HIIPM (端口高阻控制寄存器)			基地址: 0x40011000—0x40011300 偏移地址: 28H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Hiipm[15:8]							

Write:								
Reset:	1	1	1	1	1	1	1	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	Hiipm[7:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
Hiipm[15:0]	端口高阻配置位 0: 高阻关闭 1: 高阻开启 (default) 注: I/O default 由自加载控制。

注: PA. 1, PA. 2, PA. 3, PA. 4, TESTN, RSTN default 不设定高阻

7.7.13 新增复用功能生效控制寄存器 (NEWAFEN)

NEWAFEN (新增复用功能生效控制寄存器)			基地址: 0x40011000—0x40011300 偏移地址: 2CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	NewAF_EN[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	NewAF_EN[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
NewAF_EN[15:0]	新增复用功能生效控制位 0: 新增复用功能不生效 (Default) 1: 新增复用功能生效

7.8 应用场景

- GPIO 默认态高阻态;
- 芯片上电复位完成后, 或其他复位完成后, 所有 GPIO 寄存器均可任意配置, 不限制用户程序行为 (含新增 GPIO 口寄存器);
- 当从复用态切换为高阻态时, 复用功能不受到影响, 复用的功能不能被误触发 (例如: INT 管脚被复用, 切换到高阻态, INT 中断标志不能建立);
- 高阻态切换到复用态不影响复用功能;
- 处于复位一、二等级的复位, 正在复位期间由 Autoload 控制, 任何复位结束, GPIO 高阻态或上拉输入配置正确, 不会被加载错误, 或者不能被误加载;
- 处于复位三等级的复位, 不会复位 GPIO 配置。Autoload 不会加载 GPIO 配置;
- PTDAT 反映 GPIO 输入 PIN 状态或输出 register 值, 当配置为模拟功能 pin 时, 读出值为 0;
- 先配置 GPIO 复用功能, 然后再配置功能模块, 即配置复用功能到配置模块期间, 期间有外部电平发生变化, 不会引起功能模块中断标志清除不掉;

9. 一般 GPIO 的高低电平应符合数字确认的指标，需确认 3V 系统和 5V 系统最大低电平和最小高电平是否能被芯片正确识别；
10. 各个 I/O 具有 5mA 驱动能力。所有 GPIO 驱动能力之和应符合模拟给出的指标，并给出拉电流驱动能力 & VCC 电压关系图，并给出温度特性。
11. 各个 GPIO 的上拉电阻和下拉能力应符合模拟给出的指标。
12. 常用 Case:

					仅输入有效	仅输出有效	仅输出有效	仅输出有效	
	高阻	IOCFG	AFCFG	PTDIR	PTUP	PTOD	PTSET	PTCLR	说明
通用应用	0-高阻关闭	0-GPIO	可任意配置	0-输入	0-上拉	可任意配置	可任意配置	可任意配置	此时可任意配置的寄存器实际配置值无效。
	0-高阻关闭	0-GPIO	可任意配置	0-输入	1-浮空	可任意配置	可任意配置	可任意配置	此时可任意配置的寄存器实际配置值无效。
	0-高阻关闭	0-GPIO	可任意配置	1-输出	可任意配置	1-推挽	可任意配置	可任意配置	此时除 PTSET 和 PTCLR，其他可任意配置的寄存器实际配置值无效。
	0-高阻关闭	0-GPIO	可任意配置	1-输出	可任意配置	0-开漏	可任意配置	可任意配置	此时除 PTSET 和 PTCLR，其他可任意配置的寄存器实际配置值无效。
	0-高阻关闭	1-复用功能	数字输入	可任意配置	0-上拉	可任意配置	可任意配置	可任意配置	此时可任意配置的寄存器实际配置值无效。
	0-高阻关闭	1-复用功能	数字输入	可任意配置	1-浮空	可任意配置	可任意配置	可任意配置	此时可任意配置的寄存器实际配置值无效。
	0-高阻关闭	1-复用功能	数字输出	可任意配置	可任意配置	1-推挽	可任意配置	可任意配置	此时可任意配置的寄存器实际配置值无效。
	0-高阻关闭	1-复用功能	数字输出	可任意配置	可任意配置	0-开漏	可任意配置	可任意配置	此时可任意配置的寄存器实际配置值无效。
	1-高阻开启	可任意配置	可任意配置	可任意配置	可任意配置	可任意配置	可任意配置	可任意配置	此时固定为 GPIO 高阻状态，其他寄存器配置均无效。默认配置
固定应用	0-高阻关闭	0-GPIO	可任意配置	1-输出	可任意配置	1-推挽	1-输出高	0-无操作	测试 Isource
	0-高阻关闭	0-GPIO	可任意配置	1-输出	可任意配置	1-推挽	0-无操作	1-输出低	测试 Isink
	0-高阻关闭	0-GPIO	可任意配置	1-输出	可任意配置	0-开漏	可任意配置	可任意配置	客户低功耗应用

8 中断模块

8.1 中断向量说明

系统中断	中断号	中断使能	中断标志	功能描述
NMI	-14			硬件强制切换低频 RC 中断
HardFault	-13			故障/异常触发中断
SVCall	-5			软件触发中断
PendSV	-2			软件触发中断
SysTick	-1			系统定时器周期中断
以上为内核自带				
PMU	0	PMUIE. BORIE	PMUIF. BORIF	BOR检测中断
		PMUIE. VCCIE	PMUIF. VCCIF	VCC检测中断
SAR ADC	1	SADCIE. SIGIE	SADCIF. SIGIF	SADC 单次转换完成中断
		SADCIE. SEQIE	SADCIF. SEQIF	SADC 扫描序列转换完成中断
		SADCIE. TPSIE	SADCIF. TPSIF	SADC 温度转换完成中断
EXTIO-6	2-8	EXTIE. RIE[6:0]	EXTIF. RIF[6:0]	外部输入引脚上升沿中断
		EXTIE. FIE[6:0]	EXTIF. FIF[6:0]	外部输入引脚下降沿中断
UART0-1	9-10	UARTCON. RXIE	UARTSTA. RXIF	UART 接收中断
		UARTCON. TXIE	UARTSTA. TXIF	UART 发送中断
		UARTCON. PRDIE	UARTSTA. PRDIF	UART 溢出中断
CMP1-3	11-13	CMPx_CR. FIE	CMPx_CR. FIF	比较器下降沿中断
		CMPx_CR. RIE	CMPx_CR. RIF	比较器上升沿中断
ME	14	ME_IER. DIVFE	ME_IFR. DIVFF	除法器错误中断
		ME_IER. PID1E	ME_IFR. PID1F	PID1 完成中断
		ME_IER. PID2E	ME_IFR. PID2F	PID2 完成中断
		ME_IER. PID3E	ME_IFR. PID3F	PID3 完成中断
		ME_IER. CDTRE	ME_IFR. CDTRF	座标转换完成中断
		ME_IER. IPDE	ME_IFR. IPDF	IPD 完成中断
		ME_IER. SMOE	ME_IFR. SMOF	SMO 完成中断
		ME_IER. SVE	ME_IFR. SVF	SVPWM 完成中断
		ME_IER. F4E	ME_IFR. F4F	F4 完成中断
TMR0-3	15-18	TMRIE. CMPIE	TMRIF. CMPIF	比较中断
		TMRIE. CAPIE	TMRIF. CAPIF	捕获中断
		TMRIE. PRDIE	TMRIF. PRDIF	周期性溢出中断
RTC	20	RTCIE. ALMIE	RTCIF. ALMIF	RTC 闹铃中断
		RTCIE. RTC2IE	RTCIF. RTC2IF	RTC 定时器 2 中断
		RTCIE. RTC1IE	RTCIF. RTC1IF	RTC 定时器 1 中断
		RTCIE. MTHIE	RTCIF. MTHIF	RTC 月中断
		RTCIE. DAYIE	RTCIF. DAYIF	RTC 日中断
		RTCIE. HRIE	RTCIF. HRIF	RTC 小时中断
		RTCIE. MINIE	RTCIF. MINIF	RTC 分钟中断
		RTCIE. SECIE	RTCIF. SECIF	RTC 秒中断

I2C	21		I2CCON. SI	I2C 传输中断
SPI0	22		SPISTA. SPIF	SPI 传输完成中断
			SPISTA. MODF	SPI 传输错误中断
TIM8	27	TIM8_DIER. UIE	TIM8_SR. UIF	更新中断
		TIM8_DIER. CC1IE	TIM8_SR. CC1IF	捕获/比较 1 中断
		TIM8_DIER. CC2IE	TIM8_SR. CC2IF	捕获/比较 2 中断
		TIM8_DIER. CC3IE	TIM8_SR. CC3IF	捕获/比较 3 中断
		TIM8_DIER. CC4IE	TIM8_SR. CC4IF	捕获/比较 4 中断
		TIM8_DIER. CC5IE	TIM8_SR. CC5IF	捕获/比较 5 中断
		TIM8_DIER. COMIE	TIM8_SR. COMIF	COM 中断
		TIM8_DIER. TIE	TIM8_SR. TIF	触发器中断
		TIM8_DIER. BIE	TIM8_SR. BIF	刹车中断
EXTI7-8	28-29	EXTIE2. RIE[1:0]	EXTIF2. RIF[1:0]	外部输入引脚上升沿中断
		EXTIE2. FIE[1:0]	EXTIF2. FIF[1:0]	外部输入引脚下降沿中断
DMA	31	DMAIE. TCIE	DMAIF. TCIF	DMA 传输完成中断
		DMAIE. BCIE	DMAIF. BCIF	DMA 块传输完成中断
		DMAIE. TEIE	DMAIF. TEIF	DMA 传输错误中断

注:

- 对于中断号大于等于 0 的中断，每个中断都有一个对应的中断使能信号，具体配置参见 21.4 CMSIS 函数说明。
- NMI/HardFault/SVCall/PendSV/SysTick 为内核自带，这些中断没有专门的中断使能控制位。

8.2 EXTI 中断说明

EXTI 中断即为外部 INT 口管脚中断，RX32S10 系列共有 9 个 INT 口，即 INT0~INT8，每个 INT 口都可以配置为上升沿触发和下降沿触发。用户在使用 INT 功能前应先将对应 IO 口配置为 INT 复用功能。

8.3 特殊功能寄存器列表

基地址: 0x40011800				
偏移地址	名称	读写方式	复位值	功能描述
0x00	EXTIE	R/W	0x0000	外部中断边沿配置寄存器
0x04	EXTIF	R/W	0x0000	外部中断标志寄存器
0x08	PINFLT	R/W	0x0000	引脚数字滤波使能寄存器
0x10	EXTIE2	R/W	0x0000	外部中断边沿配置寄存器 2
0x14	EXTIF2	R/W	0x0000	外部中断标志寄存器 2
0x18	PINFLT2	R/W	0x0000	引脚数字滤波使能寄存器 2

8.4 特殊功能寄存器说明

8.4.1 外部中断边沿配置寄存器(EXTIE)

EXTIE (外部中断边沿配置寄存器)			基地址: 0x40011800 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:			RIE[6:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:			FIE[6:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIE[6:0]	INT0-6 外部输入引脚上升沿使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来
FIE[6:0]	INT0-6 外部输入引脚下降沿使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来

8.4.2 外部中断标志寄存器(EXTIF)

EXTIF (外部中断标志寄存器)			基地址: 0x40011800 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:			RIF[6:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:			FIF[6:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIF[6:0]	INT 外部输入引脚上升沿中断标志 0: 未产生中断 1: 产生中断
FIF[6:0]	INT 外部输入引脚下降沿中断标志 0: 未产生中断 1: 产生中断

8.4.3 引脚数字滤波使能寄存器(PINFLT)

PINFLT (引脚数字滤波使能寄存器)			基地址: 0x40011800 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:							RXFLT[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:		INTFLT[6:0]						
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RXFLT[1:0]	RX0-1 PIN 输入引脚数字滤波功能使能 0: 禁止 1: 使能
INTFLT[6:0]	INT0-6PIN 输入引脚数字滤波功能使能 0: 禁止 1: 使能

注: 用户在使用 INT 中断或 RX 中断引脚功能的时候, 必须要将 PINFLT 寄存器中的相应的引脚数字滤波功能也打开。引脚数字滤波功能对相应 PIN 脚的其它功能 (GPIO 或其它复用功能) 无效。

8.4.4 外部中断边沿配置寄存器 2(EXTIE2)

EXTIE2 (外部中断边沿配置寄存器 2)			基地址: 0x40011800 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:						X	RIE[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						X	FIE[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIE[1:0]	INT7-8 外部输入引脚上升沿中断使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来
FIE[1:0]	INT7-8 外部输入引脚下降沿中断使能 0: 禁止 1: 使能 注: 只有这些位置 1, 对应的中断标志才能置起来

8.4.5 外部中断标志寄存器 2(EXTIF2)

EXTIF2 (外部中断标志寄存器 2)			基地址: 0x40011800 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:						X	RIF[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						X	FIF[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RIF[1:0]	INT7-8 外部输入引脚上升沿中断标志 0: 未产生中断 1: 产生中断
FIF[1:0]	INT7-8 外部输入引脚下降沿中断标志 0: 未产生中断 1: 产生中断

8.4.6 引脚数字滤波使能寄存器 2(PINFLT2)

PINFLT2 (引脚数字滤波使能寄存器 2)			基地址: 0x40011800 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:								
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:						reserved	INTFLT[8:7]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
INTFLT[8:7]	INT7-8PIN 输入引脚数字滤波功能使能 0: 禁止 1: 使能

注: 用户在使用 INT 中断引脚功能的时候, 必须要将 PINFLT2 寄存器中的相应的引脚数字滤波功能也打开。引脚数字滤波功能对相应 PIN 脚的其它功能 (GPIO 或其它复用功能) 无效。

9 复位模块

9.1 复位优先级

芯片共有 8 种复位方式，可分三种复位优先级。

下表中的复位主要是指芯片中除了 RTC 模块之外的所有模块的复位。

序号	复位源	复位等级	不能复位的寄存器
1	上电复位 (POR)	一级	1, 复位状态寄存器 RSTSR
2	低电压掉电复位 (LBOR)		2, RTC 模块 (包括 RTC-stamp) 的寄存器
3	外部引脚 /RST 复位	二级	1, 复位状态寄存器 RSTSR
4	掉电复位 (BOR)		2, PMU 模块的寄存器: PMUCON, VDETCFG 3, 唤醒标志寄存器 WAKEIF 4, RTC 模块的寄存器
5	看门狗复位 (WatchDog)	三级	1, 复位状态寄存器 RSTSR
6	调试复位和软复位		2, PMU 模块的寄存器: PMUCON, VDETCFG 3, 唤醒标志寄存器 WAKEIF 4, GPIO 模块的寄存器: IOCFG, AFCFG, PTDIR, PTUP, PTDAT, PTOD, HIIPM 5, EXTI 模块寄存器: EXTIE _x , EXTIF _x , PINFLT _x
7	唤醒复位 (WakeUp Reset)		6, CMU 相关寄存器: CLKCTRL0 (HRC_EN 和 PLL_EN 被复位), CLKCTRL1, LRCADJ, HRCADJ 7, RTC 模块的寄存器 9, 低频时钟选择寄存器: LFCLKCFG

注 1: RSTSTA 的 POR 复位标志和 LBOR 复位标志之间可以互相清除

注 2: LRCADJ 会被 Watchdog 和 Debug Reset 复位

注 3: 软复位 (SoftReset) 属于 32-bit CPU Core 内核自带复位，不能复位所有寄存器

9.2 复位说明

任何复位源产生复位后，CPU 的程序指针恢复到 0000H，绝大部分寄存器恢复到缺省值：

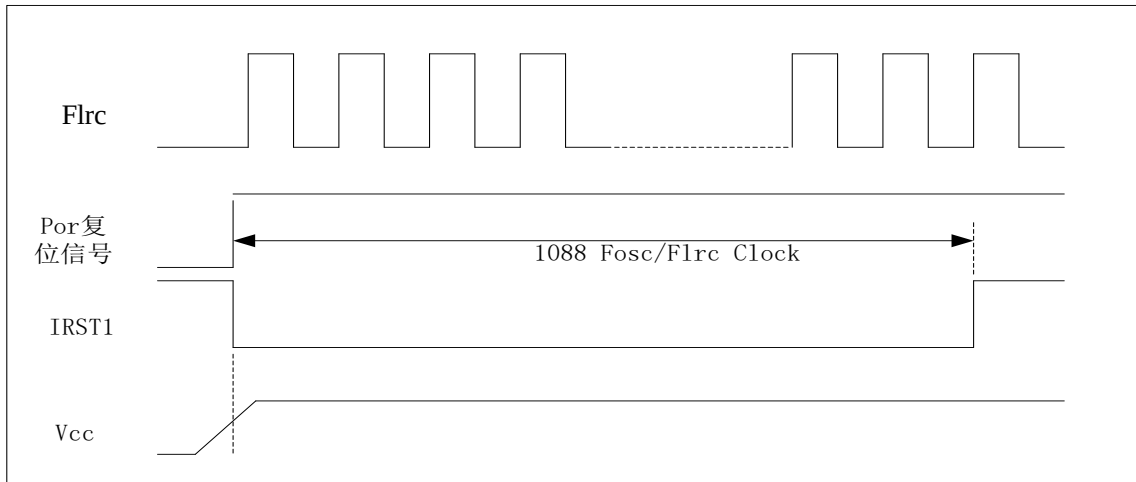
- 1) POR, LBOR 和 BOR 复位时，内部复位信号 IRST 将保持有效，并保持 1024 个 Flrc
- 2) 外部 RST, WDT Reset, SoftReset, Debug Reset, WakeUp Reset 复位时，内部 IRST 信号有效，并保持 64 个 Flrc。
- 3 复位计数时钟强制选择 Flrc。

9.2.1 上电复位

当电源第一次加到芯片上时，上电复位电路检测电源电压 V_{CC} 上升到阈值 0.3V 时，POR 输出高电平，指示发生上电。内部复位信号 $IRST1$ 保持为低电平，1024 个 $Flrc$ 后， $IRST1$ 才会变为高电平。

上电复位 POR 产生时，下面的事件将会发生：

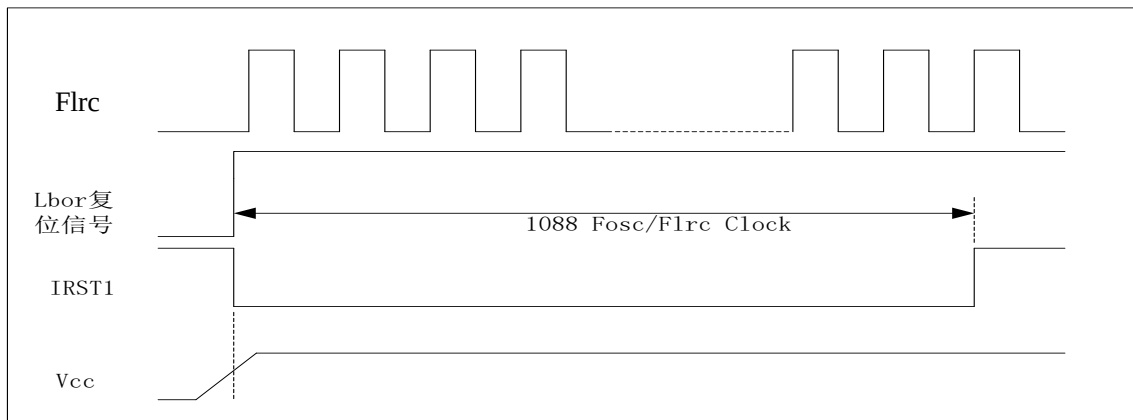
- 产生一个 POR 脉冲
- 内部复位信号 $IRST1$ 有效
- 计数 1024 个 $Flrc$
- 复位状态寄存器 $RSTSTA$ 的上电复位标志位 POR 被设置为 1，其他 $RSTSTA$ 为被清为 0。
- CPU 从地址 0000H 执行程序



上电复位 POR 说明

9.2.2 低电压检测复位

低电压检测复位 (LBOR) 在掉电后重新上电的复位过程与上电复位 (POR) 相同。

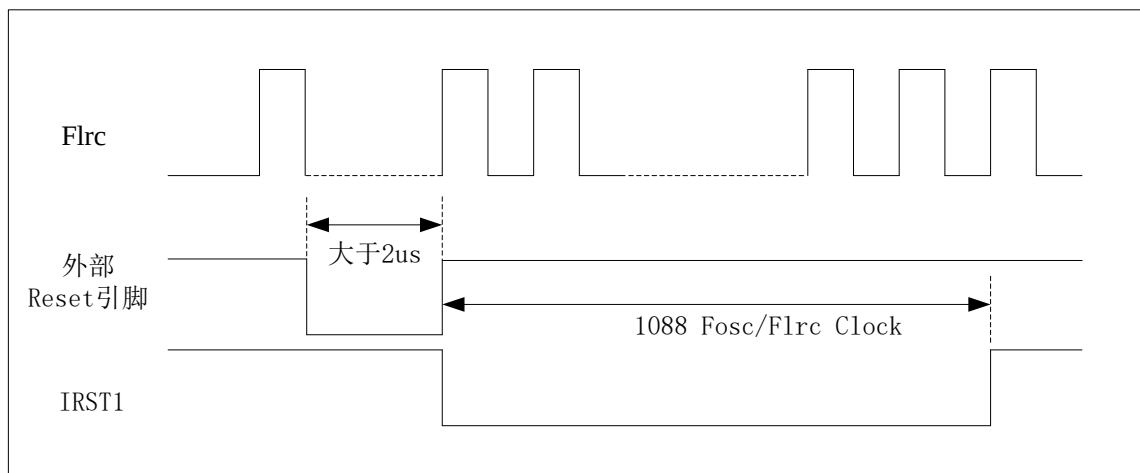


LBOR 复位说明

9.2.3 外部引脚复位

外部复位引脚/ RST 出现比 2 μs 宽的低电平时，内部复位信号 $IRST1$ 有效，复位状态寄存器的复位标志位 RST 被设置为 1；内部复位信号 $IRST1$ 有效脉宽为 1024 个 $Flrc$ 。

如果/ RST 低电平脉宽比 2 μs 窄，系统不发生复位。



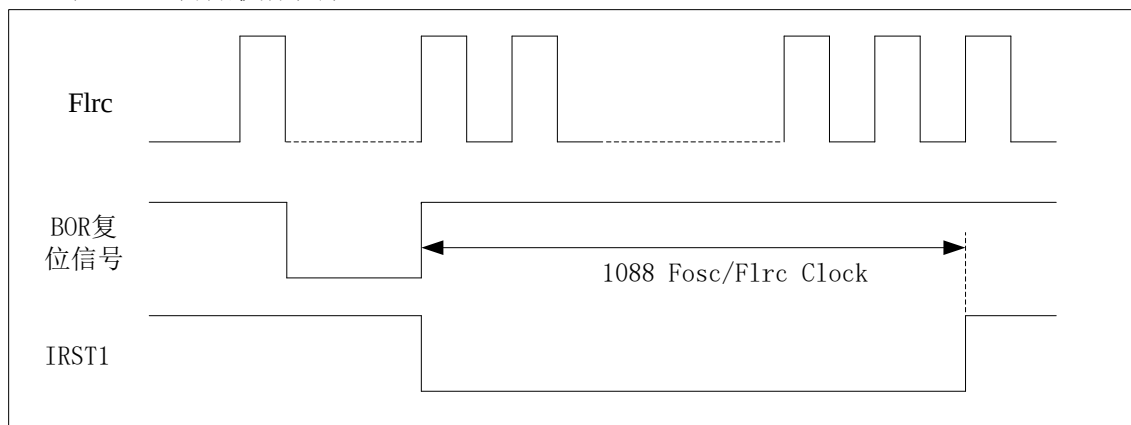
外部引脚复位说明

9.2.4 掉电复位

当掉电检测电路检测到电源电压低于电压 V_{bor} 时，BOR 输出低电平，内部复位信号 IRST1 将变为低电平，复位状态寄存器 RSTSTA 的 BOR 标志位被置为 1。当掉电检测电路检测到电源电压高于电压 V_{bor} 时，BOR 输出高电平，IRST1 在 1024 个 Flrc 时间之后变为高电平。

掉电复位 BOR 产生时，下面的事件将会发生：

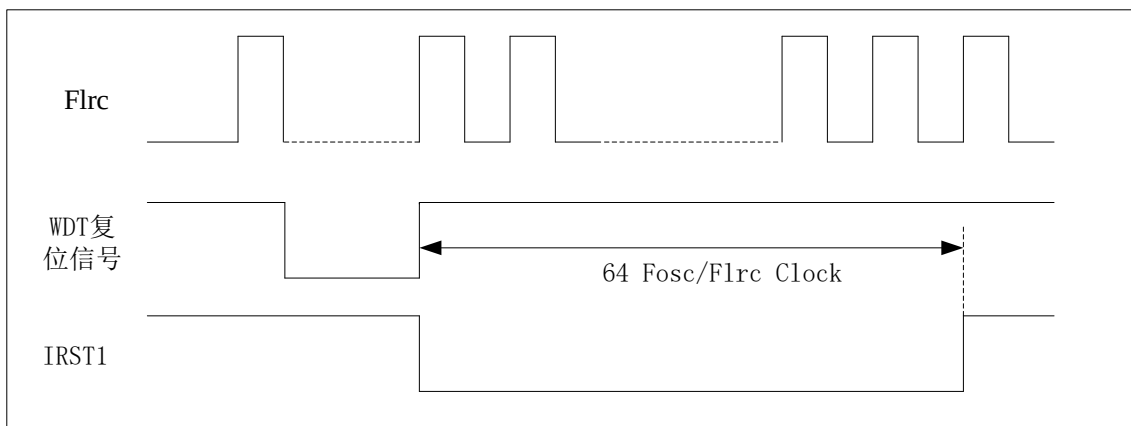
- 产生一个 BOR 脉冲
- 内部复位信号 IRST1 有效
- 计数 1024 个 Flrc
- 复位状态寄存器 RSTSTA 的掉电复位标志位 BOR 被设置为 1，其他 RSTSTA 为被清为 0。
- CPU 从 0000H 开始执行程序



BOR 复位说明

9.2.5 看门狗复位

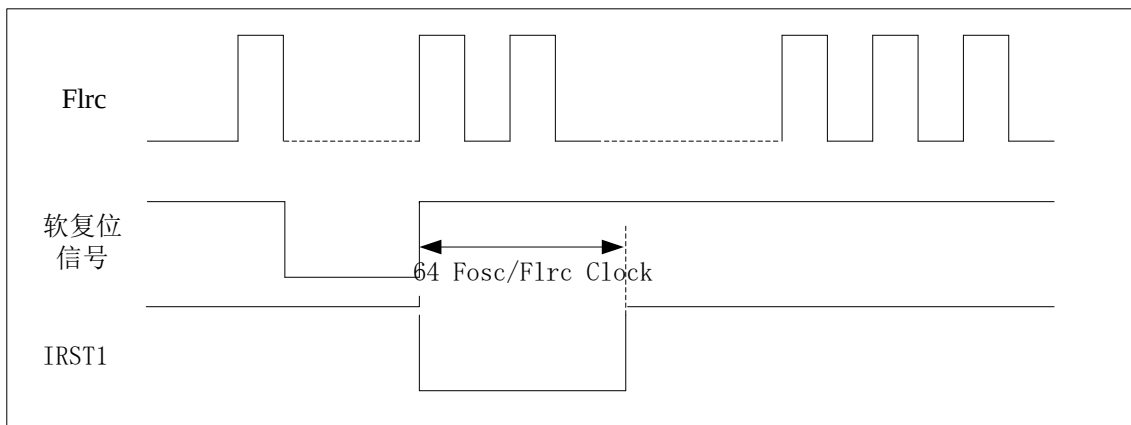
WatchDog Timer 溢出时将会产生导致内部复位 IRST1 有效，复位状态寄存器的 WDT 复位标志位 WDT 被设置为 1。WDT 的复位脉宽为 64 个 Flrc。



WDT 复位

9.2.6 软复位

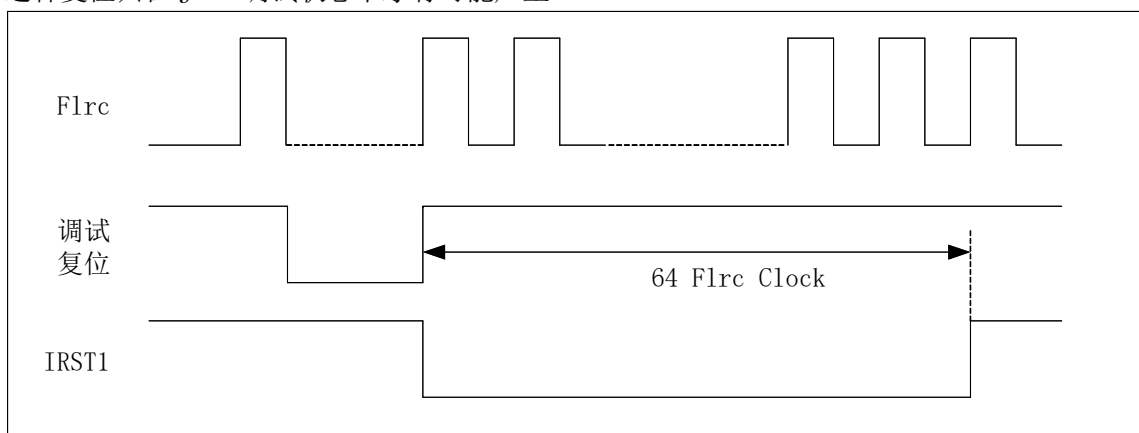
当向系统控制块的应用中断与控制状态寄存器寄存器 AIRCR 的 bit2 写入 1 时，则产生一个软复位。推荐用户使用标准的 CMSIS 函数来操作软复位，具体使用参见 19.4 CMSIS 函数说明。



软复位

9.2.7 调试复位

这种复位只在 JTAG 调试状态下才有可能产生。

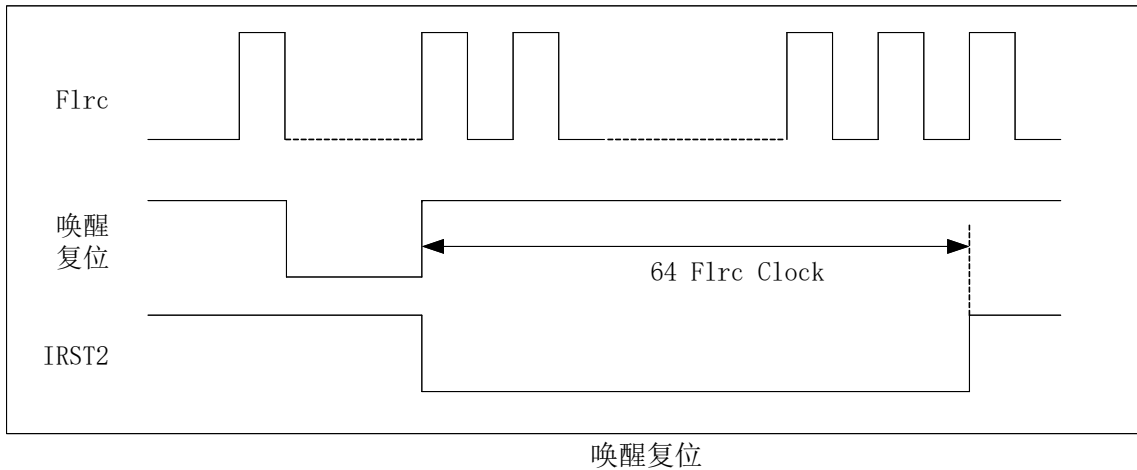


调试复位

9.2.8 唤醒复位

出现唤醒事件时，按照下面顺序执行：

- 内部复位信号 IRST2 有效
- 复位状态寄存器 RSTSTA 的复位标志位 WakeupRST 被设置为 1
- 计数 64 个 Flrc 后，释放内部复位信号 IRST2



9.3 特殊功能寄存器列表

基地值: 0x4000F400 (与 PMU 模块相同)				
偏移地址	名称	读写方式	复位值	功能描述
0x30	RSTSTA	R/W	0x0000	复位标志寄存器

9.4 特殊功能寄存器说明

9.4.1 复位标志寄存器(RSTSTA)

RSTSTA (复位标志寄存器)			基地址: 0x4000F400 偏移地址: 30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Hold_F	DS_Fl	X	X	X	X	X	BORRST
Write:	lag	ag						
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DebugR	SoftR	ExtRST	RESERV	Wakeup	WDTRS	LBORRS	PORRST
Write:	ST	ST		ED	RST	T	T	
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

Hold_flag	Hold 模式下中断唤醒标志位 0: 未发生 Hold 模式下中断唤醒标志位 1: 发生了 Hold 模式下中断唤醒标志位 写 0 清 0
DS_flag	Sleep 模式下中断唤醒标志位 0: 未发生 Sleep 模式下唤醒 1: 发生了 Sleep 模式下唤醒 写 0 清 0
BORRST	BOR 复位标志位 0: 未发生 BOR 复位 1: 发生了 BOR 复位 写 0 清 0
DebugRST	调试复位复位标志位 0: 未发生 Debug Reset 复位 1: 发生了 Debug Reset 复位 写 0 清 0
SoftRST	软复位复位标志位 0: 未发生 Soft Reset 复位 1: 发生了 Soft Reset 复位 写 0 清 0
ExtRST	外部 RST 复位标志位 0: 未发生 RST 复位 1: 发生了 RST 复位 写 0 清 0
RESERVED	该标志位对用户无意义
WakeupRST	唤醒复位复位标志位 0: 未发生 Wakeup Reset 复位 1: 发生了 Wakeup Reset 复位 写 0 清 0
WDTRST	Watch Dog 复位标志位 0: 未发生 WDT 复位 1: 发生了 WDT 复位 写 0 清 0
LBORRST	LBOR 复位标志位 0: 未发生 LBOR 复位 1: 发生了 LBOR 复位 写 0 清 0
PORRST	POR 复位标志位 0: 未发生 POR 复位 1: 发生了 POR 复位 写 0 清 0

注 1: LBOR 复位和 POR 复位发生, 置对应复位标志, 并清除其它复位标志; 除 LBOR 和 POR, 其它复位发生, 仅置对应复位标志, 不会清除其它复位标志。

10 UART 通讯模块

10.1 功能说明

UART 串行通信模块实现与外部设备的异步串行通信。

特点:

- 共2路UART
- 波特率可软件设置，最高波特率115200
- 全双工通信口
- 发送支持1个停止位或2个停止位
- 数据位宽支持7或8位
- 硬件自动完成奇偶校验，数据接收完成的同时判断并提示奇偶校验错误，给出标志。
- 接收/发送中断使能分别独立

串口提供灵活的全双工异步通信的接收器/发送器，通过寄存器 UARTCON 配置串口工作在不同的工作模式，列举如下：

- 方式 1：通过 TXD 发送或通过 RXD 接收 7 个数据位，无奇偶校验，波特率可变。
- 方式 2：通过 TXD 发送或通过 RXD 接收 7 个数据位，和 1 个奇偶校验位，波特率可变。
- 方式 3：通过 TXD 发送或通过 RXD 接收 8 个数据位，无奇偶校验，波特率可变。
- 方式 4：通过 TXD 发送或通过 RXD 接收 8 个数据位，和 1 个奇偶校验位，波特率可变。

10.2 波特率计算

在串口波特率由波特率生成器的值确定：

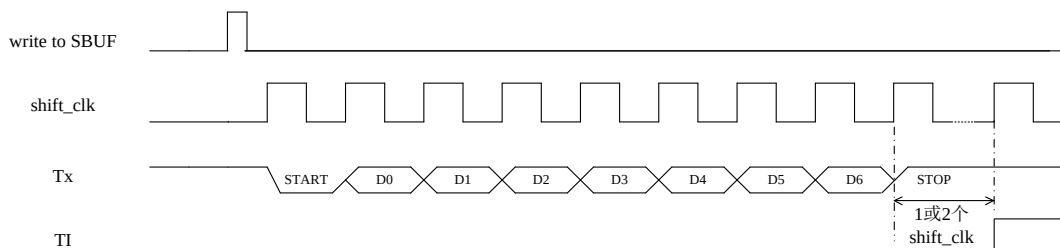
$$\text{波特率} = \frac{F_{\text{sys}}}{2 \times (SREL + 1)}$$

其中 SREL 是 16 位无符号数；Fsys 是系统时钟，波特率最高值为 115200bps。

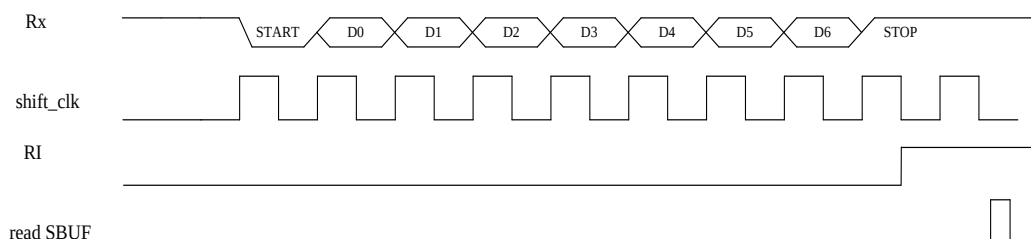
10.3 串口通讯模式说明

10.3.1 方式 1

方式 1 是一种标准的异步通信方式，每帧包含 9 或 10 位数据信息：1 位起始位（0），7 位数据位（低位在前），1 或 2 位停止位（1）。在这种方式中，TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



图示：方式 1 时串行发送数据时序

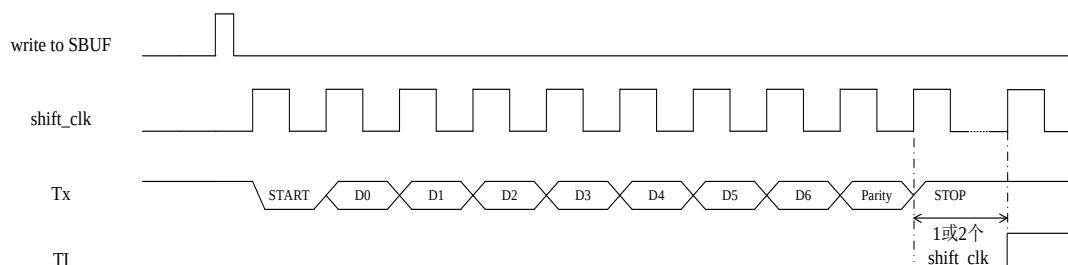


图示：方式 1 时串行接收数据时序

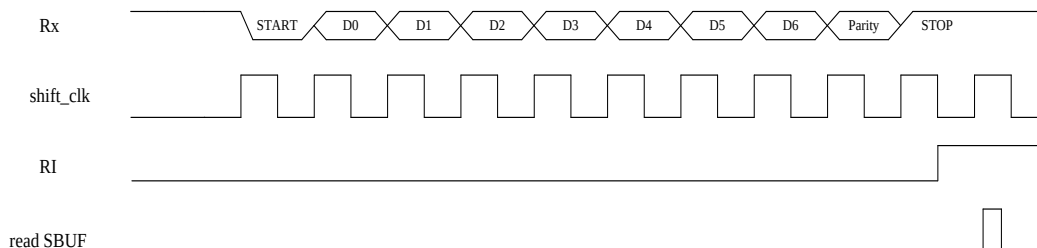
在方式 1 中，发送状态时，当一帧中最后一个数据发送完时，发送中断标志 TI 置“1”；接收状态时，接收完最后一个数据位时，接收中断标志 RI 置 1。

10.3.2 方式 2

方式 2 是每帧包含 10 或 11 位数据信息：1 位起始位（0），7 位数据位（低位在前），1 位奇偶校验数据位，1 或 2 位停止位（1）。TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



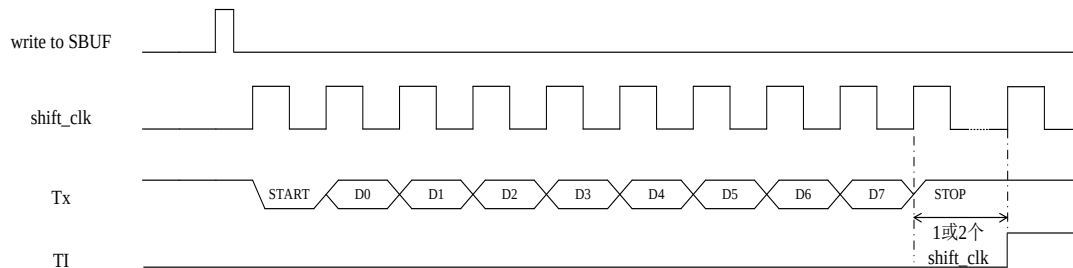
图示：方式 2 时串行发送数据时序



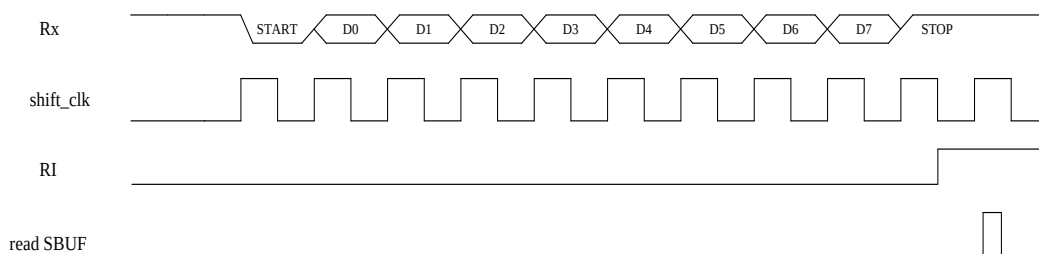
图示：方式 2 时串行接收数据时序

10.3.3 方式 3

方式 3 是一种标准的异步通信方式，每帧包含 10 或 11 位数据信息：1 位起始位（0），8 位数据位（低位在前），1 或 2 位停止位（1）。在这种方式中，TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



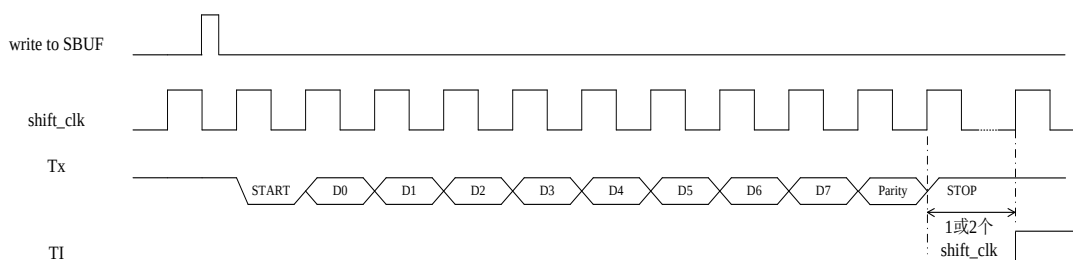
图示：方式 3 时串行发送数据时序



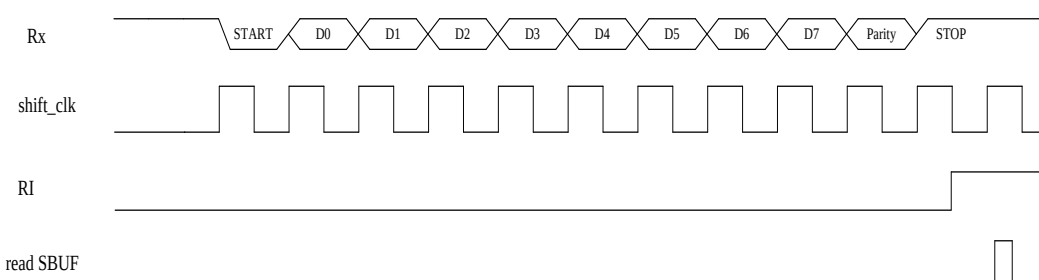
图示：方式 3 时串行接收数据时序

10.3.4 方式 4

方式 4 是使用第 9 位数据的通信方式，每帧包含 11 或 12 位数据信息：1 位起始位（0），8 位数据位（低位在前），1 个奇偶校验或自定义数据位，1 或 2 位停止位（1）。TXD 引脚为数据发送端，RXD 引脚为数据接收端，其波形如下图所示：



图示：方式 4 时串行发送数据时序



图示：方式 4 时串行接收数据时序

10.4 特殊功能寄存器列表

UART模块寄存器基地址： 0x40005000 (UART0端口)； 0x40006000 (UART1端口)；				
偏移地址	名称	读写方式	复位值	功能描述
0x00	MODESEL	R/W	0x0000	串口功能选择寄存器
0x04	UARTCON	R/W	0x0000	UART 功能配置寄存器
0x0C	SREL	R/W	0x0000	串口波特率发生寄存器
0x10	SBUF	R/W	0x0000	串口数据缓冲寄存器
0x14	UARTSTA	R/W	0x0000	UART 状态寄存器

10.5 特殊功能寄存器说明

10.5.1 串口功能选择寄存器(MODESEL)

MODESEL (串口功能选择寄存器)			基地址： 0x40005000； 0x40006000 偏移地址： 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	Mode
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Mode	串口功能选择控制位： 0: UART 功能 1: 此位设定永远为 0

10.5.2 UART 功能配置寄存器(UARTCON)

UARTCON (UART 功能配置寄存器)			基地址： 0x40005000； 0x40006000 偏移地址： 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	PRDIE	X	X	X	X	UNEGAUTO	UNEG	STOPSEL
Write:								

Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LENSEL	PARITYSEL[1:0]		PARITYEN	RXIE	TXIE	RXEN	TXEN
Write:								
Reset:	0	0	0	0	0	0	0	0
位	功能描述							
PRDIE	串口上溢中断使能控制位： 0：上溢中断禁止 1：上溢中断使能							
UNEGAUTO	UART 通讯中的正反逻辑判断： 该逻辑自动判断只针对数据接收， 0：由用户软件控制 UART 接收/发送通讯是正逻辑还是负逻辑，通过 UNEG 控制位 1：由硬件自动配置 UART 接收通讯是正逻辑还是负逻辑，UNEG 控制位无效							
UNEG	当 UNEGAUTO=0 时，用户手动配置 UART 通讯中接收和发送是正逻辑或者是负逻辑 0：正逻辑（默认） 1：负逻辑 当 UNEGAUTO=1 时，该寄存器位只能控制 UART 通讯中发送是正逻辑或者负逻辑							
	UNEGAUTO		UNEG		UART 发送和接收的逻辑状态			
	0		0		UART 发送，接收都是正逻辑			
	0		1		UART 发送，接收都是负逻辑			
	1		0		UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是正逻辑			
	1		1		UART 接收根据外部 RX 默认状态高低自动判断逻辑 UART 发送是负逻辑			
	UART 通讯中的正逻辑或者是负逻辑 0：正逻辑（默认） 1：负逻辑							
STOPSEL	UART 通讯停止位长度选择位 1：2bit 0：1bit							
LENSEL	UART 通讯数据长度选择位 1：7bit 0：8bit							
PARITYSEL	UART 奇偶校验选择位 11：固定为 1 00：固定为 0 01：奇校验 10：偶校验							
PARITYEN	UART 奇偶校验使能位 1：使能 0：禁止							
RXIE	UART 接收中断使能位 1：使能 0：禁止							
TXIE	UART 发送中断使能位 1：使能 0：禁止							
RXEN	UART 接收使能位 1：使能							

	0: 禁止
TXEN	UART 发送使能位 1: 使能 0: 禁止

10.5.3 串口波特率发生寄存器(SREL)

SREL (串口波特率发生寄存器)			基地址: 0x40005000; 0x40006000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	SREL[15:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SREL[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

串口波特率发生寄存器，是一个 16 位的波特率分频系数，其值可为 0~65535 之间的任一整数，最高波特率为 115200。波特率计算公式：

$$\text{波特率} = \frac{F_{\text{sys}}}{2 \times (\text{SREL} + 1)}$$

10.5.4 串口数据缓冲寄存器(SBUF)

SBUF (串口数据缓冲寄存器)			基地址: 0x40005000; 0x40006000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SBUF[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

低 8 位有效，对寄存器 SBUF 写操作，则串口将开始向外传输发送缓存数据；对寄存器 SBUF 读操作，则串口将从串行接收缓存中读取数据。

10.5.5 UART 状态寄存器(UARTSTA)

UARTSTA (UART 状态寄存器)			基地址: 0x40005000; 0x40006000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X

Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	PRDIF	PARITY	RXIF	TXIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
PRDIF	接收上溢中断标志 1: 产生上溢, 也就是外部输入到 SBUF 中的数据, 还未被及时读出, 然后又有新的数据传输过来把老的数据覆盖了 0: 未产生上溢; 写 0 清零, 写 1 无效。
PARITY	接收时奇偶校验的状态 1: 错误 0: 正确 写 0 清零, 写 1 无效。
RXIF	接收中断标志 1: 接收数据完成, 可从寄存器 SBUF 中读出 0: 接收数据还未完成 写 0 清零, 写 1 无效。
TXIF	发送中断标志 1: 发送完成 0: 发送未完成 写 0 清零, 写 1 无效。

11 WDT 模块

11.1 概述

RX32S10 系列有一个硬看门狗，其由内部低频 RC 时钟驱动，当计时器计满预定时间则发出溢出脉冲，产生 WDT 复位信号。在溢出脉冲发生前将 Watchdog Timer 清零，则不会发出 WDT 复位。特点如下：

- 采用硬件狗设计
- SLEEP或者HOLD模式下WDT开启/关闭可选
- 调试模式下可关闭，且随着CPU停止而停止，方便调试使用

11.2 工作模式

不同工作模式下，WDT 的控制状态也有所区别，具体参见下表

工作模式	SOFTWDT_EN	HWDT_EN	WDT 模块
测试模式	X	X	X
调试模式	0	X	关闭
	1	X	开启，且可以随着 CPU 的停止而停止
正常模式	X	X	开启
Sleep/Hold 模式	X	0	关闭
	X	1	开启

注：1) X 表示无意义

2) 工作模式的定义详见 6.1 工作模式

3) SOFTWDT_EN 为 CLKCTRL1 寄存器的 bit14

4) HWDT_EN 由 Flash 0xFC2 地址内 WDT_EN[3:0] 决定，具体参见 2.4Flash 控制功能

5) WDT 由 LRC 驱动，如果关闭了 LRC，则 WDT 也不会工作。

11.3 特殊功能寄存器列表

WDT 模块寄存器基地值：0x40010000				
偏移地址	名称	读写方式	复位值	功能描述
0x04	WDTCLR	R/W	0x0040	看门狗喂狗与时间配置寄存器
0x08	WDTCNT	R	0x0000	看门狗计数寄存器（只读）

11.4 特殊功能寄存器说明

11.4.1 WDT 喂狗与时间配置寄存器(WDTCLR)

WDTCLR (WDT 喂狗与时间配置寄存器)			基地址: 0x40010000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	0	0	0	0	0	0	0	0
Write:	CLR[7:0]							
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SET[7:0]							
Write:								
Reset:	0	1	0	0	0	0	0	0

位	功能描述
CLR[7:0]	WDT 喂狗控制位: 当该 8bit 写入 0xAA, 则清狗, 清除 WDT 内部计数器 WDTCNT, 写入其他值无效 该高 8bit 只能写入, 不能读取, 读出值永远为 0
SET[7:0]	WDT 溢出时间设置: $\text{WDT 溢出时间} = 64\text{ms} * (1 + \text{SET}[7:0])$ SET[7:0] 为 8 位无符号数, 由上面公式可以得出, 最短的定时时间为 64ms, 最长为 16384ms。默认为 4160ms。

11.4.2 WDT 计数寄存器(WDTCNT)

WDTCNT (WDT 计数寄存器)			基地址: 0x40010000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CNT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CNT[15:0]	WDT 计数寄存器: 指示当前 WDT 内部的计数值

12 Timer 通用定时器模块

12.1 定时器单元概述

RX32S10 系列共有 4 路定时器，定时器 0、1、2、3 具有所有功能，使用时钟源为 F_{sys}

定时器 0、1、2、3 的时钟源为系统时钟 (F_{sys})，可根据 $SYSCLK_SEL[2:0]$ 选择为：内部低频 RC 时钟 (F_{lrc})，内部高频 RC 时钟 (F_{hrc})，外部高频时钟 (F_{hse}) 和 PLL 时钟 (F_{pll})。

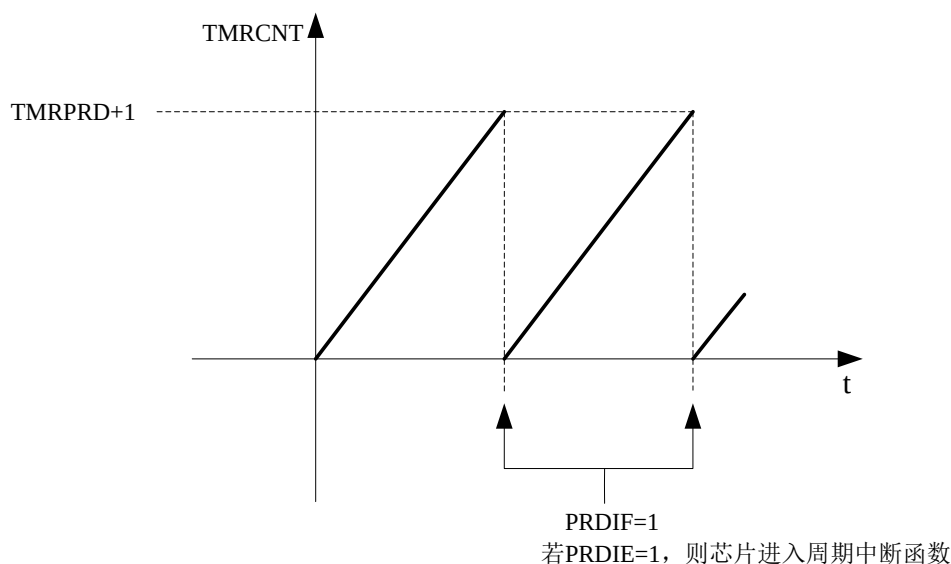
所有定时器单元相关寄存器的配置，都需要在 $CLKCTRL1$ 寄存器中使能相应的定时器模块后才能配置。

定时器主要包括以下功能：

1. 周期定时功能
2. PWM 功能
3. 捕获功能
4. 事件计数功能

12.2 周期定时功能

通用定时器包括一个 16 位计数器和周期寄存器。计数器的时钟由系统时钟 (F_{sys}) 通过定时器单元内的预分频器 ($TMRDIV$) 分频得到，当使能计数器 ($CNTEN=1$) 后，定时器的计数器从 0 开始计数，当计数寄存器 ($TMRCNT$) 的值等于设定的周期寄存器 ($TMRPRD+1$) 时会置位周期定时中断标志 ($PRDIF=1$)，如果使能周期定时中断 ($PRDIE=1$)，则会触发定时器周期中断，进入相应的周期中断服务程序。



当周期定时中断标志置位后 ($PRDIF=1$)， $TMRCNT$ 的值自动清 0，然后重新开始计数。

$TMRPRD$ 如被修改，在完成本次定时之后下一次生效。

功能主要相关寄存器： $TMRCON$ ， $TMRCNT$ ， $TMRPRD$ 。

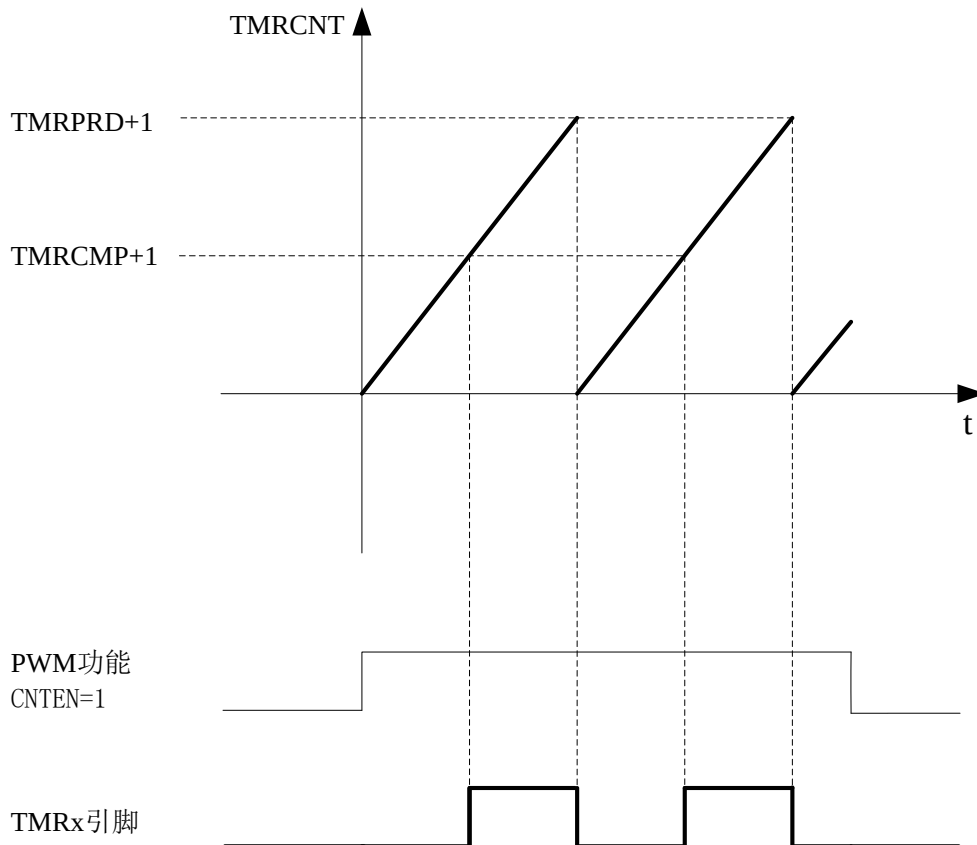
12.3 PWM 功能

PWM 功能可通过寄存器 TMRCON.MODE[1:0] 配置，同时需将对应的 GPIO 配置为 TMR 功能，配置成功后，相应的 TMR 引脚会输出 PWM 波形。PWM 的周期和占空比可通过寄存器 TMRPRD、TMRCMP 进行配置。

PWM 计数方式分为向上计数、向下计数和中央计数方式。

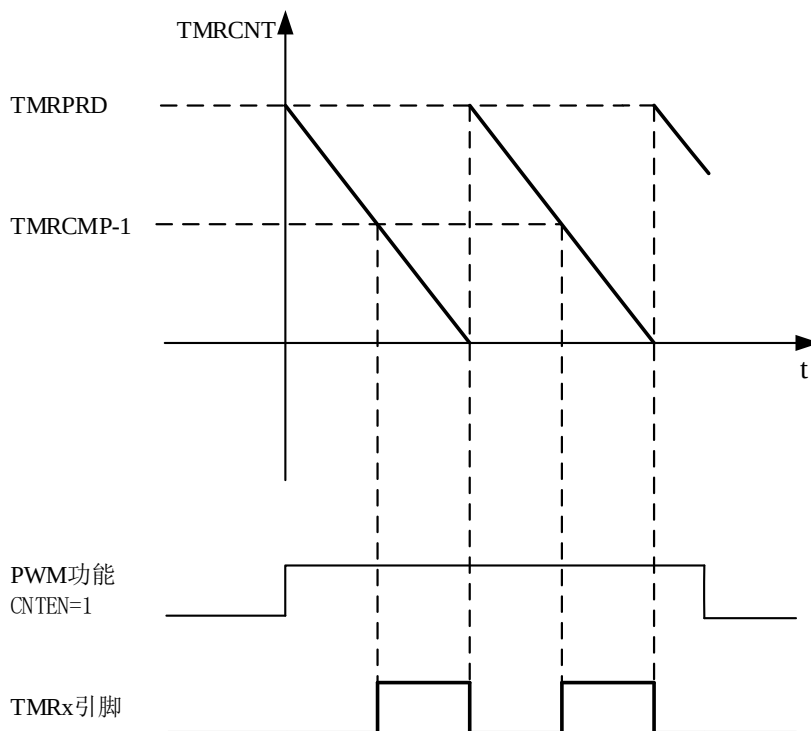
1. 向上计数方式

当使能计数器 (CNTEN=1) 之后，计数器开始从 0 计数，当计数寄存器 (TMRCNT) 的值等于设定的比较寄存器 (TMRCMP+1) 时，PWM 输出管脚发生电平翻转，同时置位比较中断标志 (CMPIF=1)。计数器继续向上计数，当计数寄存器 (TMRCNT) 的值等于设定的周期寄存器 (TMRPRD+1) 时，PWM 输出管脚再次发生电平翻转，同时置位周期定时中断标志 (PRDIF=1)。PWM 输出波形如下图所示：



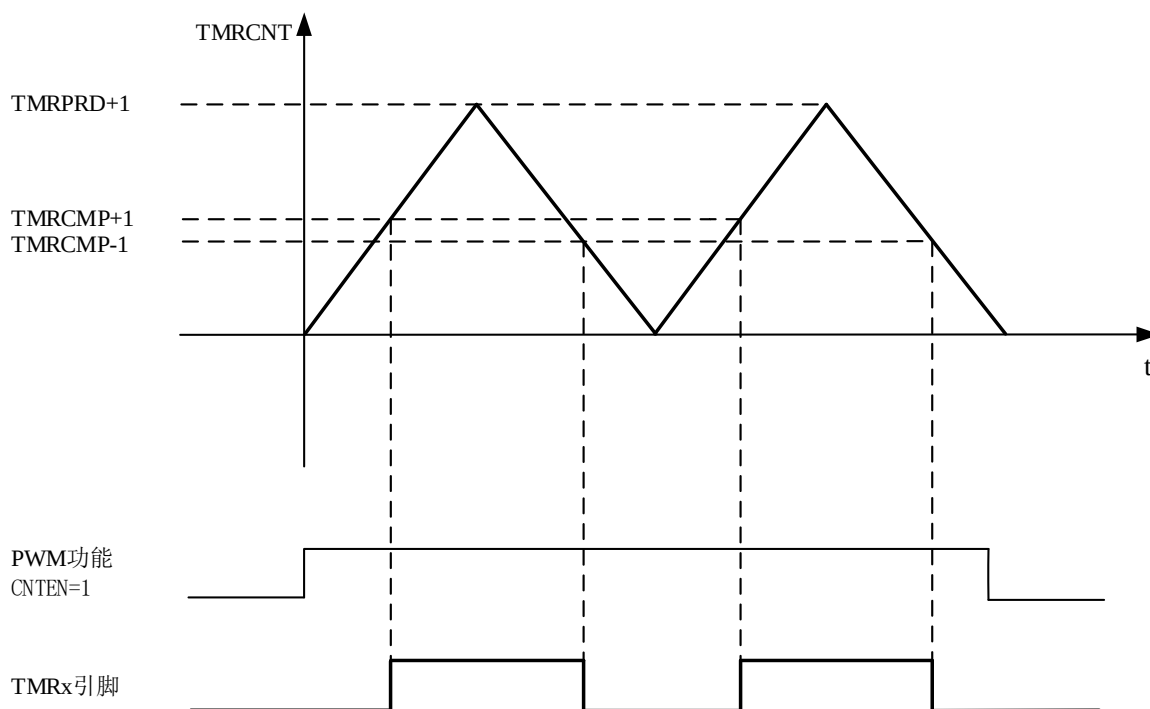
2. 向下计数方式

当使能计数器 (CNTEN=1) 之后，计数器开始从 (TMRPRD) 向下计数，当计数寄存器 (TMRCNT) 的值等于设定的比较寄存器 (TMRCMP-1) 时，PWM 输出管脚发生电平翻转，同时置位比较中断标志 (CMPIF=1)。计数器继续向下计数，当计数寄存器 (TMRCNT) 的值等于 0 时，PWM 输出管脚再次发生电平翻转，同时置位周期定时中断标志 (PRDIF=1)。PWM 输出波形如下图所示：



3. 中央计数方式

当计数寄存器 (TMRcnt) 从 0 开始向上计数, 其值等于比较寄存器 (TMRCMP+1) 时, PWM 输出管脚发生电平翻转。计数器继续向上计数, 当计数寄存器 (TMRcnt) 的值等于设定的周期寄存器 (TMRPRD+1) 时, 置位周期定时中断标志 (PRDIF=1), 但 PWM 输出管脚不发生电平翻转, 计数器从周期寄存器 (TMRPRD+1) 的值开始向下继续计数, 当计数寄存器 (TMRcnt) 的值再次等于设定的比较寄存器 (TMRCMP-1) 时, PWM 输出管脚发生电平翻转。PWM 输出波形如下图所示:



假如系统时钟选择为 PLL 产生的 21 MHz 时钟, Timer 预分频寄存器 (TMRDIV) 默认值为 0 (不分频), 定时器 0 选择 PWM 模式, 向上计数方式, 初始电平为高电平。若要想 PWM 占空比 (高电平时间与周

期之比)为30%, 则 $(\text{TMRCMP}+1)/(\text{TMRPRD}+1)=30\%$, 根据需要的周期值来确定 TMRPRD 寄存器的值, TMRPRD 所能配置的最大周期定时时间为:

$$(\text{TMRPRD}+1)/(21\text{ M}/(\text{TMRDIV}+1))=0\text{xFFFF}/(21\text{ M}/(0+1))\approx 2.9761\text{ms}$$

TMRCMP, TMRPRD 如被修改, 均是在完成本次计数之后下一次生效。

功能主要相关寄存器: TMRCON, TMRCNT, TMRCMP, TMRPRD。

12.4 捕获功能

在输入捕获模式下, 假如设定上升沿检测, 当 TMR0~TMR3 管脚检测到上升沿, 计数寄存器 (TMRCNT) 的当前值被锁定到捕获数据寄存器 (TMRCAP) 中。当捕获事件发生时, 置位捕获中断标志 (CAPIF=1), 如果使能捕获中断 (CAPIE=1), 将产生捕获中断, 进入相应的捕获中断服务程序。

捕获过程中, 如果没有检测到沿, 当计数寄存器 (TMRCNT) 的值和设定的周期寄存器 (TMRPRD+1) 相等时会置位周期定时中断标志 (PRDIF=1), 同时周期寄存器 (TMRCNT) 从 0 开始计数。如使能了周期定时中断 (PRDIE=1), 会进入相应的周期中断服务程序。

捕获检测可以选择上升沿, 下降沿方式。

功能主要相关寄存器: TMRCON, TMRCAP, TMRCNT, TMRPRD。

12.5 事件计数功能

在输入事件计数模式下, 假如设定下降沿检测, 当 Timer0~Timer3 管脚上检测到一次下降沿, 计数寄存器 (TMRCNT) 的当前值加 1。当计数寄存器 (TMRCNT) 的值等于设定的比较寄存器 (TMRCMP+1) 时, 会置位事件计数中断标志 (ACIF=1), 同时计数寄存器 (TMRCNT) 自动清 0, 重新开始计数。如果使能了事件计数中断 (ACIE=1), 将产生事件计数中断, 进入相应的事件计数中断服务程序。

事件计数过程中, 在 $\text{TMRCMP} > \text{TMRPRD}$ 情况下, 当计数寄存器 (TMRCNT) 的值等于设定的周期寄存器 (TMRPRD+1) 时会置位周期定时中断标志 (PRDIF=1), 计数寄存器 (TMRCNT) 继续计数直到等于设定的比较寄存器 (TMRCMP+1), 如果使能了周期定时中断 (PRDIE=1) 会进入周期中断服务程序。

单次最大计数个数为 0xFFFF, 可以配合周期中断实现任意次数的组合。事件计数检测可以选择上升沿, 下降沿方式, 该功能可用于 MCU 发脉冲。

比较寄存器 (TMRCMP) 如被修改, 如果修改后的值小于当前计数寄存器 (TMRCNT) 的值, 则立刻触发事件计数中断, 同时计数寄存器 (TMRCNT) 清 0, 重新开始计数; 如果修改后的值大于当前计数寄存器 (TMRCNT) 的值, 则继续本次计数。

在事件计数模式下, 若需要修改 TMRCMP、TMRPRD, 应先关闭计数器 (CNTEN=0), 修改完后再开启计数器 (CNTEN=1)。

功能主要相关寄存器: TMRCON, TMRCNT, TMRCMP, TMRPRD。

12.6 中断功能

12.6.1 周期定时中断

当计数寄存器 (TMRCNT) 的值等于设定的周期寄存器 (TMRPRD+1) 时, 如果使能周期定时中断 (PRDIE=1), 则发生周期定时中断。此中断在任何功能模式下都可以产生。

12.6.2 捕获中断

当检测到外部输入信号相应沿时，如使能了捕获中断（CMP1E=1），则发生捕获中断。计数寄存器（TMRcnt）的值被锁定到捕获数据寄存器（TMRcap）中。

12.6.3 比较中断

当计数寄存器（TMRcnt）的值等于设定的比较寄存器（TMRcmp+1）时，如使能了 PWM 比较中断（CMP1E=1），则发生比较中断。

12.6.4 事件计数中断

当检测到设定次数的外部输入信号相应沿时，如使能了事件计数中断（AC1E=1），则发生事件计数中断。

12.7 特殊功能寄存器列表

TMR 模块寄存器基地址: 0x40001000 (TMR0); 0x40002000 (TMR1); 0x40003000 (TMR2); 0x40004000 (TMR3);				
偏移地址	名称	读写方式	复位值	功能描述
0x00	TMRCON	R/W	0x0000	定时器控制寄存器
0x04	TMRDIV	R/W	0x0000	预分频寄存器
0x08	TMRPRD	R/W	0x0000	周期寄存器
0x0C	TMRCAP	R/W	0x0000	捕获数据寄存器
0x10	TMRCNT	R/W	0x0000	计数寄存器
0x14	TMRCMP	R/W	0x0000	比较寄存器
0x18	TMRIE	R/W	0x0000	定时器中断使能寄存器
0x1C	TMRIF	R/W	0x0000	定时器中断标志寄存器

12.8 特殊功能寄存器说明

12.8.1 定时器控制寄存器(TMRCON)

TMRCON (定时器控制寄存器)			基地址: 0x40001000--0x40004000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	RESERVED	PWMHL	PWMC[1:0]		CCMODE	MODE[1:0]		CNTEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
RESERVED	内部保留位, 应用中固定为 0
PWMHL	PWM 初始电平选择: (每次写入极性与原有极性比较, 不同则修改, 相同不处理) 0: 高电平 1: 低电平
PWMC[1:0]	PWM 工作模式选择: (PWM 计数方式) 00: 向上计数 01: 向下计数 1x: 中央对齐
CCMODE	捕获/事件计数电平沿选择 (当定时器配置为捕获/事件计数功能): 0: 上升沿 1: 下降沿

MODE[1:0]	Timer 功能选择: 00: 事件计数功能 01: PWM 功能 10: 捕获功能 11: 周期定时功能 需将 GPIO 配置为 TMRx 功能
CNTEN	计数器使能: 0: 关闭 1: 使能

18.8.2 预分频寄存器(TMRDIV)

TMRDIV (预分频寄存器)			基地址: 0x40001000--0x40004000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRDIV[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRDIV[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRDIV[15:0]	预分频的范围在 0-65535 之间 经预分频器后的频率等于输入频率的 1/ (TMRDIV[15:0] +1)

18.8.3 周期寄存器(TMRPRD)

TMRPRD (周期寄存器)			基地址: 0x40001000--0x40004000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRPRD[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRPRD[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRPRD[15:0]	该寄存器是一个 16 位的周期寄存器 计数的周期寄存器和 PWM 的周期寄存器都是该寄存器 在使用任何模式功能之前, 需要设置周期寄存器。 $\text{Period} = \frac{(TMRPRD[15:0] + 1)}{F/(TMRDIV[15:0] + 1)}$

18.8.4 捕获数据寄存器(TMRCAP)

TMRCAP (捕获数据寄存器)			基地址: 0x40001000—0x40004000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCAP[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCAP[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRCAP[15:0]	当发生捕获事件时, 当前计数寄存器 (TMRCNT) 的值被存到该寄存器里

18.8.5 计数寄存器(TMRCNT)

TMRCNT (计数寄存器)			基地址: 0x40001000—0x40004000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TMRCNT[15:0]	计数器当前的计数值

18.8.6 比较寄存器(TMRCMP)

TMRCMP (比较寄存器)			基地址: 0x40001000—0x40004000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	TMRCMP[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TMRCMP[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
---	------

TMRCMP[15:0]	比较寄存器有两个功能： (1)Timer 做 PWM 功能的时候，当计数器达到（TMRCMP+1）的设定值时，PWM 输出翻转，同时置位比较中断标志（CMPIF=1），如使能了 PWM 比较中断（CMPIE=1），则发生比较中断。 (2)Timer 做事件计数功能的时候，当计数寄存器（TMCNT）的值等于设定的比较寄存器（TMRCMP+1）时，会置位事件计数标志（ACIF=1），同时计数寄存器（TMCNT）会从 0 开始重新计数，如果使能了事件计数中断（ACIE=1），则芯片会产生事件计数中断。
--------------	---

18.8.7 定时器中断使能寄存器(TMRIE)

TMRIE (定时器中断使能寄存器)			基地址： 0x40001000—0x40004000 偏移地址： 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	ACIE	CMPIE	CAPIE	PRDIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ACIE	事件计数中断使能 0: 关闭 1: 使能
CMPIE	比较中断使能 0: 关闭 1: 使能
CAPIE	捕获中断使能 0: 关闭 1: 使能
PRDIE	周期定时中断使能 0: 关闭 1: 使能

18.8.8 定时器中断标志寄存器(TMRIF)

TMRIF (定时器中断标志寄存器)			基地址： 0x40001000—0x40004000 偏移地址： 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	ACIF	CMPIF	CAPIF	PRDIF
Write:								



Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
ACIF	事件计数中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
CMPIF	比较中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
CAPIF	捕获中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)
PRDIF	周期定时中断标志 0: 未产生中断 1: 产生中断 (写 0 清 0)

13 TIM8 高级定时器模块

13.1 TIM8 简介

高级控制定时器(TIM8)由一个 16 位的自动装载计数器组成,它由一个可编程的预分频器 驱动。

它适合多种用途,包含测量输入信号的脉冲宽度(输入捕获),或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。

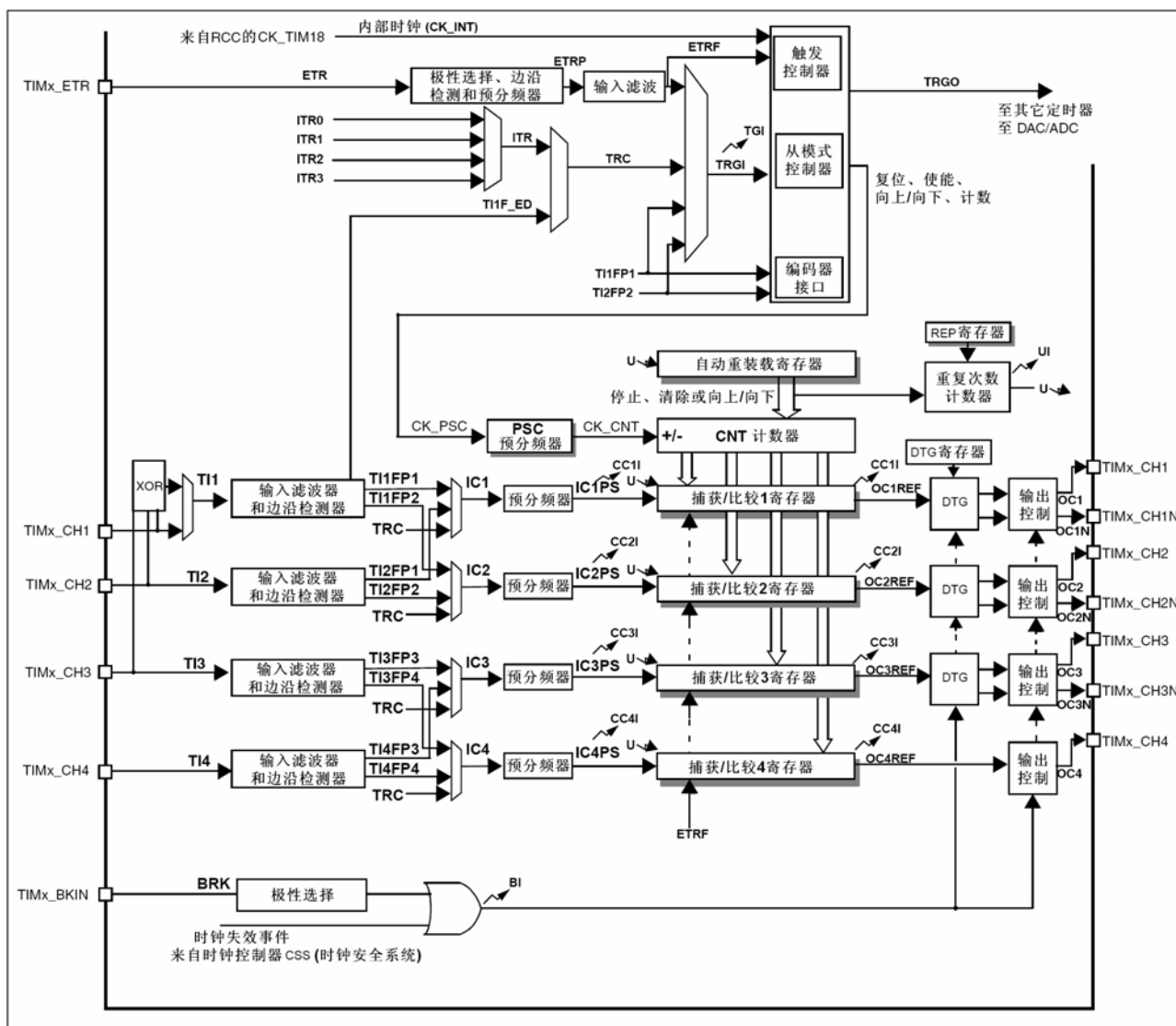
使用定时器预分频器和 RCC 时钟控制预分频器,可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

13.2 TIM8 主要特性

TIM8 定时器的功能包括:

- 16 位向上、向下、向上/下自动装载计数器
 - 16 位可编程(可以实时修改)预分频器,计数器时钟频率的分频系数为 1~65535 之间的任意 数值
- 多达 4 个独立通道:
 - 输入捕获
 - 输出比较
 - PWM 生成(边缘或中间对齐模式)
 - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA:
 - 更新:计数器向上溢出/向下溢出,计数器初始化(通过软件或者内部/外部触发)
 - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
 - 刹车信号输入
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

图 50 高级控制定时器框图



注  根据控制位的设定，在U(更新)事件时传送预加载寄存器的内容至工作寄存器

 事件

 中断和DMA 输出

13.3 TIM8 功能描述

13.3.1 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器 (TIMx_CNT)

- 预分频器寄存器 (TIMx_PSC)
- 自动装载寄存器 (TIMx_ARR)
- 重复次数寄存器 (TIMx_RCR)

自动装载寄存器是预先装载的，写或读自动重载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (向下计数时的下溢条件) 并当 TIMx_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIMx_CR1 寄存器中的计数器使能位 (CEN) 时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

注意，在设置了 TIMx_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

13.3.1.1 预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个 (在 TIMx_PSC 寄存器中的) 16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下次更新事件到来时被采用。

图 51 和图 52 给出了在预分频器运行时，更改计数器参数的例子。图 51 当预分频器的参数从 1 变到 2 时，计数器的时序图

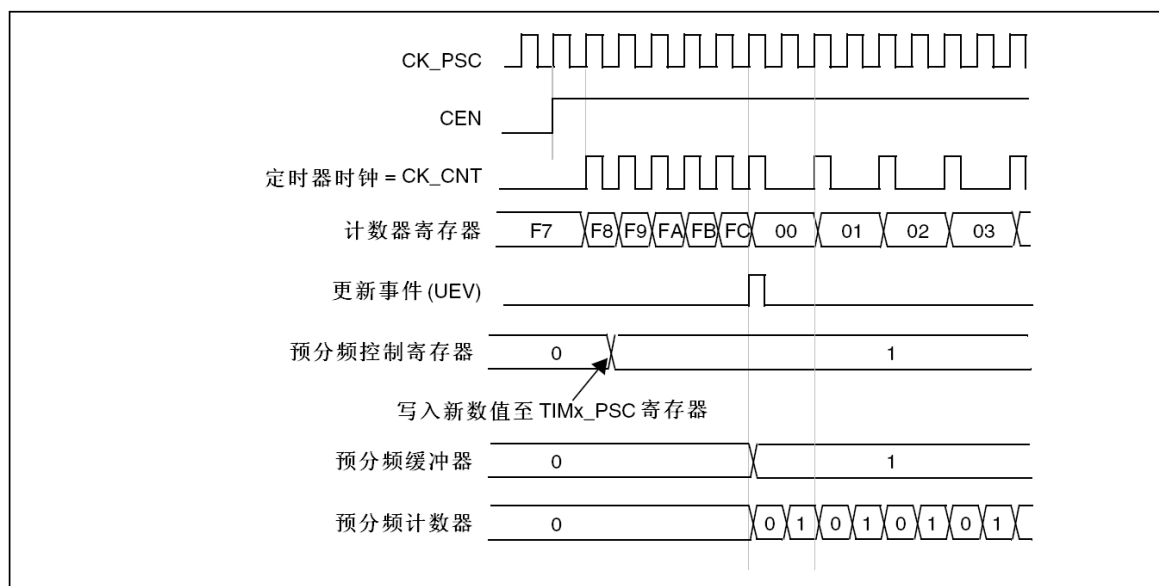
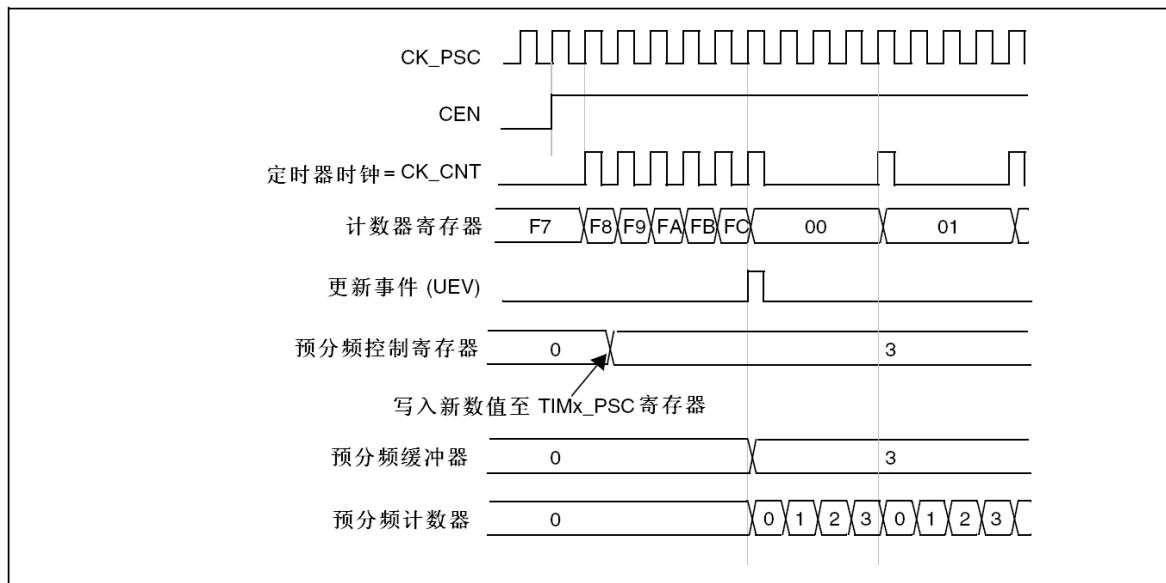


图 52 当预分频器的参数从 1 变到 4 时，计数器的时序图



13.3.2 计数器模式

13.3.2.1 向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值 (TIMx_ARR 计数器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件。

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数 (TIMx_RCR) 时，产生更新事件 (UEV)；否则每次计数器溢出时才产生更新事件。

在 TIMx_EGR 寄存器中 (通过软件方式或者使用从模式控制器) 设置 UG 位也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清 '0' 之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 '0'，同时预分频器的计数也被清 0 (但预分频器的数值不变)。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志 (即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时 (依据 URS 位) 设置更新标志位

(TIMx_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIMx_RCR 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值 (TIMx_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值 (TIMx_PSC 寄存器的内容)。下图给出一些例子，当 TIMx_ARR=0x36 时计数器在不同时钟频率下的动作。图 53 计数器时序图，内部时钟分频因子为 1

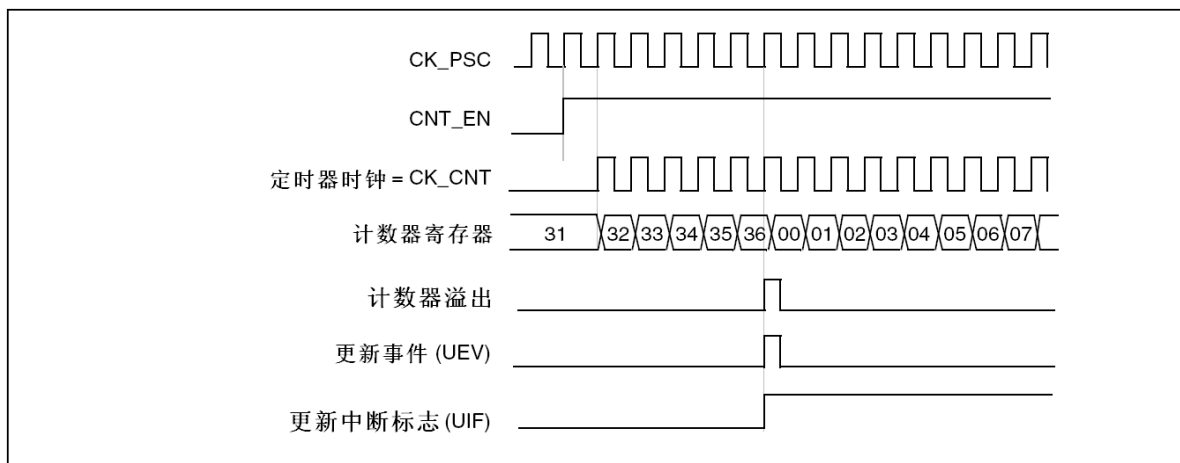


图 54 计数器时序图，内部时钟分频因子为 2

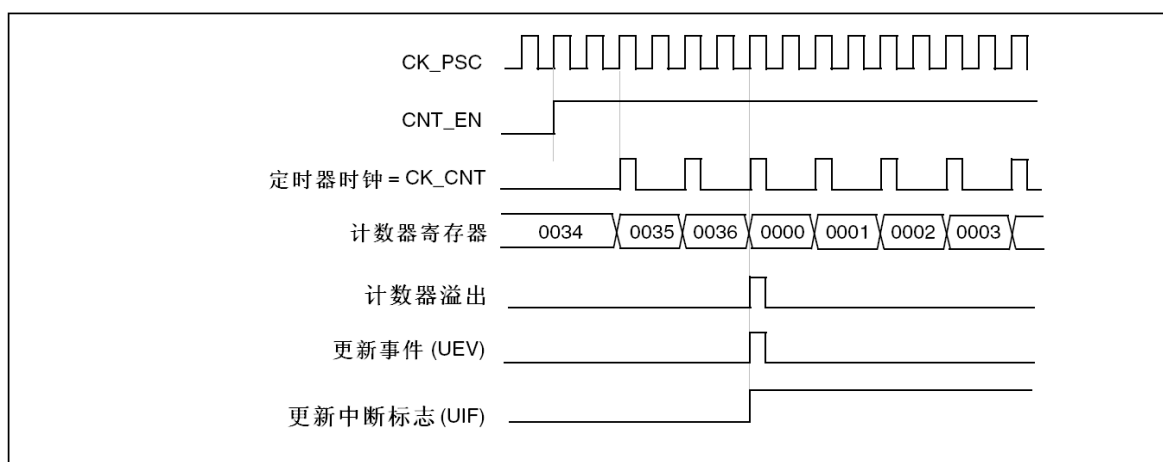


图 55 计数器时序图，内部时钟分频因子为 4

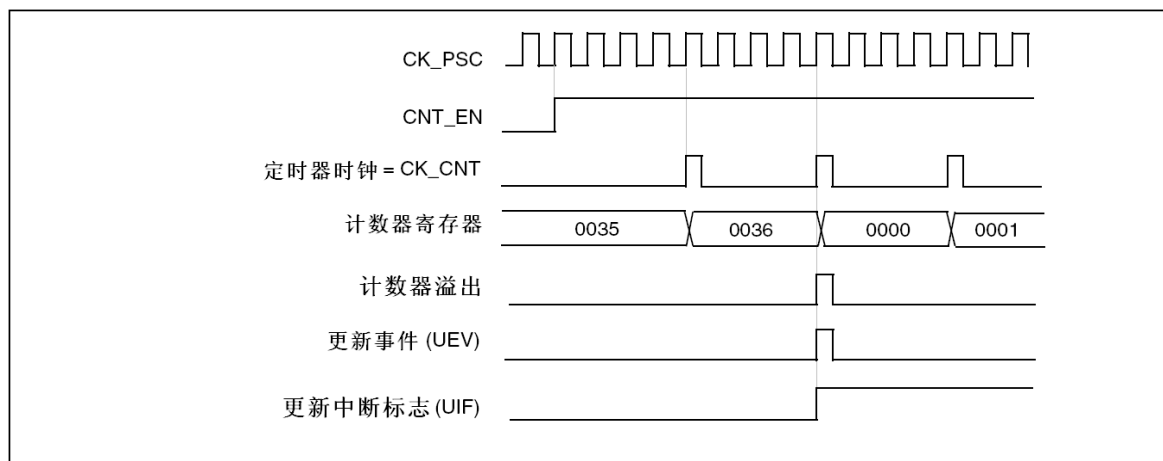


图 56 计数器时序图，内部时钟分频因子为 N

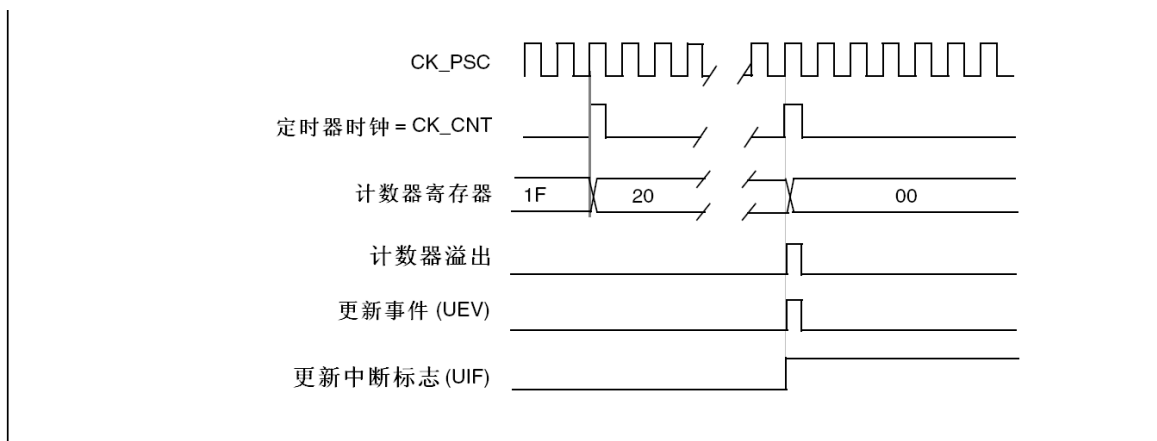
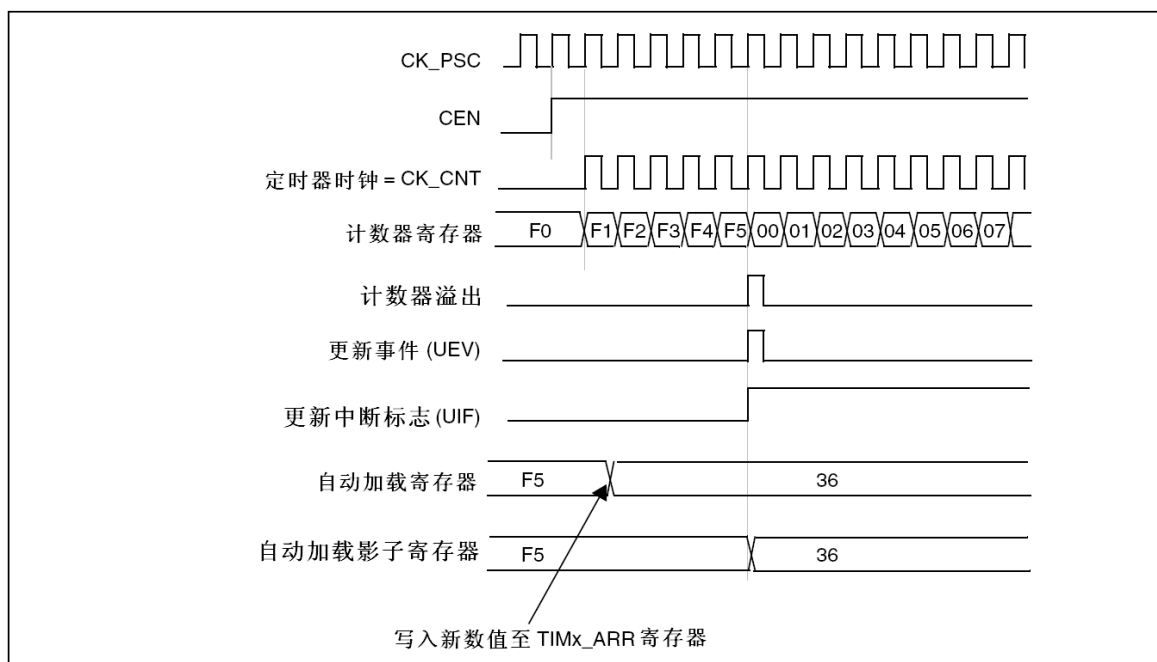
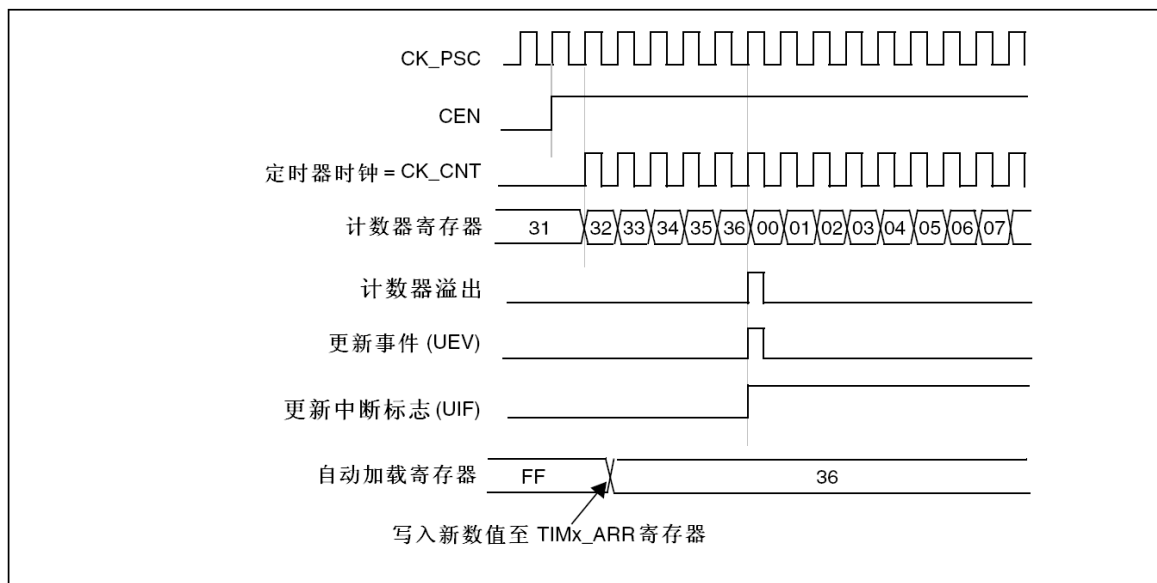


图 57 计数器时序图，当 ARPE=0 时的更新事件 (TIMx_ARR 没有预装入)



13.3.2.2 向下计数模式

在向下模式中，计数器从自动装入的值 (TIMx_ARR 计数器的值) 开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

如果使用了重复计数器，当向下计数重复了重复计数寄存器 (TIMx_RCR) 中设定的次数后，将产生更新事件 (UEV)，否则每次计数器下溢时才产生更新事件。

在 TIMx_EGR 寄存器中 (通过软件方式或者使用从模式控制器) 设置 UG 位，也同样可以产生一个更新事件。

设置 TIMx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始 (但预分频系数不变)。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志 (因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且 (根据 URS 位的设置) 更新标志位 (TIMx_SR 寄存器中的 UIF 位) 也被设置。

- 重复计数器被重置为 TIMx_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载的值 (TIMx_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值 (TIMx_ARR 寄存器中的内容)。注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIMx_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

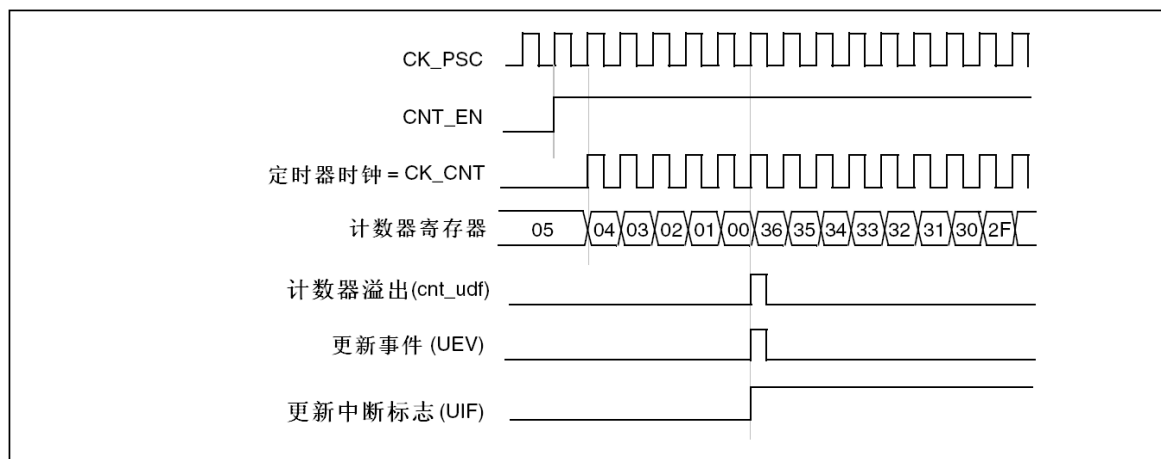


图 60 计数器时序图，内部时钟分频因子为 2

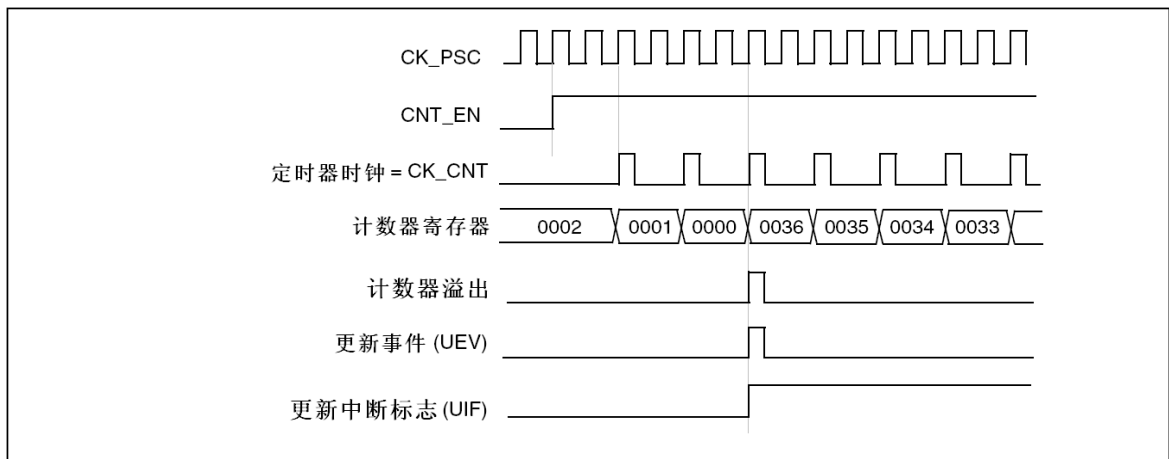


图 61 计数器时序图，内部时钟分频因子为 4

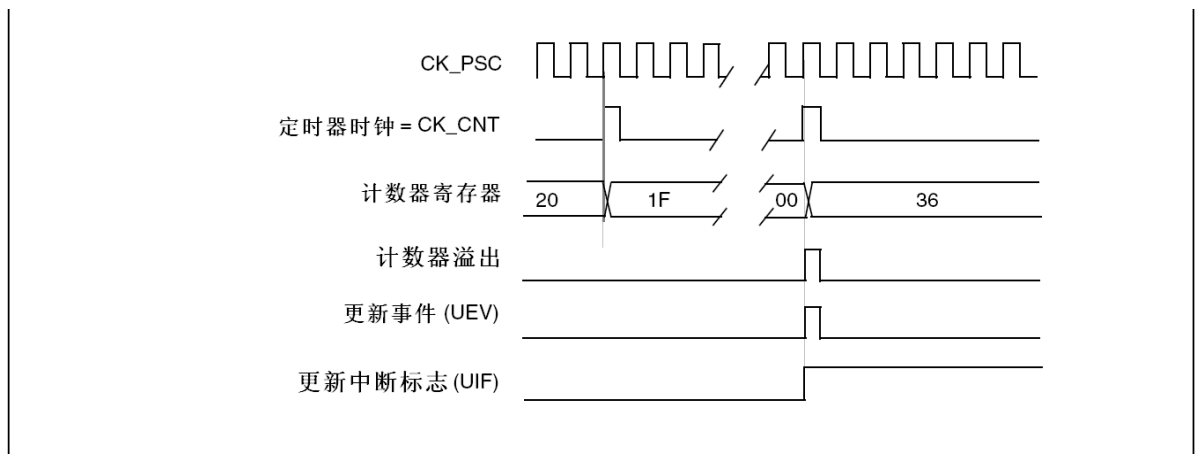
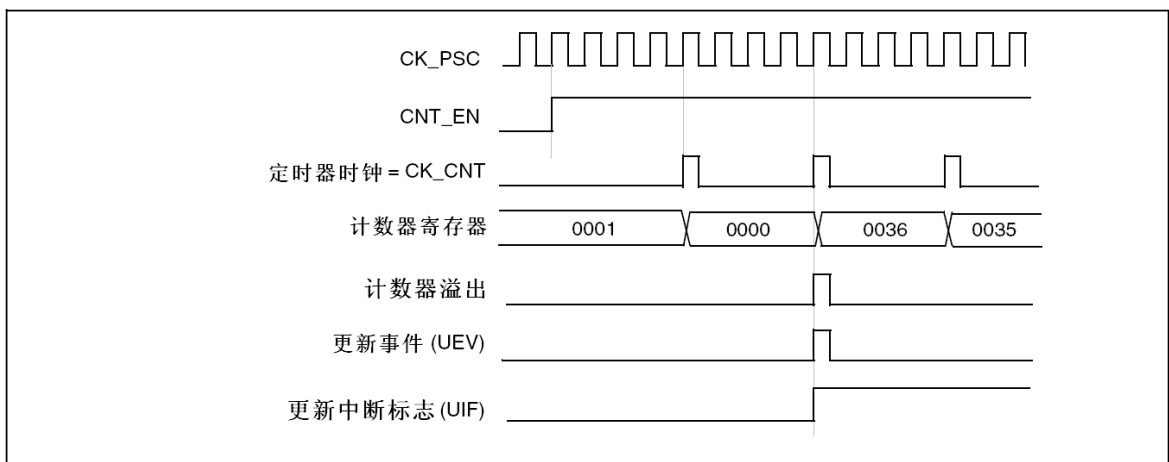
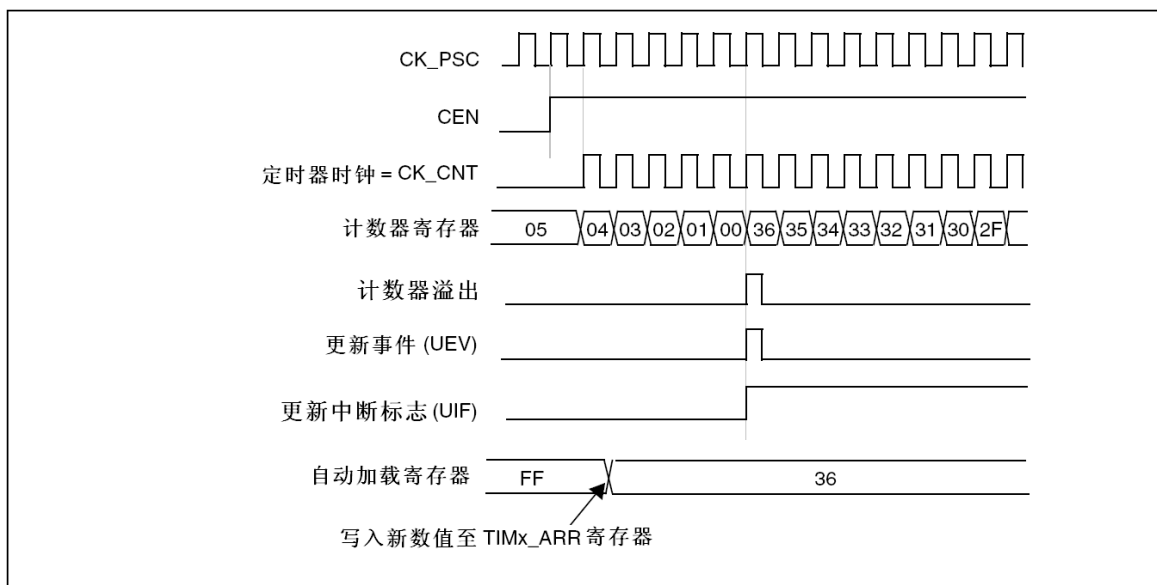


图 63 计数器时序图，当没有使用重复计数器时的更新事件



13.3.2.3 中央对齐模式(向上/向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值 (TIMx_ARR 寄存器)-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过 (软件或者使用从模式控制器) 设置 TIMx_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志 (因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

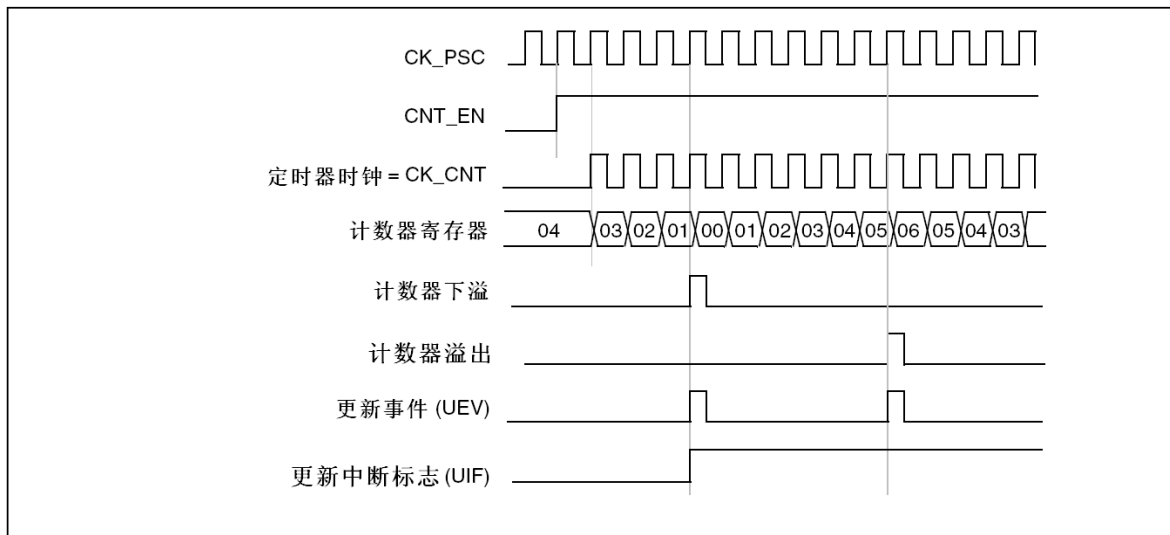
当发生更新事件时，所有的寄存器都被更新，并且 (根据 URS 位的设置) 更新标志位 (TIMx_SR 寄存器中的 UIF 位) 也被设置。

- 重复计数器被重置为 TIMx_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载 (TIMx_PSC 寄存器) 的值。

- 当前的自动加载寄存器被更新为预装载值(TIMx_ARR 寄存器中的内容)。注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值(计数器被装载为新的值)。

以下是一些计数器在不同时钟频率下的操作的例子：

图 64 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6



1. 这里使用了中心对齐模式 1 (详见 13.4.1 节)。

图 65 计数器时序图，内部时钟分频因子为 2

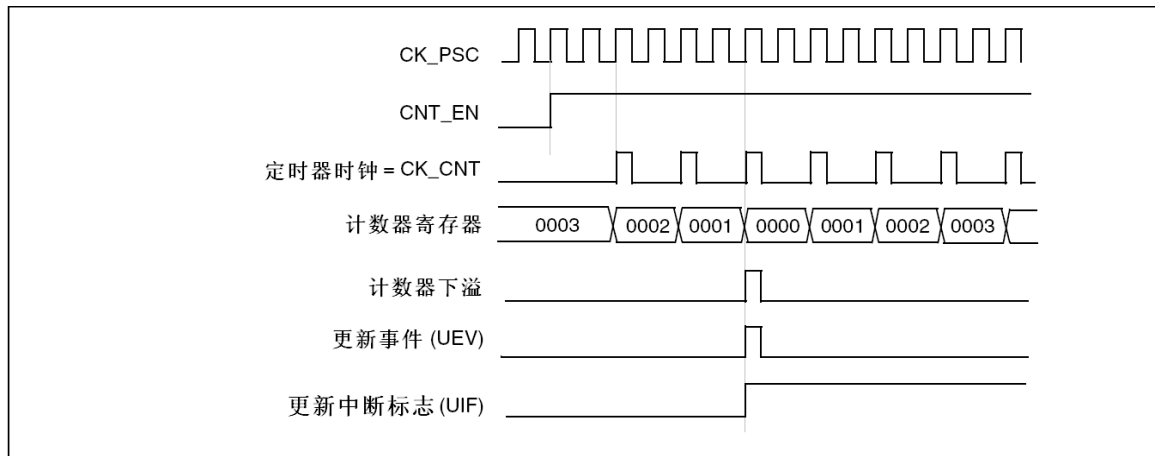


图 66 计数器时序图，内部时钟分频因子为 4，TIMx_ARR=0x36

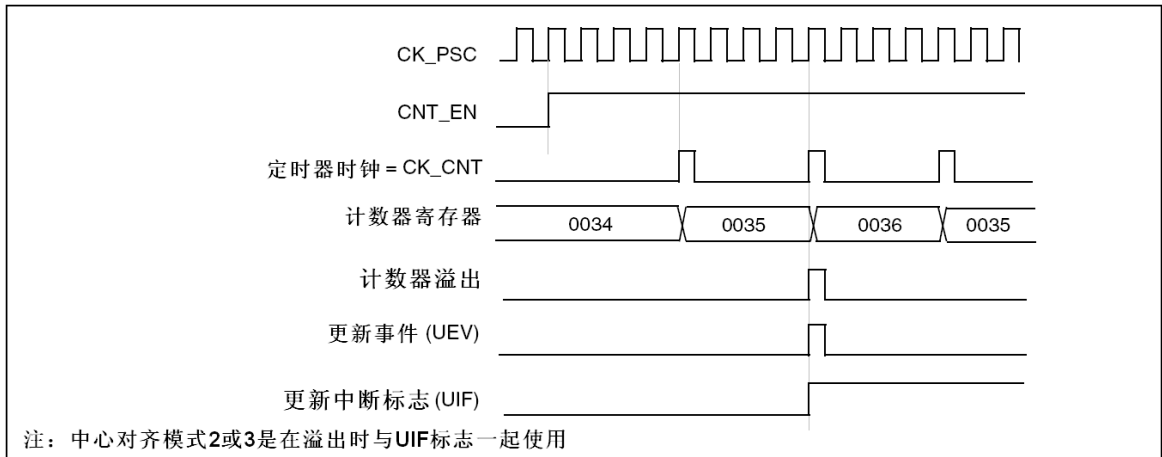


图 67 计数器时序图，内部时钟分频因子为 N

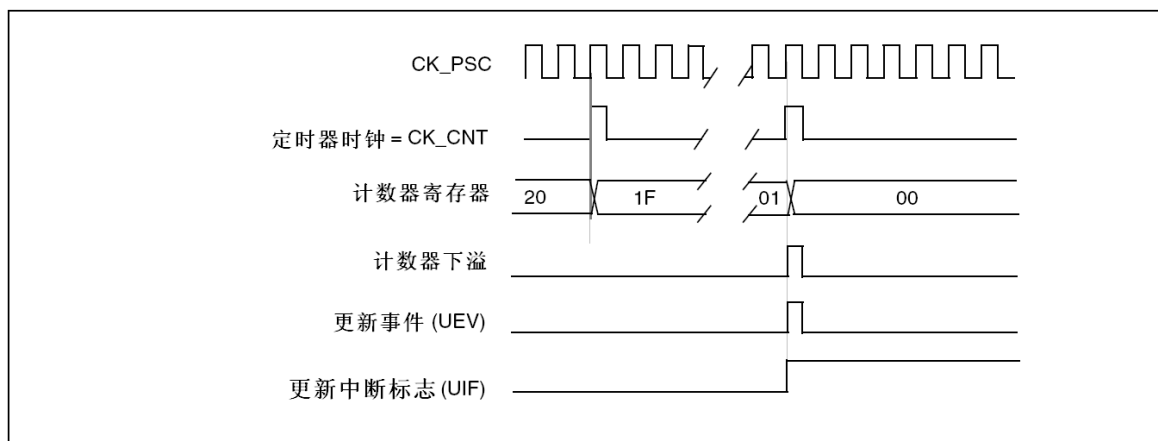
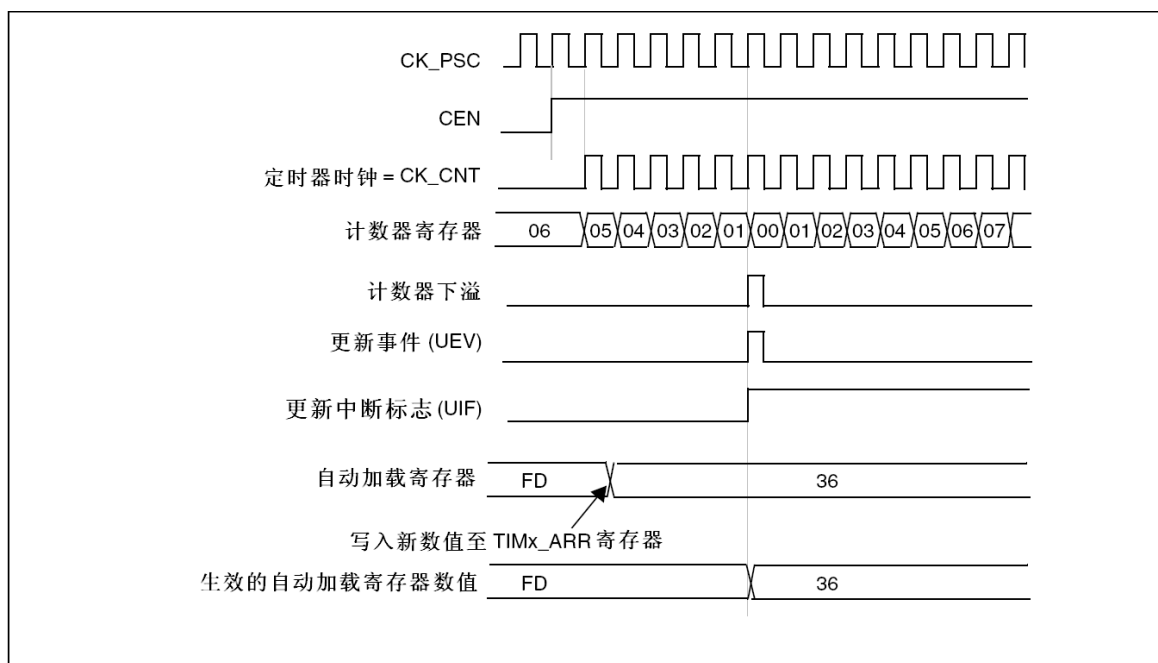
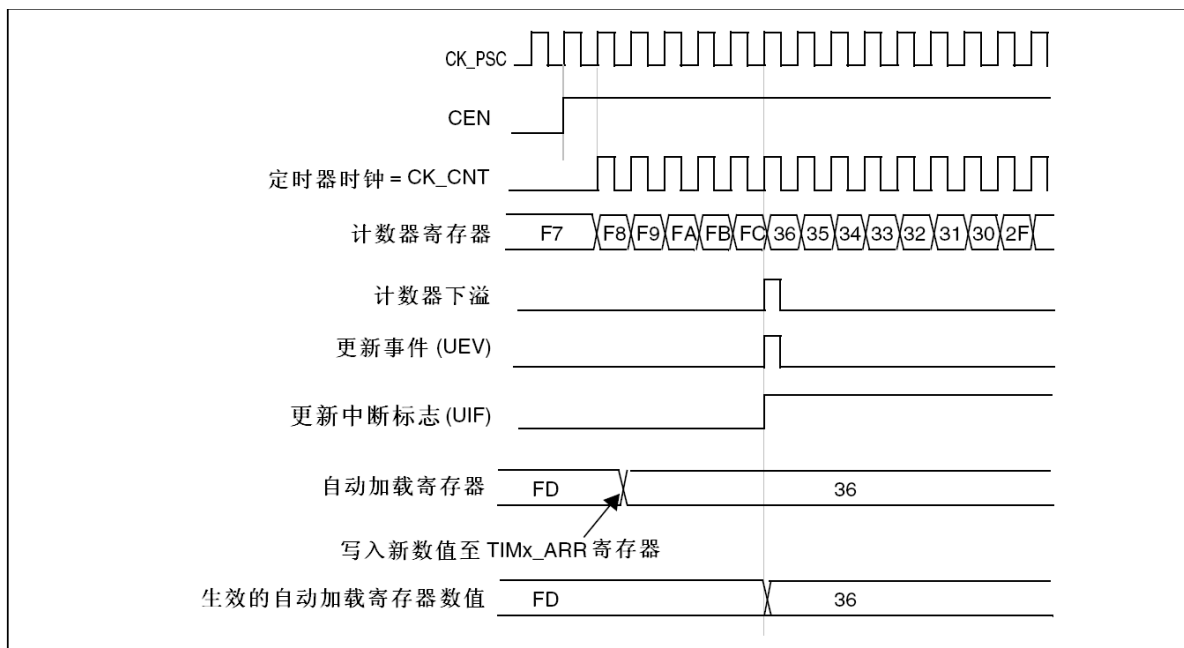


图 68 计数器时序图，ARPE=1 时的更新事件(计数器下溢)





13.3.3 重复计数器

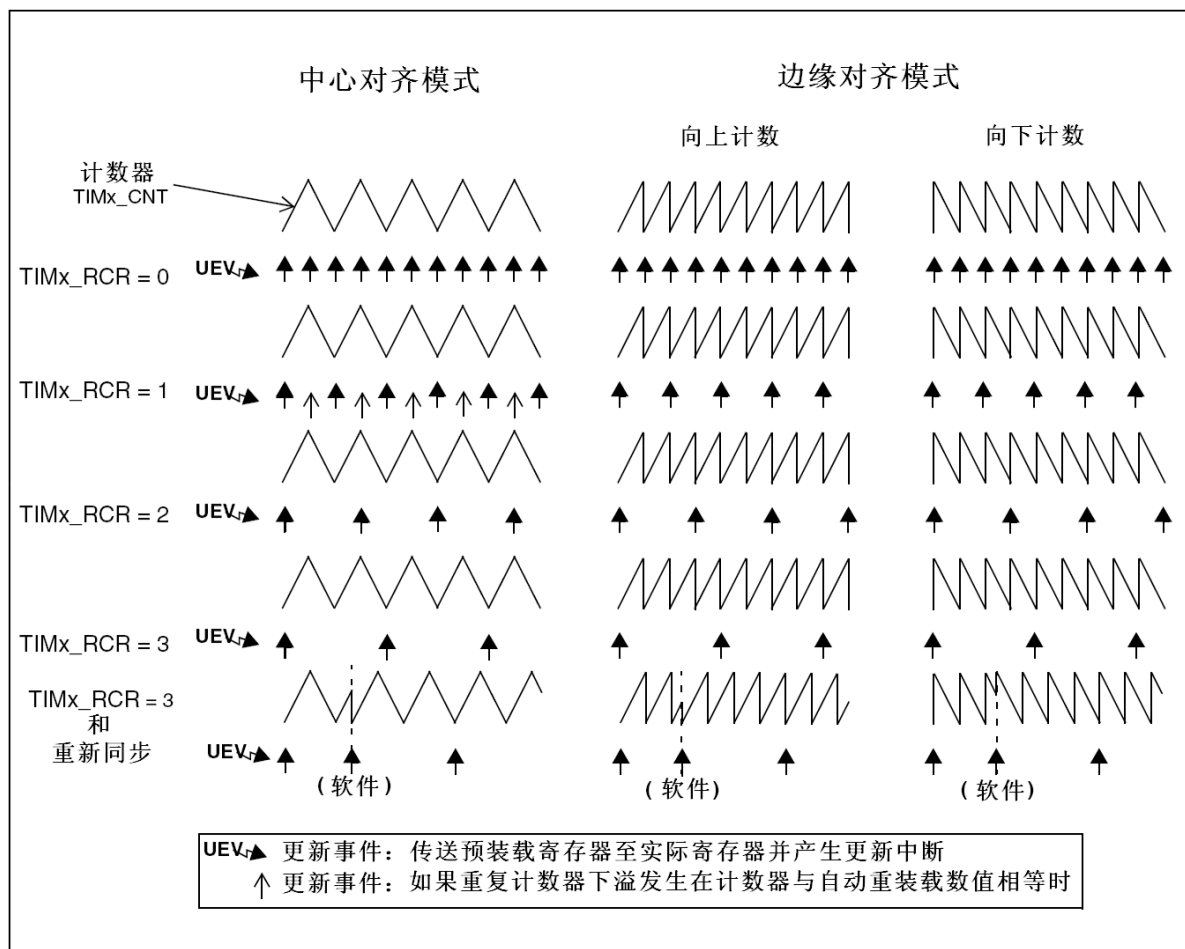
13.3.1 节“时基单元”解释了计数器上溢/下溢时更新事件(UEV)是如何产生的，然而事实上它只能在重复计数达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

这意味着在每 N 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器(TIMx_ARR 自动重载入寄存器，TIMx_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 TIMx_CCRx)，N 是 TIMx_RCR 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

- 向上计数模式下每次计数器溢出时，
- 向下计数模式下每次计数器下溢时，
- 中央对齐模式下每次上溢和每次下溢时。虽然这样限制了 PWM 的最大循环周期为 128，但它能够在每个 PWM 周期 2 次更新占空比。在中央对齐模式下，因为波形是对称的，如果每个 PWM 周期中仅刷新一次比较寄存器，则最大的分辨率为 $2 \times T_{ck}$ 。

重复计数器是自动加载的，重复速率是由 TIMx_RCR 寄存器的值定义(参看图 70)。当更新事件由软件产生(通过设置 TIMx_EGR 中的 UG 位)或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMx_RCR 寄存器中的内容被重载入到重复计数器。



13.3.4 时钟选择

计数器时钟可由下列时钟源提供：

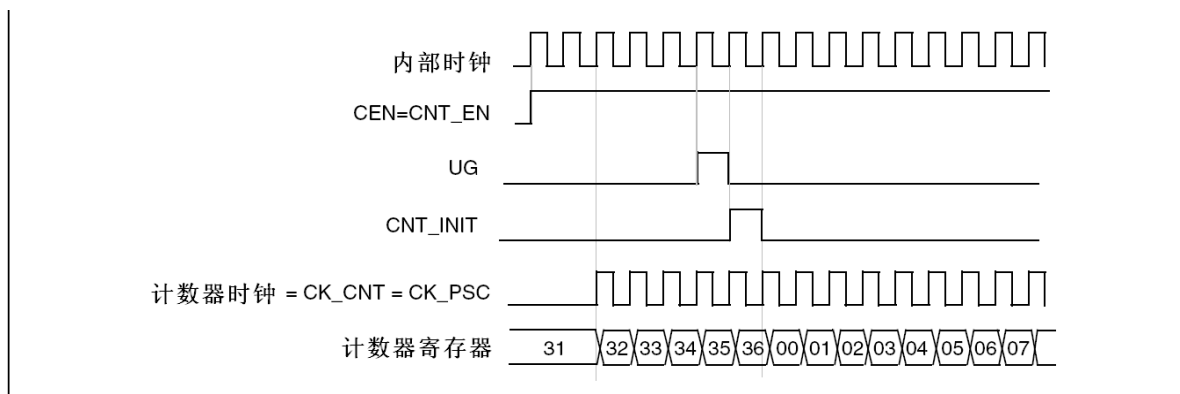
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。详见下一章。

13.3.4.1 内部时钟源 (CK_INT)

如果禁止了从模式控制器 (SMS=000)，则 CEN、DIR (TIMx_CR1 寄存器) 和 UG 位 (TIMx_EGR 寄存器) 是事实上的控制位，并且只能被软件修改 (UG 位仍被自动清除)。只要 CEN 位被写成 '1'，预

分频器的时钟就由内部时钟 CK_INT 提供。

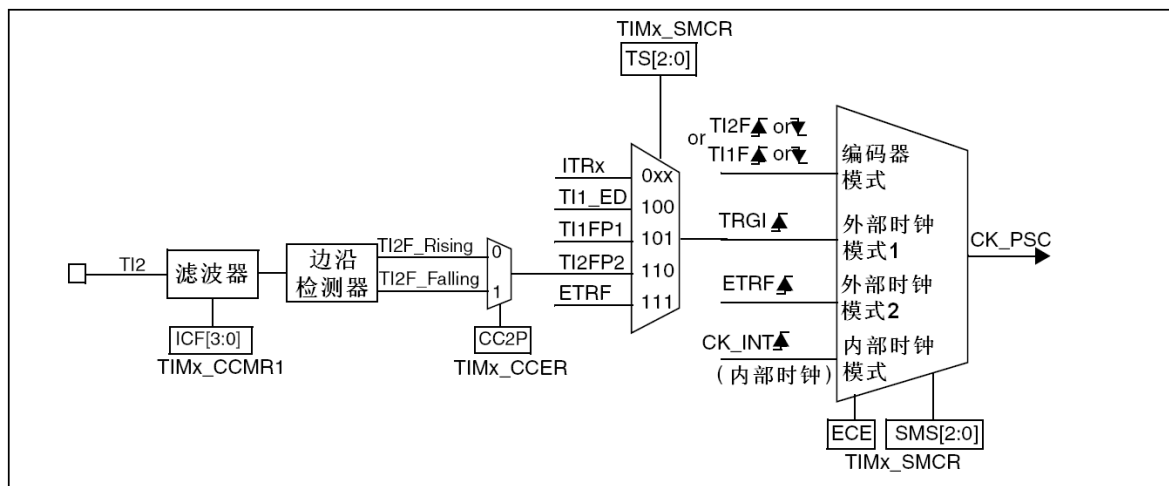
下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。



13.3.4.2 外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图 72 TI2 外部时钟连接例子



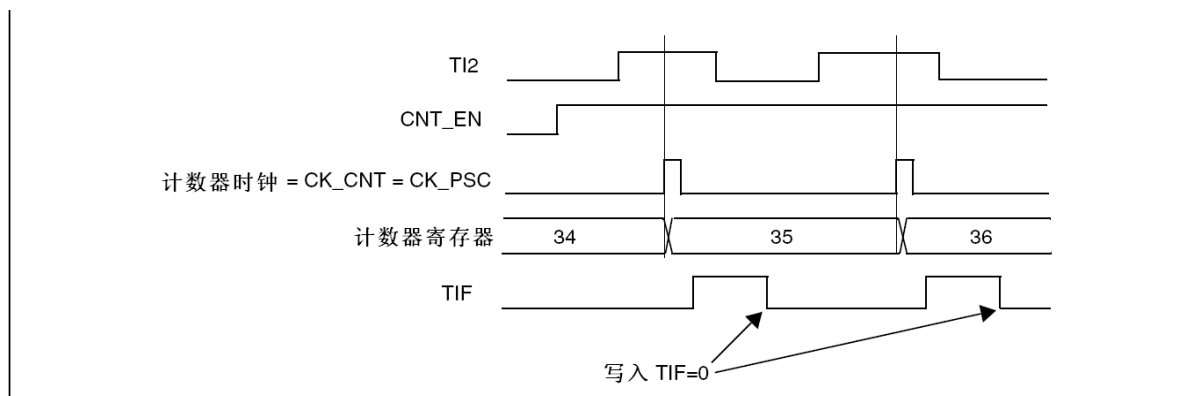
例如，要配置向上计数器在 TI2 输入端的上升沿计数，使用下列步骤： 配置 TIMx_CCMR1 寄存器 CC2S=01，配置信道 2 检测 TI2 输入的上升沿

配置 TIMx_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持

IC2F=0000)

配置 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性 配置 TIMx_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1 配置 TIMx_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源 设置 TIMx_CR1 寄存器的 CEN=1，启动计数器

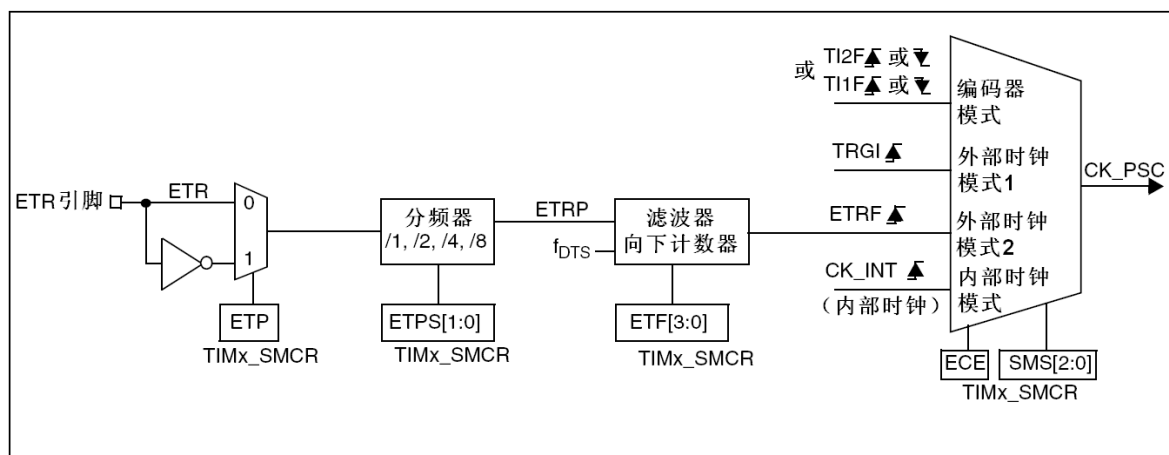
注： 捕获预分频器不用作触发，所以不需要对它进行配置 当上升沿出现在 TI2，计数器计数一次，且 TIF 标志被设置。在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。



13.3.4.3 外部时钟源模式 2

选定此模式的方法为：令 TIMx_SMCR 寄存器中的 ECE=1 计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。下图是外部触发输入的框图

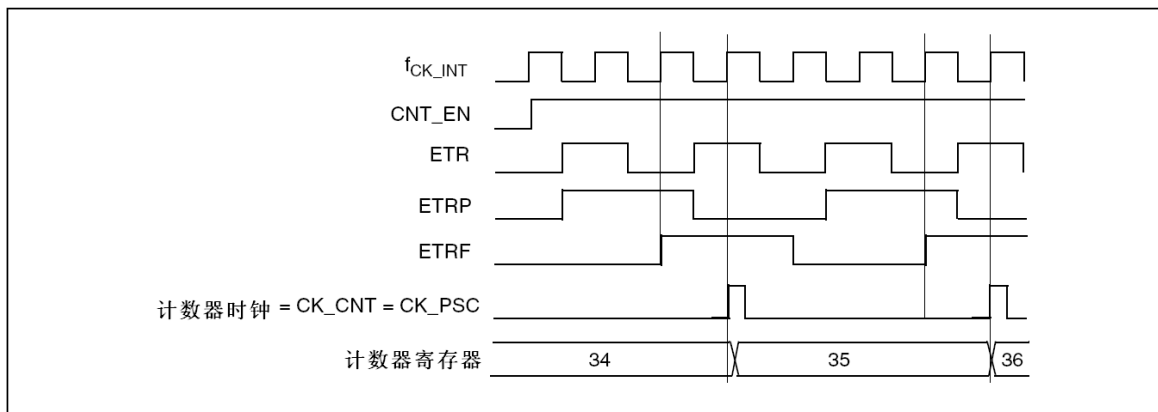
图 74 外部触发输入框图



例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

4. 本例中不需要滤波器，置 TIMx_SMCR 寄存器中的 ETF[3:0]=0000 设置预分频器，置 TIMx_SMCR 寄存器中的 ETPS[1:0]=01 选择 ETR 的上升沿检测，置 TIMx_SMCR 寄存器中的 ETP=0 开启外部时钟模式 2，写 TIMx_SMCR 寄存器中的 ECE=1 启动计数器，写 TIMx_CR1 寄存器中的 CEN=1 计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

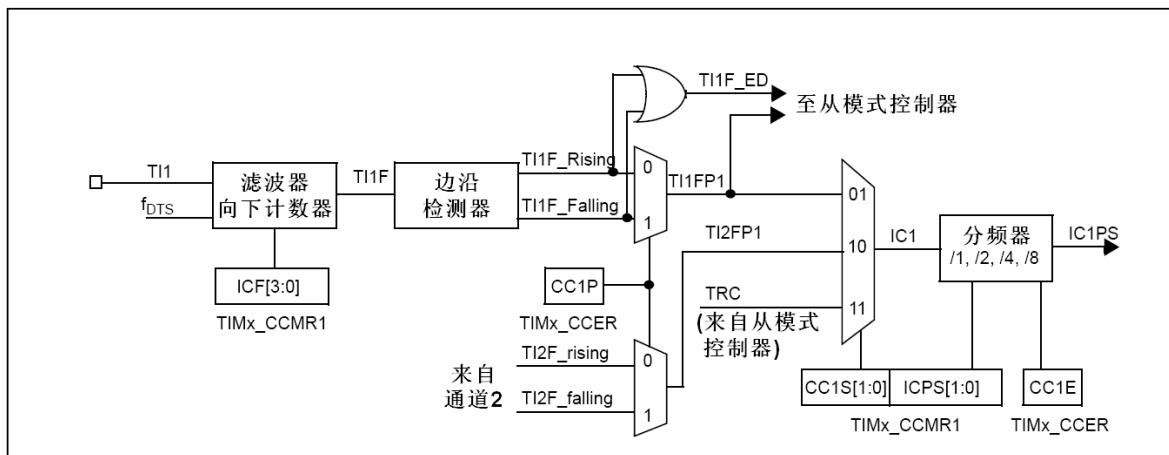


13.3.5 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器(包含影子寄存器)，包括捕获的输入部分(数字滤波、多路复用和预分频器)，和输出部分(比较器和输出控制)。图 76 至图 79 是一个捕获/比较通道概览。

输入部分对相应的 TI_x 输入信号采样，并产生一个滤波后的信号 TI_xF 。然后，一个带极性选择的边缘监测器产生一个信号 (TI_xFP_x)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器 (IC_xPS)。

图 76 捕获/比较通道(如：通道 1 输入部分)



输出部分产生一个中间波形 OC_xRef (高有效) 作为基准，链的末端决定最终输出信号的极性。

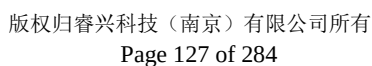
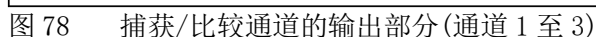
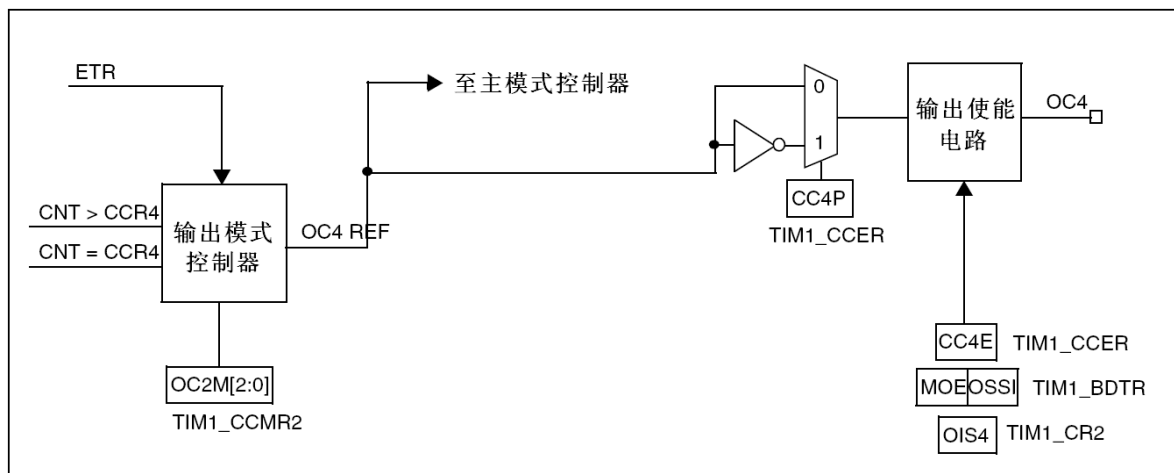


图 79 捕获/比较通道的输出部分(通道 4)



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写过程仅操作预装载寄存器。在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

13.3.6 输入捕获模式

在输入捕获模式下，当检测到 IC_x 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器 (TIM_x_CCR_x) 中。当发生捕获事件时，相应的 CC_xIF 标志 (TIM_x_SR 寄存器) 被置 1，如果开放了中断或者 DMA 操作，则将产生中断或者 DMA 请求。如果发生捕获事件时 CC_xIF 标志已经为高，那么重复捕获标志 CC_xOF (TIM_x_SR 寄存器) 被置 1。写 CC_xIF=0 可清除 CC_xIF，或读取存储在 TIM_x_CCR_x 寄存器中的捕获数据也可清除 CC_xIF。写 CC_xOF=0 可清除 CC_xOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM_x_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIM_x_CCR1 必须连接到 TI1 输入，所以写入 TIM_x_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为 '00'，信道被配置为输入，并且 TIM_x_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽 (即输入为 TI_x 时，输入滤波器控制位是 TIM_x_CCMR_x 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以 (以 fDTS 频率) 连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM_x_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIM_x_CCER 寄存器中写入 CC1P=0 (上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止 (写 TIM_x_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIM_x_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIM_x_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIM_x_DIER 寄存器中的 CC1DE 位允许 DMA 请求。当发生一个输入捕获时：
- 产生有效的电平转换时，计数器的值被传送到 TIM_x_CCR1 寄存器。
- CC1IF 标志被设置 (中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。

- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。 为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注： 设置 `TIMx_EGR` 寄存器中相应的 `CCxG` 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

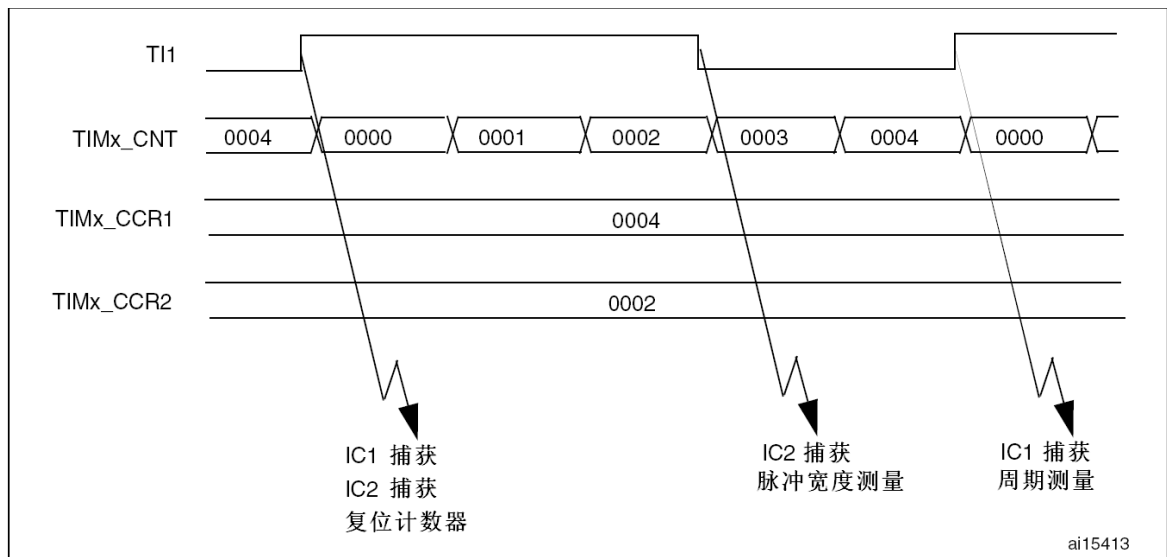
13.3.7 PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 `ICx` 信号被映射至同一个 `TIx` 输入。
- 这 2 个 `ICx` 信号为边沿有效，但是极性相反。
- 其中一个 `TIxFP` 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 `TI1` 上的 PWM 信号的长度 (`TIMx_CCR1` 寄存器) 和 占空比 (`TIMx_CCR2` 寄存器)，具体步骤如下 (取决于 `CK_INT` 的频率和预分频器的值)

- 选择 `TIMx_CCR1` 的有效输入：置 `TIMx_CCMR1` 寄存器的 `CC1S=01` (选中 `TI1`)。
 - 选择 `TI1FP1` 的有效极性 (用来捕获数据到 `TIMx_CCR1` 中和清除计数器)：置 `CC1P=0` (上升沿 有效)。
- 选择 `TIMx_CCR2` 的有效输入：置 `TIMx_CCMR1` 寄存器的 `CC2S=10` (选中 `TI1`)。
- 选择 `TI1FP2` 的有效极性 (捕获数据到 `TIMx_CCR2`)：置 `CC2P=1` (下降沿有效)。
 - 选择有效的触发输入信号：置 `TIMx_SMCR` 寄存器中的 `TS=101` (选择 `TI1FP1`)。
 - 配置从模式控制器为复位模式：置 `TIMx_SMCR` 中的 `SMS=100`。
 - 使能捕获：置 `TIMx_CCER` 寄存器中 `CC1E=1` 且 `CC2E=1`。 图 80 PWM 输入模式时序



因为只有 `TI1FP1` 和 `TI2FP2` 连到了从模式控制器，所以 PWM 输入模式只能使用 `TIMx_CH1` / `TIMx_CH2` 信号。

13.3.8 强置输出模式

在输出模式(TIMx_CCMRx 寄存器中 CCxS=00)下, 输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态, 而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRx 寄存器中相应的 OCxM=101, 即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效), 同时 OCx 得到 CCxP 极性相反的信号。

例如: CCxP=0(OCx 高电平有效), 则 OCx 被强置为高电平。置 TIMx_CCMRx 寄存器中的 OCxM=100, 可强置 OCxREF 信号为低。

该模式下, 在 TIMx_CCRx 影子寄存器和计数器之间的比较仍然在进行, 相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

13.3.9 输出比较模式

此项功能是用来控制一个输出波形, 或者指示一段给定的时间已经到时。当计数器与捕获/比较寄存器的内容相同时, 输出比较功能做如下操作:

- 将输出比较模式(TIMx_CCMRx 寄存器中的 OCxM 位)和输出极性(TIMx_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时, 输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIMx_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIMx_DIER 寄存器中的 CCxIE 位), 则产生一个中断。
 - 若设置了相应的使能位(TIMx_DIER 寄存器中的 CCxDE 位, TIMx_CR2 寄存器中的 CCDS 位选择 DMA 请求功能), 则产生一个 DMA 请求。

TIMx_CCMRx 中的 OCxPE 位选择 TIMx_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

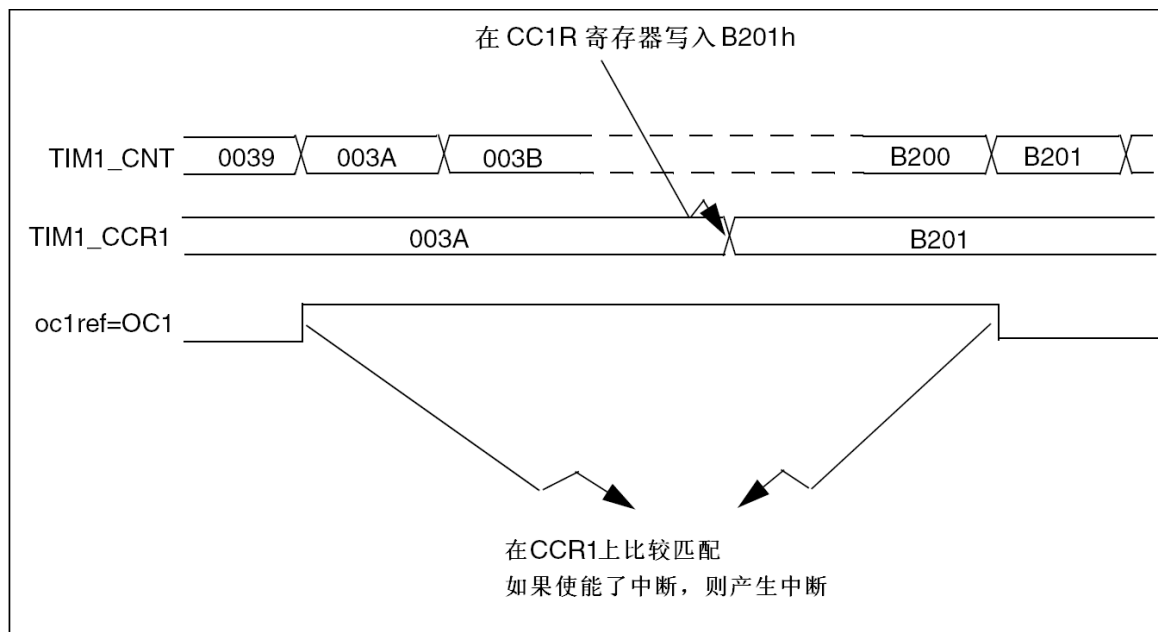
同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤:

1. 选择计数器时钟(内部, 外部, 预分频器)。
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRx 寄存器中。
3. 如果要产生一个中断请求, 设置 CCxIE 位。
4. 选择输出模式, 例如:
 - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚, 设置 OCxM=011
 - 置 OCxPE = 0 禁用预装载寄存器
 - 置 CCxP = 0 选择极性为高电平有效
 - 置 CCxE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位启动计数器

TIMx_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器 (OCxPE='0'，否则 TIMx_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 81 输出比较模式，翻转 OC1



13.3.10 PWM 模式

脉冲宽度调制模式可以产生一个由 TIMx_ARR 寄存器确定频率、由 TIMx_CCRx 寄存器确定占空比的信号。

在 TIMx_CCMRx 寄存器中的 OCxM 位写入 '110' (PWM 模式 1) 或 '111' (PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器，最后还要设置 TIMx_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中) 使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器中的 UG 位来初始化所有的寄存器。OCx 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或

低电平有效。OCx 的输出使能通过 (TIMx_CCER 和 TIMx_BDTR 寄存器中) CCxE、CCxNE、

MOE、OSSI 和 OSSR 位的组合控制。详见 TIMx_CCER 寄存器的描述。在 PWM 模式 (模式 1 或模式 2) 下，TIMx_CNT 和 TIMx_CCRx 始终在进行比较，(依据计数器的计数方向) 以确定是否符合 $TIMx_CCRx \leq TIMx_CNT$ 或者 $TIMx_CNT \leq TIMx_CCRx$ 。根据 TIMx_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的

PWM 信号。

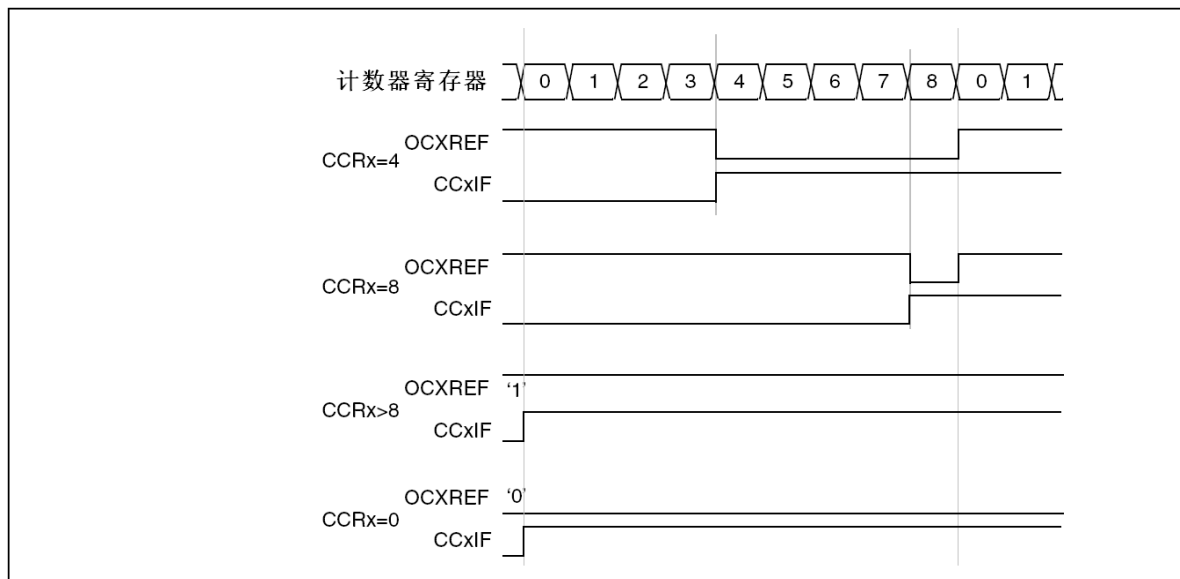
13.3.10.1 PWM 边沿对齐模式

- 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。参看 13.3.2 节。

下面是一个 PWM 模式 1 的例子。当 $TIMx_CNT < TIMx_CCRx$ 时，PWM 参考信号 $OCxREF$ 为 高，否则为低。如果 $TIMx_CCRx$ 中的比较值大于自动重装载值 ($TIMx_ARR$)，则 $OCxREF$ 保持为 '1'。如果比较值为 0，则 $OCxREF$ 保持为 '0'。下图为 $TIMx_ARR=8$ 时边沿对齐的 PWM 波形实例。

图 82 边沿对齐的 PWM 波形 ($ARR=8$)



● 向下计数的配置

当 $TIMx_CR1$ 寄存器的 DIR 位为高时执行向下计数。参看 13.3.2 节。

在 PWM 模式 1，当 $TIMx_CNT > TIMx_CCRx$ 时参考信号 $OCxREF$ 为低，否则为高。如果 $TIMx_CCRx$ 中的比较值大于 $TIMx_ARR$ 中的自动重装载值，则 $OCxREF$ 保持为 '1'。该模式下不能产生 0% 的 PWM 波形。

13.3.10.2 PWM 中央对齐模式

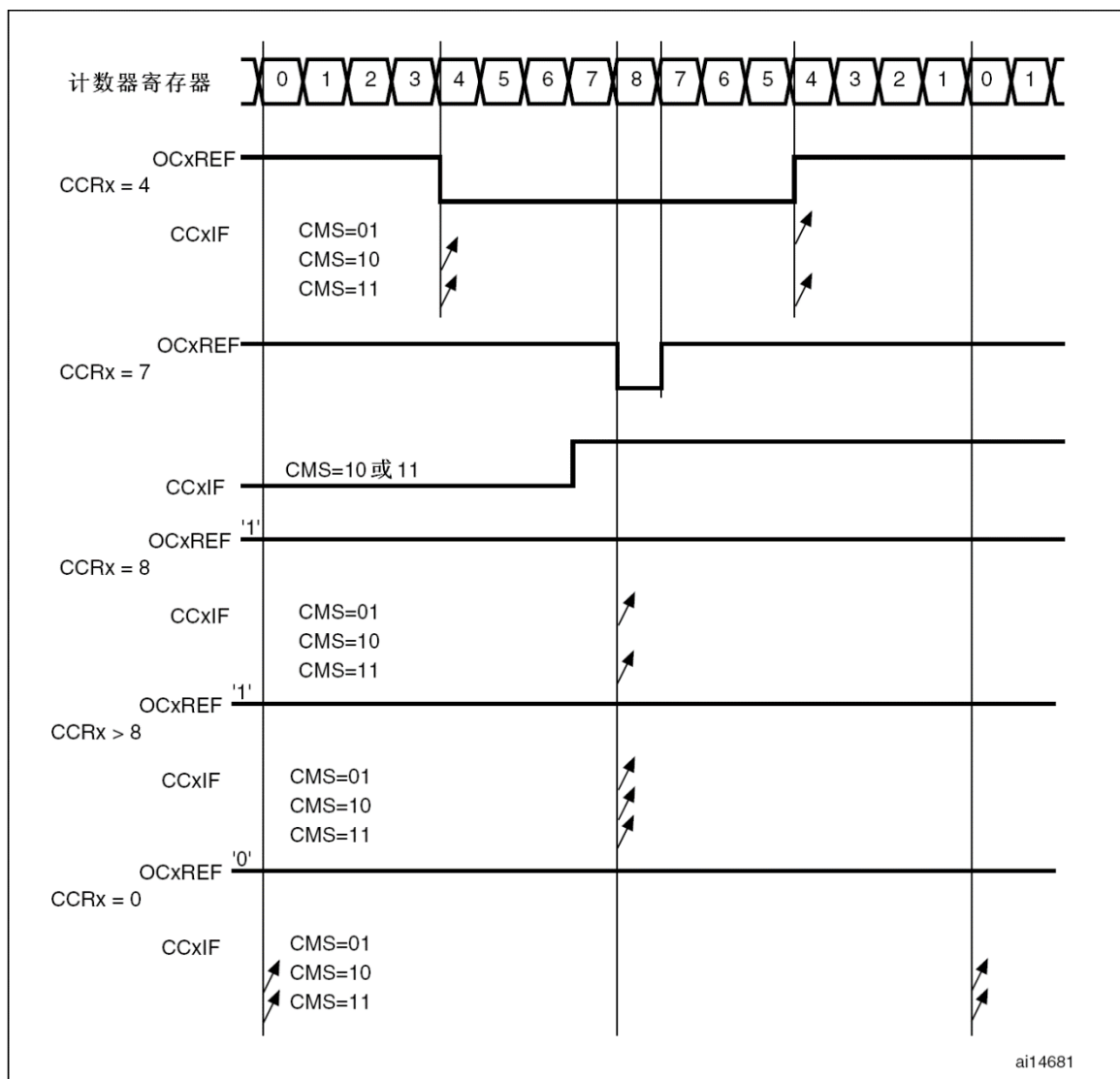
当 $TIMx_CR1$ 寄存器中的 CMS 位不为 '00' 时为中央对齐模式 (所有其他的配置对 $OCxREF/OCx$ 信号都有相同的作用)。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计

数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。 $TIMx_CR1$ 寄存器中的计数方向位 (DIR) 由硬件更新，不要软件修改它。参看 13.3.2 节的中央对齐模式。

下图给出了一些中央对齐的 PWM 波形的例子

- $TIMx_ARR=8$
- PWM 模式 1
- $TIMx_CR1$ 寄存器的 $CMS=01$ ，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

图 83 中央对齐的 PWM 波形 ($ARR=8$)



13.3.10.3 使用中央对齐模式的提示：

- 进入中央对齐模式时，使用当前的向上/向下计数配置；这意味着计数器向上还是向下计数取决于 TIM_x_CR1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这会产生不可预知的结果。特别地：
 - 如果写入计数器的值大于自动重加载的值 (TIM_x_CNT > TIM_x_ARR)，则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
 - 如果将 0 或者 TIM_x_ARR 的值写入计数器，方向被更新，但不产生更新事件 UEV
- 使用中央对齐模式最保险的方法，就是在启动计数器之前产生一个软件更新 (设置 TIM_x_EGR 位中的 UG 位)，并且不要在计数进行过程中修改计数器的值。

13.3.11 互补输出和死区插入

高级控制定时器(TIM8)能够输出两路互补信号,并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区,用户应该根据连接的输出器件和它们的特性(电平转换的延时、电源开关的延时等)来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位,可以为每一个输出独立地选择极性(主输出 OCx 或互补输出 OCxN)。

互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制: TIMx_CCER 寄存器的 CCxE 和 CCxNE 位, TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位,详见表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是,在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区,如果存在刹车电路,则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效:

- OCx 输出信号与参考信号相同,只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反,只是它的上升沿相对于参考信号的下降沿有一个延迟。如果延迟大于当前有效的输出宽度(OCx 或者 OCxN),则不会产生相应的脉冲。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

图 84 带死区插入的互补输出

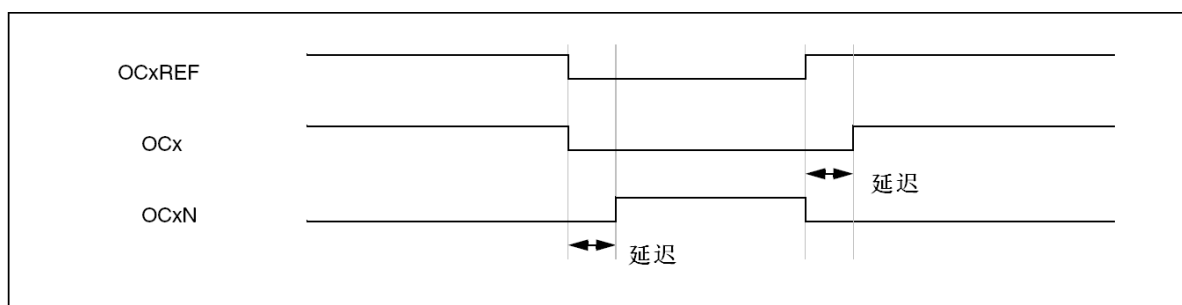


图 85 死区波形延迟大于负脉冲

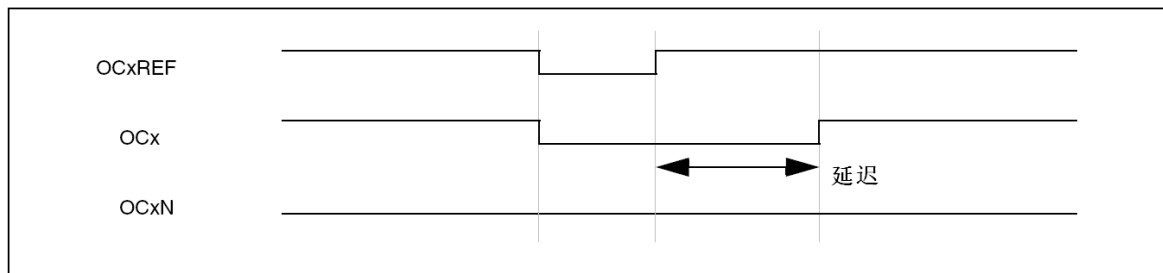
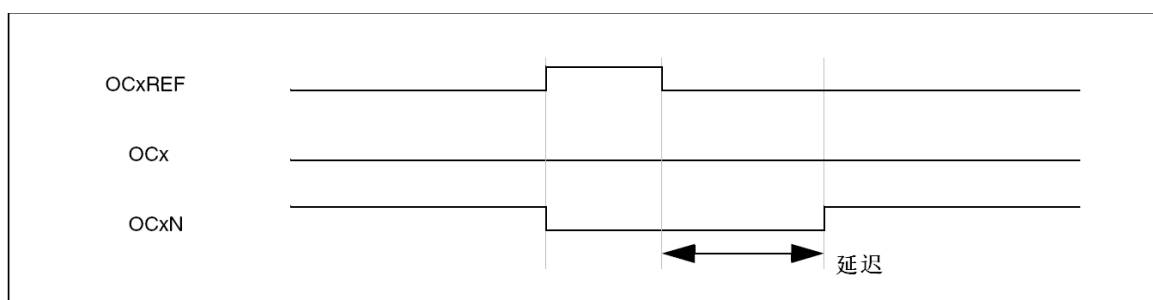


图 86 死区波形延迟大于正脉冲



每一个通道的死区延时都是相同的，是由TIMx_BDTR寄存器中的DTG位编程配置。详见13.4.18节TIM8刹车和死区寄存器(TIMx_BDTR)中的延时计算。

13.3.11.1 复位向 OCxREF 到 OCx 或 OCxN

在输出模式下(强置、输出比较或 PWM)，通过配置 TIMx_CCER 寄存器的 CCxE 和 CCxNE 位，OCxREF 可以被复位向到 OCx 或者 OCxN 的输出。

这个功能可以在互补输出处于无效电平时，在某个输出上送出一个特殊的波形(例如 PWM 或者静态有效电平)。另一个作用是，让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注：当只使能 OCxN(CCxE=0, CCxNE=1)时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时(CCxE=CCxNE=1)，当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

13.3.12 使用刹车功能

当使用刹车功能时，依据相应的控制位(TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OISx 和 OISxN 位)，输出使能信号和无效电平都会被修改。但无论何时，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。详见表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。

刹车源既可以是刹车输入引脚又可以是一个时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统产生，详见 6.2.7 节时钟安全系统(CSS)。系统复位后，刹车电路被禁止，MOE 位为低。设置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车

功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时

被修改。当写入 BKE 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOE 下降沿可以是异步的，在实际信号(作用在输出端)和同步控制位(在 TIMx_BDTR 寄存器中)之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOE=1，则读出它之前必须先插入一个延时(空指令)才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时(在刹车输入端出现选定的电平)，有下述动作：

- MOE 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态(由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOE=0，每一个输出通道输出由 TIMx_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
 - 输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作，即使定时器没有时钟时，此功能也有效。
 - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOE，死区时间比通常情况下长一些(大约 2 个 ck_tim 的时钟周期)。

- 如果 OSSR=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxE 与 CCxNE 之一变高时，使能输出变为高。
- 如果设置了 TIMx_DIER 寄存器中的 BIE 位，当刹车状态标志(TIMx_SR 寄存器中的 BIF 位)为'1'时，则产生一个中断。如果设置了 TIMx_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BDTR 寄存器中的 AOE 位，在下一个更新事件 UEV 时 MOE 位被自动置位；例如，这可以用来进行整形。否则，MOE 始终保持低直到被再次置'1'；此时，这个特性可 以被用在安全方面，你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

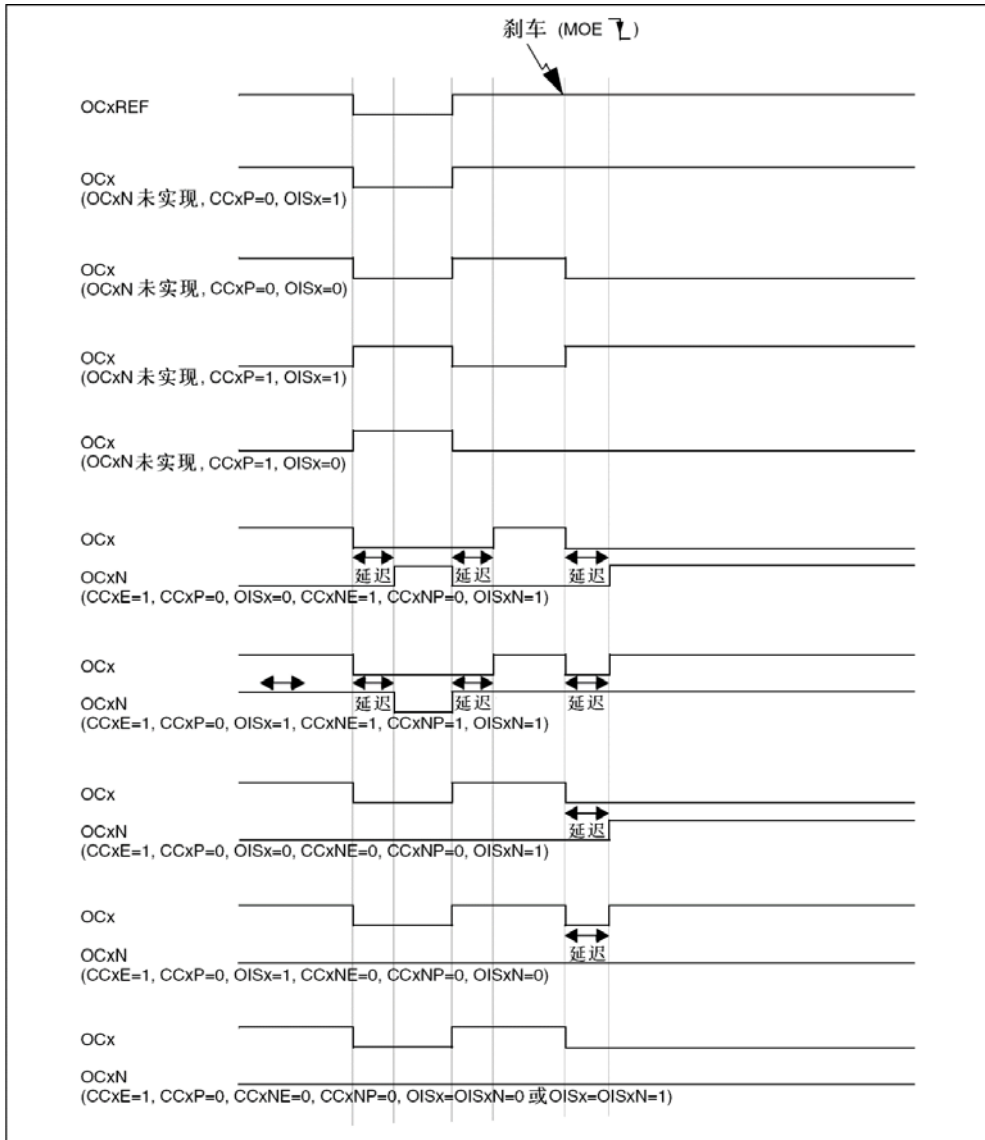
注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时(自动地或者通过软件)设置 MOE。同时，状态标志 BIF 不能被清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMx_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。

用户可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 13.4.18 节

TIM8 刹车和死区寄存器(TIMx_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。下图显示响应刹车的输出实例。

图 87 响应刹车的输出



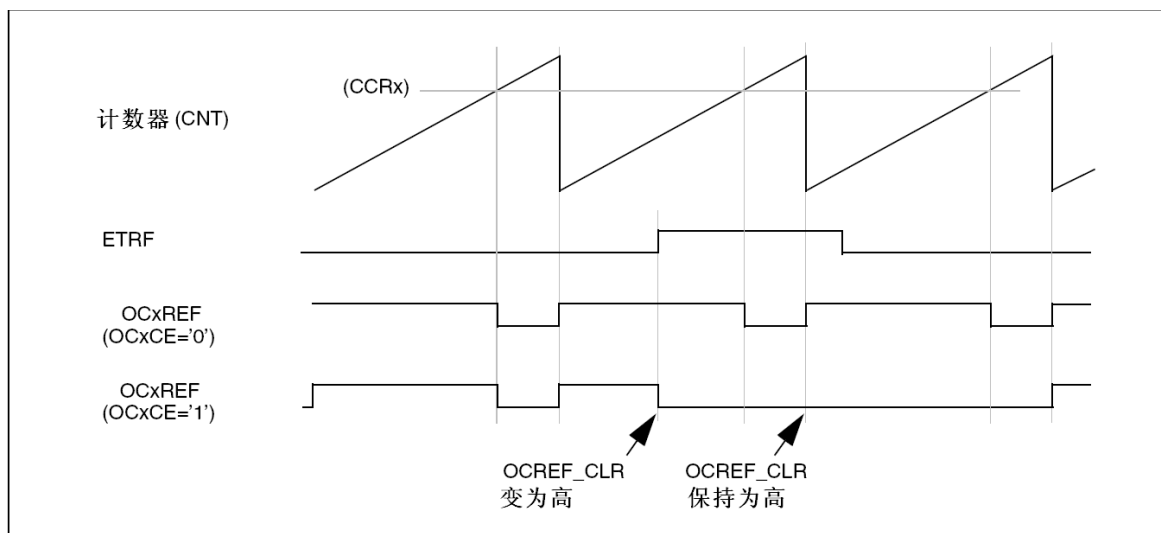
13.3.13 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIMx_CCMRx 寄存器中对应的 OCxCE 位为 '1'，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIMx_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE=0。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可以根据需要配置。下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIMx 被置于 PWM 模式。

图 88 清除 TIMx 的 OCxREF



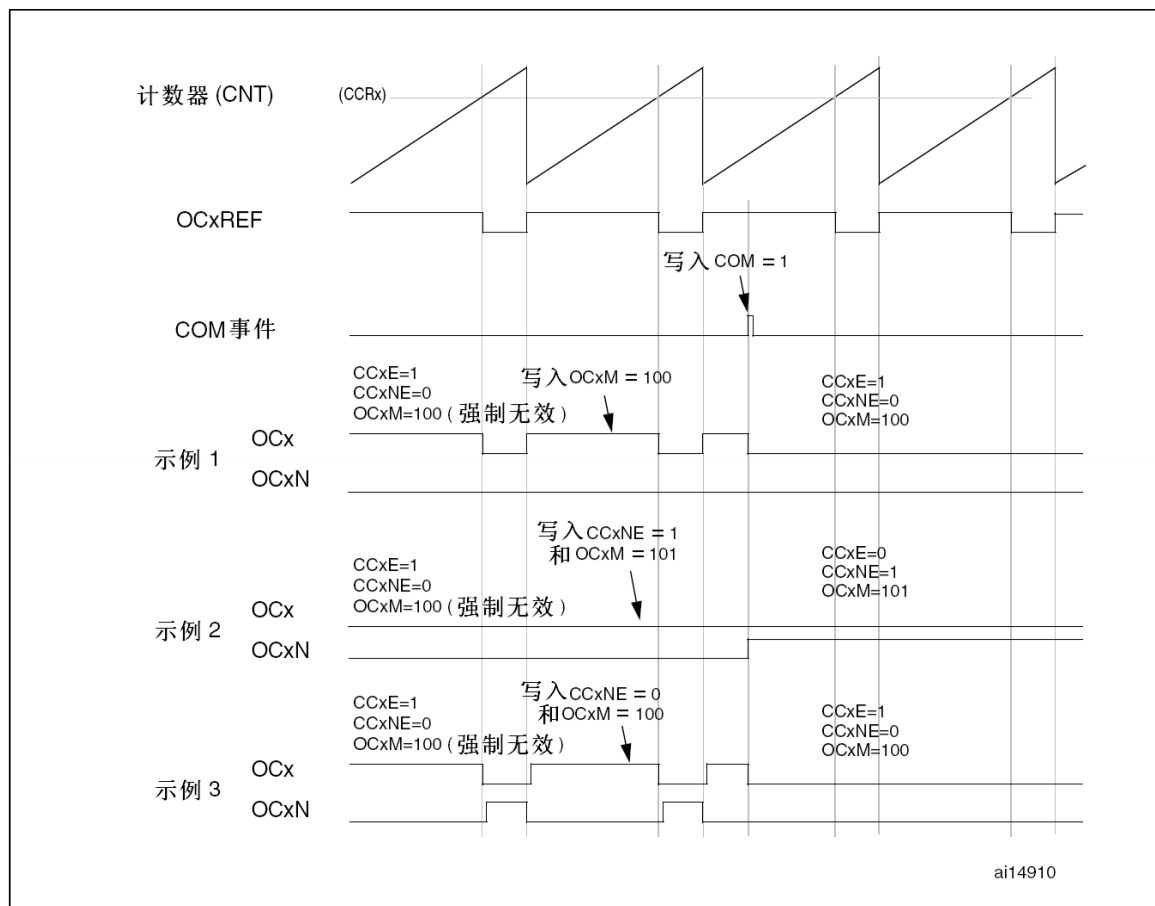
13.3.14 产生六步 PWM 输出

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样你就可以预先设置好下一步配置，并在同一个时刻同时修更改所有信道的配置。COM 可以通过设置 TIMx_EGR 寄存器的 COM 位由软件产生，或在 TRGI 上升沿由硬件产生。

当发生 COM 事件时会设置一个标志位 (TIMx_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIMx_DIER 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIMx_DIER 寄存器的 COMDE 位，则产生一个 DMA 请求。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

图 89 产生六步 PWM，使用 COM 的例子 (OSSR=1)



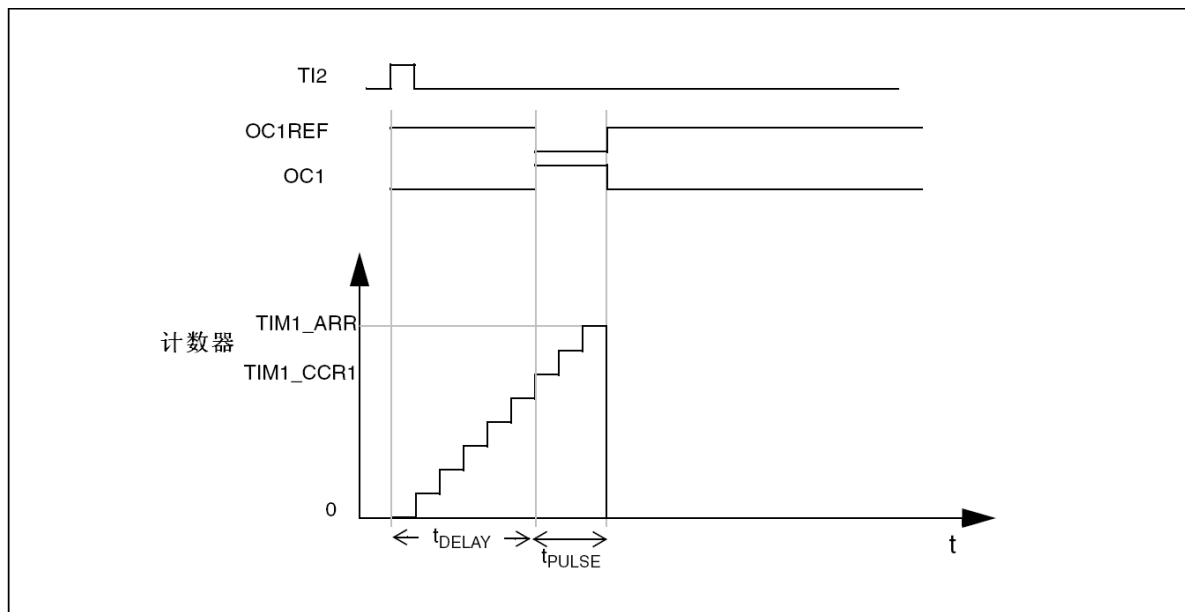
13.3.15 单脉冲模式

单脉冲模式 (OPM) 是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可过程控制的脉冲。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIM_x CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前 (当定时器正在等待触发)，必须如下配置：

- 向上计数方式：计数器 $CNT < CCR_x \leq ARR$ (特别地， $0 < CCR_x$)，
- 向下计数方式：计数器 $CNT > CCR_x$ 。



例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIMx_CCMR1 寄存器中的 CC2S=01，把 TI2FP2 映像到 TI2。
- 置 TIMx_CCER 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCR 寄存器中的 TS=110，TI2FP2 作为从模式控制器的触发 (TRGI)。
- 置 TIMx_SMCR 寄存器中的 SMS=110 (触发模式)，TI2FP2 被用来启动计数器。

OPM 的波形由写入比较寄存器的数值决定 (要考虑时钟频率和计数器预分频器)

- t_{DELAY} 由 TIMx_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义 ($TIMx_ARR - TIMx_CCR1$)。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形；首先要置 TIMx_CCMR1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要选择地使能预装载寄存器：置 TIMx_CCMR1 中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE；然后在 TIMx_CCR1 寄存器中填写比较值，在 TIMx_ARR 寄存器中填写自动装载值，设置 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P=0。

在这个例子中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应该置低。

因为只需要一个脉冲，所以必须设置 TIMx_CR1 寄存器中的 OPM=1，在下一个更新事件 (当计数器从自动装载值翻转到 0) 时停止计数。

13.3.15.1 特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。

如果要以最小延时输出波形，可以设置 TIMx_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF (和 OCx) 直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在信道配置为 PWM1 和 PWM2 模式时起作用。

13.3.16 编码器接口模式

选择编码器接口模式的方法是：如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。

通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看表 73，假定计数器已经启动 (TIMx_CR1 寄存器中的 CEN=1)，则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端 (TI1 或者 TI2) 的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数 (根据方向，或是 0 到 ARR 计数，或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIMx_ARR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表 73 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

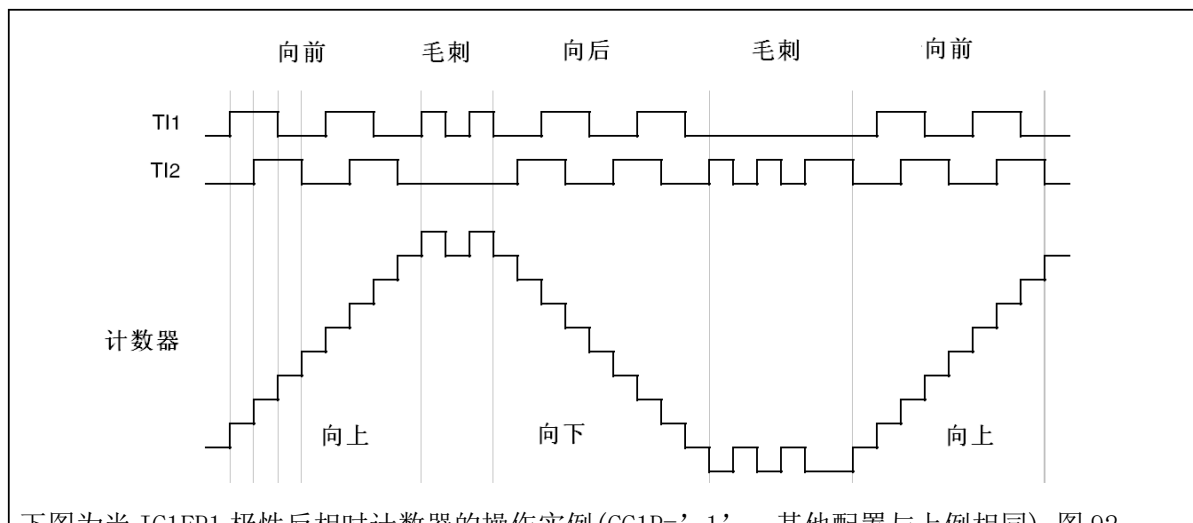
一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

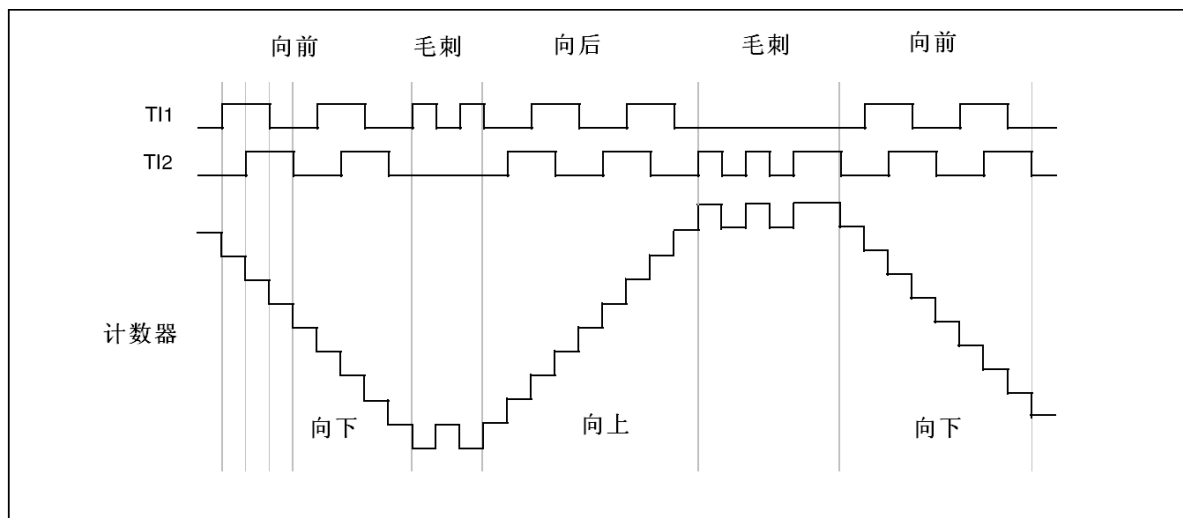
- CC1S='01' (TIMx_CCMR1 寄存器, IC1FP1 映射到 TI1)
- CC2S='01' (TIMx_CCMR2 寄存器, IC2FP2 映射到 TI2)
- CC1P='0' (TIMx_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- CC2P='0' (TIMx_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)

- SMS='011' (TIMx_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效).
- CEN='1' (TIMx_CR1 寄存器, 计数器使能)

图 91 编码器模式下的计数器操作实例



下图为当 IC1FP1 极性反相时计数器的操作实例 (CC1P='1', 其他配置与上例相同) 图 92
IC1FP1 反相的编码器接口模式实例



当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息(速度，加速度，减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器(捕获信号必须是周期的并且可以由另一个定时器产生)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

13.3.17 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。下节 [13.3.18](#) 给出了此特性用于连接霍尔传感器的例子。

13.3.18 TIMx 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

13.3.18.1 从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器 (TIMx_ARR, TIMx_CCRx) 都被更新了。

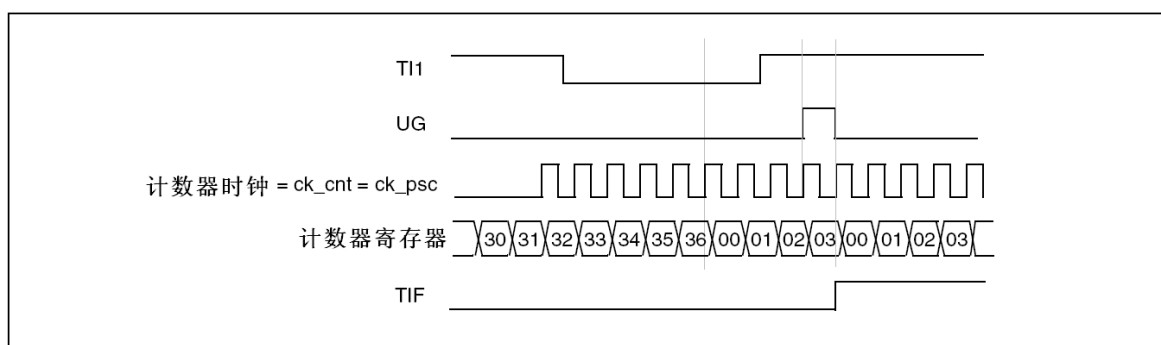
在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置信道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽 (在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=0 以确定极性 (只检测上升沿)。
- 置 TIMx_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志 (TIMx_SR 寄存器中的 TIF 位) 被设置，根据 TIMx_DIER 寄存器中 TIE (中断使能) 位和 TDE (DMA 使能) 位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

图 94 复位模式下的控制电路



13.3.18.2 从模式：门控模式

按照选中的输入端电平使能计数器。在如下的例子中，计数器只在 TI1 为低时向上计数：

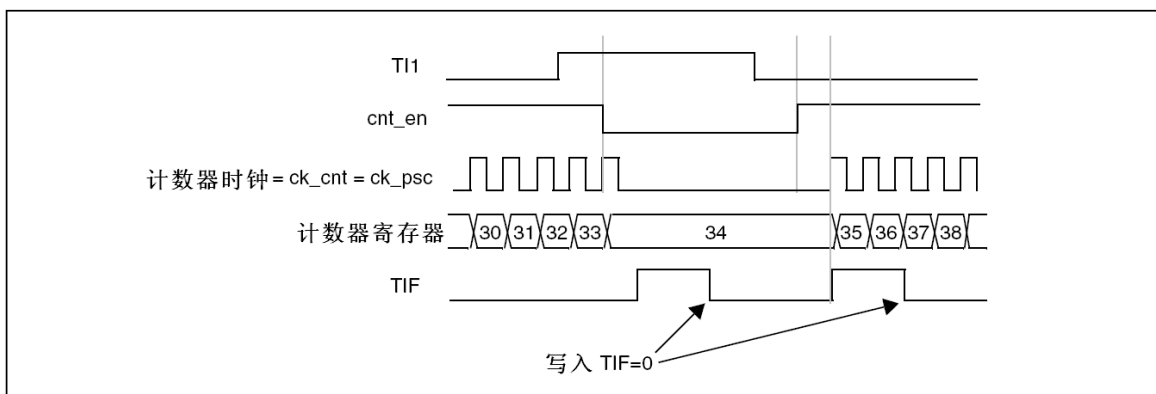
- 配置信道 1 以检测 TI1 上的低电平。配置输入滤波器带宽 (本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输

入捕获源，置 TIMx_CCMR1 寄存器中 CC1S=01。置 TIMx_CCER 寄存器中 CC1P=1 以确定极性（只检测低电平）。

- 置 TIMx_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx_SR 中的 TIF 标志。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。图 95 门控模式下的控制电路



13.3.18.3 从模式：触发模式

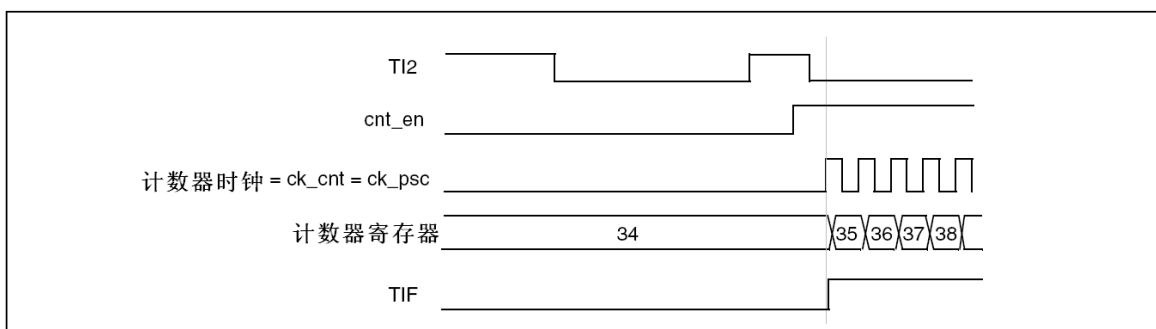
输入端上选中的事件使能计数器。在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置信道 2 检测 TI2 的上升沿。配置输入滤波器带宽（本例中，不需要任何滤波器，保持 IC2F=0000）。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIMx_CCMR1 寄存器中 CC2S=01。置 TIMx_CCER 寄存器中 CC2P=1 以确定极性（只检测低电平）。

- 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIMx_SMCR 寄存器中

TS=110，选择 TI2 作为输入源。当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 96 触发器模式下的控制电路



13.3.18.4 从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。

入。不建议使用 TIMx_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次：

1. 通过 TIMx_SMCR 寄存器配置外部触发输入电路：

- 1) ETF=0000：没有滤波
- 2) ETPS=00：不用预分频器
- 3) ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。

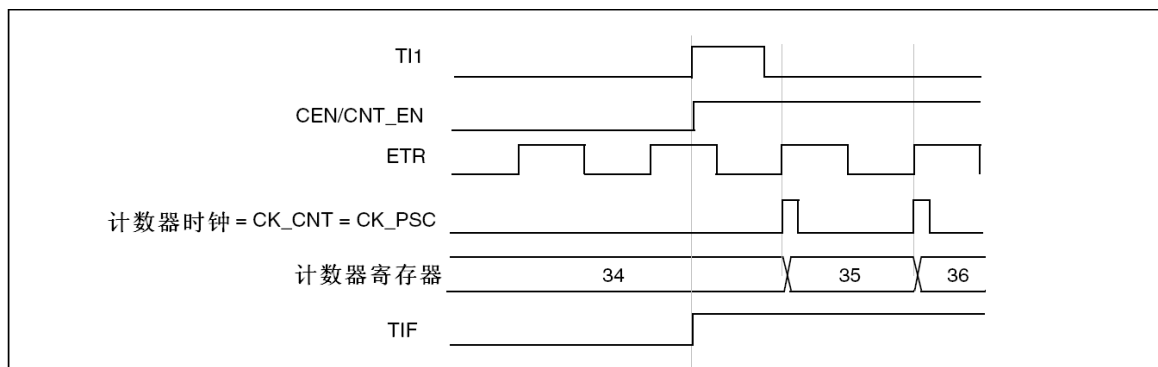
2. 按如下配置信道 1，检测 TI 的上升沿：

- 1) IC1F=0000：没有滤波
- 2) 触发操作中不使用捕获预分频器，不需要配置
- 3) 置 TIMx_CCMR1 寄存器中 CC1S=01，选择输入捕获源
- 4) 置 TIMx_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)

3. 置 TIMx_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

图 97 外部时钟模式 2+触发模式下的控制电路



13.4 TIM8 寄存器说明

13.4.1 控制寄存器(TIM8_CR1)

TIM8_CR1 (控制寄存器)			基地址: 0x40007000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X						CKD	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
Write:								
Reset:	0	0	0	0	0	0	0	0

位15:10	保留, 始终读为0。
CKD[1:0]	时钟分频因子 (Clock division) 这2位定义在定时器时钟 (CK_INT) 频率、死区时间和由死区发生器与数字滤波器 (ETR, TIx) 所用的采样时钟之间的分频比例。 00: $t_{DTS} = t_{CK_INT}$ 01: $t_{DTS} = 2 \times t_{CK_INT}$ 10: $t_{DTS} = 4 \times t_{CK_INT}$ 11: 保留, 不要使用这个配置
ARPE	自动重载预装载允许位 (Auto-reload preload enable) 0: TIMx_ARR 寄存器没有缓冲; 1: TIMx_ARR 寄存器被装入缓冲器。
CMS[1:0]	选择中央对齐模式 (Center-aligned mode selection) 00: 边沿对齐模式。计数器依据方向位 (DIR) 向上或向下计数。 01: 中央对齐模式1。计数器交替地向上和向下计数。配置为输出的信道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位, 只在计数器向下计数时被设置。 10: 中央对齐模式2。计数器交替地向上和向下计数。配置为输出的信道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位, 只在计数器向上计数时被设置。 11: 中央对齐模式3。计数器交替地向上和向下计数。配置为输出的信道 (TIMx_CCMRx 寄存器中 CCxS=00) 的输出比较中断标志位, 在计数器向上和向下计数时均被设置。 注: 在计数器开启时 (CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。
DIR	方向 (Direction) 0: 计数器向上计数; 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。
OPM	单脉冲模式 (One pulse mode) 0: 在发生更新事件时, 计数器不停止; 1: 在发生下一次更新事件 (清除 CEN 位) 时, 计数器停止。

URS	更新请求源 (Update request source) 软件通过该位选择UEV事件的源 0: 如果使能了更新中断或DMA请求, 则下述任一事件产生更新中断或DMA请求: - 计数器溢出/下溢 - 设置UG位 - 从模式控制器产生的更新 1: 如果使能了更新中断或DMA请求, 则只有计数器溢出/下溢才产生更新中断或DMA请求。
UDIS	禁止更新 (Update disable) 软件通过该位允许/禁止UEV事件的产生 0: 允许UEV。更新 (UEV) 事件由下述任一事件产生: - 计数器溢出/下溢 - 设置UG位 - 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(译注: 更新影子寄存器) 1: 禁止UEV。不产生更新事件, 影子寄存器 (ARR、PSC、CCR_x) 保持它们的值。如果设置了UG位或从模式控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。
CEN	使能计数器 (Counter enable) 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了CEN位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CEN位。

13.4.2 控制寄存器 2(TIM8_CR2)

TIM8_CR2 (控制寄存器 2)			基地址: 0x40007000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	OIS5	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	TI1S	MMS[2:0]			CCDS	CCUS	X	CCPC
Write:								
Reset:	0	0	0	0	0	0	0	0

OIS5	输出空闲状态5 (OC5输出)。参见OIS1位。
OIS4	输出空闲状态4 (OC4输出)。参见OIS1位。
OIS3N	输出空闲状态3 (OC3N输出)。参见OIS1N位。
OIS3	输出空闲状态3 (OC3输出)。参见OIS1位。
OIS2N	输出空闲状态2 (OC2N输出)。参见OIS1N位。
OIS2	输出空闲状态2 (OC2输出)。参见OIS1位。
OIS1N	输出空闲状态1 (OC1N输出) (Output Idle state 1) 0: 当MOE=0时, 死区后OC1N=0; 1: 当MOE=0时, 死区后OC1N=1。 注: 已经设置了LOCK (TIM_x_BKR寄存器) 级别1、2或3后, 该位不能被修改。

OIS1	输出空闲状态1(OC1输出) (Output Idle state 1) 0: 当MOE=0时, 如果实现了OC1N, 则死区后OC1=0; 1: 当MOE=0时, 如果实现了OC1N, 则死区后OC1=1。 注: 已经设置了LOCK(TIMx_BKR寄存器)级别1、2或3后, 该位不能被修改。
TI1S	TI1选择(TI1 selection) 0: TIMx_CH1引脚连到TI1输入; 1: TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。

MMS[2:0]	模式选择(Master mode selection) 这3位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000: 复位 - TIMx_EGR寄存器的UG位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则TRGO上的信号相对实际的复位会有一个延迟。 001: 使能 - 计数器使能信号CNT_EN被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过CEN控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO上会有一个延迟，除非选择了主/从模式(见TIMx_SMCR寄存器中MSM位的描述)。 010: 更新 - 更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲 - 在发生一次捕获或一次比较成功时，当要设置CC1IF标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)。 100: 比较 - OC1REF信号被用于作为触发输出(TRGO)。 101: 比较 - OC2REF信号被用于作为触发输出(TRGO)。 110: 比较 - OC3REF信号被用于作为触发输出(TRGO)。 111: 比较 - OC4REF信号被用于作为触发输出(TRGO)。
CCDS	捕获/比较的DMA选择(Capture/compare DMA selection) 0: 当发生CCx事件时，送出CCx的DMA请求； 1: 当发生更新事件时，送出CCx的DMA请求。
CCUS	捕获/比较控制更新选择(Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1)，只能通过设置COM位更新它们； 1: 如果捕获/比较控制位是预装载的(CCPC=1)，可以通过设置COM位或TRGI上的一个上升沿更新它们。 注：该位只对具有互补输出的通道起作用。
CCPC	捕获/比较预装载控制位(Capture/compare preloaded control) 0: CCxE, CCxNE和OCxM位不是预装载的； 1: CCxE, CCxNE和OCxM位是预装载的；设置该位后，它们只在设置了COM位后被更新。 注：该位只对具有互补输出的通道起作用。

13.4.3 控制寄存器(TIM8_SMCR)

TIM8_SMCR (控制寄存器)			基地址: 0x40007000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	ETP	ECE	ETPS[1:0]		ETF[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	MSM	TS[2:0]			X	SMS[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0

ETP	外部触发极性 (External trigger polarity) 该位选择是用ETR还是ETR的反相来作为触发操作 0: ETR不反相, 高电平或上升沿有效; 1: ETR被反相, 低电平或下降沿有效。
ECE	外部时钟使能位 (External clock enable) 该位启用外部时钟模式2 0: 禁止外部时钟模式2; 1: 使能外部时钟模式2。计数器由ETRF信号上的任意有效边沿驱动。 注1: 设置ECE位与选择外部时钟模式1并将TRGI连到ETRF (SMS=111和TS=111) 具有相同功效。 注2: 下述从模式可以与外部时钟模式2同时使用: 复位模式, 门控模式和触发模式; 但是, 这时TRGI不能连到ETRF (TS位不能是'111')。 注3: 外部时钟模式1和外部时钟模式2同时被使能时, 外部时钟的输入是ETRF。
ETPS[1:0]	外部触发预分频 (External trigger prescaler) 外部触发信号ETRP的频率必须最多是TIMxCLK频率的1/4。当输入较快的外部时钟时, 可以使用预分频降低ETRP的频率。 00: 关闭预分频; 01: ETRP频率除以2; 10: ETRP频率除以4; 11: ETRP频率除以8。
ETF[3:0]	外部触发滤波 (External trigger filter) 这些位定义了对ETRP信号采样的频率和对ETRP数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到N个事件后会产生一个输出的跳变。 0000: 无滤波器, 以 f_{DTS} 采样 0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, N=2 0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, N=4 0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, N=8 0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, N=8
MSM	主/从模式 (Master/slave mode) 0: 无作用; 1: 触发输入 (TRGI) 上的事件被延迟了, 以允许在当前定时器 (通过TRGO) 与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
TS[2:0]	触发选择 (Trigger selection) 这3位选择用于同步计数器的触发输入。 000: 内部触发0 (ITR0) 100: TI1的边沿检测器 (TI1F_ED) 001: 内部触发1 (ITR1) 101: 滤波后的定时器输入1 (TI1FP1) 010: 内部触发2 (ITR2) 110: 滤波后的定时器输入2 (TI2FP2) 011: 内部触发3 (ITR3) 111: 外部触发输入 (ETRF) <u>更多有关ITRx的细节, 参见表74。</u> 注: 这些位只能在未用到 (如SMS=000) 时被改变, 以避免在改变时产生错误的边沿检测。

SMS[2:0]	从模式选择(Slave mode selection) 当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器 和控制寄存器的说明) 000: 关闭从模式 - 如果CEN=1, 则预分频器直接由内部时钟驱动。 001: 编码器模式1 - 根据TI1FP1的电平, 计数器在TI2FP2的边沿向上/下计数。 010: 编码器模式2 - 根据TI2FP2的电平, 计数器在TI1FP1的边沿向上/下计数。 011: 编码器模式3 - 根据另一个信号的输入电平, 计数器在TI1FP1和TI2FP2的边沿向上/下计数。 100: 复位模式 - 选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。 101: 门控模式 - 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。 110: 触发模式 - 计数器在触发输入TRGI的上升沿启动(但不复位), 只有计数器的启动是受控的。 111: 外部时钟模式1 - 选中的触发输入(TRGI)的上升沿驱动计数器。 注: 如果TI1F_EN被选为触发输入(TS=100)时, 不要使用门控模式。这是因为, TI1F_ED在每次TI1F变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。
----------	--

13.4.4 控制寄存器(TIM8_DIER)

TIM8_DIER (控制寄存器)			基地址: 0x40007000 偏移地址: 0CH					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X							CC5DE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	CC5IE	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
Write:								
Reset:	0	0	0	0	0	0	0	0

CC5DE	允许捕获/比较5的DMA请求(Capture/Compare 5 DMA request enable) 0: 禁止捕获/比较5的DMA请求; 1: 允许捕获/比较5的DMA请求。
CC5IE	允许捕获/比较5中断(Capture/Compare 5 interrupt enable) 0: 禁止捕获/比较5中断; 1: 允许捕获/比较5中断。
TDE	允许触发DMA请求(Trigger DMA request enable) 0: 禁止触发DMA请求; 1: 允许触发DMA请求。
COMDE	允许COM的DMA请求(COM DMA request enable) 0: 禁止COM的DMA请求; 1: 允许COM的DMA请求。

CC4DE	允许捕获/比较4的DMA请求 (Capture/Compare 4 DMA request enable) 0: 禁止捕获/比较4的DMA请求; 1: 允许捕获/比较4的DMA请求。
CC3DE	允许捕获/比较3的DMA请求 (Capture/Compare 3 DMA request enable) 0: 禁止捕获/比较3的DMA请求; 1: 允许捕获/比较3的DMA请求。
CC2DE	允许捕获/比较2的DMA请求 (Capture/Compare 2 DMA request enable) 0: 禁止捕获/比较2的DMA请求; 1: 允许捕获/比较2的DMA请求。
CC1DE	允许捕获/比较1的DMA请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较1的DMA请求; 1: 允许捕获/比较1的DMA请求。
UDE	允许更新的DMA请求 (Update DMA request enable) 0: 禁止更新的DMA请求; 1: 允许更新的DMA请求。
BIE	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。
TIE	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
COMIE	允许COM中断 (COM interrupt enable) 0: 禁止COM中断; 1: 允许COM中断。
CC4IE	允许捕获/比较4中断 (Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。
CC3IE	允许捕获/比较3中断 (Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。
CC2IE	允许捕获/比较2中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。
CC1IE	允许捕获/比较1中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。
UIE	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。

13.4.5 状态寄存器(TIM8_SR)

TIM8_SR (状态寄存器)			基地址: 0x40007000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	CC5IF	CC50F	CC40F	CC30F	CC20F	CC10F	X

Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
Write:								
Reset:	0	0	0	0	0	0	0	0

CC5IF	捕获/比较5中断标记 (Capture/Compare 5 interrupt flag) 参考CC1IF描述。
CC5OF	捕获/比较5重复捕获标记 (Capture/Compare 5 overcapture flag) 参见CC1OF描述。
CC4OF	捕获/比较4重复捕获标记 (Capture/Compare 4 overcapture flag) 参见CC1OF描述。
CC3OF	捕获/比较3重复捕获标记 (Capture/Compare 3 overcapture flag) 参见CC1OF描述。
CC2OF	捕获/比较2重复捕获标记 (Capture/Compare 2 overcapture flag) 参见CC1OF描述。
CC1OF	捕获/比较1重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当相应的信道被配置为输入捕获时, 该标记可由硬件置1。写0可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到TIMx_CCR1寄存器时, CC1IF的状态已经为' 1' 。
BIF	刹车中断标记 (Break interrupt flag) 一旦刹车输入有效, 由硬件对该位置' 1' 。如果刹车输入无效, 则该位可由软件清' 0' 。 0: 无刹车事件产生; 1: 刹车输入上检测到有效电平。
TIF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在TRGI输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置' 1' 。它由软件清' 0' 。 0: 无触发器事件产生; 1: 触发中断等待响应。
COMIF	COM中断标记 (COM interrupt flag) 一旦产生COM事件 (当捕获/比较控制位: CCxE、CCxNE、OCxM已被更新) 该位由硬件置' 1' 。它由软件清' 0' 。 0: 无COM事件产生; 1: COM中断等待响应。
CC4IF	捕获/比较4中断标记 (Capture/Compare 4 interrupt flag) 参考CC1IF描述。
CC3IF	捕获/比较3中断标记 (Capture/Compare 3 interrupt flag) 参考CC1IF描述。
CC2IF	捕获/比较2中断标记 (Capture/Compare 2 interrupt flag) 参考CC1IF描述。

CC1IF	捕获/比较1中断标记 (Capture/Compare 1 interrupt flag) 如果信道CC1配置为输出模式： 当计数器值与比较值匹配时该位由硬件置1，但在中心对称模式下除外(参考TIMx_CR1寄存器的CMS位)。它由软件清'0'。 0：无匹配发生； 1：TIMx_CNT的值与TIMx_CCR1的值匹配。 当TIMx_CCR1的内容大于TIMx_APR的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1IF位变高 如果信道CC1配置为输入模式： 当捕获事件发生时该位由硬件置'1'，它由软件清'0'或通过读TIMx_CCR1清'0'。 0：无输入捕获产生； 1：计数器值已被捕获(拷贝)至TIMx_CCR1(在IC1上检测到与所选极性相同的边沿)。
UIF	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置'1'： – 若TIMx_CR1寄存器的UDIS=0，当重复计数器数值上溢或下溢时(重复计数器=0时产生更新事件)。 – 若TIMx_CR1寄存器的URS=0、UDIS=0，当设置TIMx_EGR寄存器的UG=1时产生更新事件，通过软件对计数器CNT重新初始化时。 – 若TIMx_CR1寄存器的URS=0、UDIS=0，当计数器CNT被触发事件重新初始化时。(参考13.4.3：TIM8从模式控制寄存器(TIMx_SMCR))。

13.4.6 控制寄存器(TIM8_EGR)

TIM8_EGR (控制寄存器)			基地址: 0x40007000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							CC5G
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
Write:								
Reset:	0	0	0	0	0	0	0	0

CC5G	产生捕获/比较5事件(Capture/Compare 5 generation) 参考CC1G描述。
BG	产生刹车事件(Break generation) 该位由软件置'1'，用于产生一个刹车事件，由硬件自动清'0'。0: 无动作; 1: 产生一个刹车事件。此时MOE=0、BIF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。
TG	产生触发事件(Trigger generation) 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。0: 无动作; 1: TIMx_SR寄存器的TIF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。
COMG	捕获/比较事件，产生控制更新(Capture/Compare control update generation) 该位由软件置'1'，由硬件自动清'0'。 0: 无动作; 1: 当CCPC=1，允许更新CCxE、CCxNE、OCxM位。 注： 该位只对拥有互补输出的通道有效。
CC4G	产生捕获/比较4事件(Capture/Compare 4 generation) 参考CC1G描述。
CC3G	产生捕获/比较3事件(Capture/Compare 3 generation) 参考CC1G描述。
CC2G	产生捕获/比较2事件(Capture/Compare 2 generation) 参考CC1G描述。
CC1G	产生捕获/比较1事件(Capture/Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。0: 无动作; 1: 在通道CC1上产生一个捕获/比较事件： 若信道CC1配置为输出： 设置CC1IF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。 若信道CC1配置为输入： 当前的计数器值被捕获至TIMx_CCR1寄存器；设置CC1IF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。若CC1IF已经为1，则设置CC1OF=1。

UG	产生更新事件(Update generation) 该位由软件置' 1' , 由硬件自动清' 0' 。 0: 无动作; 1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清' 0' (但是预分频系数不变)。若在中心对称模式下或DIR=0(向上计数)则计数器被清' 0' ; 若DIR=1(向下计数)则计数器取TIMx_ARR的值。
----	--

13.4.7 控制寄存器(TIM8_CCMR1)(输出比较模式)

信道可用于输入(捕获模式)或输出(比较模式)，信道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了信道在输出模式下的功能，ICxx 描述了信道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式

TIM8_CCMR1 (控制寄存器)		基地址: 0x40007000 偏移地址: 18H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

OC2CE	输出比较2清0使能
OC2M[2:0]	输出比较2模式
OC2PE	输出比较2预装载使能
OC2FE	输出比较2快速使能
CC2S[1:0]	捕获/比较2选择。(Capture/Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC2信道被配置为输出； 01: CC2信道被配置为输入，IC2映射在TI2上； 10: CC2信道被配置为输入，IC2映射在TI1上； 11: CC2信道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC2S仅在通道关闭时(TIMx_CCER寄存器的CC2E=0)才是可写的。
OC1CE	输出比较1清'0'使能(Output Compare 1 clear enable) 0: OC1REF 不受ETRF输入的影响； 1: 一旦检测到ETRF输入高电平，清除OC1REF=0。
OC1M[2:0]	输出比较1模式(Output Compare 1 mode) 该3位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。OC1REF是高电平有效，而OC1、OC1N的有效电平取决于CC1P、CC1NP位。 000: 冻结。输出比较寄存器TIMx_CCR1与计数器TIMx_CNT间的比较对OC1REF不起作用； 001: 匹配时设置通道1为有效电平。当计数器TIMx_CNT的值与捕获/比较寄存器1(TIMx_CCR1)相同时，强制OC1REF为高。 010: 匹配时设置通道1为无效电平。当计数器TIMx_CNT的值与捕获/比较寄存器1(TIMx_CCR1)相同时，强制OC1REF为低。

	011: 翻转。当TIMx_CCR1=TIMx_CNT时, 翻转OC1REF的电平。 100: 强制为无效电平。强制OC1REF为低。 101: 强制为有效电平。强制OC1REF为高。 110: PWM模式1—在向上计数时, 一旦TIMx_CNT<TIMx_CCR1时通道1为有效电平, 否则为无效电平; 在向下计数时, 一旦TIMx_CNT>TIMx_CCR1时通道1为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。 111: PWM模式2—在向上计数时, 一旦TIMx_CNT<TIMx_CCR1时通道1为无效电平, 否则为有效电平; 在向下计数时, 一旦TIMx_CNT>TIMx_CCR1时通道1为有效电平, 否则为无效电平。 注1: 一旦LOCK级别设为3(TIMx_BDTR寄存器中的LOCK位)并且CC1S=00(该信道配置成输出)则该位不能被修改。 注2: 在PWM模式1或PWM模式2中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时, OC1REF电平才改变。
OC1PE	输出比较1预装载使能(Output Compare 1 preload enable) 0: 禁止TIMx_CCR1寄存器的预装载功能, 可随时写入TIMx_CCR1寄存器, 并且新写入的数值立即起作用。 1: 开启TIMx_CCR1寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1的预装载值在更新事件到来时被加载至当前寄存器中。 注1: 一旦LOCK级别设为3(TIMx_BDTR寄存器中的LOCK位)并且CC1S=00(该信道配置成输出)则该位不能被修改。 注2: 仅在单脉冲模式下(TIMx_CR1寄存器的OPM=1), 可以在未确认预装载寄存器情况下使用PWM模式, 否则其动作不确定。
OC1FE	输出比较1 快速使能(Output Compare 1 fast enable) 该位用于加快CC输出对触发输入事件的响应。 0: 根据计数器与CCR1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCFE只在信道被配置成PWM1或PWM2模式时起作用。
CC1S[1:0]	捕获/比较1 选择。(Capture/Compare 1 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1信道被配置为输出; 01: CC1信道被配置为输入, IC1映射在TI1上; 10: CC1信道被配置为输入, IC1映射在TI2上; 11: CC1信道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC1S仅在通道关闭时(TIMx_CCER寄存器的CC1E=0)才是可写的。

13.4.8 控制寄存器(TIM8_CCMR1)(输入捕获模式)

输入捕获模式

TIM8_CCMR1 (控制寄存器)			基地址: 0x40007000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

	Bit7	6	5	4	3	2	1	Bit0
Read:	IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

IC2F[3:0]	输入捕获2滤波器(Input capture 2 filter)																
IC2PSC[1:0]	输入/捕获2预分频器(Input capture 2 prescaler)																
CC2S[1:0]	捕获/比较2选择(Capture/Compare 2 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2信道被配置为输出; 01: CC2信道被配置为输入, IC2映射在TI2上; 10: CC2信道被配置为输入, IC2映射在TI1上; 11: CC2信道被配置为输入, IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC2S仅在通道关闭时(TIMx_CCER寄存器的CC2E=0)才是可写的。																
IC1F[3:0]	输入捕获1滤波器(Input capture 1 filter) 这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到N个事件后会产生一个输出的跳变: <table border="0"> <tr> <td>0000: 无滤波器, 以f_{DTS}采样</td><td>1000: 采样频率f_{SAMPLING}=f_{DTS}/8, N=6</td></tr> <tr> <td>0001: 采样频率f_{SAMPLING}=f_{CK_INT}, N=2</td><td>1001: 采样频率f_{SAMPLING}=f_{DTS}/8, N=8</td></tr> <tr> <td>0010: 采样频率f_{SAMPLING}=f_{CK_INT}, N=4</td><td>1010: 采样频率f_{SAMPLING}=f_{DTS}/16, N=5</td></tr> <tr> <td>0011: 采样频率f_{SAMPLING}=f_{CK_INT}, N=8</td><td>1011: 采样频率f_{SAMPLING}=f_{DTS}/16, N=6</td></tr> <tr> <td>0100: 采样频率f_{SAMPLING}=f_{DTS}/2, N=6</td><td>1100: 采样频率f_{SAMPLING}=f_{DTS}/16, N=8</td></tr> <tr> <td>0101: 采样频率f_{SAMPLING}=f_{DTS}/2, N=8</td><td>1101: 采样频率f_{SAMPLING}=f_{DTS}/32, N=5</td></tr> <tr> <td>0110: 采样频率f_{SAMPLING}=f_{DTS}/4, N=6</td><td>1110: 采样频率f_{SAMPLING}=f_{DTS}/32, N=6</td></tr> <tr> <td>0111: 采样频率f_{SAMPLING}=f_{DTS}/4, N=8</td><td>1111: 采样频率f_{SAMPLING}=f_{DTS}/32, N=8</td></tr> </table>	0000: 无滤波器, 以f _{DTS} 采样	1000: 采样频率f _{SAMPLING} =f _{DTS} /8, N=6	0001: 采样频率f _{SAMPLING} =f _{CK_INT} , N=2	1001: 采样频率f _{SAMPLING} =f _{DTS} /8, N=8	0010: 采样频率f _{SAMPLING} =f _{CK_INT} , N=4	1010: 采样频率f _{SAMPLING} =f _{DTS} /16, N=5	0011: 采样频率f _{SAMPLING} =f _{CK_INT} , N=8	1011: 采样频率f _{SAMPLING} =f _{DTS} /16, N=6	0100: 采样频率f _{SAMPLING} =f _{DTS} /2, N=6	1100: 采样频率f _{SAMPLING} =f _{DTS} /16, N=8	0101: 采样频率f _{SAMPLING} =f _{DTS} /2, N=8	1101: 采样频率f _{SAMPLING} =f _{DTS} /32, N=5	0110: 采样频率f _{SAMPLING} =f _{DTS} /4, N=6	1110: 采样频率f _{SAMPLING} =f _{DTS} /32, N=6	0111: 采样频率f _{SAMPLING} =f _{DTS} /4, N=8	1111: 采样频率f _{SAMPLING} =f _{DTS} /32, N=8
0000: 无滤波器, 以f _{DTS} 采样	1000: 采样频率f _{SAMPLING} =f _{DTS} /8, N=6																
0001: 采样频率f _{SAMPLING} =f _{CK_INT} , N=2	1001: 采样频率f _{SAMPLING} =f _{DTS} /8, N=8																
0010: 采样频率f _{SAMPLING} =f _{CK_INT} , N=4	1010: 采样频率f _{SAMPLING} =f _{DTS} /16, N=5																
0011: 采样频率f _{SAMPLING} =f _{CK_INT} , N=8	1011: 采样频率f _{SAMPLING} =f _{DTS} /16, N=6																
0100: 采样频率f _{SAMPLING} =f _{DTS} /2, N=6	1100: 采样频率f _{SAMPLING} =f _{DTS} /16, N=8																
0101: 采样频率f _{SAMPLING} =f _{DTS} /2, N=8	1101: 采样频率f _{SAMPLING} =f _{DTS} /32, N=5																
0110: 采样频率f _{SAMPLING} =f _{DTS} /4, N=6	1110: 采样频率f _{SAMPLING} =f _{DTS} /32, N=6																
0111: 采样频率f _{SAMPLING} =f _{DTS} /4, N=8	1111: 采样频率f _{SAMPLING} =f _{DTS} /32, N=8																
IC1PSC[1:0]	输入/捕获1预分频器(Input capture 1 prescaler) 这2位定义了CC1输入(IC1)的预分频系数。一旦CC1E=0(TIMx_CCER寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每2个事件触发一次捕获; 10: 每4个事件触发一次捕获; 11: 每8个事件触发一次捕获。																
CC1S[1:0]	捕获/比较1选择(Capture/Compare 1 Selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1信道被配置为输出; 01: CC1信道被配置为输入, IC1映射在TI1上; 10: CC1信道被配置为输入, IC1映射在TI2上; 11: CC1信道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC1S仅在通道关闭时(TIMx_CCER寄存器的CC1E=0)才是可写的。																

13.4.9 控制寄存器 (TIM8_CCMR2) (输出比较模式)

输出比较模式

TIM8_CCMR2 (控制寄存器)		基地址: 0x40007000		偏移地址: 1CH				
Bit15	14	13	12	11	10	9	Bit8	

Read:	OC4CE	OC4M[2:0]				OC4PE	OC4FE	CC4S[1:0]	
Write:									
Reset:	0	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0	
Read:	OC3CE	OC3M[2:0]				OC3PE	OC3FE	CC3S[1:0]	
Write:									
Reset:	0	0	0	0	0	0	0	0	0

OC4CE	输出比较4清0使能(Output compare 4 clear enable)
OC4M[2:0]	输出比较4模式(Output compare 4 mode)
OC4PE	输出比较4预装载使能(Output compare 4 preload enable)
OC4FE	输出比较4快速使能(Output compare 4 fast enable)
CC4S[1:0]	捕获/比较4选择(Capture/Compare 4 selection) 该2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4信道被配置为输出; 01: CC4信道被配置为输入, IC4映射在TI4上; 10: CC4信道被配置为输入, IC4映射在TI3上; 11: CC4信道被配置为输入, IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC4S仅在通道关闭时(TIMx_CCER寄存器的CC4E=0)才是可写的。
OC3CE	输出比较3清0使能(Output compare 3 clear enable)
OC3M[2:0]	输出比较3模式(Output compare 3 mode)
OC3PE	输出比较3预装载使能(Output compare 3 preload enable)
OC3FE	输出比较3快速使能(Output compare 3 fast enable)
CC3S[1:0]	捕获/比较3选择(Capture/Compare 3 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3信道被配置为输出; 01: CC3信道被配置为输入, IC3映射在TI3上; 10: CC3信道被配置为输入, IC3映射在TI4上; 11: CC3信道被配置为输入, IC3映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC3S仅在通道关闭时(TIMx_CCER寄存器的CC3E=0)才是可写的。

13.4.10 控制寄存器 (TIM8_CCMR2) (输入捕获模式)

输入捕获模式

TIM8_CCMR2 (控制寄存器)		基地址: 0x40007000 偏移地址: 1CH						
	Bit15	14	13	12	11	10	9	Bit8
Read:	IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

IC4F[3:0]	输入捕获4滤波器(Input capture 4 filter)
IC4PSC[1:0]	输入/捕获4预分频器(Input capture 4 prescaler)
CC4S[1:0]	捕获/比较4选择(Capture/Compare 4 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4信道被配置为输出; 01: CC4信道被配置为输入, IC4映射在TI4上; 10: CC4信道被配置为输入, IC4映射在TI3上; 11: CC4信道被配置为输入, IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC4S仅在通道关闭时(TIMx_CCER寄存器的CC4E=0)才是可写的。
IC3F[3:0]	输入捕获3滤波器(Input capture 3 filter)
IC3PSC[1:0]	输入/捕获3预分频器(Input capture 3 prescaler)
CC3S[1:0]	捕获/比较3选择(Capture/compare 3 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3信道被配置为输出; 01: CC3信道被配置为输入, IC3映射在TI3上; 10: CC3信道被配置为输入, IC3映射在TI4上; 11: CC3信道被配置为输入, IC3映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC3S仅在通道关闭时(TIMx_CCER寄存器的CC3E=0)才是可写的。

13.4.11 捕获/比较使能寄存器 (TIM8_CCER)

TIM8_CCER (捕获/比较使能寄存器)			基地址: 0x40007000 偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CC5P	CC5E	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
Write:								
Reset:	0	0	0	0	0	0	0	0

CC5P	输入/捕获5输出极性(Capture/Compare 5 output polarity) 参考CC1P的描述。
CC5E	输入/捕获5输出使能(Capture/Compare 5 output enable) 参考CC1E 的描述。
CC4P	输入/捕获4输出极性(Capture/Compare 4 output polarity) 参考CC1P的描述。
CC4E	输入/捕获4输出使能(Capture/Compare 4 output enable) 参考CC1E 的描述。
CC3NP	输入/捕获3互补输出极性(Capture/Compare 3 complementary output polarity) 参考CC1NP的描述。

CC3NE	输入/捕获3互补输出使能(Capture/Compare 3 complementary output enable) 参考CC1NE的描述。
CC3P	输入/捕获3输出极性(Capture/Compare 3 output polarity) 参考CC1P的描述。
CC3E	输入/捕获3输出使能(Capture/Compare 3 output enable) 参考CC1E 的描述。
CC2NP	输入/捕获2互补输出极性(Capture/Compare 2 complementary output polarity) 参考CC1NP的描述。
CC2NE	输入/捕获2互补输出使能(Capture/Compare 2 complementary output enable) 参考CC1NE的描述。
CC2P	输入/捕获2输出极性(Capture/Compare 2 output polarity) 参考CC1P的描述。
CC2E	输入/捕获2输出使能(Capture/Compare 2 output enable) 参考CC1E的描述。
CC1NP	输入/捕获1互补输出极性(Capture/Compare 1 complementary output polarity) 0: OC1N高电平有效; 1: OC1N低电平有效。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为3或2且CC1S=00(信道配置为输出) 则该位不能被修改。
CC1NE	输入/捕获1互补输出使能(Capture/Compare 1 complementary output enable) 0: 关闭— OC1N禁止输出, 因此OC1N的电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。 1: 开启— OC1N信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1E位的值。
CC1P	输入/捕获1输出极性(Capture/Compare 1 output polarity) CC1信道配置为输出: 0: OC1高电平有效; 1: OC1低电平有效。 CC1信道配置为输入: 该位选择是IC1还是IC1的反相信号作为触发或捕获信号。 0: 不反相: 捕获发生在IC1的上升沿; 当用作外部触发器时, IC1不反相。 1: 反相: 捕获发生在IC1的下降沿; 当用作外部触发器时, IC1反相。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为3或2, 则该位不能被修改。
CC1E	输入/捕获1输出使能 (Capture/Compare 1 output enable) CC1信道配置为输出: 0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。 1: 开启— OC1信号输出到对应的输出引脚, 其输出电平依赖于MOE、OSSI、OSSR、OIS1、OIS1N和CC1NE位的值。 CC1信道配置为输入: 该位决定了计数器的值是否能捕获入TIMx_CCR1寄存器。 0: 捕获禁止; 1: 捕获使能。

表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态 ⁽¹⁾	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态

1	X	0	0	0	输出禁止(与定时器断开) OCx=0, OCx_EN=0	输出禁止(与定时器断开) OCxN=0, OCxN_EN=0
		0	0	1	输出禁止(与定时器断开) OCx=0, OCx_EN=0	OCxREF + 极性, OCxN= OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + 极性, OCx= OCxREF xor CCxP, OCx_EN=1	输出禁止(与定时器断开) OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性+ 死区, OCx_EN=1	OCxREF反相+ 极性+ 死区, OCxN_EN=1
		1	0	0	输出禁止(与定时器断开) OCx=CCxP, OCx_EN=0	输出禁止(与定时器断开) OCxN=CCxNP, OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP, OCx_EN=1	OCxREF + 极性, OCxN= OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性, OCx= OCxREF xor CCxP, OCx_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP, OCxN_EN=0
		1	1	1	OCxREF + 极性+ 死区, OCx_EN=1	OCxREF反相+ 极性+ 死区, OCxN_EN=1
0	0	X	0	0	输出禁止(与定时器断开) 异步地: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0; 若时钟存在: 经过一个死区时间后 OCx=OISx, OCxN=OISxN, 假设OISx与OISxN并不都对应OCx和OCxN的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	关闭状态(输出使能且为无效电平) 异步地: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1; 若时钟存在: 经过一个死区时间后 OCx=OISx, OCxN=OISxN, 假设OISx与OISxN并不都对应OCx和OCxN的有效电平。	
	1		0	1		
	1		1	0		
	1		1	1		

1. 如果一个通道的 2 个输出都没有使用 (CCxE = CCxNE = 0), 那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

注: 引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态, 取决于 OCx 和 OCxN 通道状态和 GPIO 以及 AFIO 寄存器。

13.4.12 计数器 (TIM8_CNT)

TIM8_CNT (计数器)		基地址: 0x40007000 偏移地址: 24H
	Bit15...Bit0	
Read:	CNT[15:0]	
Write:		
Reset:	0	

位	功能描述
CNT[15:0]	计数器的值 (Counter value)

13.4.13 预分频器 (TIM8_PSC)

TIM8_PSC (预分频器)	基地址: 0x40007000 偏移地址: 28H
	Bit15...Bit0
Read:	PSC[15:0]
Write:	
Reset:	0

位	功能描述
PSC[15:0]	预分频器的值 (Prescaler value) 计数器的时钟频率 (CK_CNT) 等于 $f_{CK_PSC} / (PSC[15:0] + 1)$ 。PSC 包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被 TIM_EGR 的 UG 位清 '0' 或被工作在复位模式的从控制器清 '0'。

13.4.14 自动重装载寄存器 (TIM8_ARR)

TIM8_ARR (自动重装载寄存器)	基地址: 0x40007000 偏移地址: 2CH
	Bit15...Bit0
Read:	ARR[15:0]
Write:	
Reset:	0

位	功能描述
ARR[15:0]	自动重装载的值 (Prescaler value) ARR 包含了将要装载入实际的自动重装载寄存器的值。 详细参考 13.3.1 节：有关 ARR 的更新和动作。 当自动重装载的值为空时，计数器不工作。

13.4.15 重复计数寄存器 (TIM8_RCR)

TIM8_RCR (重复计数寄存器)	基地址: 0x40007000 偏移地址: 30H
	Bit15 14 13 12 11 10 9 Bit8
Read:	X
Write:	
Reset:	0 0 0 0 0 0 0 0
	Bit7 6 5 4 3 2 1 Bit0
Read:	REP[7:0]
Write:	
Reset:	0 0 0 0 0 0 0 0

位	功能描述
REP[7:0]	: 重复计数器的值 (Repetition counter value) 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率 (即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。 每次向下计数器REP_CNT达到0, 会产生一个更新事件并且计数器REP_CNT重新从REP值开始计数。由于REP_CNT只有在周期更新事件U_RC发生时才重载REP值, 因此对TIMx_RCR寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在PWM模式中, (REP+1)对应着: — 在边沿对齐模式下, PWM周期的数目; — 在中心对称模式下, PWM半周期的数目;

13.4.16 捕获/比较寄存器 1 (TIM8_CCR1)

TIM8_CCR1 (捕获/比较寄存器 1)	基地址: 0x40007000 偏移地址: 34H
	Bit15...Bit0
Read:	CCR1[15:0]
Write:	
Reset:	0

位	功能描述
CCR1[15:0]]	捕获/比较通道1的值 (Capture/Compare 1 value) 若CC1信道配置为输出: CCR1包含了装入当前捕获/比较1寄存器的值(预装载值)。 如果在TIMx_CCMR1寄存器(OC1PE位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较1寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较, 并在OC1端口上产生输出信号。 若CC1信道配置为输入: CCR1包含了由上一次输入捕获1事件(IC1)传输的计数器值。

13.4.17 捕获/比较寄存器 2 (TIM8_CCR2)

TIM8_CCR2 (捕获/比较寄存器 2)	基地址: 0x40007000 偏移地址: 38H
	Bit15...Bit0
Read:	CCR2[15:0]
Write:	
Reset:	0

位	功能描述
CCR2[15:0]]	捕获/比较通道2的值 (Capture/Compare 2 value) 若CC2信道配置为输出: CCR2包含了装入当前捕获/比较2寄存器的值(预装载值)。 如果在TIMx_CCMR2寄存器(OC2PE位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较2寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较, 并在OC2端口上产生输出信号。 若CC2信道配置为输入: CCR2包含了由上一次输入捕获2事件(IC2)传输的计数器值。

13.4.18 捕获/比较寄存器 3 (TIM8_CCR3)

TIM8_CCR3 (捕获/比较寄存器 3)	基地址: 0x40007000 偏移地址: 3CH
---------------------------	------------------------------

	Bit15...Bit0
Read:	CCR3[15:0]
Write:	
Reset:	0

位	功能描述
CCR3[15:0]	捕获/比较通道3的值 (Capture/Compare3value) 若CC3信道配置为输出： CCR3包含了装入当前捕获/比较3寄存器的值(预装载值)。 如果在TIMx_CCMR3寄存器(OC3PE位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较3寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC3端口上产生输出信号。 若CC3信道配置为输入： CCR3包含了由上一次输入捕获3事件(IC3)传输的计数器值。

13.4.19 捕获/比较寄存器 4 (TIM8_CCR4)

TIM8_CCR4 (捕获/比较寄存器 4)	基地址: 0x40007000 偏移地址: 40H
	Bit15...Bit0
Read:	CCR4[15:0]
Write:	
Reset:	0

位	功能描述
CCR4[15:0]	捕获/比较通道4的值 (Capture/Compare 4 value) 若CC4信道配置为输出： CCR4包含了装入当前捕获/比较4寄存器的值(预装载值)。 如果在TIMx_CCMR4寄存器(OC4PE位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较4寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC4端口上产生输出信号。 若CC4信道配置为输入： CCR4包含了由上一次输入捕获4事件(IC4)传输的计数器值。

13.4.20 刹车和死区寄存器 (TIM8_BDTR)

TIM8_BDTR (刹车和死区寄存器)	基地址: 0x40007000 偏移地址: 44H							
	Bit23	22	21	20	19	18	17	Bit16
Read:	X	X	X	DOE	CMP5_OEP	CMP5_OE	CMP4_OEP	CMP4_OE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0

Read:	DTG[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

注释： 根据锁定设置，AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0]位均可被写保护，有必要在第一 次写入 TIMx_BDTR 寄存器时对它们进行配置。

DOE	MOE置0後,有效 1: 立即輸出空閑狀態,不等死區時間 0: 制車輸入後,等待一個死區時間後輸出空閑狀態
CMP5_OEP	制車輸入CMP5_out极性 (Break polarity) 0: 制車輸入低电平有效; 1: 制車輸入高电平有效。 注：一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为'1'，则该位不能被修改。 注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
CMP5_OE	制車功能CMP5_out使能 (Break enable) 0: 禁止制車輸入 1: 开启制車輸入 注：当设置了LOCK级别1时(TIMx_BDTR寄存器中的LOCK位)，该位不能被修改。 注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
CMP4_OEP	制車輸入CMP4_out极性 (Break polarity) 0: 制車輸入低电平有效; 1: 制車輸入高电平有效。 注：一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为'1'，则该位不能被修改。 注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
CMP4_OE	制車功能CMP4_out使能 (Break enable) 0: 禁止制車輸入 1: 开启制車輸入 注：当设置了LOCK级别1时(TIMx_BDTR寄存器中的LOCK位)，该位不能被修改。 注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
MOE	主輸出使能 (Main output enable) 一旦制車輸入有效,该位被硬件异步清'0'。根据AOE位的设置值,该位可以由软件清'0'或被自动置1。它仅对配置为输出的信道有效。 0: 禁止OC和OCN輸出或强制为空閑状态; 1: 如果设置了相应的使能位(TIMx_CCER寄存器的CCxE、CCxNE位)，则开启OC和OCN輸出。 有关OC/OCN使能的细节,参见13.4.9节, TIM8捕获/比较使能寄存器(TIMx_CCER)。
AOE	自动輸出使能 (Automatic output enable) 0: MOE只能被软件置'1'; 1: MOE能被软件置'1'或在下一个更新事件被自动置'1'(如果制車輸入无效)。 注：一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为'1'，则该位不能被修改。
BKP	制車輸入极性(Break polarity) 0: 制車輸入低电平有效; 1: 制車輸入高电平有效。 注：一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为'1'，则该位不能被修改。 注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
BKE	制車功能使能 (Break enable) 0: 禁止制車輸入 (BRK及CCS时钟失效事件); 1: 开启制車輸入 (BRK及CCS时钟失效事件)。 注：当设置了LOCK级别1时(TIMx_BDTR寄存器中的LOCK位)，该位不能被修改。 注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。

OSSR	运行模式下“关闭状态”选择(Off-state selection for Run mode) 该位用于当MOE=1且通道为互补输出时。没有互补输出的定时器中不存在OSSR位。 参考OC/OCN使能的详细说明(13.4.9节, <u>TIM8捕获/比较使能寄存器(TIMx_CCER)</u>)。 0: 当定时器不工作时, 禁止OC/OCN输出(OC/OCN使能输出信号=0); 1: 当定时器不工作时, 一旦CCxE=1或CCxNE=1, 首先开启OC/OCN并输出无效电平, 然后置OC/OCN使能输出信号=1。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为2, 则该位不能被修改。
OSSI	空闲模式下“关闭状态”选择(Off-state selection for Idle mode) 该位用于当MOE=0且通道设为输出时。 参考OC/OCN使能的详细说明(13.4.9节, <u>TIM8捕获/比较使能寄存器(TIMx_CCER)</u>)。 0: 当定时器不工作时, 禁止OC/OCN输出(OC/OCN使能输出信号=0); 1: 当定时器不工作时, 一旦CCxE=1或CCxNE=1, OC/OCN首先输出其空闲电平, 然后OC/OCN使能输出信号=1。 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为2, 则该位不能被修改。
LOOK[1:0]	锁定设置(Lock configuration) 该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别1, 不能写入TIMx_BDTR寄存器的DTG、BKE、BKP、AOE位和TIMx_CR2寄存器的OISx/OISxN位; 10: 锁定级别2, 不能写入锁定级别1中的各位, 也不能写入CC极位(一旦相关通道通过CCxS位设为输出, CC极位是TIMx_CCER寄存器的CCxP/CCxNP位)以及OSSR/OSSI位; 11: 锁定级别3, 不能写入锁定级别2中的各位, 也不能写入CC控制位(一旦相关通道通过CCxS位设为输出, CC控制位是TIMx_CCMRx寄存器的OCxM/OCxPE位); 注: 在系统复位后, 只能写一次LOCK位, 一旦写入TIMx_BDTR寄存器, 则其内容冻结直至复位。
UTG[7:0]	死区发生器设置(Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设DT表示其持续时间: $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times T_{dtg}, T_{dtg} = T_{DTS}$; $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times T_{dtg}, T_{dtg} = 2 \times T_{DTS}$; $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times T_{dtg}, T_{dtg} = 8 \times T_{DTS}$; $DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times T_{dtg}, T_{dtg} = 16 \times T_{DTS}$; 例: 若 $T_{DTS} = 125ns$ (8MHZ), 可能的死区时间为: 0到15875ns, 若步长时间为125ns; 16us到31750ns, 若步长时间为250ns; 32us到63us, 若步长时间为1us; 64us到126us, 若步长时间为2us; 注: 一旦LOCK级别(TIMx_BDTR寄存器中的LOCK位)设为1、2或3, 则不能修改这些位。

13.4.21 DMA 控制寄存器 (TIM8_DCR)

TIM8_DCR (DMA 控制寄存器)			基地址: 0x40007000 偏移地址: 48H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X			DBL[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X			DBA[4:0]				
Write:								

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

DBL[4:0]	DMA连续传送长度(DMA burst length) 这些位定义了DMA在连续模式下的传送长度(当对TIMx_DMAR寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的次数, 传输可以是半字(双字节)或字节: 00000: 1次传输 00001: 2次传输 00010: 3次传输 10001: 18次传输 例: 我们考虑这样的传输: DBL=7, DBA=TIM2_CR1 - 如果DBL=7, DBA=TIM2_CR1表示待传输数据的地址, 那么传输的地址由下式给出: (TIMx_CR1的地址) + DBA + (DMA索引), 其中DMA索引= DBL 其中(TIMx_CR1的地址) + DBA再加上7, 给出了将要写入或者读出数据的地址, 这样数据的传输将发生在从地址(TIMx_CR1的地址) + DBA开始的7个寄存器。 根据DMA数据长度的设置, 可能发生以下情况: - 如果设置数据为半字(16位), 那么数据就会传输给全部7个寄存器。 - 如果设置数据为字节, 数据仍然会传输给全部7个寄存器: 第一个寄存器包含第一个MSB字节, 第二个寄存器包含第一个LSB字节, 以此类推。因此对于定时器, 用户必须指定由DMA传输的数据宽度。
DBA[4:0]	DMA基地址(DMA base address) 这些位定义了DMA在连续模式下的基地址(当对TIMx_DMAR寄存器进行读或写时), DBA定义为从TIMx_CR1寄存器所在地址开始的偏移量: 00000: TIMx_CR1, 00001: TIMx_CR2, 00010: TIMx_SMCR,

13.4.22 连续模式的 DMA 地址 (TIM8_DMAR)

TIM8_DMAR (连续模式的 DMA 地址)	基地址: 0x40007000 偏移地址: 4CH
	Bit15...Bit0
Read:	DMAR[15:0]
Write:	
Reset:	0

位	功能描述
DMAB[15:0]	DMA连续传送寄存器 (DMA register for burst accesses) 对TIMx_DMAR寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIMx_CR1地址 + DBA + DMA索引, 其中: “TIMx_CR1地址”是控制寄存器1(TIMx_CR1)所在的地址; “DBA”是TIMx_DCR寄存器中定义的基地址; “DMA索引”是由DMA自动控制的偏移量, 它取决于TIMx_DCR寄存器中定义的DBL。

13.4.23 捕获/比较寄存器 5 (TIM8_CCR5)

TIM8_CCR5 (捕获/比较寄存器 5)	基地址: 0x40007000 偏移地址: 50H
	Bit15...Bit0

Read:	CCR5[15:0]
Write:	
Reset:	0

位	功能描述
CCR5[15:0]	捕获/比较通道5的值 (Capture/Compare 5 value) 若CC5信道配置为输出： CCR5包含了装入当前捕获/比较5寄存器的值 (预装载值)。 如果在TIMx_CCMR4寄存器 (OC4PE位) 中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较5寄存器中。 当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC5端口上产生输出信号。 若CC5信道配置为输入： CCR5包含了由上一次输入捕获5事件 (IC5) 传输的计数器值。

13.4.24 捕获/比较模式寄存器 3 (TIM8_CCMR3) (输出比较模式)

输出比较模式：

TIM8_CCMR3 (捕获/比较模式寄存器 3)			基地址: 0x40007000 偏移地址: 54H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	OC5CE	OC5M[2:0]			OC5PE	OC5FE	CC5S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

OC5CE	输出比较5清0使能 (Output compare 5 clear enable)
OC5M[2:0]	输出比较5模式 (Output compare 5 mode)
OC5PE	输出比较5预装载使能 (Output compare 5 preload enable)
OC5FE	输出比较5快速使能 (Output compare 5 fast enable)
CC5S[1:0]	捕获/比较5选择 (Capture/Compare 5 selection) 该2位定义通道的方向 (输入/输出)，及输入脚的选择： 00: CC5信道被配置为输出； 01: CC5信道被配置为输入，IC5映射在TI4上； 10: CC5信道被配置为输入，IC5映射在TI5上； 11: CC5信道被配置为输入，IC5映射在TRC上。 此模式仅工作在内部触发器输入被选中时 (由TIMx_SMCR寄存器的TS位选择)。 注： CC5S仅在通道关闭时 (TIMx_CCER寄存器的CC5E=0) 才是可写的。

13.4.25 控制寄存器 (TIM8_CCMR3) (输入捕获模式)

输入捕获模式：

TIM8_CCMR3 (控制寄存器)			基地址: 0x40007000 偏移地址: 54H					
	Bit15	14	13	12	11	10	9	Bit8

Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	IC5F[3:0]				IC5PSC[1:0]		CC5S[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

IC5F[3:0]	输入捕获5滤波器(Input capture 5 filter)
IC5PSC[1:0]	输入/捕获5预分频器(Input capture 5 prescaler)
CC5S[1:0]	捕获/比较5选择(Capture/Compare 5 selection) 该2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC5信道被配置为输出; 01: CC5信道被配置为输入, IC5映射在TI4上; 10: CC5信道被配置为输入, IC5映射在TI5上; 11: CC5信道被配置为输入, IC5映射在TRC上。 此模式仅工作在内部触发器输入被选中时(由TIMx_SMCR寄存器的TS位选择)。 注: CC5S仅在通道关闭时(TIMx_CCER寄存器的CC5E=0)才是可写的。

14 SPI 模块

14.1 概述

SPI 模块可以实现在 MCU 和外围设备（包含外部 MCU）之间的全双工同步串行通讯。这里提到的 MCU 或者外围设备必须包含 SPI 模块。包含下列特征：

- 全双工模式
- 三线同步传输
- 主机和从机模式
- 7 种主机波特率
- 从机时钟最高至 $F_{cpu}/4$
- 极性和相位可编程的串行时钟
- 写冲突处理机制
- 8 位数据传输，高字节在前，低字节在后
- 8 位从机选择接口，控制外部从机
- 与主机 CPU 的专用功能寄存器接口
- 无二义端口，标准的 SPI

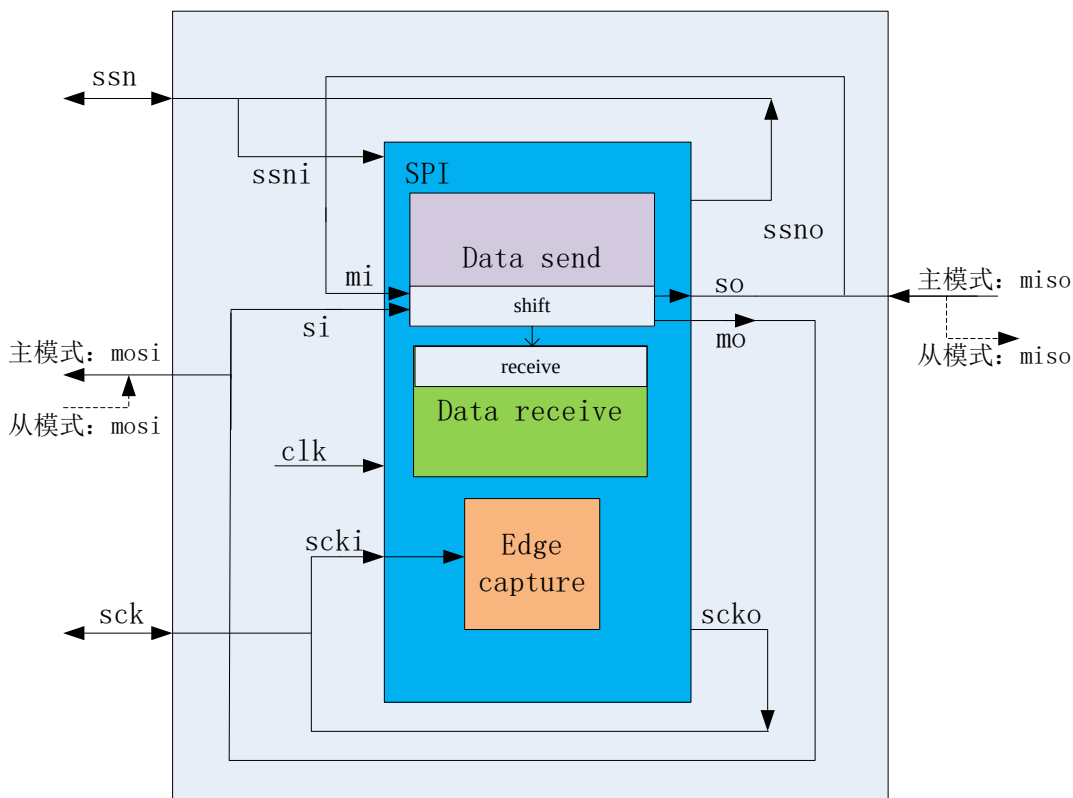
14.2 详细功能说明

串行外设接口 (SPI) 允许芯片与其他设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为从设备提供通信时钟 (SCK)。

14.2.1 SPI 主要特征

- 3 线全双工同步传输
- 主模式或从模式操作
- 7 个主模式频率 (f_{cpu} 的 2/4/8/16/32/64/128 分频)
- 在输入引脚 SPI_CS 上的电平和下降沿侦测
- 可编程的时钟极性和相位
- 可触发中断的专用发送和接收标志

14.2.2 SPI 模块框图



14.2.3 SPI 接口传输格式

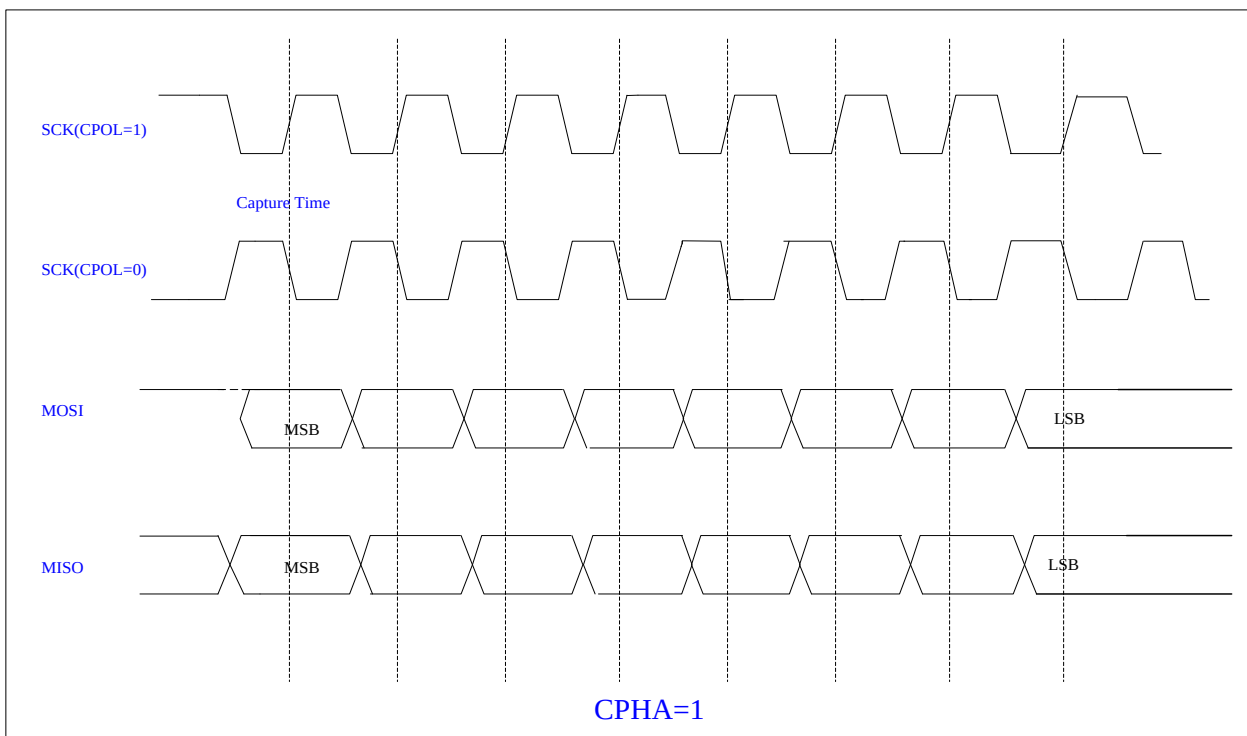
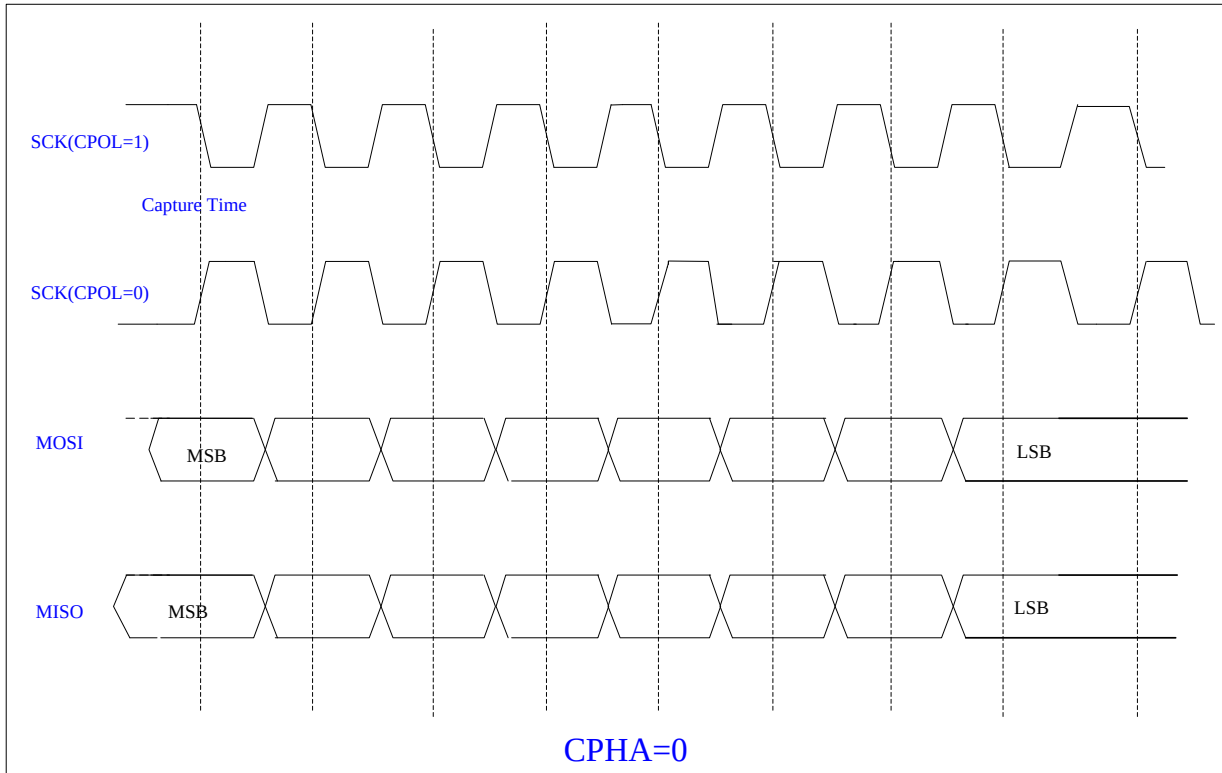
下图显示了数据传输的主要格式。根据 SPI 模块的设置，数据的每一位在主时钟（SCK）的上升沿（CPOL=0）或下降沿（CPOL=1）被传送。数据在主时钟（SCK）的下降沿（CPOL=1）或上升沿（CPOL=0）被接收。这适用于主模式或从模式的传输器/接收器，前提是 SCK 是传输过程中的主时钟。如果 CPHA 被置位，第一位（MSB）将在 SCK 的第一个动态沿时通过 MOSI/MISO 被发送。如果 CPHA 被清零，第一位（MSB）将在 SCK 的第一个动态沿之前半个周期被发送。

除此之外，输入数据在每一位传输一半时被取样，在这个时钟周期的相反的电平上，数据被移位到输出信号 MOSI 上。

14.2.4 主机模式传输格式

SPI 默认为主机模式。

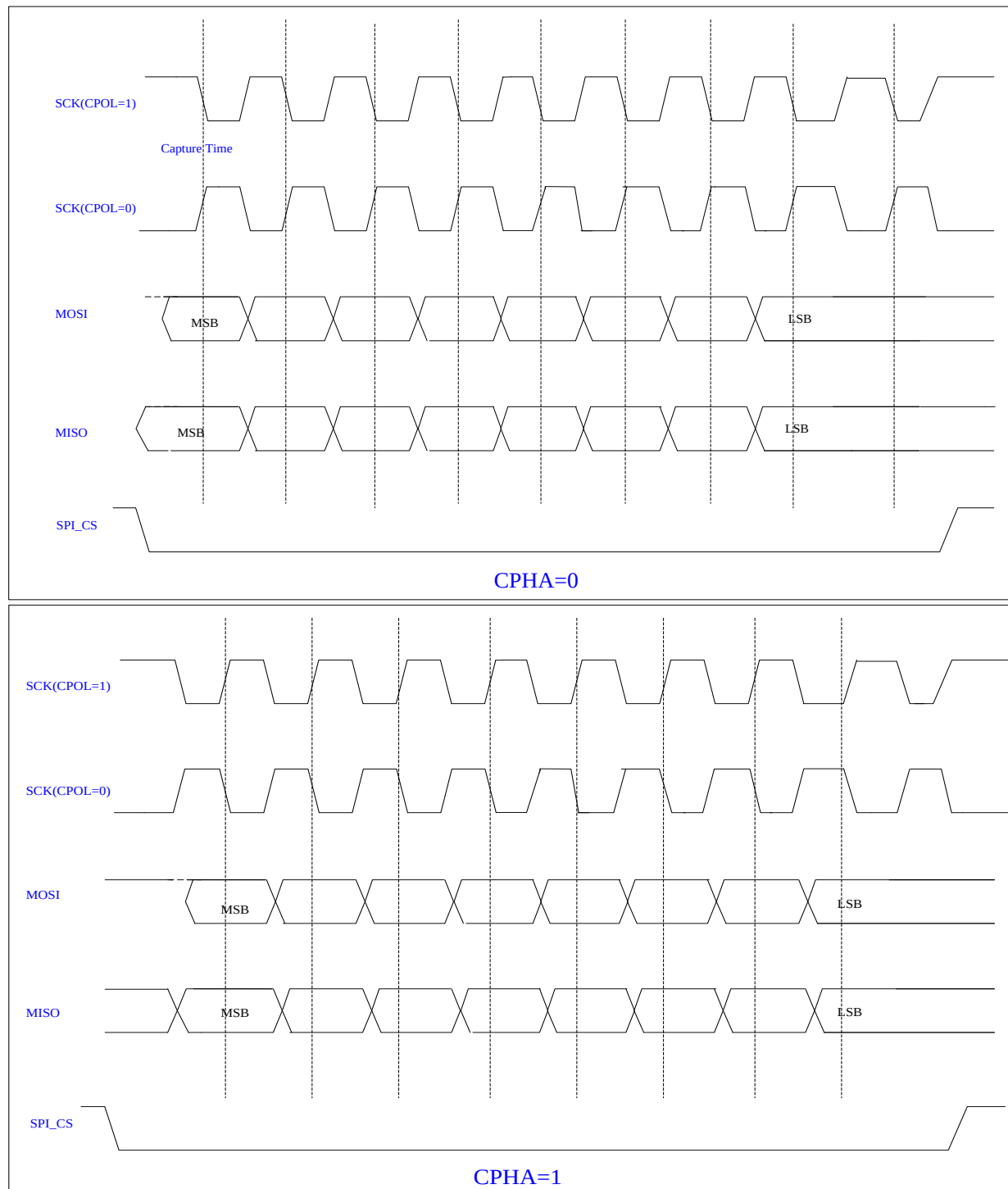
在主机模式中，SPI 等待程序向寄存器 SPIDAT 中写入数据。如果向 SPIDAT 的写入动作完成，传输就开始。在时钟 SCK 的发送沿，数据被移位到输出引脚 MOSI 上。同时，从从机传送过来的另一字节的数据被移位到主机的输入引脚 MISO 上。



主机模式数据传输格式

14.2.5 从机模式传输格式

首先，需要写寄存器 SPICON 中的 MSTR=0，以配置 SPI 进入从机模式。另外配置 SPI_EN=1 以打开 SPI 模块使能。



从机模式数据传输格式

在从机模式中，SPI 等待输入信号 SPI_CS 的低电平，当抓到 SPI_CS 的下降沿，传输开始，直到传输完成，SPI_CS 都需要保持低电平状态。寄存器 SPCON 中 CPHA 的状态决定传输的开始位置，当 CPHA 被清零，从机必须在 SCK 信号的第一个下降沿之前开始传输；当 CPHA 被置位，从机会把 SCK 信号的第一个下降沿做为传输的开始标志。

14.2.6 中断功能

名称	SPI 中断标志描述
SPIF	当传输完成，该标志位被硬件置位
MODF	当 SPI_CS 的状态与主从模式设置有冲突

14.3 特殊功能寄存器列表

SPI0 模块寄存器基地址：0x4000B000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	SPICON	R/W	0x0000	SPI 控制寄存器
0x04	SPISTA	R/W	0x0000	SPI 状态寄存器
0x08	SPIDAT	R/W	0x0000	SPI 数据寄存器
0x0C	SPISSN	R/W	0x00FF	SPI 从机选择寄存器
0x14	CSDLY	R/W	0x0000	SPI cs/delay 控制寄存器

14.3.1 控制寄存器（SPICON）

SPICON (控制寄存器)			基地址：0x4000B000 偏移地址：00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SSDIS	SPR[2:0]			CPHA	CPOL	MSTR	SPI_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SSDIS	SS控制位 0: 在主/从模式中打开SPI_CS输入 1: 在主/从模式中关闭 SPI_CS 输入，该情况下不会产生 MODF 中断请求；在从模式中，若 CPHA=0，则该位无效

SPR[2:0]	SPI时钟速率控制位(主模式时用)			
	SPR[2:0]			SPI时钟速率
	0	0	0	Fsys/2
	0	0	1	Fsys/4
	0	1	0	Fsys/8
	0	1	1	Fsys/16
	1	0	0	Fsys/32
	1	0	1	Fsys/64
	1	1	0	Fsys/128
	1	1	1	不产生主时钟
CPHA	时钟相位 0: 表示高位 (MSB) 将在SCK的第一个动态沿之前半个周期被发送 1: 表示高位 (MSB) 将在SCK的第一个动态沿通过被发送			
CPOL	时钟极性 0: SCK在空闲状态时被设置为低电平 1: SCK在空闲状态时被设置为高电平			
MSTR	SPI模式选择位 0: 从机模式 1: 主机模式			
SPI_EN	SPI使能位 0: 关闭SPI模块 1: 打开 SPI 模块			

14.3.2 状态寄存器 (SPISTA)

SPISTA (状态寄存器)			基地址: 0x4000B000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	SPIF	WCOL	SSERR	MODF
Write:								
Reset:	0	0	0	0	0	0	0	0
位	功能描述							
SPIF	数据传输完成标志位 当传输完成时由硬件置位; 传输过程中由硬件复位, 也可通过读寄存器“SPISTA” SPIDAT来复位。 增加对这位写零清零功能;							
WCOL	写冲突标志位 当写SPIDAT冲突时由硬件置位; 当传输完成无冲突发生时由硬件复位, 也可通过访问寄存器“SPISTA” SPIDAT复位。 增加对这位写零清零功能;							

SSERR	同步从机错误标志位 在接收完成前，SPI_CS被拉高，硬件置位；关闭SPI模块可清除该位（设置spen=0）。 增加对这位写零清零功能；
MODF	模式故障标志位 当SPI_CS引脚状态与设置的模式有冲突时，硬件自动置位；当SPI_CS引脚恢复合适的电平状态时，硬件自动复位；也可以由软件读“SPISTA”寄存器来复位。 增加对这位写零清零功能；

14.3.3 数据寄存器（SPIDAT）

SPIDAT (数据寄存器)			基地址: 0x4000B000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SPIDAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SPIDAT[7:0]	寄存器 SPIDAT 是“接收数据”寄存器的一个读/写缓冲。当向 SPIDAT 中写入数据，是直接写入移位寄存器中（没有传输缓冲）；从 SPIDAT 中读数据，返回的是接收缓冲中的数据，而非移位寄存器。

14.3.4 从机选择寄存器（SPISSN）

SPISSN (从机选择寄存器)			基地址: 0x4000B000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	SSN1	SSN0
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
SSN1	SPIDAT 控制 SPI_CS 使能控制 1: SSN0 控制 SPI_CS 状态，CSDLY 寄存器的 CSB/CSA 控制无效 0: CSDLY 寄存器的 CSB/CSA 控制 SPI_CS 有效，SSN0 控制无效
SSN0	当芯片做 SPI 通讯的主机时，使用 SSN0 控制位可以控制芯片外部 SPI_CS 引脚的高低电平。 在使能 SPI 情况下（且为主机），写 1 拉高 SPI_CS，写 0 拉低 SPI_CS。

14.3.5 SPI cs/delay 控制寄存器（CSDLY）

CSDLY (SPI cs/delay 控制寄存器)			基地址: 0x4000B000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CSB[1:0]		CSA[1:0]		ITDelay[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CSB[1..0]	00: 发送数据前保持CS不变 01: 发送数据前提前2个SPICLK拉低CS（即先拉低CS，等待2个SPICLK延迟后再发送数据） 10: 发送数据前提前4个SPICLK拉低CS 11: 发送数据前提前8个SPICLK拉低CS
CSA[1..0]	00: 发送数据后保持CS不变 01: 发送数据后延迟2个SPICLK拉高CS 10: 发送数据后延迟4个SPICLK拉高CS 11: 发送数据后延迟8个SPICLK拉高CS
ITDelay [3..0]	发送数据完成后等待（Bit11-Bit8）*8 个SPICLK时间给出DMA请求

注：CSB、CSA 控制位的配置，需要先配置 SPISSN→SSN1=0，才能生效。

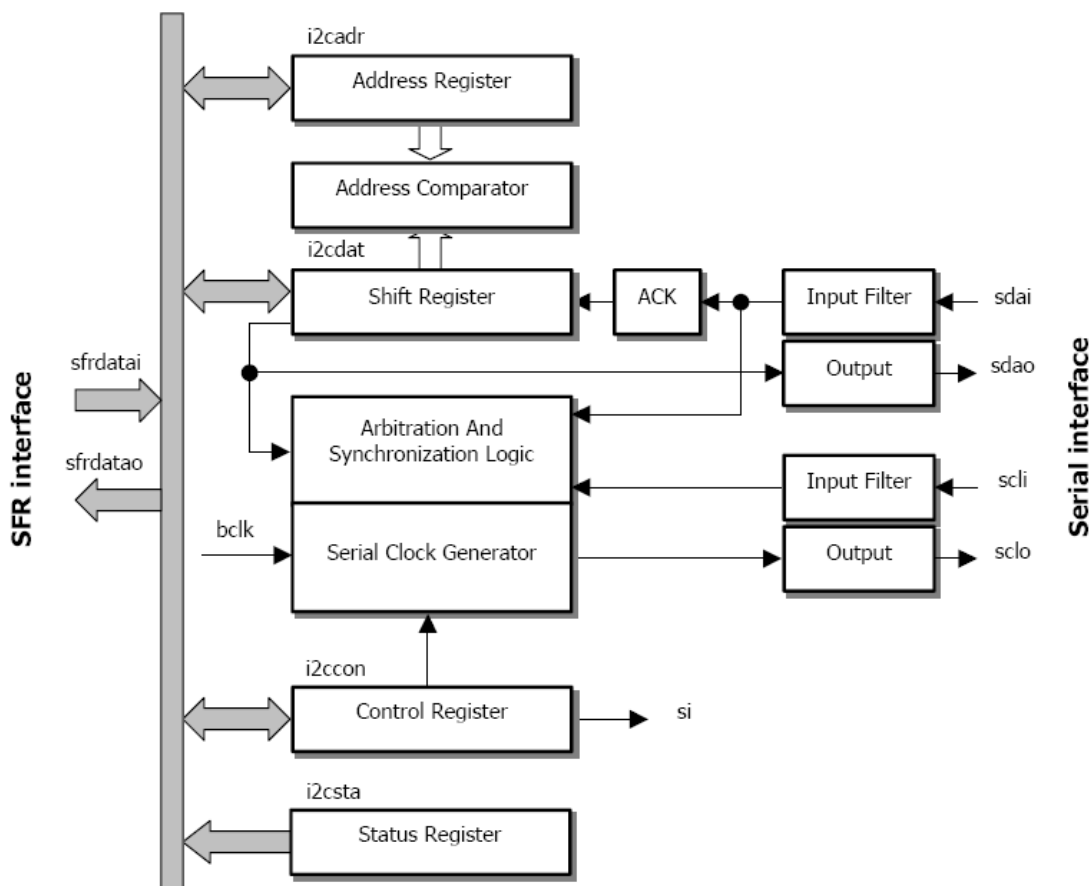
15 I2C 模块

15.1 概述

I2C 模块的芯片引脚为 PD. 8/SCL/TX1 和 PD. 9/SDA/RX1。

I2C 模块提供一个符合 Philips I2C 总线规范的串行接口，用两根线实现设备与总线之间的数据传输，通过状态寄存器 I2CSTA 反映了 I2C 总线控制器的实时状态。

15.2 框图



I2C 模块功能框图

15.3 功能描述

I2C 用两根线实现设备与总线之间的数据传输：串行时钟 SCL 和串行数据 SDA。每一个与总线相连的设备都有一个唯一的地址。I2C 是一个真正的多主机总线，它包含冲突侦测和仲裁机制，以防止多个主机同时开始数据传输时的数据丢失。

15.3.1 操作模式

I2C 数据传输是以 8-bit 进行双向数据传输，标准模式下可达 100kbit/s 的传输速率，快速模式可达 400kbit/s 的速率。它可以下边四种模式工作：

- 主机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输出
- 主机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输出
- 从机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输入
- 从机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输入

15.3.2 串行时钟生成

当 I2C 处于主机模式时，可编程的时钟发生器提供 SCL 时钟；当 I2C 处于从机模式时，时钟发生器被关闭，接收来自主机的时钟。时钟发生器的输出频率可以由寄存器 I2CCON 中的位 CR[9:0] 控制，其中包含 I2CCON[0...1]，I2CCON[8...14]。

15.3.3 中断生成

使能 ENS1，启动 I2C 模块，I2C 模块实时监测 I2C 总线状态，并根据用户设置对总线进行相应的操作及回应。当检测到总线有应用需求情况时，寄存器 I2CCON 中的标志位 SI 会被置位，并将当前应用状态写入状态寄存器 I2CSTA 中。若 I2C 中断使能打开，则产生 I2C 中断。

IIC 产生中断时，寄存器 I2CCON 中的标志位 SI 会被置位。

15.3.4 传输模式

I2C 数据传输是以 8-bit 进行双向数据传输，标准模式下可达 100kbit/s 的传输速率。它可以下边四种模式工作：

- 主机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输出
- 主机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输出
- 从机接收模式：串行数据通过 SDA 输入，串行时钟通过 SCL 输入
- 从机发送模式：串行数据通过 SDA 输出，串行时钟通过 SCL 输入

下面将分别介绍 I2C 通讯的四种主要模式，并对所有可能的状态码进行了描述。下图中有如下缩写：

- S : 开始条件
- Rs : 重复开始条件
- R : 读控制位
- W : 写控制位
- A : 应答位
- $\bar{A}$: 无应答位

DATA : 8 位数据
P : 终止条件
SLA : 从机地址

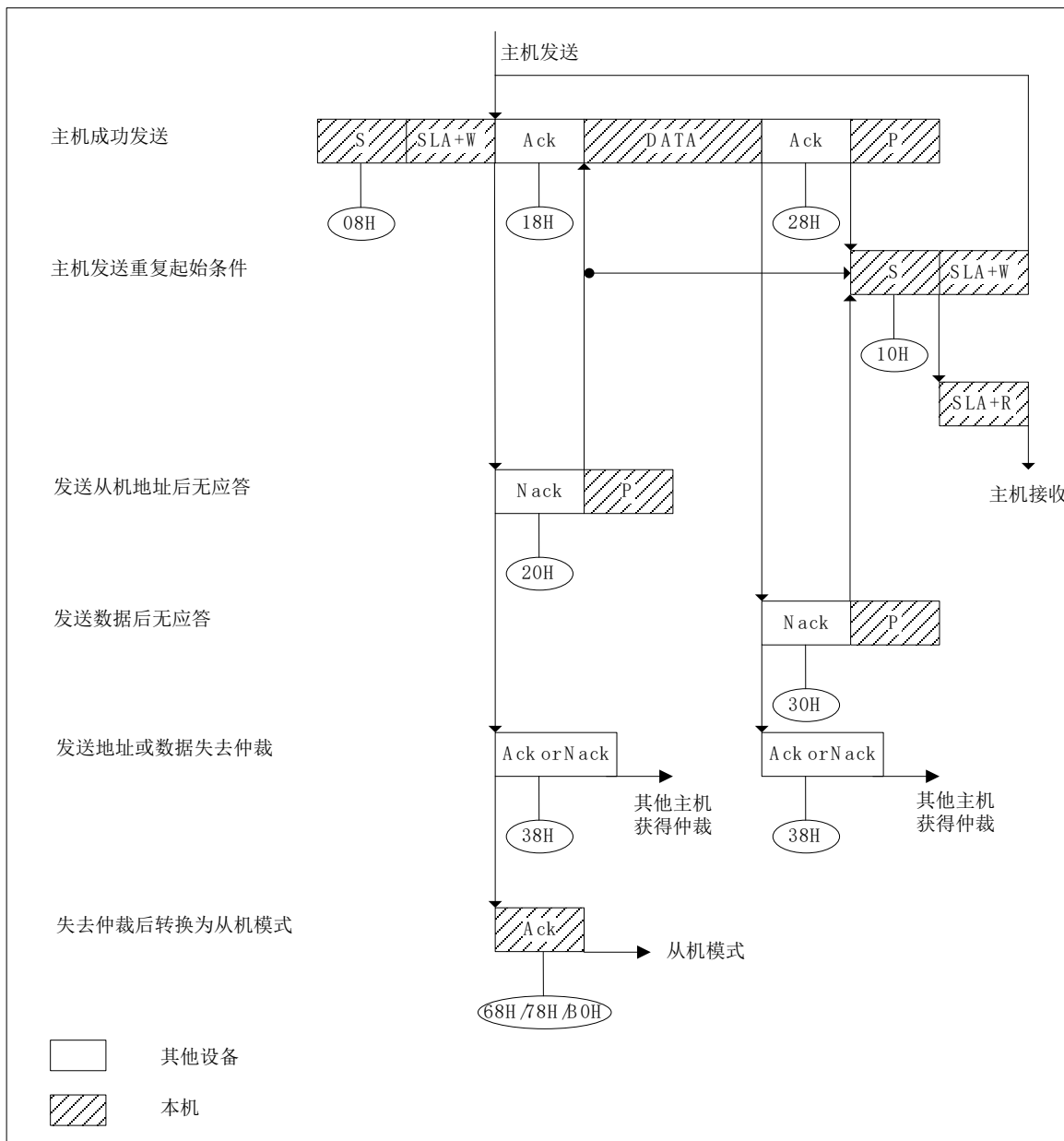
圆形用于表示中断标志已被置起。其中的数字表示当前状态寄存器 I2CSTA 中被掩去低三位的状态码。在 SI 被清除之前，I2C 通讯会暂停，应用软件必须决定是继续通讯还是终止当前传输。对每一个状态码，所需要的软件动作和随后的传输细节均有描述。

I2C 主机发送模式：

主机发送模式中，主机发送一系列数据到从机。一个开始条件(S)，随后一个从机地址(SLA)+写控制字(W)，表示进入主机发送模式。

状态 代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			st a	st o	si	aa	
08H	起始条件 已被发送	加载 SLA+W	X	0	0	X	SLA+W 将被发送 ACK 将被接收
10H	重复起始 条件已被 发送	加载 SLA+W	X	0	0	X	同上
		或者加载 SLA+R	X	0	0	X	SLA+R 将被发送 I2C 将转换为“主接收器”模式
18H	SLA+W 已 被发送； ACK 已被 接收	加载数据字 节	0	0	0	X	数据字节将被发送；ACK 将被接收
		或无动作	1	0	0	X	重复起始条件将被发送
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复 位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止 条件；sto 标志将被复位
20H	SLA+W 已 被发送； “not ACK” 已 被接收	加载数据字 节	0	0	0	X	数据字节将被发送；ACK 将被接收
		或无动作	1	0	0	X	重复起始条件将被发送
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复 位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止 条件 ； sto 标志位将被复位
28H	i2cdat 的 数据字节 已被发 送； ACK 已被 接收	加载数据字 节	0	0	0	X	数据字节将被发送；将发送 ACK 字节
		或无动作	1	0	0	X	重复起始条件将被发送。
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复 位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止 条件 ； sto 标志将被复位
30H	i2cdat 的 数据字节 已被发送	数据字节	0	0	0	X	数据字节将被发送；ACK 将被接收
		或无动作	1	0	0	X	重复起始条件将被发送；
		或无动作	0	1	0	X	终止条件将被发送；sto 标志将被复 位
		或无动作	1	1	0	X	起始条件被发送后将再发送一个终止 条件 ； sto 标志将被复位
38H	SLA+R/W 或数据字	无动作	0	0	0	X	I2C 总线将被释放；将进入“未寻址 从机”状态；
		或无动作	1	0	0	X	当总线空闲时将发送一个起始条件

	节仲裁失败					
--	-------	--	--	--	--	--

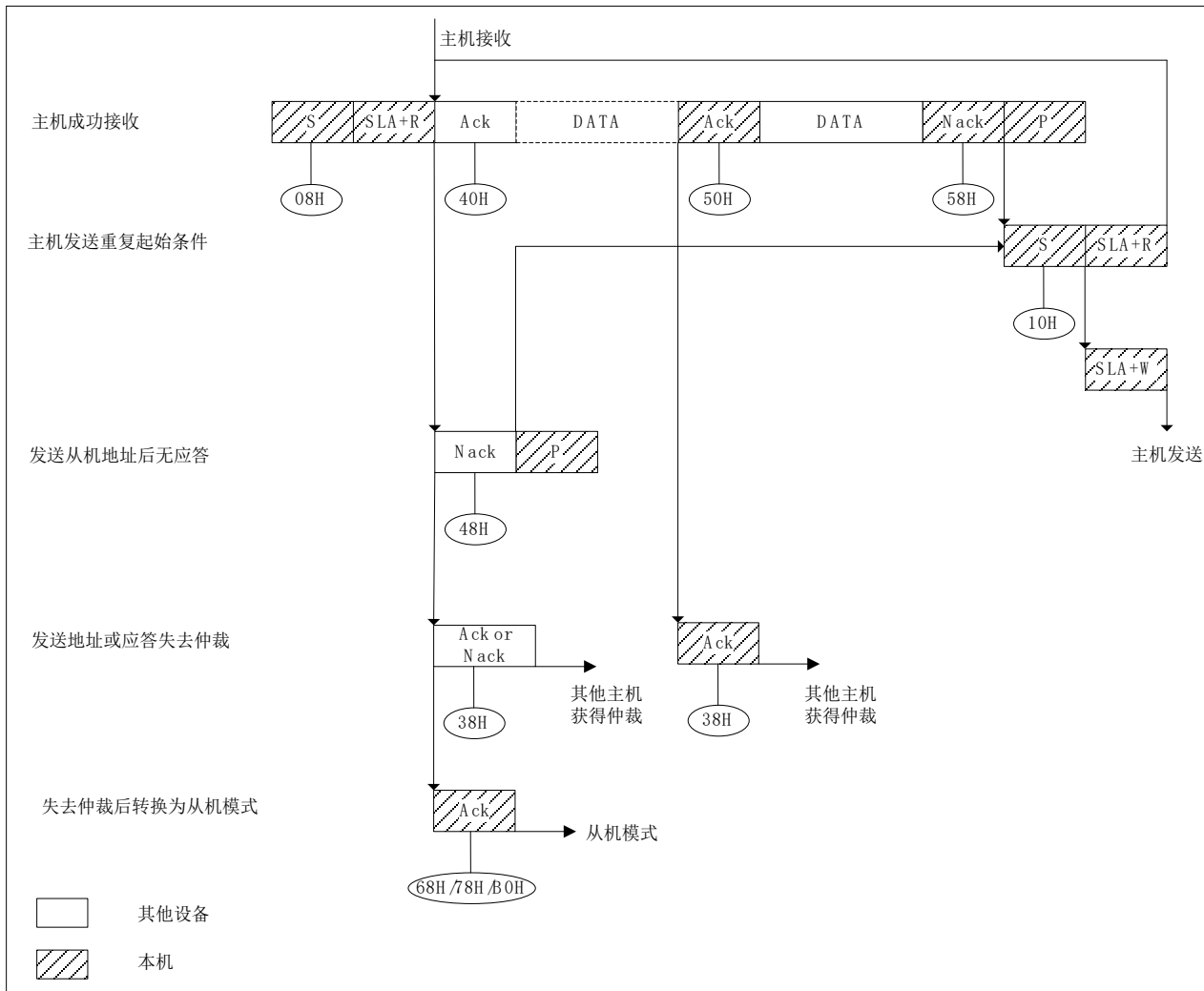


I2C 主机接受模式：

主机接收模式中，主机从从机接收一系列数据。一个开始条件（S），随后一个从机地址（SLA）+读控制字（R）表示进入主机接收模式。

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			st a	st o	si	aa	
08 H	起始条件已被发送	加载 SLA+R	X	0	0	X	SLA+R 将被发送；ACK 将被接收
		加载 SLA+R	X	0	0	X	同上

10 H	重复起始条件已被发送	或者加载 SLA+W	X	0	0	X	SLA+W 将被发送； I2C 将转换为“主接收器”模式
38 H	“not ACK”位 仲裁失败	无动作 或者无动作	0 1	0 0	0 0	X X	I2C 总线将被释放；I2C 将会进入 “从机”模式 当总线空闲时将发送一个起始条件
40 H	SLA+R 已被发送； ACK 已被接收	无动作 或者无动作	0 0	0 0	0 0	0 1	数据字节将被接收；将返回“not ACK” 数据字节将被接收；将返回“not ACK”
48 H	SLA+R 已被发送； “not ACK”已被接收	无动作 或无动作 或无动作	1 0 1	0 1 1	0 0 0	X X X	重复起始条件将被发送 终止条件将被发送；sto 标志将被 复位 起始条件被发送后将再发送一个终 止条件；sto 标志将被复位
50 H	数据字节已被接收； 已返回 ACK	读取数据字节 或读取数据字节	0 0	0 0	0 0	0 1	数据字节将被接收；将返回“not ACK” 数据字节将被接收；将返回 ACK
58 H	数据字节将被接收； 已返回“not ACK”	读取数据字节 或读取数据字节 或读取数据字节	1 0 1	0 1 1	0 0 0	X X X	重复起始条件将被发送 终止条件将被发送；sto 标志将被 复位 起始条件被发送后将再发送一个终 止条件；sto 标志将被复位



I2C 从机接受模式:

从机接收模式中，从机从主机接收一系列数据。

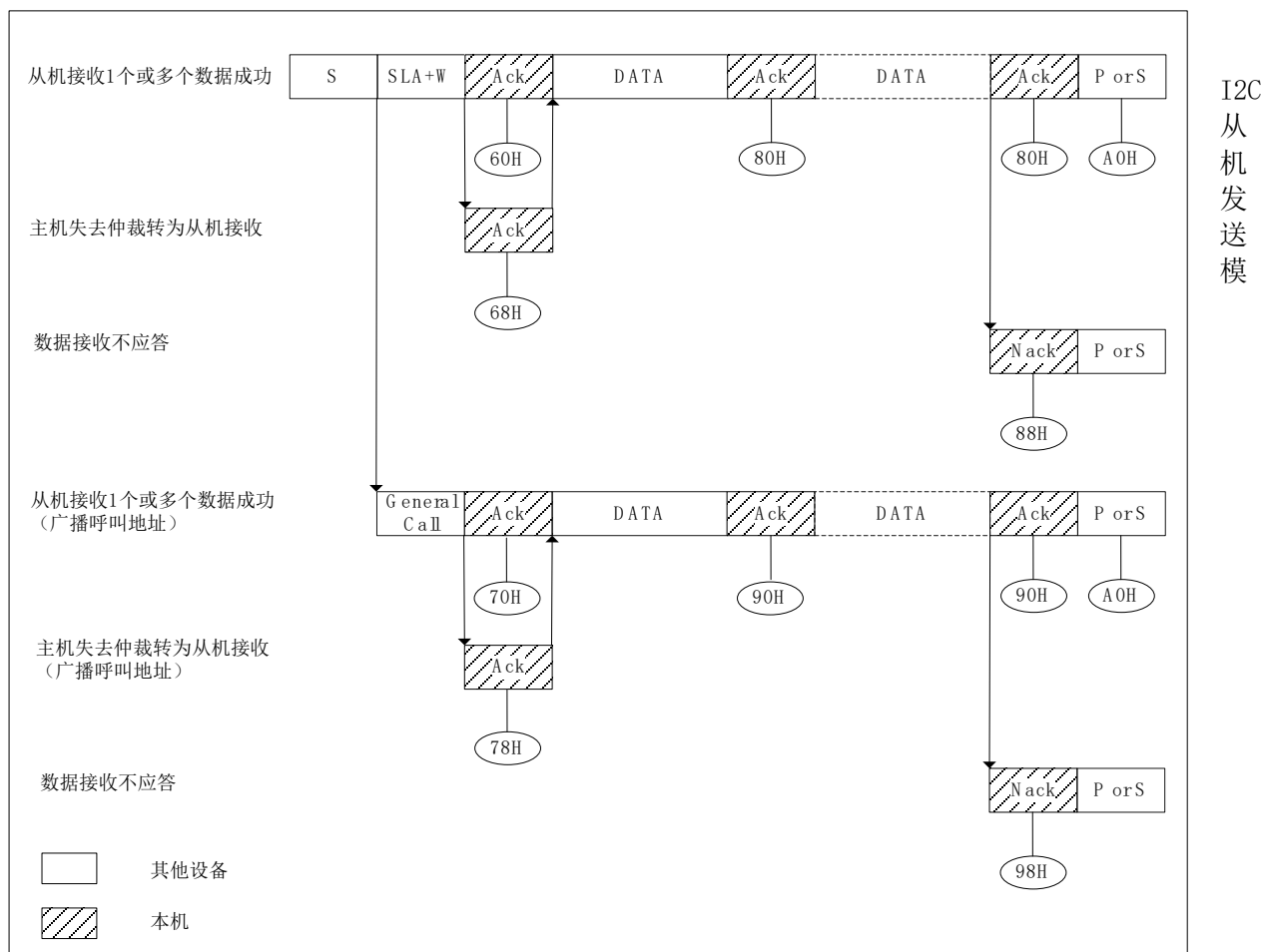
进入从机模式前，需设置从机地址，I2CADR 中 I2CADR[7..1] 位为从机地址。如果 I2CADR[0] 置位，从机也将响应广播呼叫地址 (00H)；否则将不响应广播呼叫地址。

从机模式中，I2C 模块等待总线对本机地址或广播呼叫地址 (如果 I2CADR[0] 被置位) 的寻址。如果读写数据位是‘写’，则 I2C 进入从机接收模式，否则将进入从机发送模式。

地址和读写数据位接受完成后，中断标志 (SI) 置位，状态寄存器 I2CSTA 写入当前状态。

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			st a	st o	si	aa	
60 H	自身的 SLA+W 已被接收； 已返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
68 H	主机 SLA+R/W 仲裁失败；自身的 SLA+W 已被接收，返回 ACK	无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”
		或无动作	X	0	0	1	数据字节将被接收并返回 ACK
70 H		无动作	X	0	0	0	数据字节将被接收并返回 “not ACK”

	呼叫地址 (00H) 已被接收; 已返回 ACK	或无动作	X	0	0	1	数据字节将被接收并返回 ACK
78 H	主机 SLA+R/W 仲裁失败; 呼叫地址已被接收, 返回 ACK	无动作 或无动作	X X	0 0	0 0	0 1	数据字节将被接收并返回 “not ACK” 数据字节将被接收并返回 ACK
80 H	预先写入自身 SLV 地址; DATA 字节已被接收; 返回 ACK	读取数据字节 或者 读取数据字节	X X	0 0	0 0	0 1	数据字节将被接收并返回 “not ACK” 数据字节将被接收并返回 ACK
88 H	预先写入自身 SLA; DATA 字节已被接收; 返回 “not ACK”	读取数据字节 或者 读取数据字节 或者 读取数据字节 或者 读取数据字节	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换为 “未寻址从机” 模式; 不识别自身从机地址或呼叫地址 切换为 “未寻址从机” 模式; 识别自身从机地址或呼叫地址 切换为 “未寻址从机” 模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件 切换为 “未寻址从机” 模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
90 H	预先写入呼叫地址; DATA 字节已被接收; 返回 ACK	读取数据字节 或者 读取数据字节	X X	0 0	0 0	0 1	数据字节将被接收并返回 “not ACK” 数据字节将被接收并返回 ACK
98 H	预先写入呼叫地址; DATA 字节已被接收; 返回 ACK	读取数据字节 或者 读取数据字节 或者 读取数据字节 或者 读取数据字节	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换为 “未寻址从机” 模式; 不识别自身从机地址或呼叫地址 切换为 “未寻址从机” 模式; 识别自身从机地址或呼叫地址 切换为 “未寻址从机” 模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件 切换为 “未寻址从机” 模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
A0 H	终止条件或重复起始条件在被配置为 SLV/REC 或 SLV/TRX 时被接收	无动作 或者 无动作 或者 无动作 或者 无动作	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换为 “未寻址从机” 模式; 不识别自身从机地址或呼叫地址 切换为 “未寻址从机” 模式; 识别自身从机地址或呼叫地址 切换为 “未寻址从机” 模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件 切换为 “未寻址从机” 模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件



式:

从机发送模式中，从机发送一系列数据到主机。

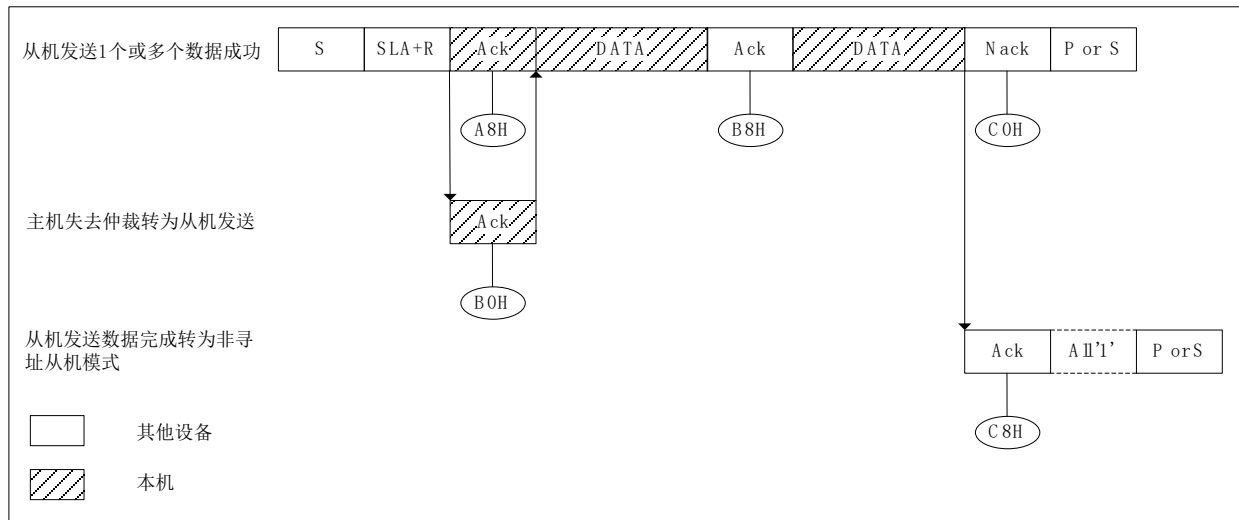
进入从机模式前，需设置从机地址，I2CADR 中 I2CADR[7..1] 位为从机地址。如果 I2CADR[0] 置位，从机也将响应广播呼叫地址 (00H)；否则将不响应广播呼叫地址。

从机模式中，I2C 模块等待总线对本机地址或广播呼叫地址 (如果 I2CADR[0] 被置位) 的寻址。如果读写数据位是‘写’，则 I2C 进入从机接收模式，否则将进入从机发送模式。

地址和读写数据位接受完成后，中断标志 (SI) 置位，状态寄存器 I2CSTA 写入当前状态。

状态代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			st a	st o	si	aa	
A8 H	自身 SLA+R 已被接收；返回 ACK	加载数据字节 或者 加载数据字节	X X	0 0	0 0	0 1	最后一个数据字节将被发送并接收 ACK 数据字节将被发送；ACK 将被接收
B0 H	主机 SLA+R 仲裁失败；自身 SLA+R 已被接收；返回 ACK	加载数据字节 或者 加载数据字节	X X	0 0	0 0	0 1	最后一个数据字节将被发送并接收 ACK 数据字节将被发送；ACK 将被接收
B8 H	数据字节已被发送；ACK 已被接收	加载数据字节 或者 加载数据字节	X X	0 0	0 0	0 1	最后一个数据字节将被发送并接收 ACK 数据字节将被发送；ACK 将被接收

C0 H	数据字节已被发送; “not ACK” 已被接收	无动作	0	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址
		或者无动作	0	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址
		或者无动作	1	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
C8 H	最后一个数据字节已被发送; ACK 已被接收	无动作	0	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址
		或者无动作	0	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址
		或者无动作	1	0	0	0	切换为“未寻址从机”模式; 不识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件
		或者无动作	1	0	0	1	切换为“未寻址从机”模式; 识别自身从机地址或呼叫地址; 当总线空闲时将发送一个起始条件



I2C 复合状态:

状态 代码	I2C 状态	应用程序配置					I2C 硬件响应
		I2CDAT	I2CCON				
			st a	st o	si	aa	
F8H	没有可利用信息的相关状态； si=0	无动作	无动作				等待或继续进行传递
00H	MST 或选择从机模式中的总线错误	无动作	0	1	0	X	只有当被配置为“主机”或“从机”模式时 I2C 硬件才会被触发 在所有情况下，总线将被释放并且 I2C 将切换到

							“未寻址从机”模式。sto 标志将被复位
--	--	--	--	--	--	--	----------------------

15.4 特殊功能寄存器列表

微控制器与 I2C 组件的接口通过以下四个特殊功能寄存器来实现：

I2C 模块寄存器基地址：0x4000A000				
偏移地址	名称	读写方式	复位值	功能描述
00H	I2CDAT	R/W	0000H	I2C数据寄存器
04H	I2CADR	R/W	0000H	I2C地址寄存器
08H	I2CCON	R/W	4000H	I2C控制寄存器
0CH	I2CSTA	R/W	00F8H	I2C状态寄存器

15.5 特殊功能寄存器说明

15.5.1 I2C 数据寄存器（I2CDAT）

I2CDAT (I2C 数据寄存器)			基地址：0x4000A000 偏移地址：00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CDAT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

寄存器I2CDAT是将被传送到总线上的数据，或者是刚从总线上接收到的数据。寄存器I2CDAT没有设置影子寄存器，也没有双缓存，所以当I2C中断发生时，MCU需要及时从它读取数据，以免数据丢失。

15.5.2 地址寄存器（I2CADR）

I2CADR (地址寄存器)			基地址：0x4000A000 偏移地址：04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CADR[7:0]							
Write:								

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
I2CADR[7:1]	I2C从机地址(7位)
I2CADR[0]	呼叫地址确认位 当此位置1时, 呼叫地址可以被识别, 否则不能被识别。

15.5.3 控制寄存器 (I2CCON)

I2CCON (控制寄存器)			基地址: 0x4000A000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	CR[9:3]						
Write:								
Reset:	0	1	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CR2	ENS1	STA	STO	SI	AA	CR[1:0]	
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CR[9:0]	I2C时钟频率控制位 $I2C\ CLOCK = F_{sys} / (CR[9:0] + 1) / 4$
ENS1	I2C 使能位 1: 使能IIC模块; 0: 关闭IIC模块;
STA	开始标志位 1: 检查IIC总线的状态, 如果空闲则生成开始信号; 0: 不会生成开始信号;
STO	停止标志位 1: 当处于主机模式, 则向总线传输停止信号 0: 不向总线传输停止信号;
SI	中断标志位 当进入25种IIC状态之一时, SI由硬件置位, 唯一不置位的状态是“F8H”; 写0清0, 写1无影响。
AA	生成应答标志位 1: 应答在以下情况下被返回: 接收到自身作为从机的地址; gc被置位的情况下接收到地址呼叫; 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成; 0: 非应答在以下情况下被返回: 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成;

15.5.4 状态寄存器 (I2CSTA)

I2CSTA (状态寄存器)			基地址: 0x4000A000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8

Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	I2CSTA[4:0]					X	X	X
Write:								
Reset:	1	1	1	1	1	0	0	0

位	功能描述
I2CSTA[4:0]	I2C状态码

寄存器 I2CSTA 反映 I2C 模块的实时状态。这个寄存器的低三位始终为 0。总共有 26 种可能的状态。当进入 25 种状态的其中一种时，都会产生中断；唯一一种不产生中断的情况是状态 F8H。

在下表中，“SLA”指从机地址，“R”指与从机地址一起传送的读/写位是读，“W”指与从机地址一起传送的读/写位是写。

16 RTC 模块

16.1 概述

RTC 单元提供实时时钟、日历功能，自动闰年调整，支持闹钟和周期性中断。

RTC 模块在各种工作模式下都不会被关闭，在低功耗下仍然正常运行。

RTC 模块的时钟源 LRC 时钟。

16.2 功能描述

- 提供时钟和日历功能：输出寄存器包含秒、分、时、日、月、年和星期
- 具有自动闰年闰月调整功能
- 1 个 RTC 闹钟中断功能
- 2 个定时器周期性中断功能
- 5 个时间中断功能（秒、分、时、日、月）

16.3 时间和万年历

RTC 提供秒、分、时、日、月、年和星期输出寄存器。

通过 RTC 的输出寄存器，可以得到自动闰年校正的万年历功能，其范围从 2000 年 1 月 1 日到 2099 年 12 月 31 日。

16.4 中断功能

RTC 一共提供 8 种中断源，共享 MCU 的 IRQ-RTC 中断向量，向量号 20。RTC 的 8 种中断源由 RTCIE 控制其使能。

具体的中断产生条件和中断清除步骤如下：

ALMF：RTC 闹钟中断标志

当小时和分钟与设定的闹钟匹配时，产生 RTC 闹钟中断，ALMF 被置为 1。

对该位写 0 清标志。

RTC1F：RTC 定时器 1 中断标志

如设置 RTCTMR1=X，使能计数 RTC1EN 后，经过 $(X+1)*1S$ 后，该标志位置位 1。

对该位写 0 清标志。

RTC2F：RTC 定时器 2 中断标志

如设置 RTCTMR2=X，使能计数 RTC2EN 后，经过 $(X+1)*0.0625S$ 后，该标志位置位 1。

对该位写 0 清标志。

MTHF：月中断

月计数器 MONTHR 加 1 时，产生一个月中断，MTHF 被置为 1。
对该位写 0 清标志。

DAYF: 日中断

日期计数器 DAYR 加 1 时，产生一个日中断，DAYF 被置为 1。
对该位写 0 清标志。

HRF: 小时中断

小时计数器 HOURR 加 1 时，产生一个小时中断，HRF 被置为 1。
对该位写 0 清标志。

MINF: 分钟中断

分钟计数器 MINR 加 1 时，产生一个分钟中断，MINF 被置为 1。
对该位写 0 清标志。

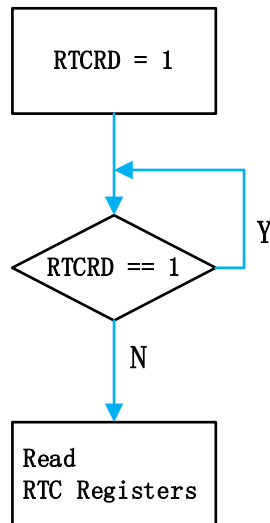
SECF: 秒中断

秒计数器 SECR 加 1 时，产生一个秒中断，SECF 被置为 1。
对该位写 0 清标志。

16.5 RTC 指示寄存器读写流程

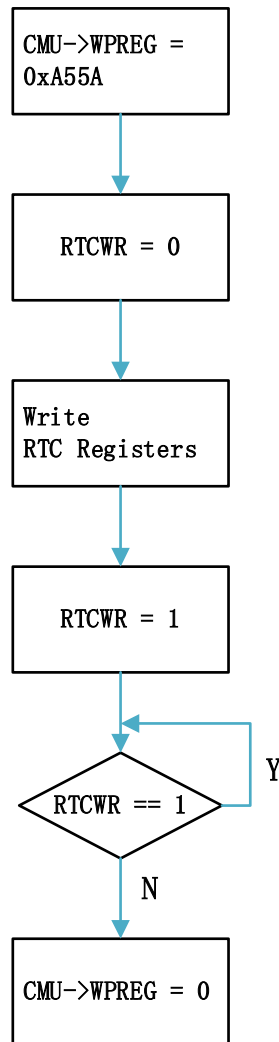
16.5.1 读取 RTC 指示寄存器流程

在用户需要读取 RTC 指示寄存器（SECR, MINR, HOURR, DAYR, MONTHR, YEARR, WEEKR）的时候，用户应该按照以下流程操作：



16.5.2 写入 RTC 指示寄存器流程

在用户需要更新 RTC 指示寄存器（SECR, MINR, HOURR, DAYR, MONTHR, YEARR, WEEKR）的时候，用户应该按照以下流程操作，请一次性写入所有 7 个 RTC 指示寄存器，并且按照年、月、日、时、分、秒、周的顺序操作：



16.6 寄存器复位的相关说明

RTC 模块 RTC 软复位不影响寄存器列表:

RTC 模块寄存器基地址: 0x4000C000				
偏移地址	名称	读写方式	复位值	功能描述
0x18	SECR	R/W	0x0000	秒寄存器 (写保护)
0x1C	MINR	R/W	0x0000	分寄存器 (写保护)
0x20	HOURR	R/W	0x0000	时寄存器 (写保护)
0x24	DAYR	R/W	0x0001	日寄存器 (写保护)
0x28	MONTHR	R/W	0x0001	月寄存器 (写保护)
0x2C	YEARR	R/W	0x0000	年寄存器 (写保护)
0x30	WEEKR	R/W	0x0001	周寄存器 (写保护)

16.7 特殊功能寄存器列表

RTC 模块寄存器基地址: 0x4000C000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	RTCCON	R/W	0x0000	RTC 控制寄存器
0x04	RTCIE	R/W	0x0000	RTC 中断使能寄存器
0x08	RTCIF	R/W	0x0000	RTC 中断标志寄存器
0x0C	ALMR	R/W	0x0000	闹钟寄存器
0x10	RTCTMR1	R/W	0x0000	RTC 定时器 1 计数设置
0x14	RTCTMR2	R/W	0x0000	RTC 定时器 2 计数设置
0x18	SECR	R/W	0x0000	秒寄存器 (写保护)
0x1C	MINR	R/W	0x0000	分寄存器 (写保护)
0x20	HOURLR	R/W	0x0000	时寄存器 (写保护)
0x24	DAYR	R/W	0x0001	日寄存器 (写保护)
0x28	MONTHR	R/W	0x0001	月寄存器 (写保护)
0x2C	YEARR	R/W	0x0000	年寄存器 (写保护)
0x30	WEEKR	R/W	0x0001	周寄存器 (写保护)
0x3C	RTCRD	R/W	0x0000	RTC 读控制寄存器
0x40	RTCWR	R/W	0x0000	RTC 写控制寄存器

16.8 特殊功能寄存器说明

16.8.1 RTC 控制寄存器 (RTCCON)

RTCCON (RTC 控制寄存器)			基地址: 0x4000C000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Reserved	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	RTC2EN	RTC1EN	Reserved	Reserved	Reserved	Reserved	Reserved
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
Bit15	保持为 0
RTC2EN	RTC 定时器 2 使能位 RTC2EN=0: RTC 定时器 2 被关闭 RTC2EN=1: RTC 定时器 2 被使能, 溢出产生 RTC2IF 标志。
RTC1EN	RTC 定时器 1 使能位 RTC1EN=0: RTC 定时器 1 被关闭 RTC1EN=1: RTC 定时器 1 被使能, 溢出产生 RTC1IF 标志。
Bit[4:1]	保持为 0

Bit0	保持为 0
------	-------

16.8.2 RTC 中断使能寄存器 (RTCIE)

RTCIE (RTC 中断使能寄存器)			基地址: 0x4000C000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMIE	RTC2IE	RTC1IE	MTHIE	DAYIE	HRIE	MINIE	SECIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMIE	RTC 闹铃中断使能位 0: 关闭 1: 打开
RTC2IE	RTC 定时器 2 中断使能位 0: 关闭 1: 打开
RTC1IE	RTC 定时器 1 中断使能位 0: 关闭 1: 打开
MTHIE	RTC 月中断使能位 0: 关闭 1: 打开
DAYIE	RTC 日中断使能位 0: 关闭 1: 打开
HRIE	RTC 小时中断使能位 0: 关闭 1: 打开
MINIE	RTC 分钟中断使能位 0: 关闭 1: 打开
SECIE	RTC 秒中断使能位 0: 关闭 1: 打开

16.8.3 RTC 中断标志寄存器 (RTCIF)

RTCIF (RTC 中断标志寄存器)			基地址: 0x4000C000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X

Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	ALMIF	RTC2IF	RTC1IF	MTHIF	DAYIF	HRIF	MINIF	SECIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMIF	RTC 闹铃中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
RTC2IF	RTC 定时器 2 中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
RTC1IF	RTC 定时器 1 中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
MTHIF	RTC 月中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
DAYIF	RTC 日中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
HRIF	RTC 小时中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
MINIF	RTC 分钟中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0
SECIF	RTC 秒中断标志位 0: 未产生中断 1: 产生中断, 写 0 清 0

16.8.4 闹钟寄存器 (ALMR)

ALMR (闹钟寄存器)			基地址: 0x4000C000 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	ALMH[5:0]				
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	ALMM[6:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
ALMH[5:0]	RTC 闹铃中断小时设置, 允许写入 0-23 以外的数, 但是 RTC 闹铃中断将永远不会产生。

ALMM[6:0]	RTC 闹铃中断分钟设置，允许写入 0-59 以外的数，但是 RTC 闹铃中断将永远不会产生。 注：当小时和分钟寄存器与闹钟寄存器中的值相匹配时（且秒寄存器为 0），才会产生中断。
-----------	---

16.8.5 RTC 定时器 1 寄存器（RTCTMR1）

RTCTMR1 (RTC 定时器 1 寄存器)			基地址: 0x4000C000 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CNT[15:0]	最小分格为 1s，即最小可以每 1 秒产生一次中断，最大可以每 65536 秒产生一次中断，当计数溢出时，置位 RTC1IF 标志。 CNT[15: 0]用来表示一个 16BIT 的二进制的无符号整数，如果设置 CNT[15: 0]=00H，表示 RTC 内部的秒表功能中断每经过 (00H+1)*1S =1*1S=1S 的计时周期后，置位 RTC1IF 标志。 注：当定时器到达设定值时，如果用户没有关闭定时器，则定时器将从 0 开始重新计数。

16.8.6 RTC 定时器 2 寄存器（RTCTMR2）

RTCTMR2 (RTC 定时器 2 寄存器)			基地址: 0x4000C000 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CNT[15:8]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CNT[7:0]							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CNT[15:0]	最小分格为 0.0625s，即最小可以每 0.0625s 秒产生一次中断，最大可以每 4096 秒产生一次中断，当计数溢出时，置位 RTC2IF 标志。 CNT[15:0]用来表示一个 16BIT 的二进制的无符号整数，如果设置 CNT[15:0]=00H，表示 RTC 内部的秒表功能中断每经过 (00H+1)*0.0625S =1*0.0625S=0.0625S 的计时周期后，置位 RTC2IF 标志。 注：当定时器溢出时，如果用户没有关闭定时器，则定时器将从 0 开始重新计数。

16.8.7 RTC 秒寄存器 (SECR)

SECR (RTC 秒寄存器)			基地址: 0x4000C000 偏移地址: 18H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	SEC[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SEC[5:0]	秒计数器: 可设范围: 0-59。写入 0-59 以外的任何数值, 对该寄存器没有影响。

16.8.8 RTC 分寄存器 (MINR)

MINR (RTC 分寄存器)			基地址: 0x4000C000 偏移地址: 1CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	MIN[5:0]					
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
MIN[5:0]	分计数器: 可设范围: 0-59。写入 0-59 以外的任何数值, 对该寄存器没有影响。

16.8.9 RTC 时寄存器 (HOURR)

HOURR (RTC 时寄存器)			基地址: 0x4000C000 偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	HOUR[4:0]				
Write:								

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
HOUR[4:0]	时计数器: 可设范围: 0-23。写入 0-23 以外的任何数值, 对该寄存器没有影响。

16.8.10 RTC 日寄存器 (DAYR)

DAYR (RTC 日寄存器)			基地址: 0x4000C000 偏移地址: 24H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	DAY[4:0]				
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
DAY[4:0]	日计数器: 可设范围: 1-28/29/30/31。写入与年、月不匹配的任何数值, 对该寄存器没有影响。

16.8.11 RTC 月寄存器 (MONTHR)

MONTHR (RTC 月寄存器)			基地址: 0x4000C000 偏移地址: 28H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	MONTH[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
MONTH[3:0]	月计数器: 可设范围: 1-12。写入 1-12 以外的任何数值, 对该寄存器没有影响

16.8.12 RTC 年寄存器 (YEARR)

YEARR (RTC 年寄存器)			基地址: 0x4000C000 偏移地址: 2CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X

Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	YEAR[6:0]						
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
YEAR[6:0]	年计数器: 可设范围: 0-99。写入 0-99 以外的任何数值, 对该寄存器没有影响。

16.8.13 RTC 周寄存器 (WEEKR)

WEEKR (RTC 周寄存器)			基地址: 0x4000C000 偏移地址: 30H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	WEEK[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	1

位	功能描述
WEEK[2:0]	周计数器: 可设范围: 1-7。写入 1-7 以外的任何数值, 对该寄存器没有影响。

16.8.14 RTC 读控制寄存器 (RTCRD)

RTCRD (RTC 读控制寄存器)			基地址: 0x4000C000 偏移地址: 3CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	READFLAG
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
READFLAG	RTC 指示寄存器读控制位, 具体使用参见“RTC 指示寄存器读写流程”

16.8.15 RTC 写控制寄存器（RTCWR）

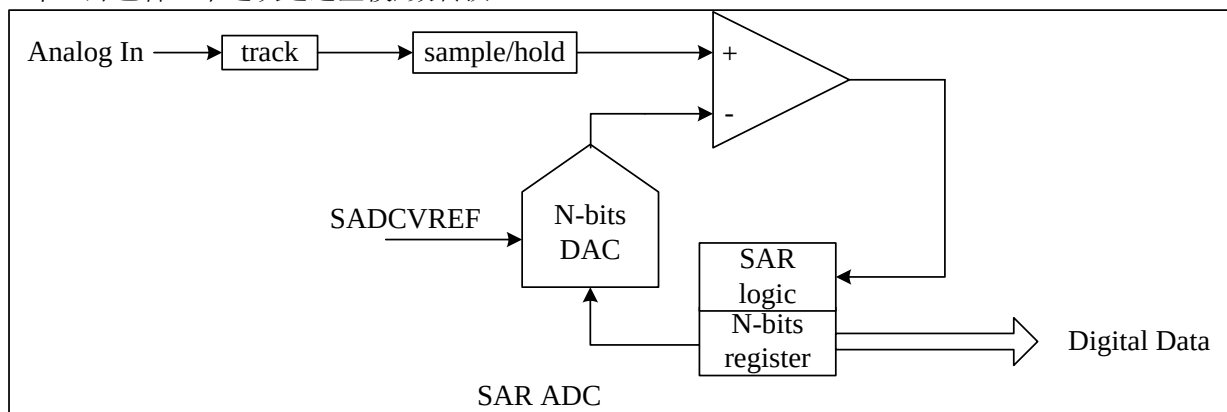
RTCWR (RTC 写控制寄存器)			基地址: 0x4000C000 偏移地址: 40H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	UPDATE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
UPDATE	RTC 指示寄存器写控制位，具体使用参见“RTC 指示寄存器读写流程”

17 SAR ADC

17.1 概述

本芯片包含 1 个逐次逼近型模/数转换 (SAR ADC)。



1.1 SAR ADC 采样原理图

17.1.1 特性

- 最高 11-Bits 分辨率
- $ENOB > 10.5$ bit for differential, 9.5 for single ended
- 内建基准电压通道
- ADC 时钟 F_{sadc} 可以选择 F_{sys}
- ADC 最大时钟频率 14MHz, 转换时间 1 μ s
- 差分采样功能: 支持两个模拟输入通道组合进行差分采样
- SADCVREF 可选外部 ADCREF、AVCC、TPSVREF 和 VREF
- 最高 16 信道, 包括 11 个外部通道 (ADC_IN 0~10) 和 5 个内部通道 (PGA0~2, VBG 和 VTPS)。
- 1 个规则转换组, 可任意设置规则序列长度和任意通道组合 (序列长度最大为 8)
- 内建滑动平均滤波器专用于温度测量模块
- 每个通道采样时间可独立设置
- ADC 大小功耗模式可选
 - 大功耗模式: ADC 高速转换, 可选高频时钟
 - 小功耗模式: ADC 低速转换, 适用于低功耗模式
- 中断: A/D 单次转换完成中断; A/D 序列全部转换完成中断
- 触发方式可设置
 - 软件触发、RTC 触发、外部中断触发、内部 TMR & TIM8 触发
- 工作模式
 - 单次转换模式
 - 连续转换模式
 - 扫描模式

17.1.3 模块结构图

- ADC 公式:
 $Volt = (code - 1024) * V_{ref} / 2048$
- ADC resolution:
 $1LSB = 5V / 2048 = 2.44mV$
- SADCxDAT 低16 位为有符号 16 位, bit19-16 为通道编号
- ADC offset 寄存器可以填入 1024, DAT 的值就会自动帮你减去, 但要注意正负号问题



图 1.2 SAR ADC 模块结构图

17.2 功能描述

17.2.1 ADC 模块开关

SAR ADC 模块的开关可以控制。当设置 SADCON=1 时，将 ADC 从关闭状态唤醒，为 A/D 转换做准备；当设置 SADCON=0 时，ADC 模块处于关闭状态，降低功耗。

SADCEN 控制模块的时钟，当 SADCEN=1 时，开启模块时钟，SAR ADC 按照配置进行转换；当 SADCEN=0 时，关闭模块时钟，SAR ADC 停止转换；

17.2.2 ADC 时钟

SAR ADC 的时钟源选择通过 SADCCLK_SEL 进行控制，Fsadc 默认时钟为 Fsys。Fsadc 由系统时钟 Fsys 通过设置 SADCCLK_DIV[3:0] 选择不同分频产生。

计算公式如下： $F_{sadc} = F_{sys} / SADCCLK_DIV$

注：

1. SAR ADC 时钟频率 Fsadc 最大可选 14MHz，最小频率为 112KHz
2. 仅在 ADC 时钟小于等于 144KHz 时，才能在序列中添加 TPS 采样通道。

17.2.3 ADC 触发方式

触发源可独立设置，配置寄存器 SADCTRGLEN 中的 ADTRIG[3:0] 可以选择不同触发源，具体对应关系如下：

- 0000: 软件触发 (SADCSTR) (default)
- 0001: RTC 周期定时完成触发
- 0010: 外部中断 INT3 触发
- 0011: 外部中断 INT7 触发
- 0100: TMR0 定时触发
- 0101: TMR1 定时触发
- 0110: TIM8_CH4 定时触发
- 0111: TIM8_CH5 定时触发
- 1000: TIM8_CH1 触发
- 1001: TIM8_CH2 触发
- 1010: TIM8_CH3 触发
- 1011: TIM8_CH4_CH5 触发

软件启动：若 SADCSTR 位被置 1，对应模式 A/D 转换开始，SADCSTR 位被硬件自动清 0；

RTC 触发：使用 RTC 定时完成中断触发

外部中断触发：外部中断触发可选择 INT3 和 INT7，捕获到上升沿触发；

内部 TMR 定时触发：TMR0/TMR1/TMR8_CH4/TMR8_CH5 可选择，定时完成触发。

内部 TMR 触发：TMR8_CH1/TMR8_CH2/TMR8_CH3/TMR8_CH4_CH5 可选择

注：（1）有关中断触发 ADC 方式，内部增加逻辑，不使用中断标志，使用触发模块内部信号，触发 ADC 工作。

（2）外部中断触发为上升沿触发，TMR 的定时触发为定时中断触发。

（3）必须首先配置触发使能控制位 (SADC_TRIG_EN=1)，ADC 的触发才会生效；若 SADC_TRIG_EN=0，则所有触发均无意义。

17.2.4 ADC 模拟输入与通道号

ADC 模块包含 1 个规则转换序列，序列长度最大为 8 个，在 ADC_IN0~10/PGA0~2/VBG/VTPS 这 16 个通道任意组合选择，可向序列信道配置寄存器 SADCSEQCFG 中 SEQx_SEL[3:0], 写入通道号（其中 x=0~15），从而设置序列中通道转换顺序。

SAR ADC 模拟输入与通道号对应关系如下表所示。

Channel No	通道名称
0	ADC_IN0
1	ADC_IN1
2	ADC_IN2
3	ADC_IN3
4	ADC_IN4
5	ADC_IN5
6	ADC_IN6
7	ADC_IN7
8	ADC_IN8
9	ADC_IN9
10	ADC_IN10
11	VBG
12	PGA0
13	PGA1
14	PGA2
15	TPS

注：序列信道配置寄存器配置的为信道号，而采样时间配置寄存器配置的对应信道号的采样时间。

17.2.5 差分输入功能

除了传统的单端采样，SAR ADC 模块还支持对两个模拟输入通道进行差分采样。要启用差分采样功能，软件必须在 SADCIFCFG 寄存器中配置 SAR_INMODE_SEL 位置位，配置指定信道的输入方式为差分输入。若配置为差分采样，必须将对应的模拟通道输入差分信号。差分信号对 0 对模拟输入端 ADC_IN0 (VP[0]) 和 ADC_IN1 (VN[0]) 进行采样，差分信号对 1 对模拟输入端 ADC_IN2 (VP[1]) 和 ADC_IN3 (VN[1]) 进行采样，依此类推（见下表）。ADC 不支持差分信号对的随意组合，如模拟输入端 0 和模拟输入端 3 无法作为一对差分信号输入。

VTPS 和 VBG 通道采样时，不能开启差分功能，否则造成 VTPS 和 VBG 采样错误。

差分模式只对外部模拟通道（ADC_IN0~ADC_IN5）有效，若转换组指定序列选择差分方式，序列中配置差分对的任意一个通道，均转换其对应的差分对，并将差分结果保存到数据寄存器中，差分对的采样时间由序列所选择的通道采样时间决定。

下表中的差分对与通道的对应关系如下所示。

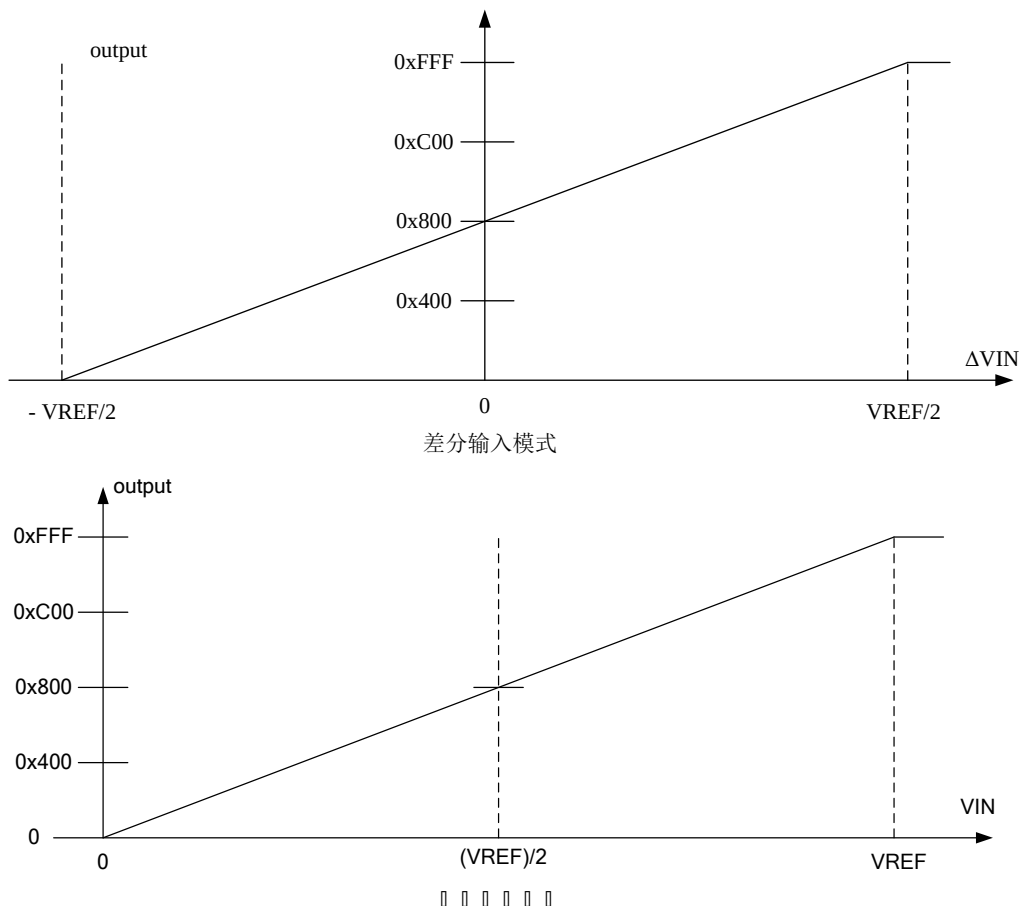
差分信号对	模拟输入通道
0	ADC_IN0 和 ADC_IN1
1	ADC_IN2 和 ADC_IN3
2	ADC_IN4 和 ADC_IN5

差分模式下，输入电压为差分信号对的电压差 $\Delta VIN = VP[x] - VN[x]$ （VP[x] 和 VN[x] 电压均大于 0），其中 x=0, 1, 2。

11-bits 分辨率下：

若输入电压 = 0，则转换结果 = 0x800

输入电压转换范围是 0x800~0xFFFF



17.2.6 ADC 工作模式

规则转换序列可以是多个通道上以任意顺序进行组合而成，并按照设定的通道和顺序完成一系列转换。序列长度可通过 SEQLENG[2:0] 进行设置，最多设定 8 个序列，优先级顺序从 0 至 7 依次递减。规则组转换序列被触发时，从最高优先级（序列 0）开始转换；例如，设定规则转换组序列有 4 个转换，则使用规则组序列 0 至 3；若设定规则组转换序列有 8 个转换，则使用规则组序列 0 至 7。

规则组序列的每个转换对应的采样通道可通过寄存器独立设置（可选通道共有 16 个，依次为 ADC_IN0-10、VBG、PGA0、PGA1、PGA2、VTPS）。例如，规则组序列 0 至 7，可以设置如下顺序完成转换：信道 3、信道 5、信道 2、信道 2、信道 0、信道 2、信道 2、信道 7。

使用规则组转换需要依照以下步骤配置：

- 1) 设定规则组的转换序列长度；
- 2) 设定规则组中各个转换序列所对应的模拟输入通道；
- 3) 使能并设定规则组转换的触发源；
- 4) 设定 ADC 转换模式，并使能 ADC 转换；

17.2.6.1 规则转换的几种模式使用说明

通过配置 SADCCON 寄存器的 SADCCONT 和 SADCSCAN 这 2 个控制位组合使用，可组合出 4 种工作模式，具体模式说明见下表。

表-1 SAR ADC 规则组转换模式说明

寄存器配置		模式说明
SADCCONT	SADCSCAN	
0	0	单通道单次采样
0	1	多通道单次采样
1	0	单通道连续采样
1	1	多通道连续采样

17.2.6.2 数据存储说明

SAR ADC 的规则转换组有一个通用数据寄存器 SADCDAT，每次转换数据都会更新到该寄存器中，为了防止数据被覆盖，建议配合 DMA 完成数据传递，当前转换完成标志置起时，此时 DMA 将 SADCDAT 中的数据保存到 RAM 中，该数据寄存器仅保存最近一次触发转换的数据。

SAR ADC 的规则转换组数据还按照序列号对应的数据寄存器存储，每个序列号都有独立的数据寄存器，转换完成一次，数据寄存器的数据更新一次，此时不需要采用 DMA 搬移数据，不会存在数据被覆盖的问题。

在 ADC 转换过程中，通用数据寄存器和启用的序列数据寄存器都会更新。

注：为了自动补偿 RTC 误差，温度测量过程中，为了提高温度测量的准确度，增加滑动平均控制位。

当 SADCTPSFLEN=1 时，此时才可选择 SADCTPSFL[1:0]的滑动滤波次数，默认为 2 次，此时温度测量的数据存储 SADCTAD 寄存器中。

当 SADCTPSFLEN=0 时，选择 SADCTPSFL[1:0]的滑动滤波次数无效，此时温度传感器采样数据存储 SADCDAT 或按照序列号对应的数据寄存器中，此时不影响 SADCTAD 寄存值。

17.2.6.3 单次转换模式

SADCCONT=0 时，ADC 工作于单次转换模式下，每次触发 ADC 转换，只对第一个序列选中的通道执行一次转换。若配置了扫描模式，则执行规则组序列的每一次转换。

例如：

(1) 在单次转换模式下，未使能扫描模式，信道序列 0 配置了 ADC_IN3，每次触发 ADC 转换只会对 ADC_IN3 通道进行一次转换，转换完成后自动结束，此时 SADCSTR 被清 0。

(2) 在单次转换模式下，使能了扫描模式，设定转换组序列 0~2 为 ADC_IN3、ADC_IN7、ADC_IN5，则每次触发 ADC 转换，自动转换 ADC_IN3、ADC_IN7、ADC_IN5 通道，转换完成后自动结束，如果转换期间 SADC_TRIG_EN 被清 0，则当前序列 n 转换完成后，自动停止。下次发生触发 ADC 转换事件时，转换从序列 0 开始。

单次转换模式下，每次转换完成后：

- 1) 转换数据储存在数据寄存器中
- 2) 转换结束标志置 1
- 3) 如果设置了转换完成中断，则产生转换完成中断。

17.2.6.4 连续转换模式

SADCCONT=1 时, ADC 工作于连续转换模式下。当发生触发 ADC 转换事件时, 当前序列 $0 \sim n$ ($0 \leq n \leq 7$) 通道转换完成后, 继续转换序列 $0 \sim n$ ($0 \leq n \leq 7$) 通道。

在连续转换模式下, 未使能扫描模式, 当发生触发 ADC 转换事件时, 当前序列 0 完成 ADC 通道转换后, 继续转换序列 0 配置的 ADC 信道。例如序列 0 配置为 ADC_IN3, 当序列 0 被触发时, ADC 自动转换 ADC_IN3 通道, 转换完成后再次对 ADC_IN3 通道进行转换。

在连续转换模式下, 使能了扫描模式, 设定转换组序列 $0 \sim 2$ 为 ADC_IN3、ADC_IN7、ADC_IN5 通道。当发生触发 ADC 转换事件时, 自动依次转换 ADC_IN3→ADC_IN7→ADC_IN5, 完成规则组序列的 3 个转换后, 立即重新从规则转换序列 0 开始转换, 且依次完成序列 $0 \sim 2$ 的通道转换, 重复上述转换流程。如果转换期间 SADC_TRIG_EN 被清 0, 完成当前转换后, 自动停止转换, 再次发生触发事件时, 转换从序列 0 开始。将 SADCCONT 写 0, 完成本次规则组序列转换后也会停止)。

在连续模式下, 每次转换后:

- 1) 转换数据储存在数据寄存器中;
- 2) 转换结束标志置 1
- 3) 如果设置了转换完成中断, 则产生转换完成中断
- 4) 启动下一次转换

17.2.6.5 扫描模式

扫描模式开启时, 扫描序列长度配置为 n (SADCTRLLEN. SEQLENG= n), 其中 $0 \leq n \leq 7$ 。当发生触发 ADC 事件, 从规则转换组的序列 0 依次转换序列 $0 \sim n$ 中的通道。

扫描模式和单次模式组合时, 当发生触发 ADC 事件, 从规则转换组的序列 0 依次转换序列 $0 \sim n$ 中的通道, 完成序列 n 的通道转换后自动停止。

扫描模式和连续模式组合时, 当发生触发 ADC 事件, 从规则转换组的序列 0 依次转换序列 $0 \sim n$ 中的通道, 完成序列 n 的通道转换后自动重新从序列 0 依次转换。

17.2.7 ADC 采样转换时间

ADC 采样转换所使用的时钟为 F_{sadc} , 每个通道采样时间可独立控制, 设置采样时间配置寄存器 SADCSAMP 中的 $CHx_SAMP[3:0]$ ($x=0 \sim 7$) 位, 可控制各个通道的采样时间, 默认采样时间为 4 个 CLK, 最小采样时间可设置为 2 个 CLK, 最大采样时间可设置为 256 个 CLK。(SADCON 使能后到 ADC 建立完成约 10us, 由模拟提供时间)

ADC 高速采样转换时间时序控制图如下 (低速采样转换详见 1.2 章节):

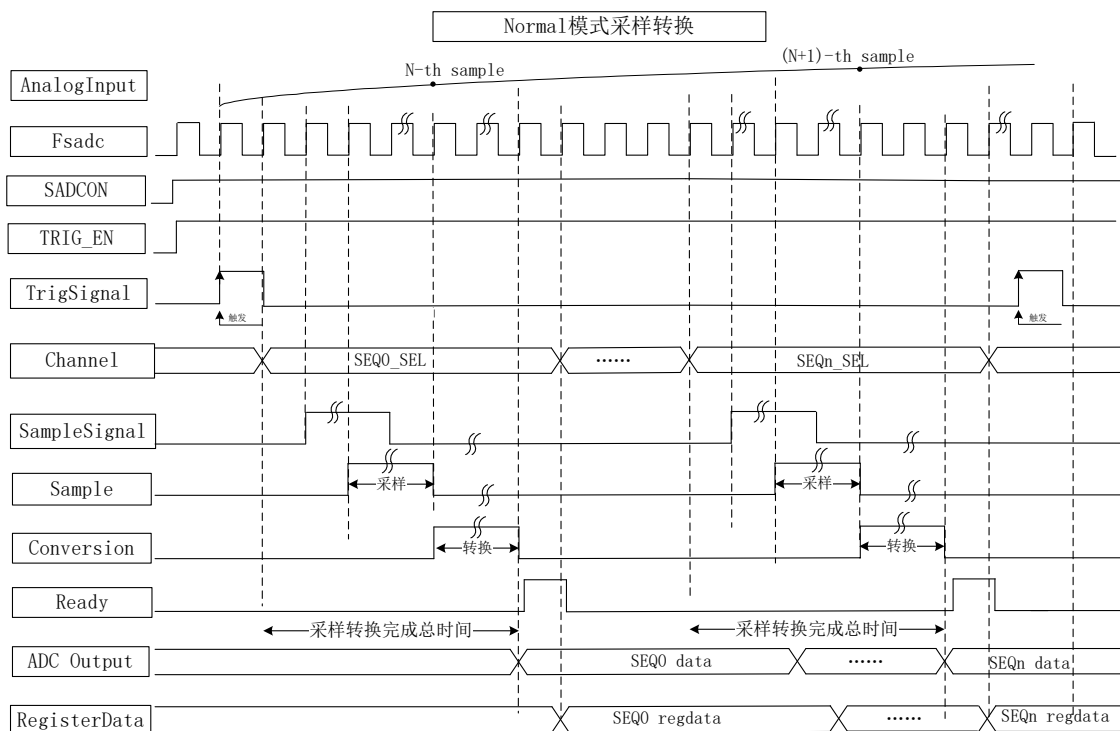


图 1.3

SAR ADC Normal 模式采样转换时序图

在高速采样转换时，SADCON 由寄存器控制，采样信号由触发使能信号开启；在低速采样转换时，SADCON 和采样信号均由触发使能信号开启（数字控制）

17.2.8 ADC 参考源

SAR ADC 的参考源为 AVCC

17.2.9 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)。

温度传感器在内部和 ADC_CH15 输入通道相连接，此通道把传感器输出的电压转换成数字值。

温度传感器模拟输入推荐单次采样时间是 8 μ s。

当没有被使用时，传感器可以置于关电模式。

17.3 特殊功能寄存器列表

SAR ADC 模块寄存器基地址: 0x4000E100				
偏移地址	名称	读写方式	复位值	功能描述
0x00	SADCCON	R/W	0x0402	SADC控制寄存器
0x04	SADCCLKCFG	R/W	0x0002	SADC 时钟配置寄存器
0x08	SADCDFCFG	R/W	0x01A2	SADC 差分输入配置寄存器
0x0C	SADCSTR	R/W	0x0000	SADC 序列转换控制寄存器
0x10	SADCIE	R/W	0x0000	SADC 中断控制寄存器
0x14	SADCIF	R/W	0x0000	SADC 中断标志寄存器
0x18	SADCSAMP	R/W	0x0000	SADC 采样时间配置寄存器
0x20	SADCTRLEN	R/W	0x0000	SADC 触发源及序列长度寄存器

0x24	SADCSEQCFG	R/W	0x0000	SADC 序列信道配置寄存器
0x2C	SADCDAT	R	0x0000	SADC 数据寄存器
0x30+4*x x=0~7	SADCxDAT	R	0x0000	SADCx 数据寄存器
0x70	SADC_CALCH	R/W	0x0000	SADC 校准通道选择寄存器
74H + (x*4) x=0~4	SADC_CALDATx	R/W	0x0000	CALDATx 校准值寄存器
0x88	SADC_TPSOffset	R/W	0x0000	TPS 校准值寄存器
0x8C	SADCCRV	R/W	0x0000	SADC 电流控制寄存器
0x90	SADCTAD	R	0x0000	SADC 温度滤波值寄存器

17.4 特殊功能寄存器说明

17.4.1 SADC 控制寄存器（SADCCON）

SADCCON (SADC 控制寄存器)			基地址: 0x4000E100 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	SADC EN	CAL	TPSON	CHOP_AD C_TPS_E N	SADCTPSFL[1:0]		X	X
Write:								
Reset:	0	0	0	0	0	1	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	SADC CONT	SADC SCAN	SADCTPS FLEN	SADC_TRI G_EN	SADC PWR	SADC ON
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述
SADCEN	SADC 时钟控制位: 0: SADC 时钟关闭 (default) 1: SADC 时钟开启
CAL	软件设置 1 开始校准, 校准完成硬件清除为 0 0: 校准完成 1: 开始校准 自校准只在 ADC 校准通道选择 PGA0-3 或 OPA 时有效
TPSON	1: TPS on 0: TPS off
CHOP_ADC_TPS_EN	设定为 0 (不更动此值)
SADCTPSFL[1:0]	SADC 温度测量转换滑动平均滤波控制位 00: 1 次平均 01: 2 次平均 (default) 10: 4 次平均 11: 8 次平均
SADCCONT	SADC 连续转换模式控制位:

	0: 单次转换模式 (default) 1: 连续转换模式
SADCSCAN	SADC 扫描模式控制位 0: 关闭扫描模式 (default) 1: 开启扫描模式
SADCTPSFLEN	SADC 温度滤波开关控制位: 0: 关闭 (default) 1: 开启
SADC_TRIG_EN	SADC 触发使能控制位 0: 触发使能关闭 (default) 1: 触发使能开启
SADCPWR	SADC 功耗模式选择位: 0: 低功耗模式 1: 大功耗模式, 大功耗模式可以高速转换 (default) 注: 连续模式下, SADCPWR 需要置 1
SADCON	SADC 开关控制位: 0: SADC 关闭 (default) 1: SADC 开启

17.4.2 SARADC 时钟配置寄存器 (SADCCLKCFG)

SADCCLKCFG (SARADC 时钟配置寄存器)			基地址: 0x4000E100 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	SADCCLK_SEL
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	SADCCLK_DIV[3:0]			
Write:								
Reset:	0	0	0	0	0	0	1	0

位	功能描述
SADCCLK_SEL	SADC 时钟源选择位: 0: 选择 Fsys 1: 选择 PLLDIV2
SADCCLK_DIV[3:0]	SADC 时钟分频控制位: ADC 时钟计算公式: $F_{sadc} = \text{SADC 时钟源} / \text{SADCCLK_DIV}$ 0000: 1 分频 0001: 2 分频 0010: 3 分频 (default) 0011: 4 分频 0100: 5 分频 0101: 6 分频 0110: 7 分频 0111: 8 分频 1000: 9 分频 1001: 10 分频 1010: 11 分频

	1011: 12 分频 1100: 16 分频 1101: 32 分频 1110: 64 分频 1111: 128 分频
--	--

17.4.3 差分输入配置寄存器 (SADCIFCFG)

SADCIFCFG (差分输入配置寄存器)			基地址: 0x4000E100 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:						SAR_VRE	SAR_IB_SEL[1:0]	
Write:	X	X	X	X	X	FBUFFER _EN		
Reset:	0	0	0	0	0	0	0	1
	Bit7	6	5	4	3	2	1	Bit0
Read:	SAR_VCM	SAR_ASY	SAR_DELAY_SEL [1:0]		X	X	SAR_DIF	SAR_INM
Write:	_SEL	N_SEL					F_SEL	ODE_SEL
Reset:	1	0	1	0	0	0	1	0

位	功能描述
SAR_VREFBUFFER_EN	SADC 基准 BUFFER 使能控制位 0: 关闭 (default) 1: 打开 注: 大功耗模式下, 在 SADCON 前 200us-300us 开启该位; 小功耗模式下, 该位固定为 0, 由硬件触发自动控制。
SAR_IB_SEL[1:0]	SADC 偏置电流大小选择位: 00: 1.5uA 01: 2uA (default) 10: 2.5uA 11: 3uA
SAR_VCM_SEL	SADC 共模信号选择位 0: 从 avdd 分压 1: 从 Vreffp 分压 (default) 注: TPS 采样转换时, 内部自动切换为 0: 从 avdd 分压
SAR_ASYN_SEL	SADC 同/异步转换模式选择位 0: 同步转换模式 (default) 1: 异步转换模式 注: 低功耗模式下, 该位固定为 1; 同步模式可在大功耗模式下开启。
SAR_DELAY_SEL [1:0]	SADC 的异步转换模式下延时选择位: 00: 30ns 01: 35ns 10: 40ns (default) 11: 45ns
SAR_DIFF_SEL	SADC 转换模式选择位: 0: 单端转换 1: 差分转换 (default)
SAR_INMODE_SEL	SADC 输入方式选择位: 0: SADC 为单端输入 (default)

1: SADC 为差分输入

17.4.4 SADC 序列转换控制寄存器 (SADCSTR)

SADCSTR (SADC 序列转换控制寄存器)			基地址: 0x4000E100 偏移地址: 0CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	X	SADC STR
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SADCSTR	SADC 序列转换触发控制位: 0: 停止序列转换 (default) 1: 开启序列转换 软件写 1, 硬件自动清 0

17.4.5 SADC 中断控制寄存器 (SADCIE)

SADCIE (SADC 中断控制寄存器)			基地址: 0x4000E100 偏移地址: 10H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	TPSIE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	SEQIE	SIGIE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TPSIE	SADC 温度转换完成中断使能控制位: 0: 不使能温度转换完成中断 (default) 1: 使能温度转换完成中断
SEQIE	SADC 扫描序列转换完成中断使能控制位: 0: 不使能序列转换完成中断 (default) 1: 使能序列转换完成中断
SIGIE	SADC 单次转换完成中断使能控制位: 0: 不使能单次转换完成中断 (default) 1: 使能单次转换完成中断

注: (1) 单信道单次模式: 只会产生 SIGIE 中断;
(2) 多信道单次模式: 每个通道转换一次, 会产生 SIGIE 中断, 整个序列转换完成, 产生 SEQIE 中断;
(3) 单信道连续模式: 只会产生 SIGIE 中断;

(4) 多通达连续模式：每个通道转换一次，会产生 SIGIE 中断，整个序列转换完成一次，产生一次 SEQIE 中断。

SEQIE 中断与是否为扫描模式有关。

17.4.6 SADC 中断标志寄存器 (SADCIF)

SADCIF (SADC 中断标志寄存器)			基地址: 0x4000E100 偏移地址: 14H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	TPSIF
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	X	SEQIF	SIGIF
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TPSIF	SADC 温度转换完成中断标志位: 温度转换完成标志置 1, 若使能温度转换完成中断, 则产生中断 注: 当启用 SAR ADC 时, SADCTADA 被更新时, 该标志会置起
SEQIF	SADC 扫描序列转换完成中断标志位 规则组序列转换完成标志置 1, 若使能序列转换完成中断, 则产生中断
SIGIF	SADC 单次转换完成中断标志位: 规则组序列中一个转换完成标志置 1, 若使能单次转换完成中断, 则产生中断

17.4.7 SADC 采样时间配置寄存器 (SADCSAMP)

SADCSAMP (SADC 采样时间配置寄存器)			基地址: 0x4000E100 偏移地址: 18H					
	Bit31	30	29	28	27	26	25	Bit24
Read:	CH7_SAMP[3:0]				CH6_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	CH5_SAMP[3:0]				CH4_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	CH3_SAMP[3:0]				CH2_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CH1_SAMP[3:0]				CH0_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CHx_SAMP[3:0] X=0~7	模拟输入通道 x 采样时间 0000:2 Fsadc (default) 0001:4 Fsadc 0010:8 Fsadc 0011:16 Fsadc 0100:32 Fsadc 0101:64 Fsadc 0110:128 Fsadc 0111:256 Fsadc 1xxx:X

注：CH0_SAMP 对应 ADC_IN0 通道的采样时间；CH1_SAMP 对应 ADC_IN1 通道的采样时间；……；CH12_SAMP 对应 TPSVREF 通道的采样时间；CH13_SAMP 对应 VTPS 通道的采样时间。

17.4.8 SADC 采样时间配置寄存器 (SADCSAMP2)

SADCSAMP2 (SADC 采样时间配置寄存器)			基地址: 0x4000E100 偏移地址: 1CH					
	Bit31	30	29	28	27	26	25	Bit24
Read:	CH15_SAMP[3:0]				CH14_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	CH13_SAMP[3:0]				CH12_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	CH11_SAMP[3:0]				CH10_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	CH9_SAMP[3:0]				CH8_SAMP[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CHx_SAMP[3:0] X=8 ~ 15	模拟输入通道 x 采样时间 0000:2 Fsadc (default) 0001:4 Fsadc 0010:8 Fsadc 0011:16 Fsadc 0100:32 Fsadc 0101:64 Fsadc 0110:128 Fsadc 0111:256 Fsadc 1xxx:X

17.4.9 SADC 触发源及扫描序列长度寄存器 (SADCTRLEN)

SADCTRLEN (SADC 触发源及扫描序列长度寄存器)			基地址: 0x4000E100 偏移地址: 20H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	SEQLENG[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	CH4_EN	CH5_EN	ADTRIG[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SEQLENG[2:0]	规则组转换扫描序列长度控制位 设置范围为 000~111 000: 扫描序列长度为 1 (default) 001: 扫描序列长度为 2 ... 111: 扫描序列长度为 8
CH4_EN	TIM8 CH4 input enable 1:TIM8 CH4 觸發使能 0:TIM8 CH4 觸發關閉
CH5_EN	TIM8 CH5 input enable 1:TIM8 CH5 觸發使能 0:TIM8 CH5 觸發關閉
ADTRIG[3:0]	ADC 触发源选择位 0000: 软件触发 (SADCSTR) (default) 0001: RTC2 周期定时完成触发 0010: 外部中断 INT3 触发 0011: 外部中断 INT7 触发 0100: TMR0 定时触发 0101: TMR1 定时触发 0110: TIM8_CH4 定时触发 0111: TIM8_CH5 定时触发 1000: TIM8_CH1 触发 1001: TIM8_CH2 触发 1010: TIM8_CH3 触发 1011: TIM8_CH4_CH5 触发 注: 选择 000 为软件触发, 屏蔽硬件触发; 若选择硬件触发, 则软件触发仍然有效。

17.4.10 SADC 序列信道配置寄存器 (SADCSEQCFG)

SADCSEQCFG (SADC 序列信道配置寄存器)			基地址: 0x4000E100 偏移地址: 24H					
	Bit31	30	29	28	27	26	25	Bit24

Read:	SEQ7_SEL[3:0]				SEQ6_SEL[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit23	22	21	20	19	18	17	Bit16
Read:	SEQ5_SEL[3:0]				SEQ4_SEL[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SEQ3_SEL[3:0]				SEQ2_SEL[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SEQ1_SEL[3:0]				SEQ0_SEL[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SEQ _x _SEL[3:0]	规则转换组序列中第 x 个转换通道选择 (x=0-15) 0:AD0 1:AD1 2:AD2 3:AD3 4:AD4 5:AD5 6:AD6 7:AD7 8:AD8 9:AD9 10:AD10 11:AD11 12:PGA0 13:PGA1 14:PGA2 15:TPS

17.4.11 SADC 通用数据寄存器 (SADCDAT)

SADCDAT (SADC 通用数据寄存器)		基地址: 0x4000E100 偏移地址: 2CH						
	Bit23	22	21	20	19	18	17	Bit16
Read:					DATA_CH[3:0]			
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SADCDAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SADCDAT[7:0]							

Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SADCDAT[15:0]	规则组通用数据寄存器 16bit 有符号数，只读，所有规则组序列通道的转换结果 实际转换值 12 位计算公式为 $((V_{in}/V_{cc}) * 2048) + 1024$ (伪差分) bit13-bit0, 为实际转换值减去对应 offset 后的 14 位有符号数 bit15-bit14 为 bit13 符号位扩展相同 2 位
DATA_CH[3:0]	转换通道寄存器 只读，对应当前转换选择的通道

17.4.12 SADCx 序列数据寄存器 (SADCxDAT) (x =0~7)

SADCxDAT (x =0~7) (SADCx 序列数据寄存器)			基地址: 0x4000E100 偏移地址: 30H+4*x (x=0~7)					
	Bit23	22	21	20	19	18	17	Bit16
Read:						DATAx_CH[3:0]		
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SADCxDAT[15:8]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	SADCxDAT[7:0]							
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

位	功能描述
SADCxDAT[15:0]	规则组序列 x 数据寄存器 16bit 有符号数，只读，所有规则组序列通道的转换结果 实际转换值 12 位计算公式为 $((V_{in}/V_{cc}) * 2048) + 1024$ (伪差分) bit13-bit0, 为实际转换值减去对应 offset 后的 14 位有符号数 bit15-bit14 为 bit13 符号位扩展相同 2 位
DATAx_CH[3:0]	转换通道寄存器 只读，对应转换规则组序列 x 选择的通道

17.4.13 SADC_CALCH (SADC 校准通道选择寄存器)

SADC_CALCH (SADC 校准通道选择寄存器)			基地址: 0x4000E100 偏移地址: 70H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X				OPACH[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X			OPACH_EN	X		SADC_CALCH[1:0]	

Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
OPACH[3:0]	OPA channel select 0:AD0 1:AD1 2:AD2 3:AD3 4:AD4 5:AD5 6:AD6 7:AD7 8:AD8 9:AD9 10:AD10 11:AD11
OPACH_EN	1: OPA channel enable 0: OPA channel disable
SADC_CALCH[1:0]	SADC 校正通道选择 0:PGA0 1:PGA1 2:PGA2 3:OPA

17.4.14 SADC_CALDATx (x=0-4) (CALDATx 校准值寄存器)

SADC_CALDATx (x=0-4) (CALDATx 校准值寄存器)	基地址: 0x4000E100 偏移地址: 74H + (x*4)
	Bit15...Bit0
Read:	SADC_CALDATx[15:0]
Write:	
Reset:	0

位	功能描述
SADC_CALDATx[15:0]	x=0,PGA0 offset (signed) x=1,PGA1 offset (signed) x=2,PGA2 offset (signed) x=3,ADC offset (signed) x=4,OPA offset (signed) bit13-bit0, 为实际 offset 值 bit15-bit14 为 bit13 符号位扩展相同 2 位, 不可写

注意: ADC offset 无自校准, 需手动填入 offset 值。只在单独转换 CHANNEL0-11 时生效。

PGA0-2 offset 有自校准, 自校准时需将校准通道 SADC_CALCH[1:0]和转换通道 SEQ0_SEL[3:0]选择对应的 PGA0-2, 也可通过手动写入 offset。Offset 只在转换对应 PGA 通道 (CHANNEL12-14) 时生效。

OPA offset 有自校准, 自校准时需将校准通道 SADC_CALCH[1:0]选择对应 OPA 且 OPACH_EN 置为 1, 转换通道 SEQ0_SEL[3:0]需和 OPACH[3:0]对应一致, 也可通过手动写入 offset。offset 只在转换选择的 OPA 通道并且 OPACH_EN 为 1 时生效。

17.4.15 SADC_TPSSOffset (TPS 校准值寄存器)

SADC_TPSSOffset (TPS 校准值寄存器)	基地址: 0x4000E100 偏移地址: 88H
	Bit15...Bit0
Read:	SADC_TPSSOffset[15:0]
Write:	
Reset:	0

位	功能描述
TPSSOffset[15:0]	TPS offset (signed) bit13-bit0, 为实际 offset 值 bit15-bit14 为 bit13 符号位扩展相同 2 位, 不可写

注意: TPSSOffset 无自校准, 需手动填入 offset 值, 只在单独转换 CHANNEL15 时生效。

17.4.16 SADCCRV (SADC 电流控制寄存器)

SADCCRV (SADC 电流控制寄存器)	基地址: 0x4000E100 偏移地址: 8CH
	Bit15 14 13 12 11 10 9 Bit8
Read:	X
Write:	
Reset:	0 0 0 0 0 0 0 0
	Bit7 6 5 4 3 2 1 Bit0
Read:	X LDO_ADC[1:0] IADC[1:0]
Write:	
Reset:	0 0 0 0 0 0 0 1

位	功能描述
IADC[1:0]	ADC 电流控制位 00:1u 01:2u(default) 10:3u 11:4u ADC 测试模式下为 01
LDO_ADC[1:0]	Hvref 输出微调控制位 00:vbg=1.263 (default) 01:vbg=1.253 10:vbg=1.263 11:vbg=1.274

17.4.17 SADCTAD (SADC 温度滤波值寄存器)

SADCTAD (SADC 温度滤波值寄存器)	基地址: 0x4000E100 偏移地址: 90H
	Bit11...Bit0

Read:	SADCTAD[11:0]
Write:	
Reset:	0

位	功能描述
SADCTAD[11:0]	温度 Moving Average 存放的值

17.5 应用说明

17.5.1 滑动滤波说明

当 SADCTPSFLEN=1 时，此时才可选择 SADCTPSFL[1:0]的滑动滤波次数，默认为 2 次，此时温度测量的数据存储在 SADCTAD 寄存器中。

当 SADCTPSFLEN=0 时，选择 SADCTPSFL[1:0]的滑动滤波次数无效，此时温度传感器采样数据存储在 SADCDAT 或按照序列号对应的数据寄存器中，此时不影响 SADCTAD 寄存值。

采用 SAR ADC 完成温度测量时，SADCTPSFLEN=1 时，若此时选择的滤波次数为 4，若只采样 1 次，则此时更新到 SADCTAD 寄存器的数据为 TPSDAT1/4；

若采样了 2 次，则更新到 SADCTAD 寄存器的数据为 (TPSDAT1+TPSDAT2)/4；

若采样了 3 次，则更新到 SADCTAD 寄存器的数据为 (TPSDAT1+TPSDAT2+TPSDAT3)/4；

若采样了 4 次，则更新到 SADCTAD 寄存器的数据为 (TPSDAT1+TPSDAT2+TPSDAT3+TPSDAT4)/4；

若采样了 5 次，则更新到 SADCTAD 寄存器的数据为 (TPSDAT1+TPSDAT2+TPSDAT3+TPSDAT4+TPSDAT5)/4；

17.5.2 ADC 工作模式切换应用说明

ADC 转换过程中，用户应避免直接修改 SADCCONT 和 SADCSCAN 寄存器进行扫描/非扫描模式和单次/连续模式切换。

当需要进行 ADC 工作模式切换时，应先将 SADC 触发使能控制位 SADC_TRIG_EN 置 0。ADC 内部会在当前一次转换完成后停止转换，建议等待当前这一次的转换完成（当前转换的数据寄存器更新或当前一次的转换完成 IRQ 到来）之后，修改 ADC 工作模式等相关配置。为了防止 SADC_TRIG_EN 置 0 和当前转换完成在同时刻发生，导致无法捕获到转换完成 IRQ 的极限情况，在使用时可增加最长等待时间限制，超过时间后可以后续进行修改配置的操作，最长等待时间 $\geq (\text{CHx_SAMP}+14)$ 个 CLK (MAX: 270 CLK)。完成配置后再将 SADC 触发使能控制位 SADC_TRIG_EN 置 1；等待下一次 ADC 使能触发信号到来时，ADC 会按照新的配置进行转换。

17.5.3 SAR ADC 使用注意说明

在使用 SARADC 时，需要先开启 SADCON=1 后，再开启 SADC_TRIG_EN=1，否则可能触发信号会在 SARADC 未完全准备好之前到来，导致模块功能异常。

18 CMP 比较器

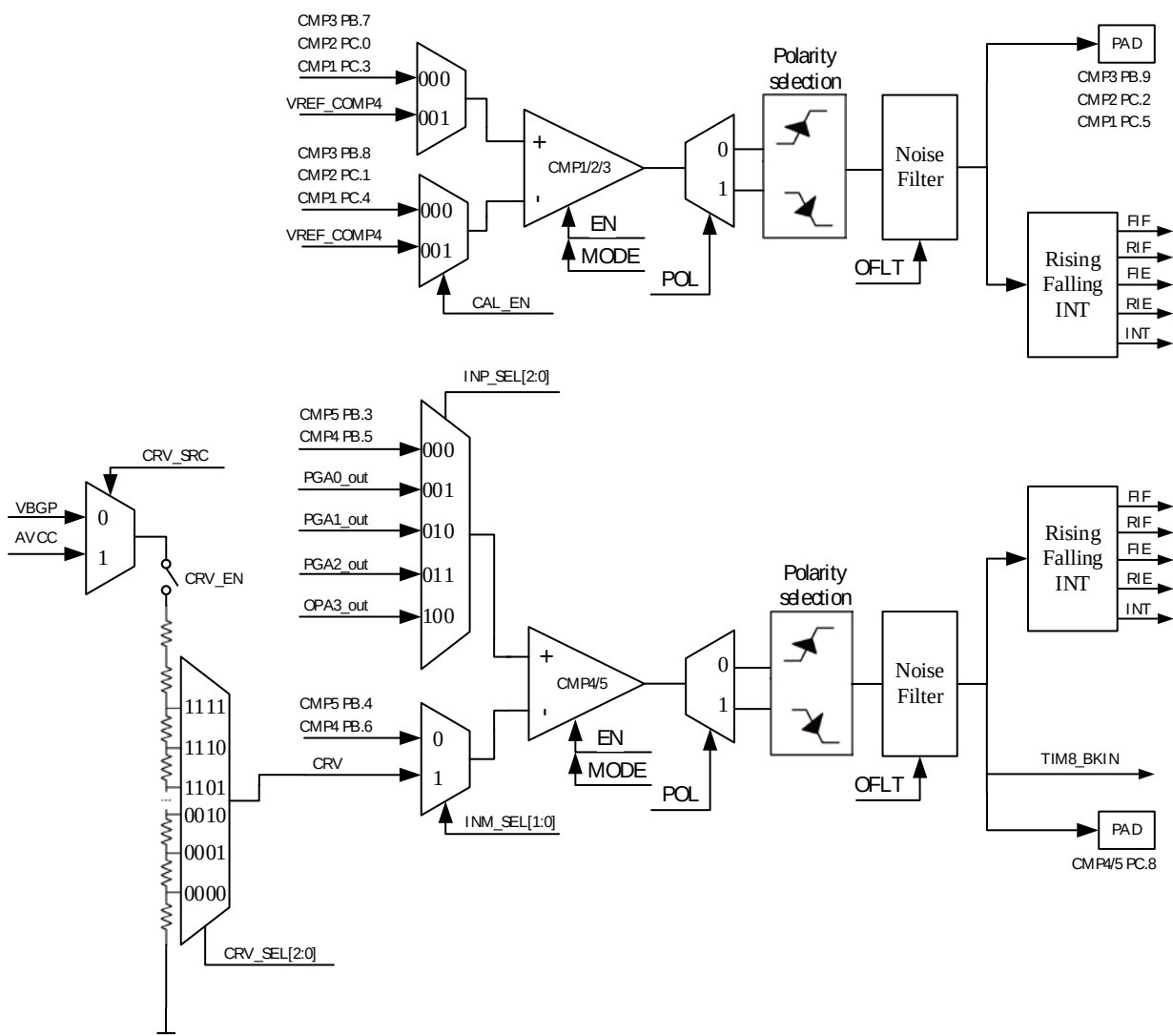
18.1 CMP 简介

芯片内嵌五个通用比较器 CMP1、CMP2、CMP3、CMP4、CMP5 可独立使用,也可与 TIM8 和 OPA 结合使用。

18.2 比较器功能描述

18.2.1 简介

下图是比较器框图:



18.2.2 比较器开关控制

通过设置CMP_x_CR 寄存器的EN 位可给CMP 上电。设置EN 位时，它将CMP 从断电状态唤醒，清除EN 位可停止比较器工作。

18.2.3 比较器输入和输出

CMP1-3 只有一个正相输入和一个反相输入通道, 不能选择输入讯号

CMP4-5 有 5 个正相输入和 2 个反向输入通道可以选择, 正相输入包含 1 个外部引脚和 4 个 OPA0-3 的输出, 反相输入包含 1 个外部引脚和 CRV 电压分压值.

CMP4-5 的输出可以选择 TIM8 的煞车输入

18.2.4 比较器用法

比较器CMP4/5 比较所选择的INP 和INM 端口上的信号，具体流程如下：

1. 配置 CMP_x_CR2 寄存器的 INP_SEL 位和 INM_SEL 位，选择所要比较的信号；
2. 配置 CMP_x_CR1 寄存器的 EN 位，比较器开始上电工作；
3. 比较的结果存放于 CMP_x_CR1 寄存器的 OUT 位。

另外，当CMP4 和CMP5 的INM_SEL 选择CRV 时，需要配置CMP_CR2 寄存器的CRV_SEL 位，然后将CRV_EN 置位。

18.2.5 比较器锁定机制

比较器能用于安全的用途，比如过流或者过热保护。在某些特定的安全需求的应用中，有必要保证比较器设置不能被无效寄存器访问或者程序计数器破坏所改变。为了这个目的，比较器控制和状态寄存器可以设为写保护（只读）。一旦设置完成，LOCK 位必须设为1，这导致整个CMP_x_CR2 寄存器变成只读，包括 LOCK 位在内。写保护只能被MCU 复位所清除。

18.2.6 迟滞现象

比较器的可配置迟滞电压能防止无效的输出生成的噪声信号。在不强制迟滞电压 的情况下迟滞现象可以被禁止。

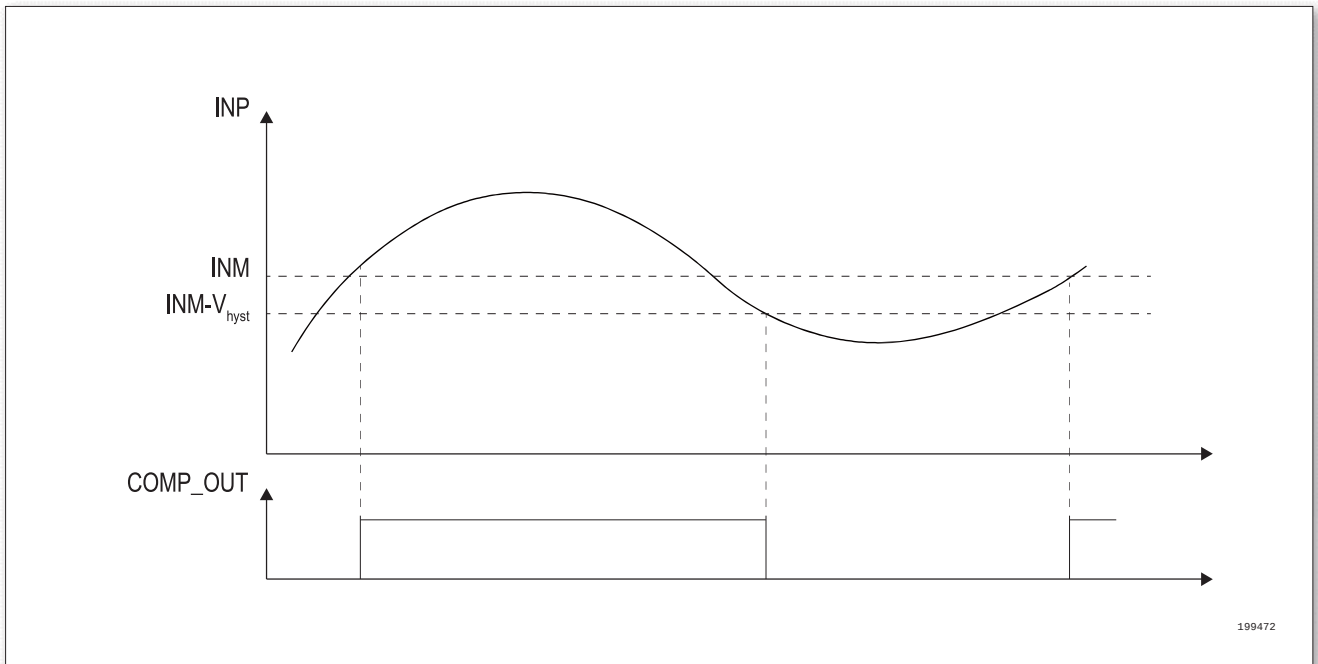


图29. 比较器的迟滞现象

18.3 比较器寄存器说明

18.3.1 控制寄存器 (CMPx_CR1)

CMPx_CR1 (控制寄存器)			基地址: 0x4000E000, 0x4000E010, 0x4000E020, 0x4000E030 0x4000E040 偏移地址: 00H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X				FHYST		RHYST	
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	FIF	RIF	FIE	RIE	X	OFLT		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	LOCK	OUT	X		POL	MODE		EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
FHYST	下降迟滞电压 00: 0mV 01: 5mV 10: 10mV

	11: 20mV
RHYST	上升迟滞电压 00: 0mV 01: 5mV 10: 10mV 11: 20mV
FIF	1: 下降沿中断事件发生 0: 无下降沿中断事件
RIF	1: 上升沿中断事件发生 0: 无上升沿中断事件
FIE	1: 下降沿中断使能 0: 下降沿中断禁止
RIE	1: 上升沿中断使能 0: 上升沿中断禁止
OFLT	比较器的输出滤波, 连续的 PCLK 时钟比较输出不变则认为有效, 否则保持不变 000b: 1 个时钟周期, 无滤波 001b: 4 个时钟周期 010b: 16 个时钟周期 011b: 32 个时钟周期 100b: 64 个时钟周期 101b: 128 个时钟周期 110b: 256 个时钟周期 111b: 512 个时钟周期
LOCK	1: CR2 只能 read 0: CR2 能 read/write Note: CMP4 和 CMP5 有效
OUT	比较器的输出, read only 1: 高输出 0: 低输出 NOTE: CPU 读取 OUT 值时, OFLT 值必须设为 001b 以上
POL	输出极性 1: 反相输出 0: 同相输出
MODE	00b: 极低功率 01b: 低功率 10b: 中等速率 11b: 高速率
EN	1: 比较器使能 0: 比较器禁止

18.3.2 控制寄存器 (CMPx_CR2)

CMPx_CR2 (控制寄存器)			基地址: 0x4000E030, 0x4000E040 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X					INP_SEL[2:0]		
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	INM_SEL[1:0]		CRV_SRC	CRV_EN	CRV_SEL[2:0]			

Write:								
Reset:	0	0	1	1	0	0	0	0

位	功能描述
INP_SEL[2:0]	0:PB5(CMP4), PB3(CMP5) 1:PGA0_out 2:PGA1_out 3:PGA2_out 4:OPA3_out
INM_SEL[1:0]	0:PB6(CMP4), PB4(CMP5) 1:CRV
CRV_SRC	比较器外部参考电压源选择 1: AVCC 0: VBGP(1.2V)
CRV_EN	1: 比较器外部参考电压使能 0: 比较器外部参考电压禁止 Note: CMP4 和 CMP5 的 CRV_EN 为相同讯号, 当 CMP4 的 CRV_EN 写入 1, CMP5 的 CRV_EN 也会变成 1
CRV_SEL[3:0]	比较器外部参考电压选择: 0000b: 1/20 AVCC 0001b: 2/20 AVCC 0010b: 3/20 AVCC 0011b: 4/20 AVCC 0100b: 5/20 AVCC 0101b: 6/20 AVCC 0110b: 7/20 AVCC 0111b: 8/20 AVCC 1000b: 9/20 AVCC 1001b: 10/20 AVCC 1010b: 11/20 AVCC 1011b: 12/20 AVCC 1100b: 13/20 AVCC 1101b: 14/20 AVCC 1110b: 15/20 AVCC 1111b: 16/20 AVCC

Note: 此寄存器只存在于 CMP4, CMP5

18.3.3 CMPx_CAL

CMPx_CAL			基地址: 0x4000E000, 0x4000E010, 0x4000E020, 0x4000E030 0x4000E040 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X				CAL_NADJ[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X		CAL_NEN	CAL_PEN	CAL_PADJ[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
CAL_NADJ	CMP NMOS offset 调节
CAL_NEN	CMP NMOS 校正使能位 1: Enable 0: Disable
CAL_PEN	CMP PMOS 校正使能位 1: Enable 0: Disable
CAL_PADJ[3:0]	CMP PMOS offset 调节

19 OPAMP 运算放大器

19.1 运算放大器简介

芯片内嵌四个运算放大器，每个运算放大器的输入输出都连接到I/O，通过共享I/O 可以与ADC，比较器相连。

19.2 运算放大器主要特征

- 轨对轨输入/输出
- 输出连接到 I/O 上
- DOffset cancel

19.3 运算放大器功能说明

19.3.1 控制寄存器（OPA_CR）

OPA_CR (控制寄存器)			基地址: 0x4000D000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	OPA3_EN	PGA2_Gain			PGA2_EN
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	PGA1_Gain			PGA1_EN	PGA0_Gain			PGA0_EN
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
OPA3_EN	1:OPA3 使能 0:OPA3 禁止
PGA2_Gain	PGA2 放大倍率选择 000b:1X 001b:2X 010b:4X 011b:8X

	100b:16X 101b:24X 110b:32X 111b:48X (测试, 视测试状况释出)
PGA2_EN	1:PGA2 使能 0:PGA2禁止
PGA1_Gain	PGA1放大倍率选择 000b:1X 001b:2X 010b:4X 011b:8X 100b:16X 101b:24X 110b:32X 111b:48X (测试, 视测试状况释出)
PGA1_EN	1:PGA1使能 0:PGA1禁止
PGA0_Gain	PGA0放大倍率选择 000b:1X 001b:2X 010b:4X 011b:8X 100b:16X 101b:24X 110b:32X 111b:48X (测试, 视测试状况释出)
PGA0_EN	1:PGA0使能 0:PGA0禁止

19.3.2 控制寄存器(OPA_CR2)

OPA_CR2 (控制寄存器)			基地址: 0x4000D000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X				OPHD3	OPHD2	OPHD1	OPHD0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
OPHD3	OPA3 High drive enable 1: High drive enable 0: High drive disable
OPHD2	PGA2 High drive enable 1: High drive enable 0: High drive disable
OPHD1	PGA1 High drive enable



	1: High drive enable 0: High drive disable
OPHD0	PGA0 High drive enable 1: High drive enable 0: High drive disable

20 CRC 模块

20.1 CRC 概述

CRC 模块含有可编程的多项式设置，支持常用的几个 CRC 标准。

- 支持三个常用的多项式：CRC-CCITT、CRC-16 和 CRC-32
CRC-CCITT: $x^{16}+x^{12}+x^5+1$
CRC-16: $x^{16}+x^{15}+x^2+1$
CRC-32: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
- 支持输入数据按字节颠倒控制
- 支持输出数据颠倒控制和输出数据异或操作选择
- 支持初始化种子数据可编程
- 接受任何字长的数据写入：8 位、16 位或 32 位
8 位数据写入：1 cpu clk
16 位数据写入：2 cpu clk
32 位数据写入：4 cpu clk

常用参数模型：

CRC-CCITT- FALSE	CRC-16/X25	CRC-16	CRC-32
Width: 16 Poly: 0x1021 Init: 0xFFFF RefIn: False RefOut: False XorOut: 0x0000	Width: 16 Poly: 0x1021 Init: 0xFFFF RefIn: True RefOut: True XorOut: 0xFFFF	Width: 16 Poly: 0x8005 Init: 0x0000 RefIn: True RefOut: True XorOut: 0x0000	Width: 32 Poly: 0x04C11DB7 Init: 0xFFFFFFFF RefIn: True RefOut: True XorOut: 0xFFFFFFFF

注：Width: 数据宽度

Poly: 多项式生成项的简写。以 16 进制表示， $x^{16}+x^{12}+x^5+1$ 即是 0x11021。忽略了最高位的“1”，即完整的生成项是 0x1021

Init: 初始化预设种子数据

RefIn: 输入数据每个字节顺序颠倒，True 表示每个字节计算前先颠倒，即 bit0 为最高位，bit7 为最低位；False 表示不颠倒

RefOut: 输出数据顺序颠倒，True 表示计算结束后，计算结果先颠倒，再进入 XorOut 处理；False 表示不颠倒

XorOut: 与所选模式位宽相同的值（全零或全 1），全 1 表示 CRCCON 寄存器的 bitXorOut 置 1，此时结果按位取反；全 0 表示 CRCCON 寄存器的 bitXorOut 清 0，此时结果不操作

20.2 特殊功能寄存器列表

CRC 模块寄存器基地址: 0x40021000				
偏移地址	名称	读写方式	复位值	功能描述
0x00	CRCCON	R/W	0x0000	CRC 控制寄存器
0x04	CRCDAT	R/W	0x0000FFFF	CRC 数据寄存器
0x08	CRCINIT	R/W	0x000FFFFF	CRC 初始化种子寄存器

20.3 特殊功能寄存器说明

20.3.1 控制寄存器(CRCCON)

CRCCON (控制寄存器)			基地址: 0x40021000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	XorOut	RefOut	RefIn	X	X	MODE1	MODE0
Write:						RESET		
Reset:	0	0	0	0	0	0	0	0

位	功能描述												
XorOut	输出数据按位异或控制 0: 输出数据不操作 1: 输出数据按位取反操作												
RefOut	输出数据顺序颠倒控制: 0: 输出数据顺序不颠倒 1: 输出数据顺序颠倒, 按所选 CRC 模式数据位长颠倒												
RefIn	输入数据顺序颠倒控制: 0: 输入数据顺序不颠倒 1: 输入数据顺序颠倒, 按每byte颠倒												
RESET	CRC计算RESET位 复位CRC计算, 但不改变当前模式, 并根据当前模式设置CRCDAT寄存器; 只能对该位写 ‘1’ , 由硬件自动清零。												
MODE[1:0]	CRC计算模式控制位 <table><tr><th colspan="2">MODE[1:0]</th><th>CRC计算模式</th></tr><tr><td>0</td><td>0</td><td>CRC-CCITT</td></tr><tr><td>0</td><td>1</td><td>CRC-16</td></tr><tr><td>1</td><td>X</td><td>CRC-32</td></tr></table>	MODE[1:0]		CRC计算模式	0	0	CRC-CCITT	0	1	CRC-16	1	X	CRC-32
MODE[1:0]		CRC计算模式											
0	0	CRC-CCITT											
0	1	CRC-16											
1	X	CRC-32											

20.3.2 数据寄存器(CRCDAT)

CRCDAT (数据寄存器)			基地址: 0x40021000 偏移地址: 04H					
	Bit31...Bit16							
Read:	DAT[31:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15...Bit0							
Read:	DAT[15:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
DAT[31:0]	CRC数据寄存器位 写入该寄存器时，作为数据输入寄存器； 读取该寄存器时，返回CRC计算的结果。

20.3.3 CRC 初始化种子寄存器(CRCINIT)

CRCINIT (CRC 初始化种子寄存器)			基地址: 0x40021000 偏移地址: 08H					
	Bit31...Bit16							
Read:	DAT[31:16]							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15...Bit0							
Read:	DAT[15:0]							
Write:								
Reset:	1	1	1	1	1	1	1	1

位	功能描述
DAT[31:0]	CRC初始化数据寄存器 若模式选择CRC-CCITT或CRC-16，则低16位（DAT[15:0]）有效，高16位无效 若模式选择CRC-32，则32位（DAT[31:0]）有效

21 DMA 功能

21.1 概述

DMA 模块提供高速的数据传输在外设和 RAM 之间或者 RAM 和 RAM 之间，在 DMA 数据搬运的过程中不需要 CPU 的参与，但是占用系统总线。

21.2 功能描述

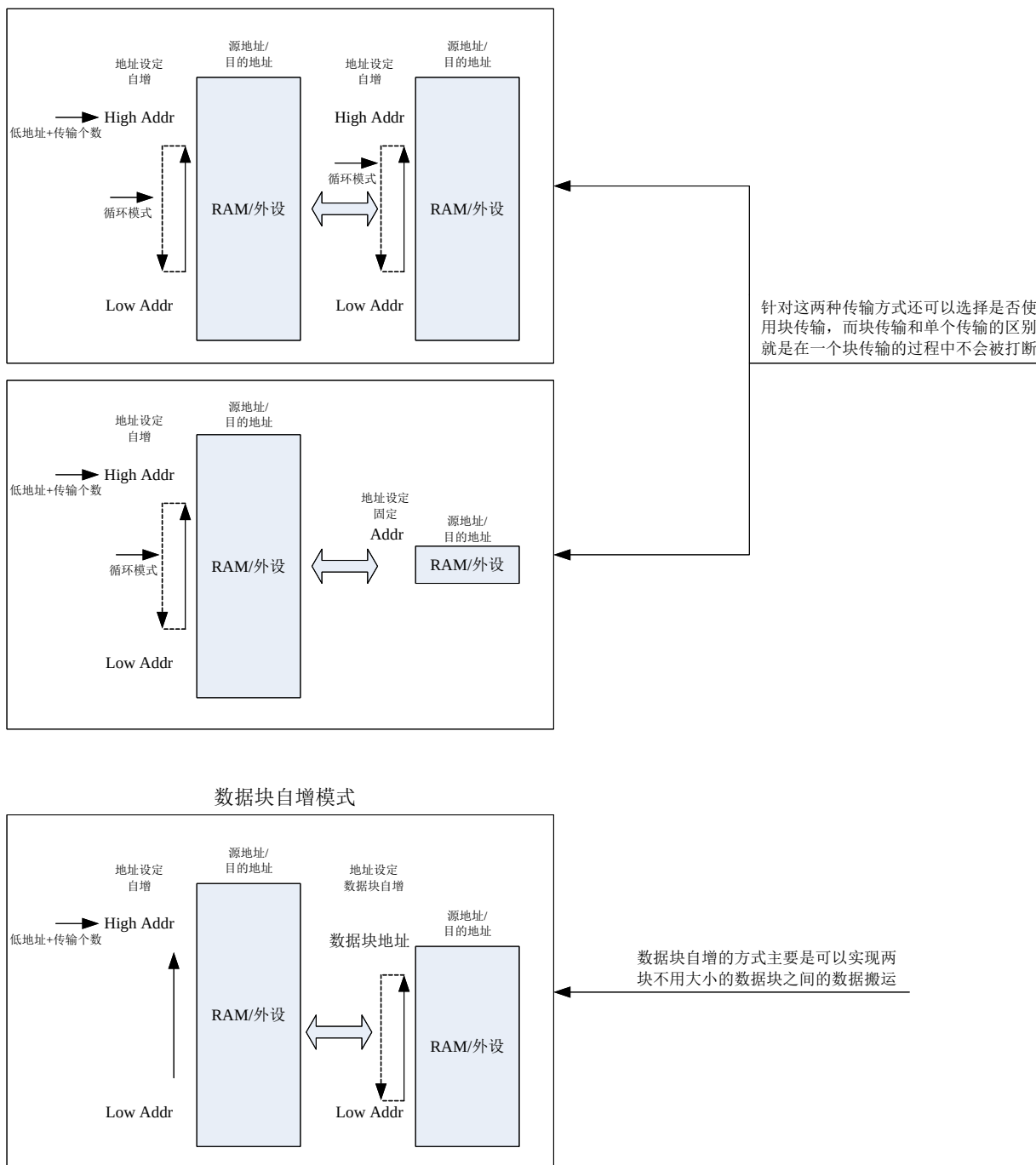
- DMA 模块共有 3 个独立的可配置信道
- DMA 的传输可以外设到外设，外设到存储器，存储器到外设，存储器到存储器
- DMA 与 CPU 之间交替占用总线
- DMA 支持最多 64 个外部请求源，每个通道还有一个软件请求源
- 三个通道有优先级的设置，一旦一个通道的一次传输开始后不能被打断。当三个通道同时请求 DMA 传输时，优先级高的被响应。优先级低的通道只有在高优先级的通道传输完成后才能被响应
- 当两个通道选择了同一个外部请求源时，只有优先级高的通道响应此请求
- DMA 支持两种传输方式：一次请求传输一个数据、一次请求传输所有的数据
- DMA 支持三种中断：传输完成中断，传输错误中断，块传输中断
- 在非循环模式下，DMA 完成数据传输后自动关闭相应通道，当选择为循环模式的时候，用户可配置循环次数 CHNxULKNUM[8:15]，达到循环次数后自动关闭相应通道

21.3 DMA 通道请求列表

DMA_CTL[8...13]	说明
0	Soft request
1	UART0发送
2	UART0接收
3	UART1发送
4	UART1接收
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	SPI0发送
18	SPI0接收
19	I2C发送
20	I2C接收
21	SAR_ADC
22	TIM8_UP
23	TIM8_TRIG
24	Timer0
25	Timer1
26	Timer2
27	Timer3
28	TIM8_COM
29	TIM8_CH1
30	TIM8_CH2
31	TIM8_CH3
32	TIM8_CH4
33	TIM8_CH5

注：当选择DMA发送或接收串口数据（UART）时，UART对应的TXIF/RXIF会置位。

21.4 DMA 数据传输说明



21.5 特殊功能寄存器列表

DMA 模块寄存器基地址: 0x40020000				
偏移地址	名称	读写方式	复位值	功能描述
00H	DMAIE	R/W	0000H	中断使能寄存器
04H	DMAIF	R/W	0000H	通道中断标志
08H	CHNSTA	R/W	0000H	通道0状态寄存器
0CH	CHNOCTL	R/W	0000H	通道0控制寄存器
10H	CHN0SRC	R/W	0000H	信道0源地址寄存器
14H	CHNOTAR	R/W	0000H	通道0目的地址寄存器
18H	CHNOCNT	R/W	0000H	通道0传输数量设置寄存器
1CH	CHNOTCCNT	R	0000H	通道0传输完成数据个数
20H	CHNOBULKNUM	R/W	0000H	通道0块传输个数设置
24H	CHN1CTL	R/W	0000H	通道1控制寄存器
28H	CHN1SRC	R/W	0000H	信道1源地址寄存器
2CH	CHN1TAR	R/W	0000H	通道1目的地址寄存器
30H	CHN1CNT	R/W	0000H	通道1传输数量设置寄存器
34H	CHN1TCCNT	R	0000H	通道1传输完成数据个数
38H	CHN1BULKNUM	R/W	0000H	通道1块传输个数设置
3CH	CHN2CTL	R/W	0000H	通道2控制寄存器
40H	CHN2SRC	R/W	0000H	信道2源地址寄存器
44H	CHN2TAR	R/W	0000H	通道2目的地址寄存器
48H	CHN2CNT	R/W	0000H	通道2传输数量设置寄存器
4CH	CHN2TCCNT	R	0000H	通道2传输完成数据个数
50H	CHN2BULKNUM	R/W	0000H	通道2块传输个数设置

21.6 特殊功能寄存器说明

21.6.1 DMA 中断使能寄存器(DMAIE)

DMAIE (DMA 中断使能寄存器)			基地址: 0x40020000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	TEIE2	TEIE1	TEIE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	BCIE2	BCIE1	BCIE0	X	TCIE2	TCIE1	TCIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TCIE[2...0]	通道 0/1/2 传输结束中断使能 0: 禁止

	1: 使能
BCIE[2...0]	通道 0/1/2 块传输中断使能 0: 禁止 1: 使能
TEIE[2...0]	通道 0/1/2 传输错误中断使能 0: 禁止 1: 使能

21.6.2 DMA 中断标志寄存器(DMAIF)

DMAIF (DMA 中断标志寄存器)			基地址: 0x40020000 偏移地址: 04H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	TEIF2	TEIF1	TEIF0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	BCIF2	BCIF1	BCIF0	X	TCIF2	TCIF1	TCIF0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
TCIF[2...0]	通道 0/1/2 传输结束中断标志 0: 未产生中断 1: 产生中断
BCIF[2...0]	通道 0/1/2 块传输完成中断标志 0: 未产生中断 1: 产生中断
TEIF[2...0]	通道 0/1/2 传输错误中断标志 0: 未产生中断 1: 产生中断

21.6.3 DMA 状态寄存器(CHNSTA)

CHNSTA (DMA 状态寄存器)			基地址: 0x40020000 偏移地址: 08H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	X	X	X	X	X	X	X
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	BUSY2	BUSY1	BUSY0
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY[2...0]	通道 0/1/2 传输 BUSY 标志 0: 空闲 1: 忙碌

21.6.4 DMA 通道控制寄存器(CHNxCTL)

CHNxCTL (DMA 通道控制寄存器)			基地址: 0x40020000 偏移地址: 0CH, 24H, 3CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X	Channe	Channe	Channe	Channe	Channe	Channe	DESTIN
Write:		15	14	13	12	11	10	_INC1
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	DESTIN_	SOURC_	SOURC_	CYCLE	MODE	SIZE1	SIZE0	DMA_CH
Write:	INC0	INC1	INC0					
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

位	功能描述
Channel[5...0]	触发通道选择, 具体参见 DMA 通道请求列表来定
DESTIN_INC[1...0]	目的地址增量模式 00: 不增加 01: 增加 10: 数据块内循环增加 11: 数据块内循环增加
SOURC_INC[1...0]	源地址地址增量模式 00: 不增加 01: 增加 10: 数据块内循环增加 11: 数据块内循环增加
MODE	传输模式 0: 单次传输模式 1: 块传输模式 (1 块数据传输过程中不会被打断)
CYCLE	循环模式 0: 不循环模式 1: 循环模式
PSIZE[1...0]	MEMORY 的传送位数 (外设的传送位数固定为 32bit): 00: 8 位 01: 16 位 10: 32 位 11: 32bit
DMA_CHNOEN	DMA 通道使能 0: 禁止 1: 使能

21.6.5 DMA 信道源地址寄存器(CHNxSRC)

CHNxSRC (DMA 信道源地址寄存器)		基地址: 0x40020000 偏移地址: 10H, 28H, 40H	
	Bit31	Bit30 ... Bit1	Bit0
Read:	ADDR31	ADDR30 ... ADDR1	ADDR0
Write:			
Reset:	0	0	0

注: x 为 0, 1, 2

位	功能描述
ADDR[31..0]	数据传输源地址寄存器

21.6.6 DMA 通道目的地址寄存器(CHNxTAR)

CHNxTAR (DMA 通道目的地址寄存器)		基地址: 0x40020000 偏移地址: 14H, 2CH, 44H	
	Bit31	Bit30 ... Bit1	Bit0
Read:	ADDR31	ADDR30 ... ADDR1	ADDR0
Write:			
Reset:	0	0	0

注: x 为 0, 1, 2

位	功能描述
ADDR[31..0]	数据传输目的地址寄存器

21.6.7 DMA 通道传输数量寄存器 (CHNxCNT)

CHNxCNT (DMA 通道传输数量寄存器)		基地址: 0x40020000 偏移地址: 18H, 30H, 48H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

位	功能描述
Num[15...0]	DMA 数据传输个数设置寄存器 最大设置到 65535 个传输数据 如果用户设置的是块传输, 那么该寄存器则表示用户需要传输的数据块个数

21.6.8 DMA 通道已传输数据个数 (CHNxTCCNT)

CHNxTCCNT	基地址: 0x40020000
-----------	-----------------

(DMA 通道已传输数据个数)			偏移地址: 1CH, 34H, 4CH					
	Bit15	14	13	12	11	10	9	Bit8
Read:	Num15	Num14	Num13	Num12	Num11	Num10	Num9	Num8
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:	X	X	X	X	X	X	X	X
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

位	功能描述
Num[15...0]	指示 DMA 已经传输完成的数据个数 如果用户设置的是块传输, 那么该寄存器则表示 DMA 已经传输完成的数据块个数

21.6.9 DMA 通道块传输设置寄存器 (CHNx BULKNUM)

CHNx BULKNUM (DMA 通道块传输设置寄存器)			基地址: 0x40020000 偏移地址: 20H, 38H, 50H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	CYCLE7	CYCLE6	CYCLE5	CYCLE4	CYCLE3	CYCLE2	CYCLE1	CYCLE0
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	Num7	Num6	Num5	Num4	Num3	Num2	Num1	Num0
Write:								
Reset:	0	0	0	0	0	0	0	0

注: x 为 0, 1, 2

位	功能描述
CYCLE[7...0]	循环次数选择, 当用户选择 DMA 传输为循环模式时起作用: =0: 无限次循环 =0x01----0xFF: 为 0x01----0xFF 次循环
NUM[7...0]	块传输数据个数选择, 表示每一个数据块内有多少个数据: 0 — 255 个 如果用户设置的是块传输, 那么该寄存器则表示每一个数据块内有多少个数据

22 Cortex-M0 内核简要说明

22.1 概述

32-bit 处理器基于一个高集成度、低功耗的 32 位处理器内核，采用一个 3 级流水线冯·诺伊曼结构 (Von Neumann architecture)。通过简单、功能强大的指令集以及全面优化的设计（提供包括一个单周期乘法器在内的高端处理硬件），32-bit 处理器可实现极高的能效。

32-bit 处理器基于 16 位的 Thumb 指令集，并包含 Thumb-2 技术。提供了一个现代 32 位结构所希望的出色性能，代码密度比其他 8 位和 16 位微控制器都要高。

32-bit CPU core 不支持内存保护模块 (MPU)。

22.2 系统定时器 SysTick

推荐用户用 CMSIS 接口函数 `SysTick_Config(uint32_t ticks)` 去配置系统定时器。

比如 `Fcpu` 为 11010048Hz，如果将 SysTick 配置为 100ms 周期中断，则只需按照如下方式调用函数即可：

```
SysTick_Config (11010048/10);
```

注意：由于 SysTick 计数器只有 24 位，用户需保证 ticks 不能超过 0xFFFFFF。

22.3 中断优先级说明

Cortex-M0 处理器紧密集成了一个可配置的嵌套向量中断处理器 (NVIC)，提供业界领先的中断性能。NVIC 具有以下功能：

- 包含一个不可屏蔽的中断 (NMI)，优先级固定为-2，为最高
- 包含一个异常处理中断 (HardFault)，优先级固定为-1，仅低于 NMI
- 其余中断可以配置 4 个优先级，但优先级都低于以上两个中断

4 个可配置中断优先级为 0x0, 0x1, 0x2, 0x3，其中 0x0 优先级最高，0x3 优先级最低。

推荐用户用 CMSIS 接口函数 `NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)` 去配置中断优先级。比如配置 DMA 中断优先级为 0（最高），则可以按照如下方式调用函数即可：

```
NVIC_SetPriority (DMA_IRQn, 0x0);
```

注：DMA_IRQn 为 DMA 中断的中断号。更多中断号参见 8.1 中断向量说明

22.4 CMSIS 函数说明

以下表格罗列了部分 CMSIS 函数，方便用户配置中断等相关功能。建议用户统一使用 CMSIS 函数去配置内核寄存器，不要直接去操作内核寄存器。

CMSIS 函数	函数说明
<code>__enable_irq()</code>	使能全局中断
<code>__disable_irq()</code>	关闭全局中断 (注: NMI 和 HardFault 不会被屏蔽)
<code>void NVIC_EnableIRQ(IRQn_Type IRQn)</code>	使能中断号为 IRQn 的中断 (注: 需满足 $IRQn \geq 0$, 对 $IRQn < 0$ 的中断, 没有中断使能位)
<code>void NVIC_DisableIRQ(IRQn_Type IRQn)</code>	屏蔽中断号为 IRQn 的中断 (注: 需满足 $IRQn \geq 0$, 对 $IRQn < 0$ 的中断, 没有中断使能位)
<code>uint32_t NVIC_GetPendingIRQ(IRQn_Type IRQn)</code>	获取中断号为 IRQn 中断的挂起状态
<code>void NVIC_SetPendingIRQ(IRQn_Type IRQn)</code>	设置中断号为 IRQn 中断的挂起状态 (注: 如果对应的中断已经使能, 则调用此函数会触发芯片进入相应的中断处理程序)
<code>void NVIC_ClearPendingIRQ(IRQn_Type IRQn)</code>	清除中断号为 IRQn 中断的挂起状态 (注: 进入相应中断处理程序后, 中断挂起状态会被自动清除)
<code>void NVIC_SetPriority(IRQn_Type IRQn, uint32_t priority)</code>	设置中断号为 IRQn 中断的优先级 (注: NMI 和 HardFault 不可设置优先级, 优先级设置范围为 0x0—0x3)
<code>uint32_t NVIC_GetPriority(IRQn_Type IRQn)</code>	获取中断号为 IRQn 的中断优先级
<code>void NVIC_SystemReset(void)</code>	系统软复位
<code>uint32_t SysTick_Config(uint32_t ticks)</code>	系统定时器配置 (注: ticks 不能超过 0xFFFFF)

注: 中断号 IRQn 详细见 8.1 中断向量说明

23 电机专用协同处理器

RX32S10 提供一电机专用模块(Motor Engine), 简称 ME.

ME 里面包含 32bit 通用型除法器, 开平方硬件加速器(SQRT), PID 硬件运算器, 坐标转换(CDTR), IPD, SMO 以及 SVPWM 的硬件运算器.

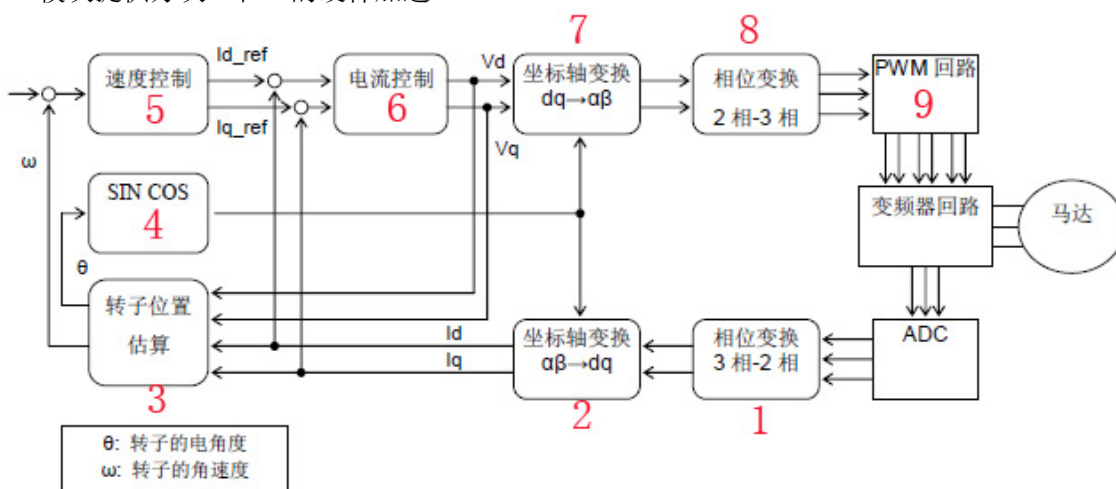
坐标转换(CDTR)包含下图的方块 1, 2, 7.

SIN COS Table 包含下图的方块 4

SVPWM 包含方块 8 和 9

SMO 包含方块 3

PID 模块提供方块 5 和 6 的硬件加速



23.1 DIV 模块

23.1.1 概述

RX32S10 系列提供一个标准的 32bit 除法器, 可以设定成有符号或无符号除法, 被除数放在 DD, 除数放在 DI, 商数放在 DQ, 余数放在 DR.

23.1.2 除法器的使用方法

1. 被除数写入到 DD, 除数写入到 DI
2. CR 的 STR 写入 1, 启动 32bit 除法运算
3. 当运算完后, 可以透过 CR 的 RDY 是否为 1 来判断, 软件可以写入 0 清除
4. 读取 DQ 为商, DR 为余数

Note:

1. 当除法器启动运算后, 在 CR. RDY=1 之前, 再次设定 CR. STR=1 则忽略, 必须等到上一笔除法运算完成才能再次启动运算
2. 当除法运算中检测到除数 DI 为 0, DIVF 会被置位为 1
3. CR. SN 选择有符号或无符号运算必须在启动 CR. STR 以前设置好

23.1.3 DIV 寄存器

DIV_CR			基地址: 0x40022000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	BUSY	DIVF	SN	RDY	STR
Write:				X				
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY	1: DIV 运算中 0: DIV 闲置
DIVF	1: 运算错误, 除数为 0, 软件可清除 0: 运算正常
SN	1: 有符号 0: 无符号
RDY	1: 除法运算完成 0: 除法器闲置或运算中
STR	1: 启动除法运算, 下一个 clock 硬件清除为 0 0: 未启动除法运算

DIV_DD		基地址:	0x40022000
		偏移地址:	04H
	Bit31...Bit0		
Read:	DD[31:0]		
Write:			
Reset:	0		

位	功能描述
DD[31:0]	除法运算的被除数, 必须在 CR. STR 设定为 1 前, 将 DD 值写入. 当 CR. STR 启动后, 修改 DD 值并不影响当前运算结果

DIV_DI		基地址:	0x40022000
		偏移地址:	08H
	Bit31...Bit0		
Read:	DI[31:0]		
Write:			
Reset:	1		

位	功能描述
DI[31:0]	除法运算的除数, 必须在 CR. STR 设定为 1 前, 将 DI 值写入. 当 CR. STR 启动后, 修改 DI 值并不影响当前运算结果. Note: DI 值不能为 0

DIV_DQ		基地址:	0x40022000
		偏移地址:	0CH
	Bit31...Bit0		
Read:	DQ[31:0]		
Write:			
Reset:	0		

位	功能描述
DQ[31:0]	除法运算后的商数

DIV_DR		基地址:	0x40022000
		偏移地址:	10H
	Bit31...Bit0		
Read:	DR[31:0]		
Write:			
Reset:	0		

位	功能描述
DR[31:0]	除法运算后的余数

23.2 SQRT 模块

23.2.1 概述

RX32S10 系列提供一个标准的开平方硬件加速器, 输入寄存器 DI 为 32bit, 输出寄存器 DO 为 16bit

23.2.2 SQRT 使用方法

1. 将需要开平方的数值写入到 DI, 自动启动开平方运算
2. 运算完成后, 可以透过 CR 的 RDY 是否为 1 来判断, 软件可以写 0 清除
3. 读出 DO 为开平方运算后的结果

Note:

1. SQRT 必须等到上一笔运算完成才能再次启动运算

23.2.3 SQRT 寄存器

SQRT_CR			基地址: 0x40023000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	X	BUSY	RDY	X
Write:						X		
Reset:	0	0	0	0	0	0	0	0

位	功能描述
BUSY	1: SQRT 运算中 0: SQRT 闲置
RDY	1: 开平方运算完成 0: 开平方硬件闲置或运算中

SQRT_DI		基地址: 0x40023000
		偏移地址: 04H
	Bit31...Bit0	
Read:	DI[31:0]	
Write:		
Reset:	0	

位	功能描述
DI[31:0]	开平方运算的输入值. 写入数据后自动启动 SQRT 运算

SQRT_D0		基地址: 0x40023000
		偏移地址: 08H
	Bit15...Bit0	
Read:	D0[15:0]	
Write:		
Reset:	0	

位	功能描述
D0[15:0]	开平方运算的输出值

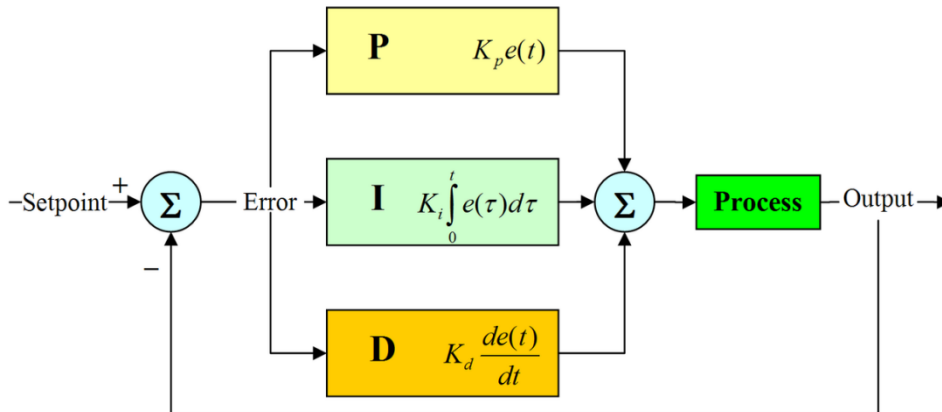
23.3 PID 模块

23.3.1 概述

PID 主要是由三个计算单元组成, 比例单元(P), 积分单元(I) 和微分单元(D), 主要公式如下:

$$OUT = K_p \times e(t) + K_i \times \int_0^t e(T) dT + K_d \frac{de(t)}{dt}$$

透过调整 K_p, K_i, K_d 的增益来调整其特性.



RX32S10 系列提供 3 组 PID 硬件加速的配置, 透过灵活的参数配置调整计算的方式.

23.3.2 PID 使用方法

1. 参考值写入到 REF, 量测到反馈值写入到 FB
2. CR. STR 写入 1, 启动 PID 运算
3. 运算完成后, 可以透过 CR 的 RDY 是否为 1 来判断, 软件可以写 0 清除
4. 读出 OUT 为 PID 运算完的结果

Note:

1. 将 KPG 写入 0 则不计算 P 单元, KIG 写入 0 则不计算 I 单元, KDG 写入 0 则不计算 D 单元.
2. 在启动 PID 运算前, 必须先将 PID 相关参数寄存器配置好 (KPG, KIG, KDG, KIMULP, DIV, INTGLIM, OUTLIM), 当启动 PID 运算过程中, 更动这些参数值并不影响当前运算.

23.3.3 PID 寄存器

PID_CR			基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 00H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X							
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	X	X	X	BUSY	RDY	CLR	STR
Write:					X			

Reset:	0	0	0	0	0	0	0	0
--------	---	---	---	---	---	---	---	---

位	功能描述
BUSY	1: PID 运算中 0: PID 闲置
RDY	1:PID 运算完成 0:PID 硬件闲置或运算中
CLR	1:清除 INTG & ERR 为 0 0:不清除 INTG & ERR
STR	1:启动 PID 运算, 下一个 clock 硬件清除为 0 0:未启动 PID 运算

PID_REF	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 04H
	Bit15...Bit0
Read:	REF[15:0]
Write:	
Reset:	0

位	功能描述
REF[15:0]	输入的 Reference 值. $ERR = REF - FB$.

PID_FB	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 08H
	Bit15...Bit0
Read:	FB[15:0]
Write:	
Reset:	0

位	功能描述
FB[15:0]	输入的 Feedback 值. $ERR = REF - FB$.

PID_OUT	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 0CH
	Bit15...Bit0
Read:	OUT[15:0]
Write:	X
Reset:	0

位	功能描述
OUT[15:0]	PID 运算的结果

PID_ERR	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 10H
	Bit15...Bit0
Read:	ERR[15:0]
Write:	
Reset:	0

位	功能描述
ERR[15:0]	PID 运算后保留的 Error 值. $ERR = REF - FB$.

PID_INTG	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 14H
	Bit31...Bit0
Read:	INTG[31:0]
Write:	
Reset:	0

位	功能描述
INTG[31:0]	PID 运算后保留的 Integral 值.

PID_KPG	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 18H
	Bit15...Bit0
Read:	KPG[15:0]
Write:	
Reset:	0

位	功能描述
KPG[15:0]	KP Gain 值

PID_KIG	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 1CH
	Bit15...Bit0
Read:	KIG[15:0]
Write:	
Reset:	0

位	功能描述
KPG[15:0]	KI Gain 值

PID_KDG	基地址: 0x40024000, 0x40025000, 0x40026000 偏移地址: 20H
	Bit15...Bit0
Read:	KDG[15:0]
Write:	
Reset:	0

位	功能描述
KDG[15:0]	KD Gain 值

PID_KIMULP	基地址: 0x40024000, 0x40025000, 0x40026000
------------	---

		偏移地址: 24H
		Bit15...Bit0
Read:	KIMULP[15:0]	
Write:		
Reset:	0	

位	功能描述
KIMULP[15:0]	KI 放大倍率值

PIDDIV			基地址: 0x40024000, 0x40025000, 0x40026000					
			偏移地址: 28H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X				KDD[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	KID[3:0]				KPD[3:0]			
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
KPD[3:0]	KP 除 2 的倍数, 默认为 0
KID[3:0]	KI 除 2 的倍数, 默认为 0
KDD[3:0]	KD 除 2 的倍数, 默认为 0

PID_INTGLIM		基地址: 0x40024000, 0x40025000, 0x40026000
		偏移地址: 2CH
		Bit31...Bit0
Read:	INTGLIM[31:0]	
Write:		
Reset:	0	

位	功能描述
INTGLIM[31:0]	限制 Integral 的上限为 INTGLIM, 下限为 -INTGLIM

PID_OUTLIM		基地址: 0x40024000, 0x40025000, 0x40026000
		偏移地址: 30H
		Bit15...Bit0
Read:	OUTLIM[15:0]	
Write:		
Reset:	0	

位	功能描述
OUTLIM [15:0]	限制 OUT 的上限为 OUTLIM, 下限为 -OUTLIM

23.4 SIN COS Table

23.4.1 使用方法

1. 写入 Angle 启动运算
2. 读出 Cos, Sin, IPDSin 值

23.5 坐标转换

23.5.1 使用方法

1. 设定坐标转换的通道 CDTR_CH[2:0]=0b111
2. 写入 Ia, Ib, Ic, Cos, Sin, Vq, Vd
3. CR.CDTR_STR 写入 1 启动坐标转换运算
4. 运算是否完成, 可以透过 CR.CDTR_RDY 为 1 来判断, 软件可以写 0 清除
5. 转换完的数值为 Ialpha, Ibeta, Iq, Id, Valpha, Vbeta

Note:

1. CDTR_CH[2:0]的配置

CDTR_CH[2:0]	项目
0b000	不启动运算
0b001	Clarke
0b010	Park
0b011	Clarke, Park
0b100	Rev Park
0b101	Clarke, Rev Park
0b110	Park, Rev Park
0b111	Clarke, Park, Rev Park

2. 当同时启动 Clarke, Park 的转换, 必须先算完 Clarke 后才能算 Park
3. Rev Clarke 运算整合到 SVPWM 的运算里面
4. 当启动坐标转换运算过程中, 更动 (Ia, Ib, Ic, Vd, Vq, Cos, Sin) 参数值并不影响当前运算.

23.6 IPD

23.6.1 使用方法

1. 写入 BEMFA 和 BEMFB
2. CR. IPD_STR 写入 1 启动 IPD 运算
3. 运算是否完成, 可以透过 CR. IPD_RDY 为 1 来判断, 软件可以写 0 清除
4. 读取 IPDTheta

Note:

1. 当 CR. IPD_STR 启动后, 修改 BEMFA 或 BEMFB 不影响当前运算结果

23.7 SMO

23.7.1 使用方法

1. 写入 SMOIalpha, SMOIbeta, Valpha, Vbeta
2. CR. SMO_STR 写入 1 启动 SMO 运算
3. 运算是否完成, 可以透过 CR. SMO_RDY 为 1 来判断, 软件可以写 0 清除
4. 读取 SMTheta

Note:

1. 在启动 SMO 运算前, 必须先将 SMO 相关参数寄存器配置好 (Kslid, Kslf, KF, KG, E0), 当启动 SMO 运算过程中, 更动这些参数值并不影响当前运算.
2. 当 CR. SMO_STR 启动后, 修改 SMOIalpha, SMOIbeta, Valpha, Vbeta 不影响当前运算结果

23.8 SVPWM

23.8.1 使用方法

1. 写入 Valpha, Vbeta
2. CR. SV_STR 写入 1 启动 SMO 运算
3. 运算是否完成, 可以透过 CR. SV_RDY 为 1 来判断, 软件可以写 0 清除
4. 读出 PDCH1, PDCH2, PDCH3, SVZone

Note:

1. 在启动 SVPWM 运算前, 必须先将 SVPWM 相关参数寄存器配置好 (FOVM, TPWM, LSMIN), 当启动 SVPWM 运算过程中, 更动这些参数值并不影响当前运算.
2. 当 CR.SV_STR 启动后, 修改 Valpha, Vbeta 不影响当前运算结果

23.9 中断功能

ME 中断列表如下

中断事件	事件标志	使能位
除法器错误事件	ME_IFR.DIVFF	ME_IER.DIVFE
PID1 完成事件	ME_IFR.PID1F	ME_IER.PID1E
PID2 完成事件	ME_IFR.PID2F	ME_IER.PID2E
PID3 完成事件	ME_IFR.PID3F	ME_IER.PID3E
坐标转换完成事件	ME_IFR.CDTRF	ME_IER.CDTRE
IPD 完成事件	ME_IFR.IPDF	ME_IER.IPDE
SMO 完成事件	ME_IFR.SMOF	ME_IER.SMOE
SVPWM 完成事件	ME_IFR.SVF	ME_IER.SVE
FOC 运算完成事件	ME_IFR.F4F	ME_IER.F4E

Note:

1. 当 ME_CR.F4_STR 写入 1, 依序启动 PID1, PID2, SVPWM, SMO 运算
2. ME_IER.F4E 启动中断, 当 PID1, PID2, SVPWM, SMO 都运算完成后, 会产生一个中断事件 (ME_IFR.F4F)

23. 10ME 寄存器

ME_CR			基地址: 0x40028000 偏移地址: 00H					
	Bit23	22	21	20	19	18	17	Bit16
Read:	X							F4_STR
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit15	14	13	12	11	10	9	Bit8
Read:	SV_BUS Y	SMO_BU SY	IPD_BU SY	CDTR_B USY	SV_RDY	SV_STR	SMO_RD Y	SMO_ST R
Write:	X	X	X	X				
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	IPD_RD	IPD_ST	CDTR_R	CDTR_S	SMO_CL	CDTR_CH[2:0]		
Write:	Y	R	DY	TR	R			
Reset:	0	0	0	0	0	1	1	1

位	功能描述
F4_STR	1: 依序启动 PID1, PID2, SVPWM, SMO 运算 0: 未启动
SV_BUSY	1: SVPWM 运算中 0: SVPWM 闲置
SMO_BUSY	1: SMO 运算中 0: SMO 闲置
IPD_BUSY	1: IPD 运算中 0: IPD 闲置
CDTR_BUSY	1: CDTR 运算中 0: CDTR 闲置
SV_RDY	1: SVPWM 运算完成 0: SVPWM 硬件闲置或运算中
SV_STR	1: 启动 SVPWM 运算, 下一个 clock 硬件清除为 0 0: 未启动 SVPWM 运算
SMO_RDY	1: SMO 运算完成 0: SMO 硬件闲置或运算中
SMO_STR	1: 启动 SMO 运算, 下一个 clock 硬件清除为 0 0: 未启动 SMO 运算
IPD_RDY	1: IPD 运算完成 0: IPD 硬件闲置或运算中
IPD_STR	1: 启动 IPD 运算, 下一个 clock 硬件清除为 0 0: 未启动 IPD 运算
CDTR_RDY	1: 坐标转换运算完成 0: 坐标转换硬件闲置或运算中
CDTR_STR	1: 启动坐标转换运算, 下一个 clock 硬件清除为 0 0: 未启动坐标转换运算
SMO_CLR	1: 清除 Ealpha, Ebeta, Zalpha, Zbeta, IalphaError, IbetaError, EstIalpha, EstIbeta 为 0

	0:不清 0 动作
CDTR_CH[2:0]	坐标转换通道选择,可透过下列组合 BIT0:写入 1,坐标转换进行 Clarke 运算,反之写 0 BIT1:写入 1,坐标转换进行 Park 运算,反之写 0 BIT2:写入 1,坐标转换进行 Rev Park 运算,反之写 0

ME_Ia	基地址: 0x40028000 偏移地址: 04H
	Bit15...Bit0
Read:	Ia[15:0]
Write:	
Reset:	0

位	功能描述
Ia[15:0]	Ia 值. Clarke 运算的输入

ME_Ib	基地址: 0x40028000 偏移地址: 08H
	Bit15...Bit0
Read:	Ib[15:0]
Write:	
Reset:	0

位	功能描述
Ib[15:0]	Ib 值. Clarke 运算的输入

ME_Ic	基地址: 0x40028000 偏移地址: 0CH
	Bit15...Bit0
Read:	Ic[15:0]
Write:	
Reset:	0

位	功能描述
Ic[15:0]	Ic 值. Clarke 运算的输入

ME_Ialpha	基地址: 0x40028000 偏移地址: 10H
	Bit15...Bit0
Read:	Ialpha[15:0]
Write:	
Reset:	0

位	功能描述
Ialpha[15:0]	Ialpha 值. Clarke 运算的输出, Park 运算的输入

ME_Ibeta		基地址:	0x40028000
		偏移地址:	14H
	Bit15…Bit0		
Read:	Ibeta[15:0]		
Write:			
Reset:	0		

位	功能描述
Ialpha[15:0]	Ibeta 值. Clarke 运算的输出, Park 运算的输入

ME_Iq		基地址:	0x40028000
		偏移地址:	18H
	Bit15…Bit0		
Read:	Iq[15:0]		
Write:			
Reset:	0		

位	功能描述
Iq[15:0]	Iq 值. Park 运算的输出

ME_Id		基地址:	0x40028000
		偏移地址:	1CH
	Bit15…Bit0		
Read:	Id[15:0]		
Write:			
Reset:	0		

位	功能描述
Id[15:0]	Id 值. Park 运算的输出

ME_Vq		基地址:	0x40028000
		偏移地址:	20H
	Bit15…Bit0		
Read:	Vq[15:0]		
Write:			
Reset:	0		

位	功能描述
Vq[15:0]	Vq 值. Rev Park 运算的输入

ME_Vd		基地址:	0x40028000
		偏移地址:	24H
		Bit15...Bit0	

Read:	Vd[15:0]
Write:	
Reset:	0

位	功能描述
Vd[15:0]	Vd 值. Rev Park 运算的输入

ME_Valpha	基地址: 0x40028000 偏移地址: 28H
	Bit15...Bit0
Read:	Valpha[15:0]
Write:	
Reset:	0

位	功能描述
Valpha[15:0]	Valpha 值. Rev Park 运算的输出, SVPWM 运算的输入

ME_Vbeta	基地址: 0x40028000 偏移地址: 2CH
	Bit15...Bit0
Read:	Vbeta[15:0]
Write:	
Reset:	0

位	功能描述
Vbeta[15:0]	Vbeta 值. Rev Park 运算的输出, SVPWM 运算的输入

ME_Angle	基地址: 0x40028000 偏移地址: 30H
	Bit15...Bit0
Read:	Angle[15:0]
Write:	
Reset:	0

位	功能描述
Angle[15:0]	输入角度值, 范围为 0~1023, 当进行坐标转换 Park 和 Rev Park 前, 要先写入此寄存器

ME_Cos	基地址: 0x40028000 偏移地址: 34H
	Bit15...Bit0
Read:	Cos[15:0]
Write:	
Reset:	0

位	功能描述
Cos[15:0]	根据 Angle 值, 输出对应的 Cos table 值

ME_Sin	基地址: 0x40028000 偏移地址: 38H
	Bit15...Bit0
Read:	Sin[15:0]
Write:	
Reset:	0

位	功能描述
Sin[15:0]	根据 Angle 值, 输出对应的 Sin table 值

ME_IPDSin	基地址: 0x40028000 偏移地址: 3CH
	Bit15...Bit0
Read:	IPDSin[15:0]
Write:	
Reset:	0

位	功能描述
IPDSin[15:0]	根据 Angle 值, 输出对应的 IPD Sin 值

ME_Kslid	基地址: 0x40028000 偏移地址: 40H
	Bit15...Bit0
Read:	Kslid[15:0]
Write:	
Reset:	0

位	功能描述
Kslid[15:0]	Kslid 参数

ME_Kslf	基地址: 0x40028000 偏移地址: 44H
	Bit15...Bit0
Read:	Kslf[15:0]
Write:	
Reset:	0

位	功能描述
Kslf[15:0]	Kslf 参数

ME_KF	基地址: 0x40028000 偏移地址: 48H
-------	------------------------------

	Bit15...Bit0
Read:	KF[15:0]
Write:	
Reset:	0

位	功能描述
KF[15:0]	KF 参数

ME_KG	基地址: 0x40028000 偏移地址: 4CH
	Bit15...Bit0
Read:	KG[15:0]
Write:	
Reset:	0

位	功能描述
KG[15:0]	KG 参数

ME_E0	基地址: 0x40028000 偏移地址: 50H
	Bit15...Bit0
Read:	E0[15:0]
Write:	
Reset:	16384

位	功能描述
E0[15:0]	E0 参数

ME_Ealpha	基地址: 0x40028000 偏移地址: 54H
	Bit31...Bit0
Read:	Ealpha[31:0]
Write:	
Reset:	0

位	功能描述
Ealpha[31:0]	Ealpha 值

ME_Ebeta	基地址: 0x40028000 偏移地址: 58H
	Bit31...Bit0
Read:	Ebeta[31:0]
Write:	
Reset:	0

位	功能描述
Ebeta[31:0]	Ebeta 值

ME_Zalpha	基地址: 0x40028000 偏移地址: 5CH
	Bit31...Bit0
Read:	Zalpha[31:0]
Write:	
Reset:	0

位	功能描述
Zalpha[31:0]	Zalpha 值

ME_Zbeta	基地址: 0x40028000 偏移地址: 60H
	Bit31...Bit0
Read:	Zbeta[31:0]
Write:	
Reset:	0

位	功能描述
Zbeta[31:0]	Zbeta 值

ME_IalphaError	基地址: 0x40028000 偏移地址: 64H
	Bit31...Bit0
Read:	IalphaError[31:0]
Write:	
Reset:	0

位	功能描述
IalphaError[31:0]	IalphaError 值

ME_IbetaError	基地址: 0x40028000 偏移地址: 68H
	Bit31...Bit0
Read:	IbetaError[31:0]
Write:	
Reset:	0

位	功能描述
IbetaError [31:0]	IbetaError 值

ME_EstIalpha	基地址: 0x40028000
--------------	-----------------

	偏移地址: 6CH
	Bit31...Bit0
Read:	EstIalpha[31:0]
Write:	
Reset:	0

位	功能描述
EstIalpha[31:0]	EstIalpha 值

ME_EstIbeta	基地址: 0x40028000 偏移地址: 70H
	Bit31...Bit0
Read:	EstIbeta[31:0]
Write:	
Reset:	0

位	功能描述
EstIbeta[31:0]	EstIbeta 值

ME_SMOTheta	基地址: 0x40028000 偏移地址: 74H
	Bit15...Bit0
Read:	SMOTheta[15:0]
Write:	
Reset:	0

位	功能描述
SMOTheta[15:0]	SMOTheta 值

ME_SMOIalpha	基地址: 0x40028000 偏移地址: 78H
	Bit15...Bit0
Read:	SMOIalpha[15:0]
Write:	
Reset:	0

位	功能描述
SMOIalpha [15:0]	SMOIalpha 值

ME_SMOIbeta	基地址: 0x40028000 偏移地址: 7CH
	Bit15...Bit0
Read:	SMOIbeta[15:0]
Write:	
Reset:	0

位	功能描述
SMOIbeta [15:0]	SMOIbeta 值

ME_BEMFA		基地址:	0x40028000
		偏移地址:	80H
	Bit15…Bit0		
Read:	BEMFA[15:0]		
Write:			
Reset:	0		

位	功能描述
BEMFA[15:0]	BEMFA 值

ME_BEMFB		基地址:	0x40028000
		偏移地址:	84H
	Bit15…Bit0		
Read:	BEMFB[15:0]		
Write:			
Reset:	0		

位	功能描述
BEMFB[15:0]	BEMFB 值

ME_IPDTheta		基地址:	0x40028000
		偏移地址:	88H
	Bit15…Bit0		
Read:	IPDTheta[15:0]		
Write:			
Reset:	0		

位	功能描述
IPDTheat[15:0]	IPDTheta 值

ME_SVZone		基地址:	0x40028000
		偏移地址:	90H
	Bit7...Bit0		
Read:	SVZone[7:0]		
Write:			
Reset:	0		

位	功能描述
SVZone[7:0]	SVPWMZone 值, SVPWM 运算的输出

ME_FOVM		基地址:	0x40028000
		偏移地址:	94H
		Bit15...Bit0	

Read:	FOVM[15:0]
Write:	
Reset:	0

位	功能描述
FOVM[15:0]	输入 PWM Full scale 值

ME_TPWM		基地址:	0x40028000
		偏移地址:	98H
	Bit15...Bit0		
Read:	TPWM[15:0]		
Write:			
Reset:	0		

位	功能描述
TPWM[15:0]	限制 PDCHx 的最大值

ME_LSMIN		基地址:	0x40028000
		偏移地址:	9CH
	Bit15…Bit0		
Read:	LSMIN[15:0]		
Write:			
Reset:	0		

位	功能描述
LSMIN[15:0]	限制 PDCHx 的最小值

ME_PDCH1		基地址:	0x40028000
		偏移地址:	A0H
	Bit31…Bit0		
Read:	PDCH1[31:0]		
Write:			
Reset:	0		

位	功能描述
PDCH1[31:0]	PwmDutyCH1 值, SVPWM 运算的输出

ME_PDCH2		基地址:	0x40028000
		偏移地址:	A4H
	Bit31…Bit0		
Read:	PDCH2[31:0]		
Write:			
Reset:	0		

位	功能描述
PDCH2[31:0]	PwmDutyCH2 值, SVPWM 运算的输出

ME_PDCH3		基地址: 0x40028000 偏移地址: A8H
	Bit31...Bit0	
Read:	PDCH3[31:0]	
Write:		
Reset:	0	

位	功能描述
PDCH3[31:0]	PwmDutyCH3 值, SVPWM 运算的输出

ME_IER		基地址: 0x40028000 偏移地址: B0H						
	Bit15	14	13	12	11	10	9	Bit8
Read:	X			F4E	SVE	SMOE	IPDE	CDTRE
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	PID3E	PID2E	PID1E	X			DIVFE
Write:								
Reset:	0	0	0	0	0	0	0	0

位	功能描述
F4E	FOC 计算中断使能 1: 使能 0: 禁止
SVE	SVPWM 中断使能 1: 使能 0: 禁止
SMOE	SMO 中断使能 1: 使能 0: 禁止
IPDE	IPD 中断使能 1: 使能 0: 禁止
CDTRE	坐标转换中断使能 1: 使能 0: 禁止
PID3E	PID3 中断使能 1: 使能 0: 禁止
PID2E	PID2 中断使能 1: 使能 0: 禁止
PID1E	PID1 中断使能 1: 使能 0: 禁止
DIVFE	DIV 错误中断使能 1: 使能 0: 禁止

ME_IFR			基地址: 0x40028000 偏移地址: B4H					
	Bit15	14	13	12	11	10	9	Bit8
Read:	X			F4F	SVF	SMOF	IPDF	CDTRF
Write:								
Reset:	0	0	0	0	0	0	0	0
	Bit7	6	5	4	3	2	1	Bit0
Read:	X	PID3F	PID2F	PID1F	X			DIVFF
Write:								
Reset:	0	0	0	0	0	0	0	0
位		功能描述						
F4F		FOC 计算旗标 1: 发生 FOC 计算中断事件, 写入 0 清除 0: 无 SVPWM 中断事件						
SVF		PID1 中断旗标 1: 发生 PID1 中断事件, 写入 0 清除 0: 无 PID1 中断事件						
SMOF		PID2 中断旗标 1: 发生 PID2 中断事件, 写入 0 清除 0: 无 PID2 中断事件						
IPDF		PID3 中断旗标 1: 发生 PID3 中断事件, 写入 0 清除 0: 无 PID3 中断事件						
CDTRF		保留						
PID3F		CDTR 中断旗标 1: 发生 CDTR 中断事件, 写入 0 清除 0: 无 CDTR 中断事件						
PID2F		IPD 中断旗标 1: 发生 IPD 中断事件, 写入 0 清除 0: 无 IPD 中断事件						
PID1F		SMO 中断旗标 1: 发生 SMO 中断事件, 写入 0 清除 0: 无 SMO 中断事件						
DIVFF		SVPWM 中断旗标 1: 发生 SVPWM 中断事件, 写入 0 清除 0: 无 SVPWM 中断事件						

24 电气规格（具体参数需芯片测试）

24.1 极限参数

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	输入电源	输入电源引脚上的电压	2.0 <i>*1</i>		6.5	V
Vi	输入电压	所有的数字引脚	-0.3		5.5	V
Via	ADCINx 输入极限电压	测试 ADCINx 引脚输入电压，高于此电压可能会导致该引脚损坏	-0.3		AVCC	V
Idd	输入电流	VCC 电源引脚	-0.3		AVCC	mA
Iss	地上电流	所有 GND 引脚				mA
Tstg	存储温度	芯片的极限存储温度				°C
VESDhbm	静态 ESD (HBM)	芯片所有的引脚				V
VESDmm	静态 ESD (MM)	芯片所有的引脚				V
VESDcdm	静态 ESD (CDM)	芯片所有的引脚				V

注：*1. 若 BOR 关闭，最小工作电压为 2.0v，若 BOR 开启，最小工作电压为当前 BOR 设置档位电压的阈值下限。

*2. 待测。

24.2 功耗参数

24.2.1 模块功耗

符号	参数说明	测试条件	最小	典型	最大	单位
Ivccdet	VCC 检测模块功耗	常温，VCC=5.0V				uA
Ibor	BOR 检测模块功耗	常温，VCC=5.0V				uA
Ihrc	HRC 功耗	常温，VCC=5.0V				uA
Ilrc	LRC 功耗	-45°C~125°C，VCC=5.0V				uA
Ipll	PLL 功耗	常温，VCC=5.0V				uA
Irtc	RTC 功耗	-45°C~125°C，VCC=2.0~5.0V				uA

24.2.2 运行功耗

符号	参数说明	测试条件	最小	典型	最大	单位
Ihold	Hold 功耗	常温, VCC=5.0v				uA
Isleep	Sleep 功耗	常温, VCC=5.0v				uA
Irunhrc	HRC 下运行功耗	高频 RC 配置为 11M, CLKOUT 关闭, 所有数字模块打开 / 关闭				mA
		高频 RC 配置为 5.5M, CLKOUT 关闭, 所有数字模块打开 / 关闭				mA
		高频 RC 配置为 2.75M, CLKOUT 关闭, 所有数字模块打开 / 关闭				mA
		高频 RC 配置为 1.375M, CLKOUT 关闭, 所有数字模块打开 / 关闭				mA
		高频 RC 配置为 700k, CLKOUT 关闭, 所有数字模块打开 / 关闭				mA
Irunpll	PLL 下运行功耗	CPU 运行 PLL 时钟 42M, 所有数字模块打开 / 关闭				mA
		CPU 运行 PLL 时钟 22M, 所有数字模块打开 / 关闭				mA
		CPU 运行 PLL 时钟 11M, 所有数字模块打开 / 关闭				mA
		CPU 运行 PLL 时钟 5.5M, 所有数字模块打开 / 关闭				mA
		CPU 运行 PLL 时钟 2.75M, 所有数字模块打开 / 关闭				mA
		CPU 运行 PLL 时钟 1.375M, 所有数字模块打开 / 关闭				mA
Irunosc	Crystal 4~16MHz 下运行功耗	常温, VCC=5.0v, 所有数字模块关闭				uA

24.3 DC/AC 参数

24.3.1 存储器

符号	参数说明	测试条件	最小	典型	最大	单位
FlashSize	Flash 空间大小			32K		bytes
InfoSize	Information Block 空间大小			2K		bytes

RamSize	Ram 空间大小			4K		bytes
Tflashrd	Flash 字节读取时间		40			ns
Tflashwr	Flash 字节写时间				20	us
Tflashper	Flash 页擦除时间				2	ms
Tflashmer	Flash 全擦除时间				10	ms
FPageSize	Code Flash 页面大小			1K		bytes/page
InPageSize	Information Block 页面大小			1K		bytes/page
Numwr	擦写次数		100,000			次
Tdat	数据保持时间		10			years
Tmprun	操作温度		-40		85	°C
Vram	RAM 数据保持电压			0.3		V

24.3.2 GPIO

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	输入电源	输入电源引脚上的电压, BOR 关闭	2.0	3.3	5.5	V
		输入电源引脚上的电压, BOR 开启	2.2	3.3	5.5	V
Vih1	高电平输入电压	Reset 引脚, TEST 引脚	0.8VCC			V
Vih2		支持 TTL 输入管脚 PC4~PC10, PC13, PC14, PH0~4, PE0, PE3, PE6,	0.6VCC <i>*1</i>			V
Vih3		除了电源和地, 除了 Reset, TEST, TTL 输入引脚 之外的所有引脚	0.7VCC			V
Vil	低电平输入电压	除了电源和地之外的所有引脚			0.2VCC	V
Ioh	高电平输出电流	VCC=5.0V I/O 口上输出高电压 Voh 降低到 0.9VCC	5	8		mA
Iol	低电平输出电流	VCC=5.0V I/O 口上输出低电压 Vol 降低到 0.1VCC	8	10		mA
Rpull	输入上拉电阻	VCC=5.0V, 除 test 其余 GPIO 引脚		88		kΩ
		VCC=5.0V, test 引脚		10		kΩ
Iif	输入漏电流	I/O 配置为输入, 上拉关闭, 引脚接 VCC 或 GND			0.1	uA
Iod	开漏输出漏电流	I/O 配置为输出高, 开漏开启, 引脚接 GND			0.1	uA

24.3.3 CMU

符号	参数说明	测试条件	最小	典型	最大	单位
Fpll	内部 PLL 时钟频率	PLL_GAIN [3:0] *1			84	MHz
	PLL 锁定时间					ms
	P11 环路倍频比	PLL_GAIN [3:0]	2		32	倍
	PLL 工作电压	工作温度-40℃~125℃	1.35	1.5	1.65	V
	PLL 模块功耗					uA
Fhse	外部高频晶体时钟频率		4		16	MHz
	Fosc 起振时间			12		ms
	OSC 工作电压	工作温度-40℃~125℃	2.5	5	5.5	V
	OSC 模块功耗					nA
Fhrc	内部高频 RC 时钟频率	Temp=-40 ~125 degree 43 degree 单点校正	20.202	21	21.126	MHz
	HRC 频率可调范围	寄存器 HRCADJ 调整, 7bit 控制位	16	21	26.668	MHz
	HRC 起振时间					us
	HRC 工作电压	工作温度-40℃~125℃				V
	HRC 模块功耗					uA
Flrc	内部低频 RC 时钟频率	工作温度-40℃~125℃ trim 后输出频率	27	32	37	KHz
	起振时间					us
	LRC 工作电压	工作温度-40℃~125℃				V
	LRC 模块功耗					uA

*1: 修改 PLL_GAIN [3:0]可调整 PLL 环路倍频比, 进而调整 PLL 输出频率, 对应关系如下表

OSC(MHz)	Feedback DIV	OUT DIV	PLL OUT(MHz)
4	2 ~ 32		8 ~ 64
6	2 ~ 14		12 ~ 84
8	2 ~ 10		16 ~80
10	2 ~ 8		20 ~ 80
12	2 ~ 6		24 ~ 72
14	2 ~ 6		28 ~ 84
16	2 ~ 4		32 ~ 64

24.3.4 PMU

符号	参数说明	测试条件	最小	典型	最大	单位
POR	上电复位阈值					V
	最慢上电速率					V/ms
LBOR	LBOR 复位阈值	上电翻转电压	1.50	1.77	2.00	V
		下点翻转电压	1.40	1.63	1.80	V
		迟滞电压				mV

VCC_DET	VCC 检测阈值电压	常温，中心值， VCC_LVL[3:0] = 0000				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0001				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0010				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0011				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0100				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0101				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0110				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 0111				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 1000				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 1001				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 1010				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 1011				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 1100				V
		迟滞电压				mV
		常温，中心值， VCC_LVL[3:0] = 1101/1110/1111				V
		迟滞电压				mV
	VCC_DET 抗噪滤波	-45℃~125℃	60		280	us
	VCC 检测模块功耗	常温，VCC=5.0V				uA
BOR_DET	BOR 检测阈值电压	常温，中心值， BOR_LVL[1:0] = 00				V
		迟滞电压				mV

		常温，中心值，BOR_LVL [1:0] = 01				V
		迟滞电压				mV
		常温，中心值，BOR_LVL [1:0] = 10				V
		迟滞电压				mV
		常温，中心值，BOR_LVL [1:0] = 11				V
		迟滞电压				mV
	BOR 抗噪滤波	-45℃~125℃				us
POW_DET	低功耗检测阈值	POW_LVL[2:0]= 000				mV
		POW_LVL[2:0]= 001				mV
		POW_LVL[2:0]= 010				mV
		POW_LVL[2:0]= 011				mV
		POW_LVL[2:0]= 100				mV
		POW_LVL[2:0]= 101				mV
		POW_LVL[2:0]= 110				mV
		POW_LVL[2:0]= 111				mV
	推荐滤波电容					uF
	推荐采样电阻					Ω
LDO_1P5	内部 1.5v 稳压源输出	-45℃~125℃，VCC=2.0~5.5v				V
	温度系数	-45℃~125℃，VCC=2.0~5.5v				ppm/℃
	外置负载电容					uF
LDO_LP	内部低功耗稳压源输出	-45℃~125℃，VCC=2.0~5.5v				V
	温度系数	-50℃~125℃				mV

24.3.5 中断&RESET

符号	参数说明	测试条件	最小	典型	最大	单位
Tana2us	外部中断/rx 2us 模拟滤波	VCC=5V，-45℃~125℃				us
Tholdwk	中断唤醒时间	Hold 唤醒				Fsys clk

***1:** 此处时间仅为唤醒信号触发到跳转至中断入口的时钟开销，详细中断处理过程及时钟开销如下（外部中断）：

Hold 唤醒过程	系统时钟开销(Fsys)
1. 开启 cpu 时钟	
2. 传递信号至 cpu	
3. 压栈	
4. 跳转至中断入口	
5. 跳转至中断处理程序并执行	
6. 出栈	
7. 返回主程序	

符号	参数说明	测试条件	最小	典型	最大	单位
----	------	------	----	----	----	----

Twarmup	复位预热时间	POR, LBOR, BOR, EXTRST, SOFTPOR		1024		clk
		除上述复位, 其他复位				clk
Textrst	复位脉冲宽度	EXTRST				us

24.3.6 UART/SPI/I2C

符号	参数说明	测试条件	最小	典型	最大	单位
Bandrate	UART 波特率	常温, Uart0/1 *1	2400		115200	Mbps
	SPI 通讯时钟	从机, 时钟 8 分频 *2		5.5	10	MHz
	I2C 通讯时钟				400	KHz

***1:** UART 通讯速率受限于 RX 引脚内建模拟 2us 滤波,
RX32S10 系列 RX0/1 内建负脉冲 2us 模拟滤波, 最大波特率 115200bps
所有的 RX 和 INTX 引脚的模拟滤波都可以控制关闭/开启, 可以通讯更高速率, 详见 GPIO 章节
附表: 常用波特率设置及误差

波特率	SREL(十进制数)	波特率误差	UART0/UART1 收发状态
VCC=3.1V, Fsys = PLL 21MHz			
			持续收发正常 (常温)
			持续收发正常 (常温)
			持续收发正常 (常温)
			持续收发正常 (常温)
VCC=3.1V, Fsys = PLL 42MHz, Catch_EN=1 (指令缓存功能)			
2.5Mbps	8	—	持续收发正常 (常温)

***2:** SPI 主机时钟可以最大 22MHz (PLL 44M/2), 从机模式受限于时钟采样开销, 最大仅为 8 分频系统时钟。

24.3.7 WDT

符号	参数说明	测试条件	最小	典型	最大	单位
	WDT 定时时间	最小时间 *1		64		ms
		最大时间		16384		ms
	WDT 误差	同 LRC 精度	-60.3		52.6	%
	功耗					uA

***1:** WDT 时钟源固定为内部 LRC 时钟

24.3.8 TIMER

符号	参数说明	测试条件	最小	典型	最大	单位
	定时器位宽			16		Bit
	时钟分频		1		65536	
	捕获沿宽度					
	事件计数频率					
	PWM 周期					Ftmer clk
	PWM 占空比					Ftmer clk
	死区宽度					Ftmer clk

24.3.9 CRC

符号	参数说明	测试条件	最小	典型	最大	单位
	CRC 支持写入数据位宽		8	16	32	bit
	CRC 运算时间	8bit 数据写入				Fcpu clk
		16bit 数据写入				Fcpu clk
		32bit 数据写入				Fcpu clk
	CRC 支持多项式	CRC-CCITT:	$x^{16}+x^{12}+x^5+1$			
		CRC-16:	$x^{16}+x^{15}+x^2+1$			
		CRC-32:	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$			

附表：常用参数模型：

CRC-CCITT- FALSE	CRC-16/X25	CRC-16	CRC-32
Width: 16 Poly: 0x1021 Init: 0xFFFF RefIn: False RefOut: False XorOut: 0x0000	Width: 16 Poly: 0x1021 Init: 0xFFFF RefIn: True RefOut: True XorOut: 0xFFFF	Width: 16 Poly: 0x8005 Init: 0x0000 RefIn: True RefOut: True XorOut: 0x0000	Width: 32 Poly: 0x04C11DB7 Init: 0xFFFFFFFF RefIn: True RefOut: True XorOut: 0xFFFFFFFF

注：Width：数据宽度

Poly：多项式生成项的简写。以 16 进制表示， $x^{16}+x^{12}+x^5+1$ 即是 0x11021。忽略了最高位的“1”，即完整的生成项是 0x1021

Init：初始化预设种子数据

RefIn：输入数据每个字节顺序颠倒，True 表示每个字节计算前先颠倒，即 bit0 为最高位，bit7 为最低位；False 表示不颠倒

RefOut：输出数据顺序颠倒，True 表示计算结束后，计算结果先颠倒，再进入 XorOut 处理；False 表示不颠倒

XorOut：与所选模式位宽相同的值（全零或全 1），True 表示 RefOut 的结果按位取反；False 表示 RefOut 的结果不操作

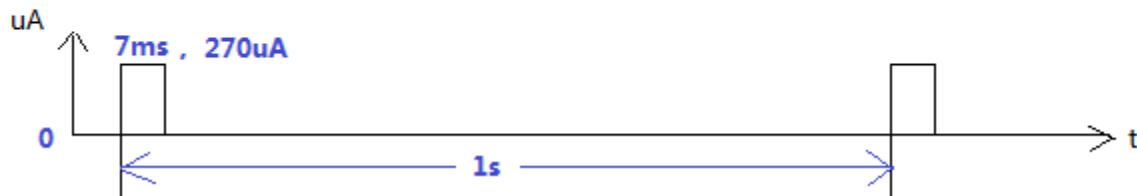
24.3.10 TPS&ADC

符号	参数说明	测试条件	最小	典型	最大	单位
VCC	工作电压		2.4		5.5	V
Iadc	工作电流					uA
Res	分辨率					Bit
ENOB	有效位					Bit
Fsample	工作频率	非快速模式				Hz

		快速模式				
Cin	输入电容					pF
Rin	输入电阻（等效）					Ω
ADCINx 通道						
Vadcin	ADC 输入范围	Vrefadc=AVCC, 常温	0		AVCC	mV
Resadcin	分辨率	理论				mV/LSB
	计算公式	常温, VCC=3.3V, 非快速模式, 实际拟合	VADCINx = 其中: VADCIN0 为实际 ADC 测量电压 (mV)			mV
温度传感器 TPS 通道						
Adjtps	温度检测精度					℃
Restps	分辨率	理论				℃/LSB
	计算公式					℃

附：分时示意图

TPS 分时1s开启一次的 功耗&开启时间 示意图



24.3.11 RTC

符号	参数说明	测试条件	最小	典型	最大	单位
	RTC 工作电压		2.0		5.5	V
	Vrtc 功耗	-45°C~125°C, VCC=2.0~5.0V		1.3	2.5 <i>*1</i>	uA
	内部高频补偿精度	PLL 时钟开启		0.06		ppm
	低频补偿精度			30		ppm
	单点校正精度	误差系数为平均系数, 多功能烧写器进行单点常温校正 -45~125°C		6	10	ppm
	TOUT 输出频率			参见下表		

注：

***1** 最大功耗发生在高温高压条件下。此功耗仅为 TPS 关闭时 VRTC 电压域下功耗, TPS 分时开启功耗详见 TBS 章节电器参数。

附：TOUT 输出配置表

TOUT[3:0]				TOUT	TOUT (PLL 使能)
0	0	0	0	0	0
0	0	0	1	1	1

0	0	1	0	32768Hz	32768Hz
0	0	1	1	未经高频补偿的 1Hz	高频补偿得到的 1Hz
0	1	0	0	未经高频补偿的 2Hz	高频补偿得到的 2Hz
0	1	0	1	未经高频补偿的 4Hz	高频补偿得到的 4Hz
0	1	1	0	未经高频补偿的 8Hz	高频补偿得到的 8Hz
0	1	1	1	未经高频补偿的 16Hz	高频补偿得到的 16Hz
1	0	0	0	未经高频补偿的 32Hz	高频补偿得到的 32Hz
1	0	0	1	未经高频补偿的 64Hz	高频补偿得到的 64Hz
1	0	1	0	未经高频补偿的 128Hz	高频补偿得到的 128Hz
其他				Reserved	

24.3.12 HRC/LRC 自校正

符号	参数说明	测试条件	最小	典型	最大	单位
	校正基准时钟源	内部 RTC 时钟		32		Hz
		外部灌入		1		Hz
	HRC 频率调整位	HRCADJ	0x00		0x7F	
	Step 精度			0.5%		1/1sb
	HRC 测量中心频率			10.5		MHz
	调整后精度			0.3%	*1	
	LRC 频率调整位	LRCADJ	0x00		0xFF	
	Step 精度					1/1sb
	LRC 测量中心频率			32.768		KHz
	调整后精度			3%	*2	

注:

*1: 当 HRC 测到的频率与中心频率误差在 0.3%以内, 则不作调整; 否则会一直测频

*2: 当 LRC 测到的频率与中心频率误差在 3%以内, 则不作调整; 否则会一直测频

24.3.13 M0 内核

符号	参数说明	测试条件	最小	典型	最大	单位
	内核位宽			32		bit
	乘法器运算时间	32bit 乘法		1		Cycle
	结构					
	指令集			Thumb		
	Systick 位宽			24		bit
	Systick 时钟源			Fcpu		

3 级流水线冯·诺伊曼结构

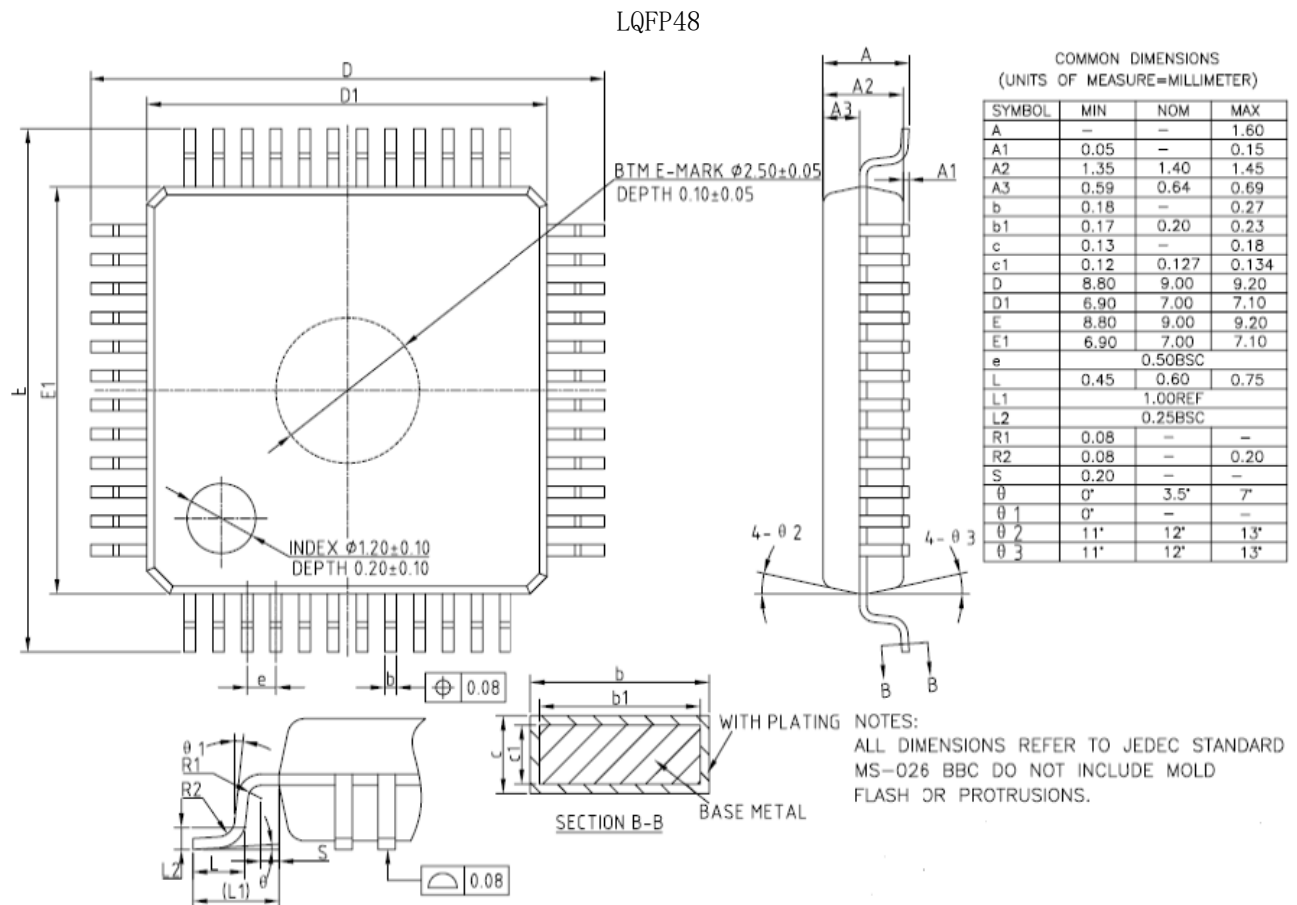


25 封装

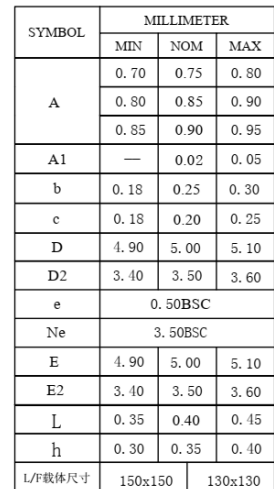
25.1 耐焊性

耐焊性：经过 260℃，不少于 5S 的回流焊，芯片能够保证正常工作。
建议过锡炉次数：不超过 3 次。

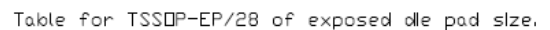
25.2 封装图



QFN32



TSS0P28



Pad Size	Symbol	Min	Nom	Max
118*232	P1	5.40	5.50	5.60
	P2	2.50	2.60	2.70