



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

# ZB32F103

**ARM® Cortex®-M3 32-bit  
Micro-Controller  
User Manual**

**Version V0.00  
6/22, 2022**



## 目录

目录 .....	2
图索引 .....	32
表索引 .....	41
【1】 简介 .....	45
【2】 产品特性 .....	46
【3】 产品功能描述 .....	49
3.1    设备概述 .....	49
3.2    内部功能框图结构 .....	51
3.3    32 位 Cortex®-M3 内核 .....	52
3.3-1    Flash .....	52
3.3-2    320K Bytes SRAM .....	52
3.4    时钟系统 .....	52
3.5    工作模式 .....	53
3.6    中断控制器(NVIC) .....	54
3.7    复位控制器 .....	54
3.8    通用 IO 端口(GPIO) .....	54
3.9    定时器和看门狗 .....	55
3.9-1    高级控制定时器(TIM1,8,12) .....	57
3.9-2    通用定时器(TIM2,3,4,5) .....	57
3.9-3    可编程计数器阵列(PCA) .....	58
3.9-4    低功耗定时器(LPTIM) .....	58
3.9-5    基础定时器(TIM10/TIM11) .....	58
3.9-6    独立看门狗(IWDG) .....	58
3.9-7    系统窗口看门狗(WWDG) .....	58
3.9-8    SysTick 定时器(SYST) .....	58
3.10    实时时钟(RTC) .....	60
3.11    通用异步收发器(USART1~3/UART4~5) .....	60
3.12    通用异步收发器(LPUART) .....	60
3.13    串行外设接口 (SPI) / 内部集成音频接口 (I2S) .....	60
3.14    I2C 接口(I2C) .....	60
3.15    控制器局域网 CAN .....	60
3.16    自唤醒定时器(AWK) .....	60



3.17	时钟校准/监测模块(CLKTRIM) .....	60
3.18	唯一 ID 号(UID).....	60
3.19	循环冗余校验计算单元(CRC16/32).....	62
3.20	模拟/数字转换器(ADC) .....	62
3.21	低电压检测器(LVD) .....	62
3.22	电压比较器(VC) .....	62
3.23	运算放大器(OPA) .....	62
3.24	嵌入式调试系统 .....	62
3.25	加密型嵌入式 Debug 支持(DBG) .....	62
3.26	高级加密标准模块 (AES) .....	63
3.27	真随机数发生器 (TRNG) .....	63
3.28	编程模式 .....	63
3.29	直接存储器存取(PDMA) .....	63
3.30	内嵌的自举程序(Bootloader) .....	63
<b>【4】</b>	<b>引脚配置及功能 .....</b>	<b>64</b>
4.1	引脚定义 .....	64
4.2	引脚功能配置 .....	65
4.3	选择芯片引脚复用说明 .....	69
4.3-1	STMD 引脚复用(AFMD=0) .....	69
4.3-2	PINMD 引脚复用(AFMD=1) .....	75
4.4	模块与引脚信号说明 .....	78
<b>【5】</b>	<b>系统和存储器概要 .....</b>	<b>81</b>
5.1	系统架构 .....	81
5.2	总线矩阵 .....	82
5.3	存储器映射 .....	83
5.4	存储空间和模块地址 .....	84
5.5	BOOT PIN 地址映射 .....	87
<b>【6】</b>	<b>电源管理和工作模式 .....</b>	<b>89</b>
6.1	电源 .....	89
6.1-1	独立的 A/D,OPA 转换器供电和参考电压 .....	89
6.1-2	电池备份区域 .....	89
6.2	工作模式 .....	91
6.2-1	运行模式(Active Mode) .....	92
6.2-2	低功耗模式 .....	93



6.2-2.1	休眠模式(Sleep Mode).....	95
6.2-2.2	停机模式(Stop Mode).....	96
6.2-2.3	待机模式(Standby mode).....	98
6.2-2.4	关机模式(power down mode).....	99
6.3	PMU 寄存器列表.....	100
6.4	PMU 寄存器說明.....	101
6.4-1	电源控制寄存器(PWR_CR) .....	101
6.4-2	掉電喚醒始能寄存器(PWR_PDWK).....	103
6.4-3	电源控制/状态寄存器(PWR_CSR).....	105
<b>【7】</b>	<b>备份寄存器(BKP) .....</b>	<b>107</b>
7.1	BKP 简介 .....	107
7.2	BKP 特性 .....	107
7.3	BKP 功能描述.....	107
7.3-1	侵入检测 .....	107
7.3-2	RTC 校准 .....	108
7.4	BKP 寄存器列表 .....	109
7.5	BKP 寄存器說明 .....	111
7.5-1	备份数据寄存器 x(BKP_DRx) (x = 1 ... 10) .....	111
7.5-2	RTC 时钟校准寄存器(BKP_RTCCR).....	112
7.5-3	备份控制寄存器(BKP_CR) .....	114
7.5-4	备份控制/状态寄存器(BKP_CSR) .....	115
<b>【8】</b>	<b>系统复位与时钟(RCC) .....</b>	<b>117</b>
8.1	复位.....	117
8.1-1	系统复位 .....	117
8.1-1.1	软件复位 .....	117
8.1-1.2	低功耗管理复位 .....	117
8.1-2	备份域复位 .....	117
8.1-3	电源复位 .....	118
8.2	系统时钟 .....	119
8.2-1	系统时钟模块图 .....	120
8.2-2	内部高速 RC 时钟 (HIRC) .....	121
8.2-3	内部低速 RC 时钟 (SIRC) .....	121
8.2-4	外部高速晶振时钟 (HXT) .....	121
8.2-5	外部低速晶振时钟 (LXT) .....	122



8.2-6	PLL 时钟 .....	123
8.2-7	系统时钟启动过程 .....	124
8.2-8	系统时钟切换 .....	125
8.2-8.1	内部高速切换到外部低速 .....	125
8.2-8.2	内部高速切换到外部高速 .....	125
8.2-8.3	内部低速切换到外部高速 .....	126
8.2-8.4	内部时钟切换到 PLL 时钟 .....	126
8.2-9	系统时钟输出 .....	126
8.2-10	系统时钟安全控制 .....	127
8.2-11	IWDG 时钟 .....	127
8.2-12	RTC 时钟 .....	127
8.2-13	AWK 时钟 .....	127
8.2-14	低功耗模式 .....	127
8.3	寄存器列表 .....	128
8.4	寄存器说明 .....	129
8.4-1	时钟控制寄存器(RCC_CR) .....	129
8.4-2	时钟配置寄存器(RCC_CFGR) .....	131
8.4-3	时钟中断寄存器(RCC_CIR) .....	134
8.4-4	APB2 外设复位寄存器(RCC_APB2RSTR) .....	137
8.4-5	APB1 外设复位寄存器(RCC_APB1RSTR) .....	139
8.4-6	AHB 外设时钟使能寄存器(RCC_AHBENR) .....	142
8.4-7	APB2 外设时钟使能寄存器(RCC_APB2ENR) .....	144
8.4-8	APB1 外设时钟使能寄存器(RCC_APB1ENR) .....	146
8.4-9	备份域控制寄存器 (RCC_BDCR) .....	149
8.4-10	控制/状态寄存器 (RCC_CSR) .....	151
8.4-11	内部高速 RC 振荡器控制寄存器(RCC_HIRCCR) .....	153
8.4-12	外部高速晶体振荡器控制寄存器(RCC_HXTCR) .....	154
8.4-13	内部低速 RC 振荡器控制寄存器(RCC_SIRCCR) .....	155
8.4-14	外部低速晶体振荡器控制寄存器(RCC_LXTCR) .....	157
8.4-15	SysTick Timer 控制寄存器(RCC_STICKCR) .....	159
8.4-16	SWDIO 端口控制寄存器(RCC_SWDIOCR) .....	160
8.4-17	RTC 复位控制寄存器(RCC_RTCRST) .....	161
8.4-18	MASK 中断时间控制寄存器(RCC_MASK_INT_T) .....	162
8.4-19	LPM 控制寄存器(RCC_LPM) .....	163



8.4-20	周边模块复位控制 1 寄存器(RCC_PERIRST1) .....	164
8.4-21	PLL 控制寄存器 2(RCC_PLLCR1).....	166
8.4-22	PLL 控制寄存器 1(RCC_PLLCR2).....	168
8.4-23	寄存器写保护控制寄存器(RCC_UNLOCK) .....	170
<b>【9】</b>	<b>系统控制(SYSCON) .....</b>	<b>171</b>
9.1	寄存器列表 .....	171
9.2	寄存器说明 .....	172
9.2-1	系统配置寄存器 0(SYSCON_CFGR0).....	172
9.2-2	端子 Deep Sleep 中断模式控制寄存器(SYSCON_PORTINTCR) .....	173
9.2-3	端子控制寄存器(SYSCON_PORTCR) .....	174
9.2-4	PCA 捕获通道控制寄存器(SYSCON_PCACR) .....	176
9.2-5	TIM1,8,12 通道输入源选择(SYSCON_TIMxCR,X=1,8,12).....	177
9.2-6	TIM2/3/4/5 通道输入源选择(SYSCON_TIMXCR).....	181
9.2-7	SYSCON 寄存器写保护(SYSCON_UNLOCK) .....	184
9.2-8	SYSCON BOOT 状态暂存器(SYSCON_BOOT).....	185
<b>【10】</b>	<b>中断和事件 .....</b>	<b>186</b>
10.1	嵌套向量中断控制器.....	186
10.1-1	统滴嗒(SysTick)校准值寄存器.....	186
10.1-2	中断和异常向量表.....	187
10.2	外部中断/事件控制器(EXTI) .....	191
10.2-1	主要特性 .....	191
10.2-2	框图 .....	192
10.2-3	唤醒事件管理 .....	193
10.2-4	功能说明 .....	194
10.2-5	外部中断/事件线路映像 .....	195
10.3	EXTI 寄存器列表.....	196
10.4	寄存器说明 .....	197
10.4-1	中断屏蔽寄存器(EXTI_IMR).....	197
10.4-2	事件屏蔽寄存器(EXTI_EMR).....	198
10.4-3	上升沿触发选择寄存器(EXTI_RTSR) .....	199
10.4-4	下降沿触发选择寄存器(EXTI_FTSR) .....	200
10.4-5	软件中断事件寄存器(EXTI_SWIER) .....	201
10.4-6	挂起寄存器(EXTI_PR).....	202
<b>【11】</b>	<b>通用和复用功能 I/O(GPIO 和 AFIO) .....</b>	<b>203</b>



---

11.1	GPIO 简介 .....	203
11.2	GPIO 主要特性 .....	203
11.3	GPIO 功能描述 .....	204
11.3-1	通用 I/O(GPIO) .....	207
11.3-2	单独的位设置或位清除 .....	207
11.3-3	外部中断/唤醒线 .....	207
11.3-4	复用功能(AF) .....	207
11.3-5	软件重新映射 I/O 复用功能 .....	209
11.3-6	输出配置 .....	209
11.3-7	输入配置 .....	210
11.3-8	GPIO 锁定机制 .....	210
11.3-9	复用功能配置 .....	211
11.3-10	模拟输入配置 .....	212
11.3-11	外设的 GPIO 配置 .....	213
11.3-11.1	TIM1,8,12 的引脚配置 .....	213
11.3-11.2	TIM2,3,4,5 的引脚配置 .....	213
11.3-11.1	PCA 的引脚配置 .....	213
11.3-11.2	USART 的引脚配置 .....	214
11.3-11.3	SPI 的引脚配置 .....	214
11.3-11.4	I2S 的引脚配置 .....	216
11.3-11.5	CAN 的引脚配置 .....	216
11.3-11.6	USB 的引脚配 .....	216
11.3-11.7	SDIO 的引脚配置 .....	218
11.3-11.8	ADC/DAC 的引脚配置 .....	218
11.3-11.1	其它 I/O 功能的引脚配置 .....	218
11.4	复用功能 I/O 和调试配置(AFIO) .....	219
11.4-1	使用 OSC32_IN/OSC32_OUT 作为 GPIO 端口 PC14/PC15 .....	220
11.4-2	使用 OSC_IN/OSC_OUT 引脚作为 GPIO 端口 PD0/PD1 .....	220
11.4-3	CAN 复用功能重映射 .....	220
11.4-4	JTAG/SWD 复用功能重映射 .....	220
11.4-5	ADC 复用功能重映射 1 .....	222
11.4-6	定时器复用功能重映射 .....	223
11.4-7	USART 复用功能重映射 .....	226
11.4-8	I2C1 复用功能重映射 .....	227

---



11.4-9	SPI 1 复用功能重映射 .....	227
11.4-10	SPI3 复用功能重映射 .....	227
11.5	GPIOX 和 AFIO 寄存器列表 .....	228
11.6	GPIO 寄存器说明 .....	230
11.6-1	GPIO 端口置低寄存器(GPIOx_CRL)(x = A..D) .....	230
11.6-2	端口配置高寄存器(GPIOx_CRH) (x = A..D) .....	231
11.6-3	端口输入数据寄存器(GPIOx_IDR))(x = A..D) .....	233
11.6-4	Port output data register (GPIOx_ODR) (x = A..D) .....	234
11.6-5	Port bit set/reset register (GPIOx_BSRR) (x = A..D) .....	235
11.6-6	Port bit reset register (GPIOx_BRR) (x = A..D) .....	236
11.6-7	Port configuration lock register (GPIOx_LCKR) (x = A..D) .....	237
11.7	AFIO 寄存器描述 .....	238
11.7-1	事件控制寄存器(AFIO_EVCR) .....	238
11.7-2	复用重映射和调试 I/O 配置寄存器(AFIO_MAPR) .....	240
11.7-3	外部中断配置寄存器 1(AFIO_EXTICR1) .....	246
11.7-4	外部中断配置寄存器 2(AFIO_EXTICR2) .....	247
11.7-5	外部中断配置寄存器 3(AFIO_EXTICR3)) .....	248
11.7-6	外部中断配置寄存器 4(AFIO_EXTICR4) .....	249
11.7-7	GPIOA 端口复用功能寄存器(GPIOA_AFR ) .....	250
11.7-7.1	PORTA 端口复用配置 .....	250
11.7-8	GPIOB 端口复用功能寄存器(GPIOB_AFR ) .....	251
11.7-8.1	PORTB 端口复用配置 .....	251
11.7-9	GPIOC 端口复用功能寄存器(GPIOC_AFR ) .....	252
11.7-9.1	PORTC 端口复用配置 .....	252
11.7-10	GPIOD 端口复用功能寄存器(GPIOA_AFR ) .....	253
11.7-10.1	PORTD 端口复用配置 .....	253
【12】	循环冗余校验计算单元(CRC) .....	254
12.1	概述 .....	254
12.2	功能描述 .....	254
12.2-1	CRC 编码模式 .....	254
12.2-2	CRC 检验模式 .....	255
12.3	CRC 寄存器列表 .....	256
12.4	寄存器说明 .....	257
12.4-1	CRC 结果寄存器(CRC_RESULT) .....	257



---

12.4-2	CRC 控制寄存器寄存器(CRC_CR) .....	258
12.4-3	CRC 数据寄存器(CRC_DATA).....	259
<b>【13】</b>	<b>高级控制定时器 TIM1X(TIM1/1A/1B) .....</b>	<b>260</b>
13.1	Advanced Timer 简介 .....	260
13.2	Advanced Timer 主要特性.....	260
13.3	Advanced Timer 功能描述.....	262
13.3-1	时基单元 .....	262
13.3-2	计数器模式 .....	263
13.3-2.1	向上计数模式 .....	263
13.3-2.2	向下计数模式 .....	268
13.3-2.3	中央对齐模式(向上/向下计数).....	271
13.3-3	重复计数器 .....	275
13.3-4	时钟选择 .....	276
13.3-4.1	外部时钟源模式 1 .....	278
13.3-4.2	外部时钟源模式 2 .....	279
13.3-5	捕获/比较通道 .....	281
13.3-6	输入捕获模式.....	283
13.3-7	PWM 输入模式 .....	285
13.3-8	强置输出模式.....	286
13.3-9	输出比较模式 .....	286
13.3-10	PWM 模式.....	288
13.3-10.1	PWM 边沿对齐模式.....	288
13.3-10.2	PWM 中央对齐模式 .....	290
13.3-11	互补输出和死区插入.....	292
13.3-11.1	复位向 OCxREF 到 OCx 或 OCxN.....	293
13.3-12	使用刹车功能 .....	294
13.3-13	在外部事件时清除 OCxREF 信号 .....	296
13.3-14	产生六步 PWM 输出 .....	297
13.3-15	单脉冲模式 .....	298
13.3-16	编码器接口模式 .....	299
13.3-17	定时器输入异或功能.....	301
13.3-18	与霍尔传感器的接口.....	303
13.3-19	TIM1X* 定时器和外部触发的同步 .....	306
13.3-19.1	从模式: 复位模式.....	306

---



13.3-19.2	模式：门控模式.....	307
13.3-19.3	从模式：触发模式.....	308
13.3-19.4	从模式：外部时钟模式 2 + 触发模式 .....	309
13.3-20	定时器同步 .....	309
13.3-21	调试模式 .....	309
13.4	高级控制定时器 TIM1X(TIM1/1A/1B)寄存器列表 .....	310
13.5	TIM1X*寄存器说明.....	311
13.5-1	TIM1X* 控制寄存器 1(TIM1X*_CR1) .....	311
13.5-2	TIM1X* 控制寄存器 2(TIM1X*_CR2) .....	313
13.5-3	TIM1X* 从模式控制寄存器(TIM1X*_SMCR).....	316
13.5-4	TIM1X* 中断与 DMA 请求使能寄存器(TIM1X*_DIER) .....	319
13.5-5	TIM1X* 状态寄存器(TIM1X*_SR).....	321
13.5-6	TIM1X* 事件产生寄存器(TIM1X*_EGR).....	324
13.5-7	TIM1X* 捕获/比较模式寄存器 1 (TIM1X*_CCMR1) .....	326
13.5-8	TIM1X* 捕获/比较模式寄存器 2(TIM1X*_CCMR2) .....	331
13.5-9	TIM1X* 捕获/比较使能寄存器(TIM1X*_CCER) .....	334
13.5-10	TIM1X* 计数器(TIM1X*_CNT).....	337
13.5-11	TIM1X* 预分频器(TIM1X*_PSC) .....	337
13.5-12	TIM1X* 自动重装载寄存器(TIM1X*_ARR) .....	338
13.5-13	TIM1X* 重复计数寄存器(TIM1X*_RCR).....	338
13.5-14	TIM1X* 捕获/比较寄存器 1(TIM1X*_CCR1).....	340
13.5-15	TIM1X* 捕获/比较寄存器 2(TIM1X*_CCR2).....	341
13.5-16	TIM1X* 捕获/比较寄存器 3(TIM1X*_CCR3).....	341
13.5-17	TIM1X* 捕获/比较寄存器 4(TIM1X*_CCR4).....	342
13.5-18	TIM1X* 刹车和死区寄存器(TIM1X*_BDTR).....	343
13.5-19	TIM1X*DMA 控制寄存器(TIM1X*_DCR).....	347
13.5-20	TIM1X*连续模式的 DMA 地址(TIM1X*_DMAR).....	349
<b>【14】</b>	<b>通用定时器 TIM2X*(TIM2/2A/2B/2C) .....</b>	<b>350</b>
14.1	通用定时器 简介 .....	350
14.2	通用定时器主要功能.....	350
14.3	TIM2X* 功能描述.....	352
14.3-1	时基单元 .....	352
14.3-2	计数器模式 .....	354
14.3-2.1	向上计数模式 .....	354



---

14.3-2.2	向下计数模式 .....	357
14.3-2.3	中央对齐模式(向上/向下计数).....	359
14.3-3	时钟选择 .....	363
14.3-3.1	内部时钟源(CK_INT) .....	363
14.3-3.2	外部时钟源模式 1 .....	364
14.3-3.3	外部时钟源模式 2 .....	365
14.3-4	捕获/比较通道 .....	366
14.3-5	输入捕获模式 .....	368
14.3-6	PWM 输入模式 .....	369
14.3-7	强置输出模式 .....	370
14.3-8	输出比较模式 .....	370
14.3-9	PWM 模式.....	372
14.3-9.1	PWM 边沿对齐模式.....	372
14.3-9.1.1	向上计数配置 .....	372
14.3-9.1.2	向下计数的配置 .....	373
14.3-9.2	PWM 中央对齐模式 .....	373
14.3-9.2.1	使用中央对齐模式的提示 .....	374
14.3-10	单脉冲模式 .....	375
14.3-11	在外部事件时清除 OCxREF 信号 .....	377
14.3-12	编码器接口模式 .....	378
14.3-13	定时器输入异或功能 .....	380
14.3-14	定时器和外部触发的同步 .....	380
14.3-14.1	从模式: 复位模式 .....	380
14.3-14.2	从模式: 门控模式 .....	381
14.3-14.3	从模式: 触发模式 .....	382
14.3-14.4	从模式: 外部时钟模式 2 + 触发模式 .....	382
14.3-15	定时器同步 .....	383
14.3-15.1	使用一个定时器作为另一个定时器的预分频器 .....	383
14.3-15.2	使用一个定时器使能另一个定时器 .....	384
14.3-15.3	使用一个定时器去启动另一个定时器 .....	385
14.3-15.4	使用一个定时器作为另一个的预分频器 .....	387
14.3-15.5	使用一个外部触发同步地启动 2 个定时器 .....	387
14.3-16	调试模式 .....	388
14.4	TIM2X* 寄存器列表 .....	389

---



14.5	TIM2X* 寄存器说明 .....	390
14.5-1	TIM2X* 控制寄存器 1(TIM2X*_CR1) .....	390
14.5-2	TIM2X* 控制寄存器 2(TIM2X*_CR2) .....	392
14.5-3	TIM2X* 从模式控制寄存器(TIM2X*_SMCR) .....	393
14.5-4	TIM2X* 中断与 DMA 请求使能寄存器(TIM2X*_DIER) .....	396
14.5-5	TIM2X* 状态寄存器(TIM2X*_SR) .....	398
14.5-6	TIM2X* 事件产生寄存器(TIM2X*_EGR) .....	400
14.5-7	TIM2X* 捕获/比较模式寄存器 1(TIM2X*_CCMR1) .....	402
14.5-8	TIM2X* 捕获/比较模式寄存器 2(TIM2X*_CCMR2) .....	407
14.5-9	TIM2X* 捕获/比较使能寄存器(TIM2X*_CCER) .....	410
14.5-10	TIM2X* 计数器(TIM2X*_CNT) .....	411
14.5-11	TIM2X* 预分频器(TIM2X*_PSC) .....	411
14.5-12	TIM2X* 自动重装载寄存器(TIM2X*_ARR) .....	413
14.5-13	TIM2X* 捕获/比较寄存器 1(TIM2X*_CCR1) .....	413
14.5-14	TIM2X* 捕获/比较寄存器 2(TIM2X*_CCR2) .....	414
14.5-15	TIM2X* 捕获/比较寄存器 3(TIM2X*_CCR3) .....	415
14.5-16	TIM2X* 捕获/比较寄存器 4(TIM2X*_CCR4) .....	415
14.5-17	TIM2X*DMA 控制寄存器(TIM2X*_DCR) .....	417
14.5-18	TIM2X* 连续模式的 DMA 地址(TIM2X*_DMAR) .....	418
<b>【15】</b>	<b>可编程计数阵列(PCA)</b> .....	<b>419</b>
15.1	PCA 简介 .....	419
15.2	PCA 功能描述 .....	420
15.2-1	PCA 定时/计数器 .....	420
15.2-2	捕获功能 .....	421
15.2-3	PCA 比较功能 .....	422
15.2-3.1	16 位软件计数器模式 .....	423
15.2-3.2	高速输出模式 .....	423
15.2-3.3	8 位脉宽调制(PWM)功能 .....	424
15.3	PCA 模块与其他模块互连及控制 .....	426
15.3-1	ECI 互连 .....	426
15.3-2	PCACAP0 .....	426
15.3-3	PCACAP1/2/3/4 .....	426
15.4	PCA 寄存器列表 .....	427
15.5	寄存器说明 .....	428



15.5-1	控制寄存器(PCA_CR).....	428
15.5-2	模式寄存器(PCA_MOD).....	430
15.5-3	计数寄存器(PCA_CNT).....	431
15.5-4	中断清除寄存器(PCA_INTCLR).....	432
15.5-5	比较捕获模式寄存器(PCA_CCAPM0~4).....	433
15.5-6	比较捕获数据寄存器低 8 位(PCA_CCAP0~4L) .....	434
15.5-7	比较捕获数据寄存器高 8 位(PCA_CCAP0~4H) .....	436
15.5-8	比较高速输出标志寄存器(PCA_CCAPO) .....	437
15.5-9	端子输出控制寄存器(PCA_POCR) .....	438
15.5-10	比较捕获 16 位寄存器(PCA_CCAP0~4).....	440
<b>【16】</b>	<b>基础定时器 Base Timer(TIM10/TIM11).....</b>	<b>441</b>
16.1	Base Timer 简介 .....	441
16.2	Base Timer 功能描述 .....	442
16.2-1	计数功能 .....	444
16.2-2	定时功能 .....	445
16.2-3	Buzzer 功能 .....	445
16.3	Base Timer 互连 .....	446
16.3-1	GATE 互联 .....	446
16.3-2	Toggle 输出互联 .....	446
16.4	Base Timer(TIM10/TIM11) 寄存器列表 .....	447
16.5	Base Timer 寄存器说明 .....	448
16.5-1	控制寄存器(TIMx_CR) .....	448
16.5-2	立即重载寄存器(TIMx_LOAD) .....	450
16.5-3	计数器寄存器(TIMx_CNT) .....	450
16.5-4	原始中断状态寄存器(TIMx_RAWINTSR) .....	451
16.5-5	中断标志寄存器(TIMx_MSKINTSR) .....	451
16.5-6	中断清除寄存器(TIMx_INTCLR) .....	452
16.5-7	周期重载寄存器(TIMx_BGLOAD) .....	452
<b>【17】</b>	<b>低功耗定时器(LPTIM).....</b>	<b>453</b>
17.1	LPTIM 功能描述 .....	453
17.1-1	计数功能 .....	455
17.1-2	定时功能 .....	455
17.2	LPTIM 互连 .....	456
17.2-1	GATE 互联 .....	456



17.2-2	EXT 互联	456
17.2-3	TOGGLE 输出互联	456
17.3	LPTIM 寄存器列表	457
17.4	LPTIM 寄存器说明	458
17.4-1	LPTIM 计数值只读寄存器(LPTIM_CNTVAL)	458
17.4-2	LPTIM 控制寄存器(LPTIM_CR)	459
17.4-3	LPTIM 立即重载寄存器(LPTIM_LOAD)	461
17.4-4	LPTIM 中断寄存器(LPTIM_INTSR)	462
17.4-5	LPTIM 中断寄存器(LPTIM_INTCLR)	463
17.4-6	LPTIM 周期重载寄存器(LPTIM_BGLOAD)	464
<b>【18】</b>	<b>自唤醒定时器(AWK)</b>	<b>465</b>
18.1	寄存器列表	466
18.2	寄存器说明	467
18.2-1	自唤醒定时器控制寄存器(AWK_CR)	467
18.2-2	自唤醒定时器重装载数据寄存器(AWK_RLOAD)	468
18.2-3	自唤醒定时器状态寄存器(AWK_SR)	469
18.2-4	自唤醒中断清除寄存器(AWK_INTCLR)	470
<b>【19】</b>	<b>独立看门狗(IWDG)</b>	<b>471</b>
19.1	IWDG 概述	471
19.2	IWDG 的功能	471
19.2-1	超时周期	472
19.2-2	IWDG 溢出后产生中断	472
19.2-3	IWDG 溢出后产生复位	473
19.3	寄存器列表	473
19.4	寄存器说明	474
19.4-1	IWDG 控制命令寄存器(IWDG_CMDCR)	474
19.4-2	IWDG 配置寄存器(IWDG_CFGR)	475
19.4-3	IWDG 计数器重装载寄存器(IWDG_RLOAD)	476
19.4-4	IWDG 计数器值寄存器(IWDG_CNTVAL)	477
19.4-5	IWDG 中断状态寄存器(IWDG_SR)	478
19.4-6	IWDG 中断清除寄存器(IWDG_INTCLR)	479
19.4-7	IWDG 保护寄存器(IWDG_UNLOCK)	480
19.5	注意	480
<b>【20】</b>	<b>系统窗口看门狗(WWDG)</b>	<b>481</b>



20.1	概述 .....	481
20.2	特征 .....	481
20.3	结构框图 .....	481
20.4	基本配置 .....	481
20.5	功能描述 .....	482
20.5-1	窗口看门狗定时器的计数 .....	482
20.5-2	窗口看门狗定时器比较中断 .....	483
20.5-3	窗口看门狗定时器复位系统 .....	483
20.5-4	窗口看门狗定时器的窗口设置限制 .....	483
20.6	与独立看门狗定时器(IWDG)比较 .....	483
20.6-1	复位条件和复位延时 .....	483
20.6-2	唤醒功能 .....	483
20.7	寄存器列表 .....	484
20.8	寄存器说明 .....	485
20.8-1	窗口看门狗定时器重载计数寄存器(WWDG_RLOAD) .....	485
20.8-2	窗口看门狗定时器控制寄存器(WWDG_CR) .....	486
20.8-3	窗口看门狗定时器中断使能寄存器(WWDG_INTEN) .....	487
20.8-4	窗口看门狗定时器状态寄存器(WWDG_SR) .....	488
20.8-5	窗口看门狗定时器中断清除寄存器(WWDG_INTCLR) .....	489
20.8-6	窗口看门狗定时器计数器值寄存器(WWDG_CNTVAL) .....	490
<b>【21】</b>	<b>通用异步/同步收发器(USART0/1/3, USART4/5) .....</b>	<b>491</b>
21.1	概述 .....	491
21.2	结构框图 .....	493
21.3	UART 工作模式 .....	495
21.3-1	Mode 0(同步模式, 半双工) .....	495
21.3-1.1	发送数据 .....	495
21.3-1.2	接收数据 .....	495
21.3-2	Mode 1(异步模式, 全双工) .....	496
21.3-2.1	发送数据 .....	496
21.3-2.2	接收数据 .....	497
21.3-3	Mode 2(异步模式, 全双工) .....	497
21.3-3.1	发送数据 .....	497
21.3-3.2	发送数据 .....	498
21.3-4	Mode 3(异步模式, 全双工) .....	499



---

21.3-5	波特率编程 .....	499
21.3-5.1	Mode 0.....	499
21.3-5.2	Mode 1/3 .....	499
21.3-5.3	Mode 2.....	500
21.3-6	帧错误检测 .....	500
21.3-7	多机通讯 .....	500
21.3-8	自动地址识别 .....	500
21.3-9	给定地址 .....	501
21.3-10	广播地址 .....	501
21.3-11	收发端缓存 .....	501
21.3-11.1	接收缓存 .....	501
21.3-11.2	发送缓存 .....	501
21.3-12	IrDA 红外功能 .....	502
21.3-12.1	IrDA 低功耗模式 .....	503
21.4	USART 工作模式 .....	505
21.4-1	USART 特征描述 .....	506
21.4-2	发送器 .....	507
21.4-3	接收器 .....	510
21.4-4	多处理器通信 .....	512
21.4-5	校验控制 .....	513
21.4-6	LIN (局域互联网) 模式 .....	514
21.4-6.1	LIN 发送 .....	514
21.4-6.2	LIN 接收 .....	514
21.4-7	USART 同步模式 .....	516
21.4-8	单线半双工通信模式 .....	518
21.4-9	智能卡模式 .....	519
21.4-10	红外线数据传输(IrDA) .....	520
21.4-10.1	IrDA 低功耗模式 .....	522
21.4-11	使用 DMA 通信 .....	523
21.4-12	硬件流控制 .....	523
21.4-12.1	RTS 流控制 .....	524
21.4-12.2	CTS 流控制 .....	524
21.4-13	中断请求 .....	525
21.5	不同波特率的分频设置 .....	526

---



21.6	UART/USART 寄存器列表.....	532
21.7	UART4,5 寄存器说明 .....	533
21.7-1	UART 控制寄存器(UARTx_SCON).....	533
21.7-2	UART 数据寄存器(UARTx_SBUF) .....	535
21.7-3	UART 地址寄存器(UARTx_SADDR) .....	536
21.7-4	UART 地址掩码寄存器(UARTx_SADEN) .....	537
21.7-5	UART 标志位寄存器(UARTx_INTSR).....	538
21.7-6	UART 标志位清除寄存器(UARTx_INTCLR).....	539
21.7-7	UART 波特率控制寄存器(UARTx_BAUDCR) .....	540
21.7-8	rDA 控制寄存器(UARTx_IRDACR).....	541
21.8	USART2,3 寄存器说明.....	542
21.8-1	状态寄存器(USARTx_SR) .....	542
21.8-2	数据寄存器(USARTx_DR).....	545
21.8-3	波特比率寄存器(USARTx_BRR) .....	546
21.8-4	控制寄存器 1 (USARTx_CR1).....	547
21.8-5	控制寄存器 2(USART_CR2).....	550
21.8-6	控制寄存器 2(USART_CR3).....	553
21.8-7	保护时间和预分频寄存器(USART_GTPR).....	556
<b>【22】</b>	<b>低功耗通用异步收发器(LPUART) .....</b>	<b>558</b>
22.1	概述.....	558
22.2	结构框图 .....	558
22.3	工作模式 .....	559
22.3-1	Mode 0(同步模式， 半双工).....	559
22.3-1.1	发送数据 .....	559
22.3-1.2	接收数据 .....	560
22.3-2	Mode 1(异步模式， 全双工).....	560
22.3-2.1	发送数据 .....	560
22.3-2.2	接收数据 .....	561
22.3-3	Mode 2(异步模式， 全双工).....	561
22.3-3.1	发送数据 .....	561
22.3-3.2	接收数据 .....	562
22.3-4	Mode 3(异步模式， 全双工).....	562
22.4	波特率编程 .....	562
22.4-1	Mode 0.....	562



22.4-2	Mode 1/3 .....	563
22.4-3	Mode 2.....	563
22.5	帧错误检测 .....	564
22.6	多机通讯 .....	564
22.7	自动地址识别 .....	564
22.8	给定地址 .....	564
22.9	广播地址 .....	565
22.10	收发端缓存 .....	565
22.10-1	接收缓存 .....	565
22.10-2	发送缓存 .....	565
22.11	寄存器列表 .....	566
22.12	寄存器说明 .....	566
22.12-1	LPUART 数据寄存器(LPUART_SBUF).....	566
22.12-2	LPUART 控制寄存器(LPUART_SCON).....	567
22.12-3	LPUART 地址寄存器(LPUART_SADDR).....	570
22.12-4	LPUART 地址掩码寄存器(LPUART_SADEN).....	571
22.12-5	LPUART 标志位寄存器(LPUART_INTSR).....	571
22.12-6	LPUART 标志位清除寄存器(LPUART_INTCLR) .....	572
22.12-7	LPUART 波特率控制寄存器(LPUART_BAUDCR) .....	572
<b>【23】 I2C 接口(I2C1/2-L032)</b>	<b>.....</b>	<b>573</b>
23.1	I2C 简介 .....	573
23.2	I2C 主要特性.....	573
23.3	I2C 协议描述.....	573
23.3-1	I2C 总线上数据传输 .....	574
23.3-2	起始位或重复起始信号 .....	574
23.3-3	从机地址传输 .....	575
23.3-4	数据传输 .....	575
23.4	I2C 功能描述.....	576
23.5	I2C 工作模式.....	577
23.5-1	仲裁与同步逻辑 .....	577
23.5-2	串行时钟发生器 .....	578
23.5-3	输入滤波器 .....	578
23.5-4	地址比较器 .....	578
23.5-5	中断产生器 .....	578



23.5-6	I2C 主机发送模式 .....	579
23.5-7	I2C 主机接收模式 .....	583
23.5-8	I2C 从机接收模式 .....	586
23.5-9	I2C 从机发送模式 .....	590
23.5-10	I2C 其他杂项状态 .....	593
23.6	I2C 操作模式 .....	593
23.6-1	初始化程序 .....	593
23.6-2	端口配置程序 .....	594
23.6-3	启动主机发送功能 .....	594
23.6-3.1	正常发送 .....	594
23.6-3.2	使用 DMA 发送 .....	594
23.6-4	启动主机接收功能 .....	596
23.6-4.1	正常接收 .....	596
23.6-4.2	使用 DMA 接收 .....	597
23.6-5	I2C 中断程序 .....	597
23.6-6	无指定模式状态 .....	597
23.6-7	主发送状态 .....	599
23.6-8	主接收状态 .....	600
23.6-9	从接收状态 .....	601
23.6-10	从发送状态 .....	603
23.7	I2CX* 寄存器列表 .....	604
23.8	I2C 寄存器说明 .....	605
23.8-1	I2C 配置寄存器(I2CX*_CR) .....	605
23.8-2	I2C 数据寄存器(I2CX*_DATA) .....	606
23.8-3	I2C 地址寄存器(I2CX*_ADDR) .....	606
23.8-4	I2C 状态寄存器(I2C_SR) .....	607
23.8-5	I2C 计数器 DMA 使能寄存器(I2CX*_TIMDMA) .....	608
23.8-6	I2C 波特率计数器配置寄存器(I2CX*_BAUDCR) .....	609
<b>【24】</b>	<b>串行外设接口(SPI/I2S) .....</b>	<b>610</b>
24.1	SPI 简介 .....	610
24.2	SPI/I2S 主要特性 .....	610
24.3	功能描述 .....	612
24.3-1	概述 .....	612
24.3-2	配置 SPI 为从模式 .....	616



24.3-3	配置 SPI 为主模式.....	617
24.3-4	配置 SPI 为单工通信.....	618
24.3-5	数据发送与接收过程.....	619
24.3-6	CRC 计算.....	626
24.3-7	状态标志.....	627
24.3-8	关闭 SPI.....	628
24.3-9	使用 DMA 的 SPI 通信.....	629
24.3-10	错误标志 .....	632
24.3-11	SPI 中断.....	633
24.4	I2S 功能描述 .....	633
24.4-1	I2S 功能描述 .....	633
24.4-2	支持的音频协议 .....	635
24.4-3	时钟发生器 .....	642
24.4-4	I2S 主模式.....	645
24.4-5	I2S 从模式.....	647
24.4-6	状态标志位 .....	650
24.4-7	错误标志位 .....	651
24.4-8	I2S 中断.....	651
24.4-9	DMA 功能.....	651
24.5	SPI/I2S 寄存器列表 .....	652
24.6	SPI/I2S 寄存器说明 .....	653
24.6-1	SPI 控制寄存器 1(SPI_CR1)(I2S 模式下不使用).....	653
24.6-2	SPI 控制寄存器 2(SPI_CR2).....	657
24.6-3	SPI 状态寄存器(SPI_SR).....	658
24.6-4	SPI 数据寄存器(SPI_DR).....	661
24.6-5	SPI CRC 多项式寄存器(SPI_CRCPCR)(I2S 模式下不使用).....	662
24.6-6	SPI Rx CRC 寄存器(SPI_RXCRCR)(I2S 模式下不使用).....	662
24.6-7	SPI Tx CRC 寄存器(SPI_TXCRCR).....	663
24.6-8	SPI_I2S 配置寄存器(SPI_I2S_CFGR).....	665
24.6-9	SPI_I2S 预分频寄存器(SPI_I2SPR).....	668
<b>【25】</b>	<b>One-Wire 接口(OWIRE).....</b>	<b>錯誤! 尚未定義書籤。</b>
25.1	单总线协议(One-Wire) .....	錯誤! 尚未定義書籤。
25.1-1	特点 .....	錯誤! 尚未定義書籤。
25.1-2	优点 .....	錯誤! 尚未定義書籤。



25.2	单总线通信过程	錯誤! 尚未定義書籤。
25.2-1	初始化	錯誤! 尚未定義書籤。
25.2-2	写时间隙	錯誤! 尚未定義書籤。
25.2-3	读时间隙	錯誤! 尚未定義書籤。
25.3	配置说明	錯誤! 尚未定義書籤。
25.3-1	初始化配置说明	錯誤! 尚未定義書籤。
25.3-2	读数据配置说明	錯誤! 尚未定義書籤。
25.3-3	写数据配置说明	錯誤! 尚未定義書籤。
25.4	One-Wire 接口寄存器列表	錯誤! 尚未定義書籤。
25.5	寄存器说明	錯誤! 尚未定義書籤。
25.5-1	1-Wire 模块控制寄存器(OWIRE_CR)	錯誤! 尚未定義書籤。
25.5-2	1-Wire 输入端子滤波控制寄存器(OWIRE_NFCR)	錯誤! 尚未定義書籤。
25.5-3	1-Wire RESET 宽度控制寄存器(OWIRE_RSTCNT)	錯誤! 尚未定義書籤。
25.5-4	1-Wire Presence Pulse 宽度计数寄存器(OWIRE_PRESCNT)	錯誤! 尚未定義書籤。
25.5-5	1-Wire Bit rate 设计计数器(OWIRE_BITRATECNT)	錯誤! 尚未定義書籤。
25.5-6	1-Wire 主器件读/写 PULL0 驱动时间(OWIRE_DRVCNT)	錯誤! 尚未定義書籤。
25.5-7	1-Wire 主器件读采样时间设定(OWIRE_RDSMPCNT)	錯誤! 尚未定義書籤。
25.5-8	1-Wire Recover Time 计数区间值(OWIRE_RECCNT)	錯誤! 尚未定義書籤。
25.5-9	1-Wire 数据寄存器(OWIRE_DATA)	錯誤! 尚未定義書籤。
25.5-10	1-Wire 总线操作命令寄存器(OWIRE_CMD)	錯誤! 尚未定義書籤。
25.5-11	1-wire 中断使能寄存器(OWIRE_INTEN)	錯誤! 尚未定義書籤。
25.5-12	1-wire 中断使能寄存器(OWIRE_INTEN)	錯誤! 尚未定義書籤。
25.5-13	1-wire 状态清除寄存器(OWIRE_INTCLR)	錯誤! 尚未定義書籤。
<b>【26】</b>	<b>时钟校准/监测模块(CLKTRIM)</b>	<b>669</b>
26.1	简介	669
26.2	主要特性	669
26.3	CLKTRIM 功能描述	670
26.3-1	CLKTRIM 校准模式	670
26.3-1.1	操作流程	670
26.3-2	CLKTRIM 监测模式	671
26.3-2.1	操作流程	671
26.4	CLKTRIM 寄存器列表	672
26.5	CLKTRIM 寄存器说明	673
26.5-1	配置寄存器(CLKTRIM_CR)	673



26.5-2	参考计数器处置配置寄存器(CLKTRIM_REFCON).....	674
26.5-3	参考计数器值寄存器(CLKTRIM_REFCNT) .....	675
26.5-4	校准计数器值寄存器(CLKTRIM_CALCNT).....	675
26.5-5	中断标志位寄存器(CLKTRIM_IFR) .....	676
26.5-6	中断标志位清除寄存器(CLKTRIM_ICLR) .....	677
26.5-7	校准计数器溢出值配置寄存器(CLKTRIM_CALCON) .....	678
<b>【27】</b>	<b>实时时钟(RTC-L032) .....</b>	<b>679</b>
27.1	简介 .....	679
27.2	主要特性 .....	679
27.3	RTC 功能描述.....	680
27.3-1	RTC 结构框图.....	680
27.3-2	RTC 时钟 .....	681
27.3-3	复位过程 .....	681
27.3-4	寄存器的写保护 .....	681
27.3-5	日历初始化及配置 .....	682
27.3-6	读出计数寄存器 .....	682
27.3-7	写入计数寄存器 .....	683
27.3-8	闹钟设定 .....	683
27.3-9	校准 1Hz 输出 .....	683
27.3-10	RTC 时钟校准 .....	683
27.4	RTC 中断 .....	684
27.4-1	RTC 闹钟中断 .....	684
27.4-2	RTC 周期中断 .....	684
27.5	RTC 寄存器列表 .....	685
27.6	RTC 寄存器说明 .....	686
27.6-1	RTC 控制寄存器(RTC_CR) .....	686
27.6-2	RTC 时钟控制寄存器(RTC_CLKCR) .....	687
27.6-3	RTC 时间寄存器(RTC_TIME) .....	689
27.6-4	RTC 日期寄存器(RTC_DATE) .....	691
27.6-5	RTC 时间闹钟寄存器(RTC_ALM1TIME) .....	692
27.6-6	RTC 日期闹钟寄存器(RTC_ALM1DATE) .....	693
27.6-7	RTC 周期闹钟寄存器(RTC_ALM2PRD) .....	695
27.6-8	RTC 时钟调校寄存器(RTC_CLKTRIM) .....	696
27.6-9	RTC 初始化和状态寄存器(RTC_ISR) .....	697



---

27.6-10	RTC 状态清除寄存器(RTC_INTCLR) .....	699
27.6-11	RTC 写保护寄存器(RTC_WPR).....	700
<b>【28】</b>	<b>实时时钟(RTC-F103) .....</b>	<b>701</b>
28.1	简介 .....	701
28.2	主要特性 .....	701
28.3	RTC 功能描述.....	702
28.3-1	RTC 概述與结构框图 .....	702
28.3-2	复位过程 .....	703
28.3-3	读 RTC 寄存器。 .....	703
28.3-4	配置 RTC 寄存器.....	703
28.3-5	RTC 标志的设置.....	704
28.4	RTC 寄存器列表 .....	705
28.5	RTC 寄存器说明 .....	706
28.5-1	控制寄存器高位(RTC_CRH) .....	706
28.5-2	RTC 控制寄存器低位(RTC_CRL) .....	707
28.5-3	RTC 预分频装载寄存器高位(RTC_PRLH).....	709
28.5-4	RTC 预分频装载寄存器高位(RTC_PRLH).....	710
28.5-5	RTC 预分频器余数寄存器高位(RTC_DIVH) .....	711
28.5-6	RTC 预分频器余数寄存器高位(RTC_DIVH) .....	712
28.5-7	RTC 计数器寄存器高位(RTC_CNTH).....	713
28.5-8	RTC 计数器寄存器低位(RTC_CNTL) .....	714
28.5-9	RTC 闹钟寄存器高位(RTC_ALRH) .....	715
28.5-10	RTC 闹钟寄存器低位(RTC_ALRL) .....	716
<b>【29】</b>	<b>模拟/数字转换器(ADC1,2,3) .....</b>	<b>717</b>
29.1	模块简介 .....	717
29.2	ADC 框图.....	718
29.2-1	ADC 输入阻抗 .....	718
29.3	转换时序及速度 .....	720
29.4	单次转换模式 .....	721
29.5	连续转换模式 .....	721
29.6	连续转换累加模式.....	724
29.7	ADC 转换结果比较 .....	726
29.8	ADC 中断.....	727
29.9	ADCX 寄存器列表.....	727

---



29.10	寄存器说明 .....	728
29.10-1	ADCX 配置寄存器 0(ADCX_CR0).....	728
29.10-2	ADCX 配置寄存器 1(ADCX_CR1).....	731
29.10-3	ADCX 配置寄存器 2(ADCX_CR2).....	734
29.10-4	ADCX 通道 0 转换结果(ADCX_RESULT0).....	735
29.10-5	ADCX 通道 1 转换结果(ADCX_RESULT1).....	735
29.10-6	ADCX 通道 2 转换结果(ADCX_RESULT2).....	736
29.10-7	ADCX 通道 3 转换结果(ADCX_RESULT3).....	736
29.10-8	ADCX 通道 4 转换结果(ADCX_RESULT4).....	737
29.10-9	ADCX 通道 5 转换结果(ADCX_RESULT5).....	737
29.10-10	ADCX 通道 6 转换结果(ADCX_RESULT6).....	738
29.10-11	ADCX 通道 7 转换结果(ADCX_RESULT7).....	738
29.10-12	ADCX 转换结果(ADCX_RESULT).....	739
29.10-13	ADCX 转换结果累加值(ADCX_RESULT_ACC).....	739
29.10-14	ADCX 比较上阈值(ADCX_LT).....	740
29.10-15	ADCX 比较下阈值(ADCX_LT).....	740
29.10-16	ADCX 中断使能寄存器(ADCX_INTEN).....	741
29.10-17	ADCX 中断清除寄存器(ADCX_INTCLR).....	742
29.10-18	ADCX 掩码前中断状态寄存器(ADCX_RAWINTSR).....	743
29.10-19	ADCX 掩码后中断状态寄存器(ADCX_MSKINTSR).....	744
29.10-20	ADC/VTSE TRIM 寄存器 2(ADCVTSETRIM).....	745
<b>【30】</b>	<b>低电压检测器(LVD).....</b>	<b>746</b>
30.1	LVD 简介 .....	746
30.2	LVD 框图 .....	746
30.3	数字滤波 .....	747
30.4	配置示例 .....	747
30.4-1	LVD 配置为低电压复位 .....	747
30.4-2	LVD 配置为电压变化中断 .....	747
30.5	LVD 寄存器列表 .....	748
30.6	寄存器说明 .....	749
30.6-1	LVD 控制寄存器(LVD_CR) .....	749
30.6-2	LVD 状态寄存器(LVD_SR) .....	751
<b>【31】</b>	<b>电压比较器(VC) .....</b>	<b>752</b>
31.1	VC 简介 .....	752



---

31.2	VC 框图 .....	752
31.3	数字滤波 .....	753
31.4	配置示例 .....	753
31.5	VC 寄存器列表 .....	754
31.6	VC 寄存器说明 .....	755
31.6-1	VC0 电压控制寄存器(VC0_CR0) .....	755
31.6-2	VC0 控制寄存器(VC0_CR1) .....	758
31.6-3	VC0 输出配置寄存器(VC0_OUTCFG) .....	760
31.6-4	VC0 状态寄存器(VC0_SR) .....	762
31.6-5	VC1 电压控制寄存器(VC1_CR0) .....	763
31.6-6	VC1 控制寄存器(VC1_CR1) .....	766
31.6-7	VC1 输出配置寄存器(VC1_OUTCFG) .....	767
31.6-8	VC1 状态寄存器(VC1_SR) .....	769
<b>【32】</b>	<b>运算放大器 (OPA) .....</b>	<b>770</b>
32.1	OPA 简介 .....	770
32.2	OPA 特性 .....	770
32.3	OPA 框图 .....	771
32.4	PGA 增益 .....	772
32.4-1	单位增益 PGA .....	772
32.4-2	正向输入 PGA .....	773
32.4-3	互补输入 PGA .....	773
32.4-4	串级 PGA .....	775
32.5	OPA 寄存器列表 .....	776
32.6	寄存器说明 .....	777
32.6-1	OP1 控制寄存器 0 (OP1_CR0) .....	777
32.6-2	OP2 控制寄存器 0 (OP2_CR0) .....	778
32.6-3	OPDAC 控制寄存器 (OPDAC_CR) .....	779
<b>【33】</b>	<b>PDMA 控制器 .....</b>	<b>780</b>
33.1	PDMA 概述 .....	780
33.2	PDMA 主要特性 .....	780
33.3	功能框图 .....	781
33.4	功能描述 .....	782
33.4-1	通道设置 .....	783
33.4-2	PDMA 仲裁率 .....	786

---



33.4-3	通道优先级 .....	786
33.4-4	DMA 通道控制数据结构.....	789
33.4-4.1	控制数据配置( CHNL_CFG).....	792
33.4-5	DMA 传输周期的类型.....	795
33.4-5.1	无效.....	796
33.4-5.2	基本.....	796
33.4-5.3	自动请求模式 .....	797
33.4-5.4	Ping-pong 模式.....	798
33.4-5.5	存储器分散/聚集式 .....	802
33.4-5.6	外设分散/聚集式 .....	805
33.4-6	错误管理 .....	807
33.5	中断.....	807
33.6	DMAX 寄存器列表(X=1,2).....	808
33.7	寄存器说明 .....	810
33.7-1	DMAX 状态寄存器(DMAX_STA) .....	810
33.7-2	DMAX 配置寄存器(DMAX_CFG) .....	811
33.7-3	DMAX 主要控制数据地址指针寄存器(DMAX_PDBPTR) .....	812
33.7-4	DMAX 备用控制数据地址指针寄存器(DMAX_ADBPTR) .....	813
33.7-5	DMAX 通道等待请求状态寄存器(DMAX_CHWTREQ) .....	814
33.7-6	DMAX 通道软件请求寄存器(DMAX_CHSWREQ) .....	815
33.7-7	DMAX 通道使用多笔传输设定寄存器 (DMAX_CHBTSET) .....	816
33.7-8	DMAX 通道使用多笔传输清除寄存器(DMAX_CHBTCLR) .....	817
33.7-9	DMAX 通道请求屏蔽设置寄存器(DMAX_CHRMKSET) .....	818
33.7-10	DMAX 通道请求屏蔽清除寄存器(DMAX_CHRMKCLR) .....	819
33.7-11	DMAX 通道使能设置寄存器(DMAX_CHENSET) .....	820
33.7-12	DMAX 通道使能清除寄存器 (DMAX_CHENCLR) .....	821
33.7-13	DMAX 通道备用设置寄存器(DMAX_CHALTSET) .....	822
33.7-14	DMAX 通道备用清除设置寄存器(DMAX_CHALTCLR) .....	823
33.7-15	DMAX 通道优先级设置寄存器(DMAX_CHPRISET) .....	824
33.7-16	DMAX 通道优先级清除寄存器(DMAX_CHPRICLR) .....	825
33.7-17	DMAX 总线错误清除寄存器寄存器(DMAX_CHERRCLR) .....	826
33.7-18	DMAX 通道传输完成中断设置寄存器(DMAX_CHINTSET) .....	827
33.7-19	DMAX 通道中断使能清除寄存器(DMAX_CHINTCLR) .....	828
33.7-20	DMAX 通道传输完成中断旗标寄存器(DMAX_CHTCIF) .....	829



33.7-21	DMAX 通道等待请求使能寄存器(DMAX_CHWTONREG).....	830
33.7-22	DMAX 通道选择请求寄存器(DMAX_CHSELREQ).....	832
33.7-23	DMAX 集成配置寄存器(DMAX_INTCFG).....	833
33.7-24	DMAX 停滞状态寄存器(DMAX_STALL).....	834
33.7-25	DMAX 请求状态 (DMAX_CHREQ).....	835
33.7-26	DMAX 通道单笔请求状态 (DMAX_CHSREQ).....	836
33.7-27	DMAX 通道传输完成的状态设置寄存器(DMAX_CHDONESET).....	837
33.7-28	DMAX 通道传输完成状态清 0 寄存器(DMAX_CHDONECLR).....	838
33.7-29	DMAX 通道正在传输中的状态设置寄存器(DMAX_CHACTSET).....	839
33.7-30	DMAX 通道正在传输中的状态清 0 寄存器(DMAX_CHACTCLR).....	840
33.7-31	DMAX 总线错误清除寄存器寄存器(DMAX_ERRSET).....	841
<b>【34】</b>	<b>QUAD-SPI 接口(QSPI).....</b>	<b>842</b>
34.1	概述 .....	842
34.2	QSPI 功能框图 .....	842
34.3	QSPI 主要特性 .....	844
34.4	QSPI 命令序列 .....	844
34.4-1	地址阶段 .....	844
34.4-2	四线交替字节阶段 .....	844
34.4-3	空指令周期阶段 .....	845
34.4-4	数据阶段 .....	845
34.5	QUADSPI 信号接口协议模式 .....	846
34.5-1	单线 SPI 模式(Single SPI Mode) .....	846
34.5-2	双线 SPI 模式(Dual SPI Mode) .....	846
34.5-3	四线 SPI 模式(Quad SPI Mode) .....	846
34.5-4	双闪存模式 .....	847
34.6	QUADSPI 间接模式 .....	847
34.6-1	触发命令启动 .....	848
34.6-2	FIFO 和数据管理 .....	848
34.7	QUADSPI 状态轮询模式 .....	849
34.8	QUADSPI 内存映像模式 .....	849
34.9	QUADSPI Flash 配置 .....	850
34.10	QUADSPI 延迟数据采样数据进行采样 .....	850
34.11	QUADSPI 配置 .....	850
34.12	QUADSPI 用法 .....	851



34.13	只发送一次指令 .....	853
34.14	QUADSPI 错误管理.....	853
34.15	QUADSPI 忙位和中止功能 .....	853
34.16	nCTS 行为 .....	854
34.17	QUADSPI 中断.....	855
34.18	QSPI 寄存器列表.....	856
34.19	QSPI 寄存器列表.....	857
34.19-1	QSPI 控制寄存器 1(QUADSPI_CR) .....	857
34.19-2	QSPI 控制寄存器 1(QUADSPI_DCR) .....	862
34.19-3	QSPI 状态寄存器(QUADSPI_DCR).....	864
34.19-4	QSPI 标志清除寄存器(QUADSPI_FCR) .....	866
34.19-5	QSPI 数据长度寄存器(QUADSPI_DLR) .....	867
34.19-6	QSPI 通信配置寄存器(QUADSPI_CCR) .....	868
34.19-7	QSPI 地址寄存器(QUADSPI_AR) .....	871
34.19-8	QSPI 可选字节寄存器(QUADSPI_ABR) .....	872
34.19-9	QSPI 数据寄存器(QUADSPI_DR).....	873
34.19-10	QSPI 轮询屏蔽寄存器(QUADSPI_PSMKR) .....	874
34.19-11	QSPI 轮询状态匹配寄存器(QUADSPI_PSMAR).....	875
34.19-12	QSPI 轮询间隔寄存器(QUADSPI_PIR) .....	876
34.19-13	QSPI 超时寄存器(QUADSPI_LPTR) .....	877
<b>【35】</b>	<b>真随机数发生器 (TRNG) .....</b>	<b>878</b>
35.1	概述 .....	878
35.2	功能框图 .....	878
35.3	功能描述 .....	878
35.4	TRNG 寄存器列表 .....	879
35.5	寄存器说明 .....	880
35.5-1	TRNG 控制寄存器(TRNG_CR) .....	880
35.5-2	TRNG 状态寄存器(TRNG_SR).....	881
35.5-3	TRNG 数据寄存器(TRNG_DR) .....	882
<b>【36】</b>	<b>高级加密标准模块 (AES) .....</b>	<b>883</b>
36.1	概述 .....	883
36.2	特性.....	883
36.3	AES 中断 .....	883
36.4	AES 功能描述.....	884



36.4-1	加密和派生密钥(Encryption and derivation keys).....	885
36.4-2	加密和 AES 链接算法(AES chaining algorithms).....	885
36.4-2.1	电子密码本 (ECB).....	886
36.4-2.2	密码块链接 (CBC).....	887
36.4-2.3	计数器模式(CTR).....	888
36.4-3	数据型态(Data Type).....	890
36.5	AES 寄存器列表 .....	893
36.6	寄存器说明 .....	894
36.6-1	AES 控制寄存器(AES_CR) .....	894
36.6-2	AES 状态寄存器(AES_SR).....	897
36.6-3	AES 输入数据寄存器(AES_DINR) .....	899
36.6-4	AES 输出数据寄存器(AES_OUTR) .....	900
36.6-5	AES KEY0 寄存器(AES_KEY0R) (LSB: key[31:0]) .....	901
36.6-6	AES KEY1 寄存器(AES_KEY1R) (LSB: key[63:32]) .....	902
36.6-7	AES KEY2 寄存器(AES_KEY1R) (MSB: key[95:64]) .....	903
36.6-8	AES KEY3 寄存器(AES_KEY1R) (MSB: key[127:96]).....	903
36.6-9	AES IVR0 寄存器(AES_IVR0)( IVR[31:0]) .....	904
36.6-10	AES IVR1 寄存器(AES_IVR1)( IVR[63:32]) .....	904
36.6-11	AES IVR2 寄存器(AES_IVR2)( IVR[95:64]) .....	905
36.6-12	AES IVR3 寄存器(AES_IVR3)( IVR[127:96]) .....	905
<b>【37】</b>	<b>选项字节区(Option Bytes).....</b>	<b>906</b>
37.1	用户配置寄存器 0(USERCFG0) .....	921
37.2	用户配置寄存器 1(USERCFG1) .....	923
37.3	用户配置寄存器 2(USERCFG2) .....	924
37.4	注意 .....	924
<b>【38】</b>	<b>Debug 支持(DBG) .....</b>	<b>925</b>
38.1	SWD 调试接口说明 .....	925
38.1-1	SWD 调试接口的引脚分配.....	926
38.1-2	SWD 引脚的内部上拉和下拉 .....	926
38.2	SWD 保护位工作原理 .....	927
38.3	在低功耗模式下使用 SWD.....	927
38.4	DBG 寄存器列表 .....	929
38.5	DBG 寄存器列表 DBG 寄存器说明 .....	929
38.5-1	Debug 模式控制寄存器(DBG_APBFZ) .....	929



---

【39】 在线电路编程(ICP) .....	934
【40】 附录 1: SysTick 定时器(SYST) .....	935
40.1    SysTick 定时器简介 .....	935
40.2    设置 SysTick 定时器.....	935
40.3    SysTick 定时器寄存器列表 .....	936
40.4    SysTick 定时器寄存器说明 .....	936
40.4-1    SysTick 定时器控制和状态寄存器(SYST_CSR).....	936
40.4-2    SysTick 定时器重载值寄存器(SYST_RVR).....	938
40.4-3    SysTick 定时器当前值寄存器(SYST_CVR) .....	938
40.4-4    SysTick 定时器校准值寄存器(SYST_CALIB) .....	938
【41】 Flash 控制器(Flash) .....	939
41.1    Flash 控制器概述 .....	939
41.2    Flash 结构框图 .....	939
41.3    功能描述 .....	940
41.3-1    擦除操作 .....	940
41.3-1.1    Sector 擦除 .....	940
41.3-1.2    Chip 擦除 .....	942
41.3-2    写操作.....	943
41.3-3    读操作.....	943
41.3-4    擦写时间 .....	944
41.3-5    擦写保护 .....	944
41.3-5.1    擦写保护位 .....	944
41.3-5.2    PC 地址擦写保护 .....	944
41.3-5.3    寄存器写保护 .....	944
41.3-6    系统 BOOT 地址映像 .....	946
41.3-6.1    APP 程序 boot.....	946
41.3-7    系选项字节区(Option Bytes).....	947
41.3-8    在系统编程(ISP) .....	948
41.3-8.1    内部配置与软件进入 ISP .....	948
41.3-8.2    外部 BOOT PIN 选择 .....	949
41.4    Flash 寄存器列表 .....	951
41.5    寄存器说明 .....	952
41.5-1    Flash_CR(Flash 控制寄存器).....	952
41.5-2    Flash_IFR(Flash 中断标记寄存器) .....	953

---



---

41.5-3	Flash_ICLR(Flash 中断标记清除寄存器).....	954
41.5-4	Flash_BYPASS(BYPASS 序列寄存器) .....	955
41.5-5	Flash_SLOCK0(sector 擦写保护寄存器 0) .....	956
41.5-6	Flash_SLOCK1(sector 擦写保护寄存器 1) .....	957
41.5-7	Flash_ISPCON(Flash ISPCON 配置寄存器).....	958
41.5-8	Flash 读取周期寄存器(Flash_RCR).....	959
<b>【42】</b>	<b>版本修订纪录 .....</b>	<b>960</b>

ZBT CONFIDENTIAL



## 图索引

Figure 3-1 内部功能框图结构.....	51
Figure 4-1 ZB32F103 LQFP64.....	64
Figure 5-1 总线矩阵图 .....	82
Figure 5-2 储存器映像图 .....	83
Figure 5-3 BOOT PIN 地址映射图 .....	88
Figure 6-1 电源框图 .....	89
Figure 8-1 复位来源示意图.....	118
Figure 8-2 系统时钟控制模块图.....	120
Figure 8-3HXT/LXT 时钟源 .....	121
Figure 8-4 PLL 时钟源 .....	123
Figure 8-5 内部高速时钟启动示意图 .....	124
Figure 8-6 时钟切换示意图 .....	125
Figure 10-1MCU 的 各種異常源框图 .....	186
Figure 10-2EXTI 功能框图 .....	192
Figure 10-3 外部中断通用 I/O 映像图.....	195
Figure 11-1 标准 I/O 端口的位基本结构.....	204
Figure 11-2 5 伏兼容 I/O 端口位的位基本结构 .....	205
Figure 11-3 输出配置 .....	209
Figure 11-4 输出配置 .....	210
Figure 11-5 复用功能配置 .....	211
Figure 11-6 高阻抗的模拟输入配置 .....	212
Figure 11-7 PAD output enable logic.....	219
Figure 11-8 PAD output data logic .....	219
Figure 11-9 PAD input enable logic.....	219
Figure 11-10 PAD input data logic .....	219
Figure 12-1CRC 应用示意图 .....	254
Figure 13-1 高级控制定时器框图 .....	261
Figure 13-2 当预分频器的参数从 1 变到 2 时，计数器的时序图 .....	263
Figure 13-3 当预分频器的参数从 1 变到 4 时，计数器的时序图 .....	263
Figure 13-4 计数器时序图：内部时钟分频因子为 1 .....	264
Figure 13-5 计数器时序图：内部时钟分频因子为 2 .....	265
Figure 13-6 计数器时序图：内部时钟分频因子为 4 .....	265
Figure 13-7 计数器时序图：内部时钟分频因子为 N .....	265



---

Figure 13-8 计数器时序图: 当 ARPE=0 时的更新事件(TIM1X*_ARR 没有预装入) .....	266
Figure 13-9 计数器时序图: 当 ARPE=1 时的更新事件(预装入了 TIM1X*_ARR) .....	267
Figure 13-10 计数器时序图: 内部时钟分频因子为 1 .....	268
Figure 13-11 计数器时序图: 内部时钟分频因子为 2 .....	269
Figure 13-12 计数器时序图: 内部时钟分频因子为 4 .....	269
Figure 13-13 计数器时序图: 内部时钟分频因子为 N .....	270
Figure 13-14 计数器时序图: 当没有使用重复计数器时的更新事件 .....	270
Figure 13-15 计数器时序图: 内部时钟分频因子为 1, TIM1X*_ARR=0x6 .....	272
Figure 13-16 计数器时序图: 内部时钟分频因子为 2 .....	272
Figure 13-17 计数器时序图: 内部时钟分频因子为 4, TIM1X*_ARR=0x36 .....	273
Figure 13-18 计数器时序图: 内部时钟分频因子为 N .....	273
Figure 13-19 计数器时序图: ARPE=1 时的更新事件(计数器下溢) .....	274
Figure 13-20 计数器时序图: ARPE=1 时的更新事件(计数器溢出) .....	274
Figure 13-21 不同模式下更新速率的例子, 及 TIM1X*_RCR 的寄存器设置 .....	276
Figure 13-22 一般模式下的控制电路, 内部时钟分频因子为 1 .....	277
Figure 13-23 TI2 外部时钟连接例子 .....	278
Figure 13-24 外部时钟模式 1 下的控制电路 .....	279
Figure 13-25 外部触发输入框图 .....	279
Figure 13-26 外部时钟模式 2 下的控制电路 .....	280
Figure 13-27 捕获/比较通道(如: 通道 1 输入部分) .....	281
Figure 13-28 捕获/比较通道 1 的主电路 .....	282
Figure 13-29 捕获/比较通道的输出部分(通道 1 至 3) .....	282
Figure 13-30 捕获/比较通道的输出部分(通道 4) .....	283
Figure 13-31 PWM 输入模式时序 .....	285
Figure 13-32 输出比较模式, 翻转 OC1 .....	287
Figure 13-33 边沿对齐的 PWM 波形(ARR=8) .....	289
Figure 13-34 中央对齐的 PWM 波形(APR=8) .....	290
Figure 13-35 带死区插入的互补输出 .....	292
Figure 13-36 死区波形延迟大于负脉冲 .....	292
Figure 13-37 死区波形延迟大于正脉冲 .....	293
Figure 13-38 响应刹车的输出 .....	295
Figure 13-39 清除 TIM1X* 的 OCxREF .....	296
Figure 13-40 产生六步 PWM, 使用 COM 的例子(OSSR=1) .....	297
Figure 13-41 单脉冲模式的例子 .....	298

---



---

Figure 13-42 编码器模式下的计数器操作实例 .....	301
Figure 13-43 IC1FP1 反相的编码器接口模式实例 .....	301
Figure 13-44 霍尔传感器接口的实例 .....	304
Figure 13-45 复位模式下的控制电路 .....	306
Figure 13-46 门控模式下的控制电路 .....	307
Figure 13-47 外部时钟模式 2 + 触发模式下的控制电路 .....	309
Figure 14-1 通用定时器框图 .....	351
Figure 14-2 当预分频器的参数从 1 变到 2 时, 计数器的时序图 .....	353
Figure 14-3 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	353
Figure 14-4 计数器时序图: 内部时钟分频因子为 1 .....	354
Figure 14-5 计数器时序图: 内部时钟分频因子为 2 .....	355
Figure 14-6 计数器时序图: 内部时钟分频因子为 4 .....	355
Figure 14-7 计数器时序图: 内部时钟分频因子为 N .....	355
Figure 14-8 计数器时序图: 当 ARPE=0 时的更新事件(TIM2X*_ARR 没有预装入) .....	356
Figure 14-9 计数器时序图: 当 ARPE=1 时的更新事件(预装入了 TIM2X*_ARR) .....	356
Figure 14-10 计数器时序图: 内部时钟分频因子为 1 .....	357
Figure 14-11 计数器时序图: 内部时钟分频因子为 2 .....	358
Figure 14-12 计数器时序图: 内部时钟分频因子为 4 .....	358
Figure 14-13 计数器时序图: 内部时钟分频因子为 N .....	358
Figure 14-14 计数器时序图: 当没有使用重复计数器时的更新事件 .....	359
Figure 14-15 计数器时序图: 内部时钟分频因子为 1, TIM2X*_ARR=0x6 .....	360
Figure 14-16 计数器时序图: 内部时钟分频因子为 2 .....	361
Figure 14-17 计数器时序图: 内部时钟分频因子为 4, TIM2X*_ARR=0x36 .....	361
Figure 14-18 计数器时序图: 内部时钟分频因子为 N .....	361
Figure 14-19 计数器时序图: ARPE=1 时的更新事件(计数器下溢) .....	362
Figure 14-20 计数器时序图: ARPE=1 时的更新事件(计数器溢出) .....	362
Figure 14-21 一般模式下的控制电路, 内部时钟分频因子为 1 .....	363
Figure 14-22 TI2 外部时钟连接例子 .....	364
Figure 14-23 外部时钟模式 1 下的控制电路 .....	365
Figure 14-24 外部触发输入框图 .....	365
Figure 14-25 外部时钟模式 2 下的控制时序 .....	366
Figure 14-26 捕获/比较通道(如: 通道 1 输入部分) .....	366
Figure 14-27 捕获/比较通道 1 的主电路 .....	367
Figure 14-28 捕获/比较通道的输出部分(通道 1) .....	367

---



---

Figure 14-29 PWM 输入模式时序 .....	369
Figure 14-30 输出比较模式, 翻转 OC1.....	371
Figure 14-31 边沿对齐的 PWM 波形(ARR=8).....	373
Figure 14-32 中央对齐的 PWM 波形(APR=8) .....	374
Figure 14-33 单脉冲模式的例子 .....	375
Figure 14-34 清除 TIM2X* 的 OCxREF.....	377
Figure 14-35 编码器模式下的计数器操作实例 .....	379
Figure 14-36 IC1FP1 反相的编码器接口模式实例 .....	379
Figure 14-37 复位模式下的控制电路 .....	381
Figure 14-38 门控模式下的控制电路 .....	381
Figure 14-39 触发器模式下的控制电路 .....	382
Figure 14-40 外部时钟模式 2 + 触发模式下的控制电路 .....	383
Figure 14-41 主/从定时器的例子 .....	383
Figure 14-42 的 OC1REF 控制定时器 2,3,4,5 .....	384
Figure 14-43 通过使能定时器 1 可以控制定时器 2 .....	385
Figure 14-44 使用定时器 1 的更新触发定时器 2 .....	386
Figure 14-45 利用定时器 1 的使能触发定时器 2 .....	386
Figure 14-46 利用定时器 1 的使能触发定时器 2 .....	388
Figure 15-1 PCA 整体框图 .....	419
Figure 15-2 PCA 计数器框图 .....	421
Figure 15-3 PCA 捕获功能框图 .....	422
Figure 15-4 PCA 比较功能框图 .....	423
Figure 15-5 PCA PWM 功能框图 .....	424
Figure 15-6 PCA PWM 输出波形 .....	425
Figure 16-1 Base Timer 框图 .....	441
Figure 16-2 Timer 模式 1 框图 .....	443
Figure 16-3 Timer 模式 2 框图 .....	444
Figure 16-4 32 位模式 1 时序图(max=0xFFFF FFFF) .....	445
Figure 16-5 32 位模式 2 时序图(PCLK 二分频, max=0xFFFF FFFF) .....	445
Figure 17-1 LPTIMER 结构框图 .....	453
Figure 17-2 LPTIMER 模式 1 .....	454
Figure 17-3 LPTIMER 模式 2 .....	455
Figure 18-1 自唤醒定时器结构图 .....	465
Figure 19-1 IWDG 整体框图 .....	471

---



---

Figure 20-1 WWDG 结构框图.....	481
Figure 21-1 UART 结构图.....	493
Figure 21-2 USART 结构图 .....	493
Figure 21-3 Mode0 发送数据 .....	495
Figure 21-4 Mode0 接收数据 .....	495
Figure 21-5 Mode1 发送数据 .....	496
Figure 21-6 Mode1 接收数据 .....	497
Figure 21-7 Mode2 发送数据 .....	497
Figure 21-8 Mode2 接收数据 .....	498
Figure 21-9 接收缓存.....	501
Figure 21-10 IrDA 结构框图 .....	503
Figure 21-11 IrDA 收发脉冲 .....	504
Figure 21-12 字长设置 .....	507
Figure 21-13 配置停止位 .....	508
Figure 21-14 配置停止位 .....	509
Figure 21-15 利用空闲总线检测的静默模式 .....	512
Figure 21-16 利用地址标记检测的静默模式 .....	513
Figure 21-17 LIN 模式下的断开检测(11 位断开长度 – 设置了 LBDL 位) .....	515
Figure 21-18 LIN 模式下的断开检测与帧错误的检测 .....	515
Figure 21-19 USART 同步传输的例子 .....	516
Figure 21-20 USART 数据时钟时序示例(M=0).....	517
Figure 21-21 USART 数据时钟时序示例(M=1).....	517
Figure 21-22 RX 数据采样/保持时间 .....	518
Figure 21-23 ISO7816-3 异步协议 .....	519
Figure 21-24 IrDA 框图 .....	522
Figure 21-25 IrDA 数据调制 .....	522
Figure 21-26 两个 USART 间的硬件流控制 .....	523
Figure 21-27 RTS 流控制.....	524
Figure 21-28 CTS 流控制 .....	524
Figure 21-29 USART 中断映像图 .....	525
<b>Figure 22-1 LPUART 结构框图 .....</b>	<b>558</b>
Figure 22-2 Mode 0 发送数据 .....	559
Figure 22-3 Mode 0 接收数据 .....	560
Figure 22-4 Mode 1 发送数据 .....	560

---



---

Figure 22-5 Mode 1 接收数据 .....	561
Figure 22-6 Mode 2 发送数据 .....	561
Figure 22-7 Mode 2 接收数据 .....	562
Figure 23-1 I2C 传输协议 .....	573
Figure 23-2 主机向从机传输数据 .....	574
Figure 23-3 主机向从机传输数据 .....	574
Figure 23-4 START 和 STOP 条件 .....	574
Figure 23-5 I2C 总线上位传输 .....	575
Figure 23-6 I2C 总线上应答信号 .....	575
Figure 23-7 I2C 功能模块图 .....	576
Figure 23-8 总线上的仲裁 .....	577
Figure 23-9 I2C 主机发送状态图 .....	582
Figure 23-10 I2C 主机接收状态图 .....	585
Figure 23-11 I2C 从机接收状态图 .....	589
Figure 23-12 I2C 从机发送状态图 .....	592
Figure 23-13 I2C1 Master DMA TX .....	595
Figure 23-14 I2C0 Master DMA RX .....	597
Figure 24-1 SPI 的方框图 .....	612
Figure 24-2 单主和单从应用 .....	613
Figure 24-3 硬件/软件的从选择管理 .....	613
Figure 24-4 数据时钟时序图 .....	615
Figure 24-5 主模式、全双工模式下(BIDIMODE=0 并且 RXONLY=0)连续传输时， TXE/RXNE/BSY 的变化示意图 .....	621
Figure 24-6 从模式、全双工模式下(BIDIMODE=0 并且 RXONLY=0)连续传输时， TXE/RXNE/BSY 的变化示意图 .....	622
Figure 24-7 主设备只发送模式(BIDIMODE=0 并且 RXONLY=0)下连续传输时， TXE/BSY 变化示意图 .....	623
Figure 24-8 从设备只发送模式(BIDIMODE=0 并且 RXONLY=0)下连续传输时， TXE/BSY 变化示意 .....	623
Figure 24-9 只接收模式(BIDIMODE=0 并且 RXONLY=1)下连续传输时， RXNE 变化示意图 .....	624
Figure 24-10 非连续传输发送(BIDIMODE=0 并且 RXONLY=0)时， TXE/BSY 变化示意图 .....	625
Figure 24-11 使用 DMA 发送 .....	630
Figure 24-12 使用 DMA 接收 .....	630

---



---

Figure 24-13 I2S 框图 .....	634
Figure 24-14 I2S 飞利浦协议波形(16/32 位全精度, CPOL = 0) .....	636
Figure 24-15 I2S 飞利浦协议标准波形(24 位帧, CPOL = 0) .....	636
Figure 24-16 发送 0x8EAA33.....	636
Figure 24-17 接收 0x8EAA33.....	637
Figure 24-18 I2S 飞利浦协议标准波形(16 位扩展至 32 位包帧, CPOL = 0) .....	637
Figure 24-19 MSB 对齐 16 位或 32 位全精度, CPOL = 0 .....	638
Figure 24-20 MSB 对齐 24 位数据, CPOL = 0.....	638
Figure 24-21 MSB 对齐 16 位数据扩展到 32 位包帧, CPOL = 0 .....	638
Figure 24-22 LSB 对齐 16 位或 32 位全精度, CPOL = 0 .....	639
Figure 24-23 LSB 对齐 24 位数据, CPOL = 0.....	639
Figure 24-24 要求发送 0x3478AE 的操作 .....	639
Figure 24-25 要求接收 0x3478AE 的操作 .....	640
Figure 24-26 LSB 对齐 16 位数据扩展到 32 位包帧, CPOL = 0 .....	640
Figure 24-27 PCM 标准波形(16 位).....	641
Figure 24-28 PCM 标准波形(16 位扩展到 32 位包帧).....	641
Figure 24-29 音频采样频率定义 .....	642
Figure 24-30 I2S 时钟发生器结构 .....	642
Figure 25-1 初始化过程中的复位与应答脉冲 .....	錯誤! 尚未定義書籤。
Figure 25-2 单总线通信协议中写时间隙时序图 .....	錯誤! 尚未定義書籤。
Figure 25-3 单总线通信协议中读时间隙时序图 .....	錯誤! 尚未定義書籤。
Figure 27-1 RTC 框图 .....	680
Figure 28-1 RTC 框图 .....	702
Figure 28-2 RTC 秒和闹钟波形示例图 .....	704
Figure 28-3 RTC 溢出波形图示例, PR=0003 示例图 .....	704
Figure 29-1 ADC 架构方块图 .....	718
Figure 29-2 ADC 转换时序图 .....	720
Figure 29-3 ADC 连续转换过程示例 .....	722
Figure 29-4 连续转换累加过程示例 .....	724
Figure 30-1 LVD 结构框图 .....	746
Figure 30-2 LVD 滤波输出 .....	747
Figure 31-1 VC0,VC1 结构框图 .....	752
Figure 31-2 VC 滤波输出 .....	753
Figure 32-1 OPA1,OPA2 结构框图 .....	771

---



---

Figure 33-1 PDMA 框图 .....	781
Figure 33-2 DMA1 通道请求框图 .....	785
Figure 33-3 DMA2 通道请求框图 .....	785
Figure 33-4 Polling flowchart .....	788
Figure 33-5 使用 16 通道的储存器映像含盖主要结构与备用结构 .....	790
Figure 33-6 基本模式图 .....	796
Figure 33-7 自动请求模式范例图 .....	797
Figure 33-8 Ping-pong 模式范例图 .....	799
Figure 33-9 存储器分散/聚集式 图例 .....	803
Figure 33-10 复制 Task A,B,C 描述元的配置数据图 .....	804
Figure 33-11 外设分散/聚集式范例图 .....	806
Figure 33-12 CHWTIONREQ[C]( DMAx_waitonreg[C]=1 的时序图 .....	830
Figure 33-13 CHx_CHWTIONREQ=0 的时序图 .....	831
Figure 34-1 QSPI 功能框图(BANK1) .....	842
Figure 34-2 QSPI 功能框图(BANK1,BANK2) .....	843
Figure 34-3 四线模式下的读命令时序 .....	844
Figure 34-4 nCS when CKMODE=0 (T = CLK period) .....	854
Figure 34-5 nCS when CKMODE=1 in SDR mode (T=CLK period) .....	854
Figure 34-6 nCS when CKMODE=1 with an abort (T=CLK period) .....	854
Figure 35-1 真随机数框图 .....	878
Figure 36-1 ECB 加密模式 .....	886
Figure 36-2 ECB 解密模式 .....	886
Figure 36-3 CBC 加密模式 .....	887
Figure 36-4 CBC 解密模式 .....	888
Figure 36-5 CTR 加密模式 .....	888
Figure 36-6 CTR 解密模式 .....	889
Figure 36-7 32 位计数器+随机数组织 .....	890
Figure 36-8 Data Type= 2' b00 .....	891
Figure 36-9 Data Type= 2' b01 .....	891
Figure 36-10 Data Type= 2' b10 .....	891
Figure 36-11 Data Type= 2' b11 .....	892
Figure 38-1 SDIO(多)数据块读操作图 .....	912
Figure 38-2 SDIO(多)数据块写操作图 .....	912
Figure 38-3 SDIO Block Diagram .....	914

---



---

Figure 38-4 S Data Transmit State Machine.....	917
Figure 38-4 S Data-Receive State Machine .....	917
Figure 39-1 ZB32F103 级别和 Cortex™-M3 级别的调试框图 .....	925
Figure 39-1 ICP 接口连接方式示意图 .....	934
Figure 41-1Flash 结构框图.....	939
Figure 41-2 擦除操作步骤.....	940
Figure 41-3Chip 擦除操作步骤 .....	942
Figure 41-4 写操作步骤 .....	943
Figure 41-5 写寄存器 BYPASS 序列 .....	945
Figure 41-6APP 程序区 boot 地址映像图 .....	946
Figure 41-7 内部配置与软件进入 ISP 流程图 .....	948
Figure 41-8 外部 BOOT0,BOOT1 PIN 选择流程图 .....	949
Figure 41-9BOOT0,BOOT1 PIN 时序图 .....	950

---

ZBT CONFIDENTIAL



## 表索引

Table 3-1 ZB32F103 芯片特性与周边配备 .....	50
Table 3-2 定时器菜单 .....	56
Table 4-1 引脚定義说明表.....	68
Table 4-2 引脚 ST MD 功能说明表 .....	74
Table 4-3 引脚 PIN MD 复用功能说明表.....	77
Table 4-4 模块与引脚信号说明表 .....	80
Table 5-1 地址对应表 .....	86
Table 5-2BOOT 启动模式.....	87
Table 6-1 运行模式下的菜单.....	92
Table 6-2 低功耗模式表 .....	94
Table 6-3 SLEEP-NOW 模式表.....	95
Table 6-4 SLEEP-ON-EXIT 模式.....	96
Table 6-5 停止模式表 .....	97
Table 6-6 待机模式表 .....	98
Table 6-7 PMU 寄存器列表和复位值 .....	100
Table 7-1 BKP 寄存器列表和复位值 .....	110
Table 8-1 PLL 参数表.....	124
Table 8-2 RCC 寄存器列表和复位值 .....	128
Table 9-1 SYSCON 寄存器列表和复位值 .....	171
Table 10-1 中断向量表 .....	187
Table 10-2 其他 EXTI 线的连接表 .....	195
Table 10-3 寄存器列表和复位值 .....	196
Table 11-1 端口位配置表.....	205
Table 11-2 高级定时器 TIM1/TIM/TIM1B 引脚配置表 .....	213
Table 11-3 通用定时器 TIM2/3/4/5 引脚配置表 .....	213
Table 11-4 通用定时器 TIM2/3/4/5 引脚配置表 .....	213
Table 11-5USART 引脚配置表 .....	214
Table 11-6 SPI 引脚配置表 .....	215
Table 11-7 I2S 引脚配置表.....	216
Table 11-8 CAN 引脚配置表.....	216
Table 11-9 USB 引脚配置表.....	217
Table 11-10 SDIO 引脚配置表 .....	218
Table 11-11 SDIO 引脚配置表 .....	218



---

Table 11-12 SDIO 引脚配置表 .....	218
Table 11-13 CAN 引脚配置表 .....	220
Table 11-14 调试接口信号引脚配置表 .....	221
Table 11-15 调试端口映像表 .....	222
Table 11-16 ADC1 外部触发注入转换复用功能重映射表 .....	222
Table 11-17 ADC1 外部触发注入转换复用功能重映射表 .....	222
Table 11-18 ADC2 外部触发注入转换复用功能重映射表 .....	223
Table 11-19 ADC2 外部触发注入转换复用功能重映射表 .....	223
Table 11-20 TIM2C 复用功能重映像表 .....	223
Table 11-21 TIM2B 复用功能重映像表 .....	223
Table 11-22 TIM2A 复用功能重映像表 .....	224
Table 11-23 TIM2 复用功能重映像表 .....	225
Table 11-24 TIM1 复用功能重映像表 .....	225
Table 11-25 USART3 复用功能重映像表 .....	226
Table 11-26 USART2 复用功能重映像表 .....	226
Table 11-27 USART1 复用功能重映像表 .....	226
Table 11-28 I2C1 复用功能重映像表 .....	227
Table 11-29 SPI1 复用功能重映像表 .....	227
Table 11-30 SPI3 复用功能重映像表 .....	227
Table 11-31 GPIOxAFIO 寄存器列表和复位值 .....	228
Table 12-1 CRC 寄存器列表和复位值 .....	256
Table 13-1 计数方向与编码器信号的关系 .....	300
Table 13-2 TIM1,TIM1A,TIM1B 寄存器列表和复位值 .....	310
Table 13-3 TIM1X* 内部触发连接 .....	318
Table 13-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位 .....	336
Table 14-1 计数方向与编码器信号的关系 .....	378
Table 14-2 TIM2X* 寄存器列表和复位值 .....	389
Table 14-3 TIM2X* 内部触发连接 .....	395
Table 15-1 PCA 比较/捕获功能模块设置 .....	425
Table 15-2 PCA 寄存器列表和复位值 .....	427
Table 16-1 PCA 寄存器列表和复位 .....	447
Table 17-1 LPTIM 寄存器列表和复位值 .....	457
Table 18-1 AWK 寄存器列表和复位值 .....	466
Table 19-1 IWDG 寄存器列表和复位值 .....	473

---



---

Table 20-1 WWDG 寄存器列表和复位值.....	484
Table 21-1 USART 工作模式 选择表 .....	505
Table 21-2 USART 工作模式控制表.....	505
Table 21-3 USART 帧格式.....	513
Table 21-4 USART 中断请求 .....	525
Table 21-5 UART 寄存器列表.....	532
Table 21-6 USART 寄存器列表.....	532
Table 22-1 LPUART 寄存器列表.....	566
Table 23-1 I2CX* 寄存器列表 .....	604
Table 24-1 SPI 中断请求.....	633
Table 24-2 使用标准的 8MHz HXT 时钟得到精确的音频频率 .....	643
Table 24-3 I2S 中断请求.....	651
Table 24-4 SPI/I2S 寄存器列表和复位值.....	652
Table 25-1 One-Wire 接口寄存器列表和复位值.....	錯誤! 尚未定義書籤。
Table 26-1 CLKTRIM 寄存器列表和复位值.....	672
Table 27-1 RTC 寄存器列表和复位值 .....	685
Table 28-1 RTC 寄存器列表和复位值 .....	705
Table 29-1 ADCX 寄存器列表和复位值 .....	727
Table 30-1 LVD 寄存器列表.....	748
Table 31-1 VC 寄存器列表和复位值 .....	754
Table 32-1 OPA 寄存器列表和复位值 .....	776
Table 33-1 DMA1 使用的通道与请求列表 .....	783
Table 33-2 DMA2 使用的通道与请求列表 .....	784
Table 33-3 AHB bus transfer arbitration interval .....	786
Table 33-4 通道控制数据结构表 .....	789
Table 33-5 控制数据配置表 .....	792
Table 33-6 存储器分散/聚集式-信道配置表 .....	802
Table 33-7 外设分散/聚集式 -信道配置表 .....	805
Table 33-8 DMA 寄存器列表和复位值.....	809
Table 34-1 QSPI 中断表 .....	855
Table 34-2 QSPI 寄存器列表和复位值 .....	856
Table 35-1 寄存器列表和复位值 .....	879
Table 36-1 AES 中断表 .....	883
Table 38-1 Debug 寄存器映象和复位值 .....	929

---



---

Table 40-1 SysTick 定时器寄存器列表和复位值 .....	936
Table 41-1 寄存器列表和复位值 .....	951

ZBT CONFIDENTIAL

---



## 【1】 简介

ZB32F103 是一款内嵌 32 位 ARM® Cortex®-M3 内核的超低功耗和宽电压工作范围(1.8V~5.5V)的微控制器，最高可运行在 108MHz，内置 512K 字节的Flash，320K 字节的 SRAM。

处理器支持增强的高效 DSP 指令集，包含扩展的单周期 16/32 位乘法累加器（MAC）、双 16 位 MAC 指令、优化的 8/16 位 SIMD 运算及饱和运算指令，并且具有单精度（IEEE-754）浮点运算单元（FPU）。

ZB32F103集成了3個 12 位 1Msps 高精度 SAR 型 ADC(16 channel)、DAC(12-bits)X2、OPAX3,、RTC、比较器 X2、多路 UARTX2、USARTX3、LPUART、SPI/I2S、QSPIX2、I2CX2、CANX1,USB2.0(FS) 和 PWMx (可达独立出口或互补式出口)等丰富的外设接口，具有高整合度、高抗干扰、高可靠性的特点。

ZB32F103 系列具有宽电压工作范围、低功耗、低待机电流、高集成度外设、高操作效率、快速唤醒及高性价比等优势，广泛适用于满足数位讯号控制市场对兼具高效且容易使用之控制和讯号处理功能的需求，这些产业包括马达控制、汽车、电源管理、嵌入式音讯和工业自动化市场



## 【2】 产品特性

### ■ ARM® Cortex®-M3 内核

- M3 CPU 最高运行到108MHz
- 内建MPU,NVIC .
- 一个24位系统定时器
- 支持低功耗睡眠模式
- 内建单周期32位硬件乘法器,除法器
- 内建硬件浮点运算,DSP单元.

### ■ 内置储存器

- 程序执行高达 320 kB 的 SRAM 用于code和数据使用(0-wait).
- 4KB Retention SRAM.
- 8KB OTP 包含boot code 与配置设定.
- 高达4Mb(512KB) 的Flash code和数据使用.

### ■ 工作条件

- 宽电压工作范围1.8V至3.6V
- 宽工作频率12M高至108MHz
- 工作温度: -40°C至+85°C

### ■ 时钟源

#### ● 5路可选时钟源

- 外部4MHz~24MHz高速晶振
- 外部32.768KHz晶振
- 内部12/11.059 MHz高速时钟
- 内部低速38.4KHz/32.768KHz时钟
- PLL时钟 : Max 192 MHz
- 支持硬件时钟监视

#### ● RTC

- 支持RTC计数(秒/分/小时)及万年历功能(日/月/年)
- 支持闹铃功能寄存器(秒/分/小时/日/月/年)
- 支持RTC从Deep Sleep模式唤醒系统

### ■ 电源管理

- POR,PDR,LVR



- RTC 及后备存储器（128Byte）专用供电区 VBAT
- 三种低功耗工作模式：Sleep、Deep Sleep Mode, standby mode
- 低电压检测，可配置为中断或复位
- 唤醒@sleep：所有中断源都可唤醒
- 唤醒@Deep sleep: 16 个外部中断GPIO PIN 与 RTC(低速时钟工作)
- 唤醒@ standby mode: wake up PIN. 4KB SRAM 保存数据.

■ 中断

- 嵌套向量中断控制器(NVIC)用于控制240个中断源，  
每个中断源可设置为 4 个优先级
- 支持串行调试(SWD)带2个观察点/4个断点

■ 通用I/O引脚

- 在64-Pin封装下51个I/O
- 5V-tolrence 输入.

■ 内置ISP Bootloader

- 支持通过UART 进行程序升级.

■ 定时器/计数器

- 通用定时器:4X16 bits
- 高级定时器:3X16 bits,1X24bits(SysTick)
- 可编程的计数器:1X16 bits
- 看门狗计数器 : WWDTX1,IWDTX1
- 基础定时器 : 2X16bits
- 低功耗定时器:1X16bits

■ 通信接口

- USART1,2,3,USART4,5标准通讯接口, 支持 ISO7816, LIN, IrDA, 支持最高速率  
4.5Mbit/s, 其中一个支持 MODEM 控制, 串口带低功耗监听功  
能并能唤醒 MCU
- 支持低速时钟的超低功耗LPUART
- 3 路 SPI 接口, 其中 2 路与 I2S 功能复用, 支持最高 24Mbit/S 传输速率.
- 2 路 I2S 接口, 支持 8KHz~48KHz 音频输入
- I2CX2标准通讯接口, 主模式最高支持1Mbps, 从模式最高支持800Kbps
- 1 路独立 CAN2.0B 接口, 支持 1Mbit/S 传输速率
- 1 路独立 USB2.0 全速接口, 支持 12Mbit/S 传输速率
- 1 路 SDIO 接口, 支持 MMC 4.2 规范, 支持 SD CARD 2.0 规范, 支持 SDIO 2.0 规范, 支  
持 CE-ATA 1.1 规范



- 1路QSPI支持133Mbps,高速访问(XIP)

- **ADCX3**

- 12位1Msps采样速率，12位SAR型ADC
- 24通道: 16路的外部引脚,1路内部温度传感器电压,3路的OPA输出,1路的1/3\*VDD,1路的BGR 1.2V,1路的DAC
- 外部参考电压: VDD,GPIO(PB01) PIN
- 内置参考电压: 1.5V,2.048V
- 

- **DMAX2 控制器**

- 支持 12 chs : SPI,I2C,USART,Timer,ADC

- **PWM**

- 支持最多6个独立出口
- 支持最多3个互补式出口

- **DAC(12-bits)X2**

- **运算放大器(OPA)X3**

- **内部温度传感器**

- **电压比较器(VCX2) / 低电压检测器(LVD)**

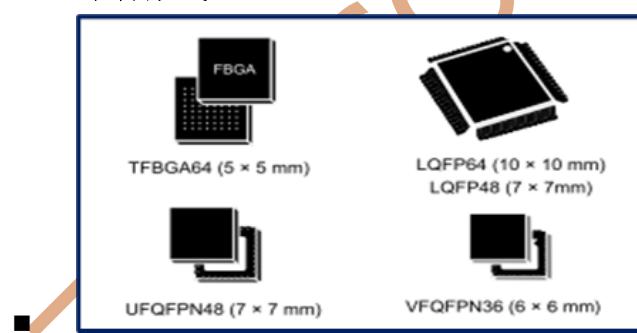
- **硬件CRC-16/32模块,3DES,AES 硬件单元, TRNG真随机数发生器**

- **16字节(128位)的芯片唯一ID (UID)**

- **开发工具**

- SWD全功能的嵌入式调试解决方案

- **封装形式:**





## 【3】 产品功能描述

### 3.1 设备概述

		ZB32F103XXX
引脚数		64
GPIO		51
MCU	内核	Cortex M3
	CPU 频率	96~108 MHz
	Instruction/cycle	96~108 MHZ
Flash		256K Bytes (ExFlash 2Mb~4Mb)
SRAM		64K (+256KB)
PDMA		2(16 ch for SPI,I2C,USART,Timer,ADC)
Timer	基础 (16-bits)	2(TIM10,TIM11)
	通用(16-bits)	4(TIM2,TIM2A,TIM2B,TIM2C)
	高级(16 bits)	3(TIM1,TIM1A,TIM1B)
	编程(16 bits)	1(PCA)
	SysTick(24 bits)	1
	省电(LPTIMER)	1
RTC/SYSTICK/IWDG/WWDG		1/1/1/1
AWK		1
工作电压范围		1.8~3.6V (1*)
工作温度		-40~85 °C
调试功能		SWD
唯一标识符(UID)		16 Byts
通信界面	UART, USART	2,3
	LPUART	1
	SPI,SPI/I2S	1,2(16/20Mhz)
	QSPI	2(1 : Internal Flash)
	I2C	2
	SDIO	1
	USB 2.0(FS)	1
	CAN	2
比较器		2
3DES/AES		1/1 (128 bytes)
TRNG		1
CRC16/32		1
内部温度传感器		1
	内部高速晶振	HIRC : 12/11.059 MHz



时钟	内部低速晶振	SIRC : 32.8/38.4KHz
	外部高速晶振	HXT : 4M~24MHz
	外部低速晶振	LXT : 32.768MHz
	PLL	PLL: 8M~192Mhz
12 Bits A/D	3(16 CH)	
12 BitsDAC	2(2 CH)	
VC	2	
OPAX3	3	
封装	LQFP64	

Table 3-1 ZB32F103 芯片特性与周边配备

注 1\*: 部分引脚容忍 5 伏输入



### 3.2 内部功能框图结构

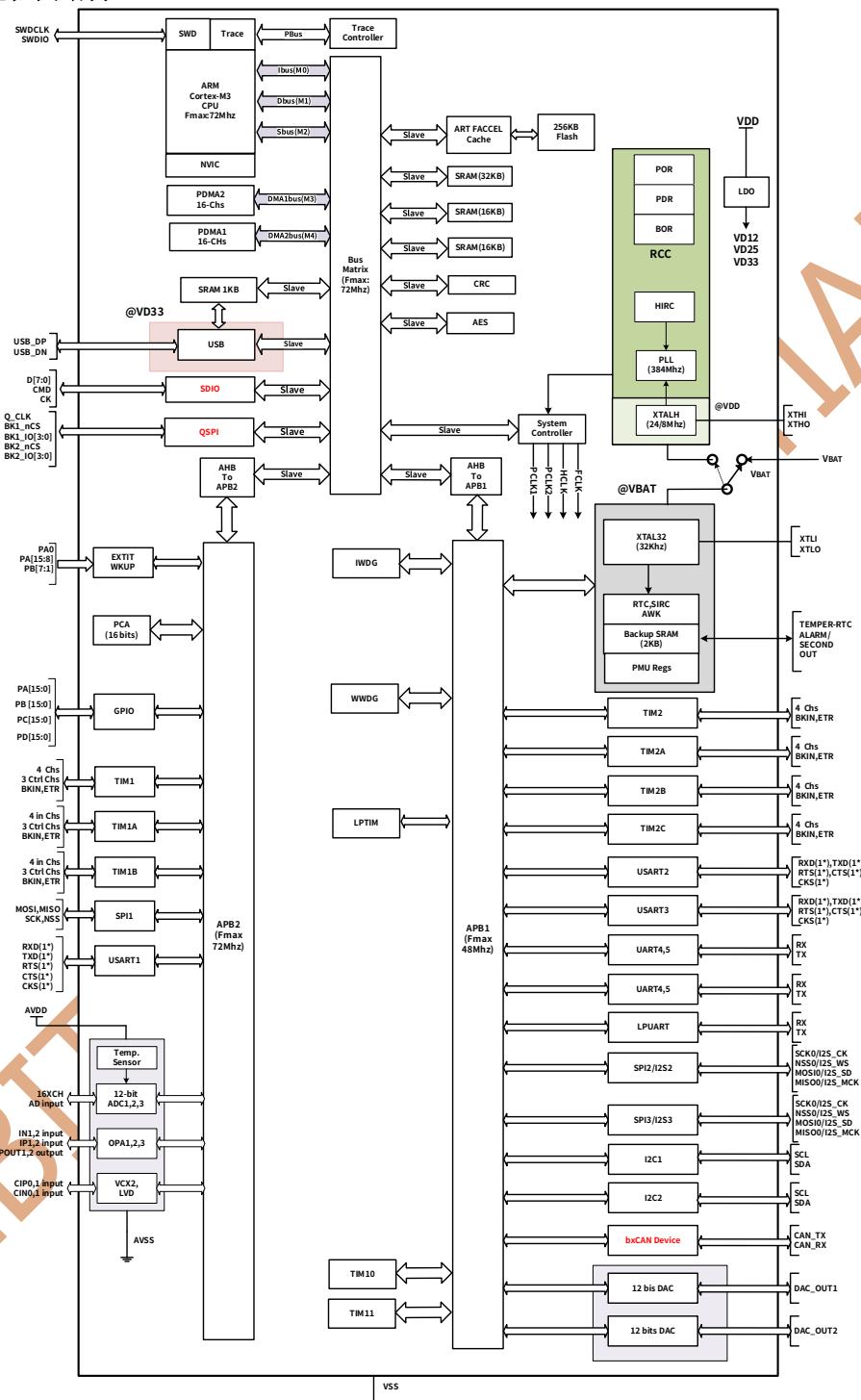


Figure 3-1 内部功能框图结构



### 3.3 32 位 Cortex®-M3 内核

ZB32F103 集成最新一代的嵌入式 ARM Cortex® -M3 32 位处理器内核带有浮点运算器(FPU)与 DSP 指令集。

Cortex-M3 内核的主要特点有

- ARMv7E-M 架构
- 三层管线以及分支预测
- 指令集：
  - ◆ 32-bit 硬件整数乘法, 乘积 32/64-BIT. 32-bit 乘法及乘积累加 (MAC) 是一个周期。
  - ◆ 32-bit 硬件整数除法 (2-12 周期)
  - ◆ 支援饱和运算
  - ◆ DSP 延伸：单一周期 16/32-bit MAC, 单一周期双 16-bit MAC, 8/16-bit SIMD 运算
- 12 个周期的中断延迟
- 整合休眠模式

#### 3.3-1 Flash

闪存存储器(flash), 用于存放程序和数据。内建全集成 Flash 控制器, 无需外部高压输入, 由全内置电路产生高压来编程, 支持 ISP 功能, 最大支持 256/512 KB 字节。

#### 3.3-2 320K Bytes SRAM

根据客户选择不同的功耗模式, RAM 数据都会被保留。自带硬件奇偶校验位, 万一数据被意外破坏, 在数据被读取时, 硬件电路会立刻产生中断, 保证系统的可靠性。它可以以字节(8 位)、半字(16 位)或字(32 位)进行访问。

### 3.4 时钟系统

一个频率为 4M~24MHz 的外部高速晶振 HXT。

一个频率为 32.768KHz 的外部低速晶振 LXT。

一个频率为 12/11.059 MHz 的内部高速晶振 HIRC。

一个频率为 32.768KHz/38.4KHz 的内部低速时钟 SIRC。

一个频率为 192/288MHz 输出的 PLL。



### 3.5 工作模式

ZB32F103 支持 5 种工作模式：

1. 运行模式 (**Active**) :

CPU 运行，周边功能模块运行。

2. 睡眠模式(**Sleep Mode**) :

在睡眠模式，只有 CPU 停止，所有外设处于工作状态并可在发生中断/事件时唤醒 CPU。

3. 停机模式(**Stop Mode**) :

在保持 SRAM 和寄存器内容不丢失的情况下，停机模式可以达到最低的电能消耗。在停机模式下，停止所有内部 1.8V 部分的供电，PLL、HSI 的 RC 振荡器和 HSE 晶体振荡器被关闭，调压器可以被置于普通模式或低功耗模式。可以通过任一配置成 EXTI 的信号把微控制器从停机模式中唤醒，EXTI 信号可以是 16 个外部 I/O 口之一、PVD 的输出、RTC 闹钟或 USB 的唤醒信号。

4. 待机模式(**Standby mode**) :

在待机模式下可以达到最低的电能消耗。内部的电压调压器被关闭，因此所有内部 1.8V 部分的供电被切断；PLL、HSI 的 RC 振荡器和 HSE 晶体振荡器也被关闭；进入待机模式后，SRAM 和寄存器的内容将消失，但后备寄存器的内容仍然保留，待机电路仍工作。

从待机模式退出的条件是：NRST 上的外部复位信号、IWDG 复位、WKUP 引脚上的一个上升边沿或 RTC 的闹钟到时。在进入停机或待机模式时，RTC、IWDG 和对应的时钟不会被停止。

5. 关机模式(**power down mode**) :

所有供电被切断RTC、IWDG 和对应的时钟会被停止,僅保持 I/O 状态(关机模式前設定的狀態), 关机模式退出的条件是NRST 上的外部复位信号,WKUP-A0 引脚上的一个上升边沿.启动时间较长.



### 3.6 中断控制器(NVIC)

Cortex®-M3 处理器内置了嵌套向量中断控制器(NVIC)，NVIC 支持多达 240 个中断，每个中断具有多达 256 个可以动态更改的优先级，处理器和 NVIC 可以进入非常低功耗的睡眠模式，让唤醒控制器(WIC) 识别中断并确定中断的优先级。此外，处理器支持电平和脉冲中断。

详情请参考 “ARM® Cortex®-M3 Technical Reference Manual” 与 “ARM® v7-M Architecture Reference Manual”。

### 3.7 复位控制器

本产品具有 9 个复位信号来源，每个复位信号可以让 CPU 重新运行，绝大多数寄存器会被重新复位，程序计数器 PC 会被复位指向复位地址(0x0000 0000)。

编号	中断源
1	上电/掉电复位
2	外部 Reset Pin 复位
3	IWDG 复位
4	WWDG 复位
5	系统软件复位
6	欠电压(LVD)复位
7	LOCKUP 复位
8	寄存器 RCC_APB2RSTR 复位
9	寄存器 RCC_APB1RSTR 复位

### 3.8 通用 IO 端口(GPIO)

最多可提供 51 个 GPIO 端口，其中部分 GPIO 与模拟端口复用。每个端口由独立的控制寄存器位来控制。支持边沿触发中断和电平触发中断，可从各种功耗模式下把 MCU 唤醒到工作模式。支持 Push-Pull CMOS 推挽输出、Open-Drain 开漏输出。内置上拉电阻、下拉电阻，带有施密特触发器输入滤波功能。输出驱动能力可配置，最大支持 12mA 的电流驱动能力。51 个通用 IO 可支持外部异步中断



### 3.9 定时器和看门狗

ZB32F103 产品包含 3 个高级控制定时器、4 个通用定时器、1 个可编程计数器阵列、2 个基础定时器、1 个低功耗基础定时器、1 个系统窗口看门狗定时器、1 个独立看门狗定时器和 1 个系统嘀嗒(SysTick)定时器。

下表比较了高级控制定时器、通用定时器和基础定时器的功能：

定时器类型	名称	计数器位宽	预分频系数	计数方向	PWM输出	捕捉 / 比较通道	互补输出
高级	TIM1	16 位	1/2/4/8/16/64/256/1024	递增	4	4	3 对
				递减			
				递增/递减			
通用	TIM1A	16 位	1/2/4/8/16/64/256/1024	递增	4	4	3 对
				递减			
				递增 / 递减			
基础	TIM1B	16 位	1/2/4/8/16/64/256/1024	递增	4	4	3 对
				递减			
				递增 / 递减			
通用	TIM2	16 位	1/2/4/8/16/64/256/1024	递增	4	4	0 对
				递减			
				递增 / 递减			
基础	TIM2A	16 位	1/2/4/8/16/64/256/1024	递增	4	4	0 对
				递减			
				递增 / 递减			
基础	TIM2B	16 位	1/2/4/8/16/64/256/1024	递增	4	4	0 对
				递减			
				递增 / 递减			
基础	TIM2C	16 位	1/2/4/8/16/64/256/1024	递增	4	4	0 对
				递减			
				递增 / 递减			
可编程计数器阵列	PCA	16 位	2/4/8/16/32	递增 / 递减	5	5	无
低功耗	LPTIM	16 位	无	递增	无	无	无
基础	TIM10	16/32 位	1/2/4/8/16/32/64/128	递增	无	无	无
	TIM11	16/32 位	1/2/4/8/16/32/64/128	递增	无	无	无



Table 3-2 定时器菜单

ZBT CONFIDENTIAL



### 3.9-1 高级控制定时器(TIM1,8,12)

3 个高级控制定时器(TIM1,8,12),每个定时器可以被看成是分配到 6 个通道的三相 PWM 发生器, 它具有带死区插入的互补 PWM 输出, 还可以被当成完整的通用定时器。四个独立的通道可以用于:

- 输入捕获
- 输出比较
- 产生 PWM(边缘或中心对齐模式)
- 单脉冲输出配置为 16 位标准定时器时, 它与 TIMx 定时器具有相同的功能。配置为 16 位 PWM 发生器时, 它具有全调制能力(0~100%)。

在调试模式下, 计数器可以被冻结, 同时 PWM 输出被禁止, 从而切断由这些输出所控制的开关。很多功能都与通用 TIM 定时器相同, 内部结构也相同, 因此高级控制定时器可以通过定时器链接功能与其他 TIM 定时器协同操作, 提供同步或事件链接功能。

高级控制定时器 基本的功能及特性如表所示:

PWM 波形产生模式	锯齿波、三角波
基本功能	<ul style="list-style-type: none"><li>1. 递加、递减计数方向</li><li>2. 软件同步</li><li>3. 硬件同步</li><li>4. 缓存功能</li><li>5. 正交编码计数</li><li>6. 通用 PWM 输出</li><li>7. 保护机制</li><li>8. AOS 关联动作</li></ul>
中断类型	计数比较匹配中断 计数周期匹配中断

### 3.9-2 通用定时器(TIM2,3,4,5)

通用定时器(TIM2,3,4,5),每个计数器都有一个 16 位的自动加载递加/递减计数器、一个 16 位的预分频器和 4 个独立的信道, 每个信道都可用于输入捕获、输出比较、PWM 和单脉冲模式输出, 它们还能通过定时器链

接功能与高级控制定时器共同工作, 提供同步或事件链接功能。在调试模式下, 计数器可以被冻结。

任一标准定时器都能用于产生 PWM 输出。



### 3.9-3 可编程计数器阵列(PCA)

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可以用作一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个通道都可以进行独立编程，提供输入捕捉/输出比较或脉冲宽度调制。

### 3.9-4 低功耗定时器(LPTIM)

低功耗定时器为 1 个异步的 16 位可选定时器。在系统时钟关闭后仍然可以通过内部低速 SIRC 或者外部低速晶体振荡器计时/计数。通过中断可以在低功耗模式下唤醒系统

### 3.9-5 基础定时器(TIM10/TIM11)

基础定时器包含 2 个 16/32 位可选定时器 TIM10/TIM11。TIM10/TIM11 功能完全相同，都是同步定时/计数器，可以选择工作在重载模式和非重载模式。TIM10/TIM11 可以对外部脉冲进行计数或者实现系统定时。

### 3.9-6 独立看门狗(IWDG)

独立的看门狗是一个 20 位递减计数器。它由内部独立的 SIRC 提供时钟；由于内部 SIRC 独立于主时钟，因此它可在停机和待机模式下工作。它既可用作看门狗，以便在发生问题时复位器件，也可用作自由运行的定时器，以便为应用程序提供超时管理。通过选项字节，可对其进行硬件或软件配置。在调试模式下，计数器可以被冻结。

### 3.9-7 系统窗口看门狗(WWDG)

系统窗口看门狗基于一个 8 位递减计数器，支持 20 位的预分频，它由 APB 时钟(PCLK)提供动作时钟。它可以作为看门狗，以便在系统发生问题时复位器件，同时具有早期警告中断功能，并且计数器可以在调试模式下被冻结。

### 3.9-8 SysTick 定时器(SYST)

此定时器专用于实时操作系统，但也可用作标准递减计数器。它具有以下特性：

- (1) 24 位递减计数器



- (2) 自动重载功能
- (3) 当计数器计为 0 时，产生可屏蔽系统中断
- (4) 可编程时钟源(HCLK 或 HCLK/4)

ZBT CONFIDENTIAL

---

---



### 3.10 实时时钟(RTC)

1. 支持 RTC 计数(秒、分、小时)及万年历功能(日、月、年)
2. 支持闹铃寄存器(秒、分、小时、日、月、年)
3. RTC 可以从 Sleep 模式唤醒系统

### 3.11 通用异步收发器(USART1~3/UART4~5)

2 路通用异步收发器(Universal Asynchronous Receiver/Transmitter)

3 路通用同步/异步收发器(Universal Synchronous /Asynchronous Receiver/Transmitter)

### 3.12 通用异步收发器(LPUART)

1 路低功耗模式下可以工作的异步发 器 (Low Power Universal Asynchronous Receiver/Transmitter)

### 3.13 串行外设接口 (SPI) / 内部集成音频接口 (I2S)

2 路串行外设接口(Serial Peripheral Interface), 支持主从模式。

据传输最大速度可达 20/16Mbps。

一个标准 I2S 接口 (与 SPI 复用) 支持四种不同的音频标准, 能以主或从半双工通信模式工作。它可配置为 16、24、32 位传输, 有 16 位或 32 位数据分辨率, 由专用信号同步。可由 8 位可编程线性预分频设置 8 kHz 至 192 kHz 的音频采样频率。当工作于主模式时, 它可为外部音频组件输出采样频率 256 倍的时钟。

### 3.14 I2C 接口(I2C)

2 路 I2C 接口, 支持主从模式。采用串行同步时钟, 可实现设备之间以不同的速率传输数据, 串行 8 位双向数据传输最大速度可达 1Mbps。

### 3.15 控制器局域网 CAN

支持 控制器局域网(CAN)协议。

### 3.16 自唤醒定时器(AWK)

AWK 是用于当 MCU 进入低功耗模式时提供一个内部的唤醒时间基准。该时间基准的时钟是由内部的低速 RC 振荡器时钟(SIRC)或者通过预分频的 HXT 晶振时钟来提供的

### 3.17 时钟校准/监测模块(CLKTRIM)

内建时钟校准电路, 可以通过外部精准的晶振时钟来校准内部 RC 时钟, 亦可使用内部 RC 时钟去检测外部晶振时钟是否工作正常。

### 3.18 唯一 ID 号(UID)

每颗芯片出厂时都具备唯一的 16 字节设备标识号, 包括 wafer lot 信息, 以及芯片坐标信息等。



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

---

---

ID 地址 0x180000F0-0x180000FF。

ZBIT CONFIDENTIAL

---

---



### 3.19 循环冗余校验计算单元(CRC16/32)

符合 ISO/IEC13239 中给出的多项式：

CRC16 :  $F(x) = X^{16} + X^{12} + X^5 + 1$ 。

CRC32 :  $F(x) = X^{32} + X^{26} + X^{23} X^{22} + X^{16} + X^{12} X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

### 3.20 模拟/数字转换器(ADC)

单调不失码的 12 位逐次逼近型模数转换器，在 24MHz ADC 时钟下工作时，采样率达到 1Msps。

参考电压可选择电源电压。15 个外部通道，可以实现单次，扫描，循环转换。在扫描/循环模式下，自动进行在选定的一组模拟输入上的转换。

1. 输入电压范围：0 to VDD
2. 转换周期：16/20 clock cycles
3. 可以从外部端子，内部 TIM1、TIM2、TIM10/TIM11、VC 等模块来触发 ADC 采样
4. 采样完成(EOC)中断
5. 4 种参考源：AVDD 电压、EXVREF(PB01)引脚、内置 VREF(2.048V,1.8V) 参考电压

### 3.21 低电压检测器(LVD)

对芯片电源电压或芯片引脚电压进行检测。8 档电压监测值(2.5-4.4V)。可根据上升/下降边沿产生异步中断或复位。具有硬件迟滞电路和可配置的软件防抖功能。

### 3.22 电压比较器(VC)

芯片引脚电压监测/比较电路。3 个可配置的正/负外部输入通道；1 个内部 BGR 2.5V 参考电压。

VC 输出可供定时器 TIM1、TIM10/TIM11、LPTimer 与可编程计数阵列 PCA 捕获、门控、外部计数使用。可根据上升/下降边沿产生异步中断，从低功耗模式下唤醒 MCU。可配置软件防抖。

### 3.23 运算放大器(OPA)

OPA 模块可以灵活配置，适用于简易滤波器和 Buffer 应用。

内部的三个运放可以配置为反相、同相具有不同增益的组合运放，也可以使用外部电阻进行级联。

### 3.24 嵌入式调试系统

嵌入式调试解决方案，提供全功能的实时调试器，配合标准成熟的 Keil/IAR 等调试开发软件。支持 4 个硬断点以及多个软断点。

### 3.25 加密型嵌入式 Debug 支持(DBG)

加密型嵌入式调试解决方案，提供全功能的实时调试器，详见用户手册相关章节。



### 3.26 高级加密标准模块 (AES)

AES (The Advanced Encryption Standard) 是美国联邦政府采用的一种区块加密标准。AES 的分组长度固定为 128 Bit，而密钥长度支持 128 Bit。

### 3.27 真随机数发生器 (TRNG)

TRNG 是一个真随机数发生器，用来产生真随机数。

### 3.28 编程模式

支持两种编程模式：在线编程、离线编程。

支持两种编程协议：ISP 协议、SWD 协议。

ISP 协议编程接口：PC12、PD2 或 PD13、PC14。

SWD 协议编程接口：PA13(SWDIO)、PA14(SWCLK) 或 PA11(SWDIO)、PA12(SWCLK)

当复位时 **BOOT0** (PD03) 管脚为高电平，芯片工作于 SRAM，可通过 SWD 协议对 FLASH 进行编程。

当复位时 **BOOT0** 管脚为低电平，芯片工作于用户模式，执行的程序如下

**Step1**：ISP 编程模式，→ 可通过 ISP 协议对 FLASH 进行编程。

**Step2**：芯片执行 main FLASH 内的程序代码，可通过 SWD 协议对 FLASH 进行编程。

### 3.29 直接存储器存取(PDMA)

PDMA 控制器可以从一个地址到另一个地址传输数据，无需 CPU 介入。

### 3.30 内嵌的自举程序(Bootloader)

内嵌的自举程序存放在 ROM 区域。该程序可以通过 UART5 对 Flash 进行重新编程。



## 【4】 引脚配置及功能

### 4.1 引脚定义

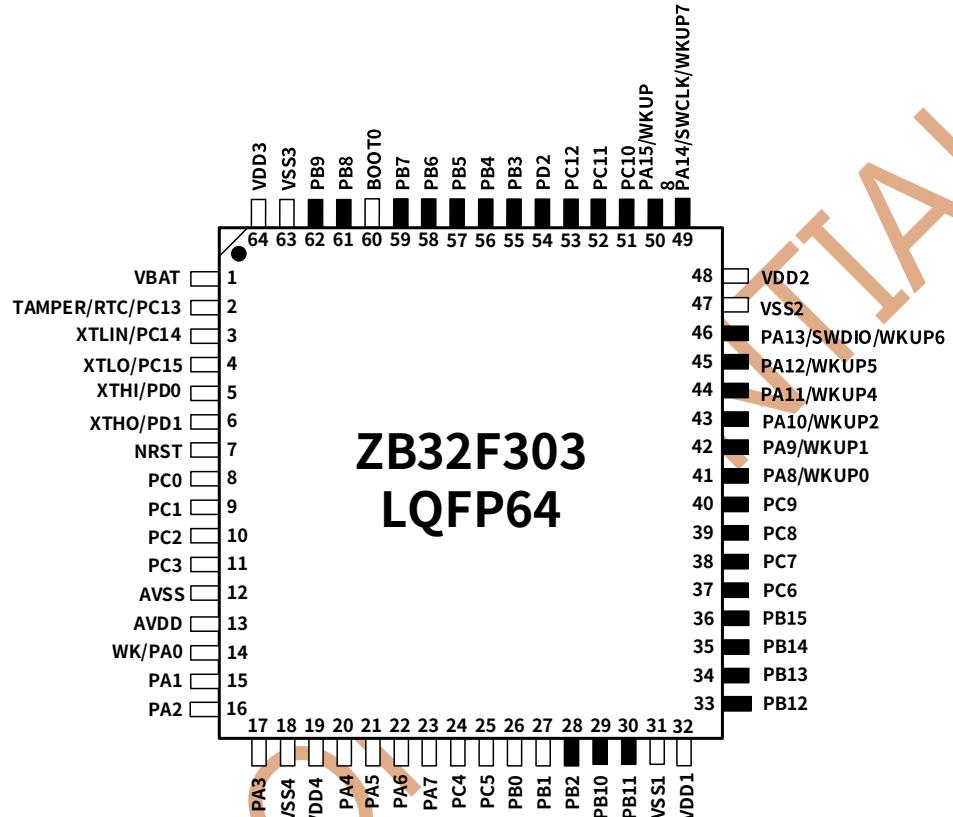


Figure 4-1 ZB32F103 LQFP64

Note : ■ → Signal 5V Tolerant.



#### 4.2 引脚功能配置

LQFP 64	LQFP 48	Pin name	Type	Main function (after reset)	CONFIG/WKUP	Default
1	1	VBAT	S	VBAT		
2	2	PC13-TAMPER-RTC	I/O	PC13		
3	3	PC14-OSC32_IN	I/O	PC14		
4	4	PC15-OSC32_OUT	I/O	PC15		
5	5	OSC_IN-PD0	I	OSC_IN		
6	6	OSC_OUT-PD1	O	OSC_OUT		
7	7	NRST	I/O	NRST		
8		PC0	I/O	PC0		ADC123_IN10 /OP3_INP
9		PC1	I/O	PC1		ADC123_IN11 /OP3_INN
10		PC2	I/O	PC2		ADC123_IN12 /OP2_INP
11		PC3	I/O	PC3		ADC123_IN13 /OP2_INN
12	8	VSSA	S	VSSA		
		EXVREF-	S			ADCEXVREF-
		EXCREF+	S			ADCEXVREF+
13	9	VDDA	S	VDDA		
14	10	PA0	I/O	PA0	PA0-WKUP	ADC123_IN0/VC0_N0 /VC0_P0/OP1_INP
15	11	PA1	I/O	PA1		ADC123_IN1 /VC0_N1 /VC0_P1/OP1_INN
16	12	PA2	I/O	PA2		ADC123_IN2 /VC0_N2 /VC0_P2



LQFP 64	LQFP 48	Pin name	Type	Main function (after reset)	CONFIG/WKUP	Default
17	13	PA3	I/O	PA3		ADC123_IN3 /VC0_N3 /VC0_P3
18		VSS_4	S	VSS_4		
19		VDD_4	S	VDD_4		
20	14	PA4	I/O	PA4		ADC12_IN4 /VC0_N4 VC0_P4
21	15	PA5	I/O	PA5		ADC12_IN5 /VC0_N5 VC0_P5
22	16	PA6	I/O	PA6		ADC12_IN6 /VC0_N6 /VC0_P6
23	17	PA7	I/O	PA7		ADC12_IN7/VC0_N7 VC0_P7
24		PC4	I/O	PC4		ADC12_IN14 /VC1_N0 /VC1_P0
25		PC5	I/O	PC5		ADC12_IN15 /VC1_N1 /VC1_P1
26	18	PB0	I/O	PB0		ADC12_IN8 /VC1_N2 /VC1_P2
27	19	PB1	I/O	PB1	PB1-WKUP	ADC12_IN9 /VC1_N3 /VC1_P3
28	20	PB2	I/O (FT)	PB2/BOOT1	PB2-WKUP	VC1_N4 /VC1_P4
29	21	PB10	I/O (FT)	PB10		VC1_N5 /VC1_P5
30	22	PB11	I/O (FT)	PB11		VC1_N6



LQFP 64	LQFP 48	Pin name	Type	Main function (after reset)	CONFIG/WKUP	Default
						/VC1_P6
31	23	VSS_1	S	VSS_1		
32	24	VDD_1	S	VDD_1		
33	25	PB12	I/O (FT)	PB12		VC1_N7 /VC1_P7
34	26	PB13	I/O (FT)	PB13		
35	27	PB14	I/O (FT)	PB14		
36	28	PB15	I/O (FT)	PB15		
37		PC6	I/O (FT)	PC6		
38		PC7	I/O (FT)	PC7		
39		PC8	I/O (FT)	PC8		
40		PC9	I/O (FT)	PC9		
41	29	PA8	I/O (FT)	PA8	PA8 -WKUP	
42	30	PA9	I/O (FT)	PA9	PA9 -WKUP	
43	31	PA10	I/O (FT)	PA10	PA10 -WKUP	
44	32	PA11	I/O (FT)	PA11	SWDIO/ PA11 -WKUP	
45	33	PA12	I/O (FT)	PA12	SWCLK/ PA12 -WKUP	
46	34	PA13	I/O (FT)	SWDIO	SWDIO/ JTMS/ PA13-WKUP	
47	35	VSS_2	S	VSS_2		
48	36	VDD_2	S	VDD_2		
49	37	PA14	I/O (FT)	SWCLK	SWCLK/ JTCK/ PA14-WKUP	
50	38	PA15	I/O (FT)	PA15	JTDI / PA15- WKUP	
51		PC10	I/O (FT)	PC10		
52		PC11	I/O (FT)	PC11		
53		PC12	I/O (FT)	PC12		



LQFP 64	LQFP 48	Pin name	Type	Main function (after reset)	CONFIG/WKUP	Default
54		PD2	I/O (FT)	PD2		
55	39	PB3-WKUP	I/O (FT)	PB3	JTDO/ PB3-WKUP	
56	40	PB4	I/O (FT)	PB4	JNTRST/ PB4-WKUP	
57	41	PB5	I/O (FT)	PB5	PB5-WKUP	
58	42	PB6	I/O (FT)	PB6	PB6-WKUP	
59	43	PB7	I/O (FT)	PB7	PB7-WKUP	
60	44	BOOT0	I	BOOT0		
61	45	PB8	I/O (FT)	PB8		
62	46	PB9	I/O (FT)	PB9		
63	47	VSS_3	S	VSS_3		
64	48	VDD_3	S	VDD_3		

Table 4-1 引脚定義说明表

Note : (1) I = 输入, O = 输出, S = 电源。

(2) FT = 5V 容忍。



#### 4.3 选择芯片引脚复用说明

ZB32F103 选择芯片引脚复用可區分兩種模式:

- (1) ST 相容模式 : AFMD=0
- (2) PIN 多選模式:AFMD=1

##### 4.3-1 STMD 引脚复用(AFXMLD=0)

LQFP 64	LQFP 48	Main function (after reset)	Default AFIO_MAPR=0	Remap AFIO_MAPR Define
1	1	VBAT		
2	2	PC13	TAMPER-RTC	
3	3	PC14	OSC32_IN	
4	4	PC15	OSC32_OUT	
5	5	OSC_IN		PD0
6	6	OSC_OUT		PD1
7	7	NRST		
8		PC0	ADC123_IN10/ OP3_INP	
9		PC1	ADC123_IN11/ OP3_INN	
10		PC2	ADC123_IN12/ OP2_INP	
11		PC3	ADC123_IN13/ OP2_INN	
12	8	VSSA		
13	9	VDDA		
14	10	PA0	WKUP/USART2_CTS/ADC123_IN0/TIM2_CH1_ETR/TIM2C_CH1/TIM1A_ETR	OP1_INP VC0_N0 VC0_P0
15	11	PA1	USART2_RTS/ADC123_IN1/TIM2C_CH2/TIM2_C_H2	OP1_INN VC0_N1 VC0_P1
16	12	PA2	USART2_TX/TIM2C_CH3/ADC123_IN2/TIM2_CH3	VC0_N2 VC0_P2



LQFP 64	LQFP 48	Main function (after reset)	Default AFIO_MAPR=0	Remap AFIO_MAPR Define
17	13	PA3	USART2_RX/TIM2C_CH4/ADC123_IN3/TIM2_CH4	VC0_N3 VC0_P3
18		V <sub>SS_4</sub>		
19		V <sub>DD_4</sub>		
20	14	PA4	SPI1 NSS/USART2 CK/DAC_OUT1/ADC12_IN4	OP1O VC0_N4 VC0_P4 <b>LPT_TOG</b>
21	15	PA5	SPI1_SCK/DAC_OUT2/ADC12_IN5	VC0_N5 VC0_P5 <b>LPT_TOGN</b>
22	16	PA6	SPI1_MISO/TIM1A_BKIN/ADC12_IN6/TIM2A_CH1	TIM1_BKIN VC0_N6 VC0_P6 <b>TXD_LP</b>
23	17	PA7	SPI1_MOSI/TIM1A_CH1N/ADC12_IN7/TIM2A_CH2	TIM1_CH1N VC0_N7 VC0_P7 <b>RXD_LP</b>
24		PC4	ADC12_IN14	OP3O VC1_N0 VC1_P0 <b>LPT_EXT</b> <b>PCA_ECI</b>
25		PC5	ADC12_IN15	OP2O VC1_N1 VC1_P1 <b>LPT_GATE</b> <b>PCA_CH0</b>
26	18	PB0	ADC12_IN8/TIM2A_CH3/TIM1A_CH2N	TIM1_CH2N VC1_N2 VC1_P2



LQFP 64	LQFP 48	Main function (after reset)	Default AFIO_MAPR=0	Remap AFIO_MAPR Define
				PCA_CH1 TXD_LP
27	19	PB1	ADC12_IN9/TIM2A_CH4/TIM1A_CH3N	TIM1_CH3N VC1_N3 VC1_P3 PCA_CH2 RXD_LP
28	20	PB2/BOOT1	SCAN_CLK/SMB_BIST_CLK/FB_SCLK/ANAT_COD E0/USBPHYT_PONRST	VC1_N4 VC1_P4 PCA_CH3
29	21	PB10	I2C2_SCL/USART3_TX/SCAN_IN5/ANAT_CTLI2/A NAT_CTL02/USBPHYT_BCKPD	TIM2_CH3 VC1_N5 VC1_P5 PCA_CH4
30	22	PB11	I2C2_SDA/USART3_RX/SCAN_OUT0/ANAT_CTLI3 /ANAT_CTL03/USBPHYT_RCV	TIM2_CH4 VC1_N6 VC1_P6
31	23	Vss_1		
32	24	Vdd_1		
33	25	PB12	SPI2 NSS/I2S2_WS/I2C2_SMBA/USART3_CK/TIM 1_BKIN/SCAN_OUT1/ANAT_CTLI4/ANAT_CTL04/ USBPHYT_VOP	VC1_N7 VC1_P7
34	26	PB13	SPI2_SCK/I2S2_CK/USART3_CTS/TIM1_CH1N/SC AN_OUT2/ANAT_CTLI5/ANAT_CTL05/USBPHYT_ VON	
35	27	PB14	SPI2_MISO/TIM1_CH2N/USART3_RTS/SCAN_OU T3/ANAT_CTLI6/ANAT_CTL06/USBPHYT_CK48M	



LQFP 64	LQFP 48	Main function (after reset)	Default AFIO_MAPR=0	Remap AFIO_MAPR Define
36	28	PB15	SPI2_MOSI/I2S2_SD/TIM1_CH3N/SCAN_OUT4/A NAT_CTLI7/ANAT_CTL07/USBPHYT_CLK_RDY	
37		PC6	I2S2_MCK/TIM1A_CH1/SDIO_D6/QSPI_BK2_IO2/ ANAT_CTLI14/ANAT_CTL014	TIM2A_CH1
38		PC7	I2S3_MCK/TIM1A_CH2/SDIO_D7/QSPI_BK2_IO3/ ANAT_CTLI15/ANAT_CTL015	TIM2A_CH2
39		PC8	TIM1A_CH3/SDIO_D0/QSPI_BK1_IO0/ANAT_CTLI 16/ANAT_CTL016	TIM2A_CH3
40		PC9	TIM1A_CH4/SDIO_D1/QSPI_BK1_IO1/ANAT_CTLI 17/ANAT_CTL017	TIM2A_CH4 VC1_O
41	29	PA8	USART1_CK/TIM1_CH1/MCO/SCAN_OUT5/ANA T_CTLI8/ANAT_CTL08	VC1_O
42	30	PA9	USART1_TX/TIM1_CH2/ANAT_CTLI9/ANAT_CTL 9	VC1_O
43	31	PA10	USART1_RX/TIM1_CH3/ANAT_CTLI10/ANAT_CTL O10	VC0_O/
44	32	PA11	USART1_CTS/USBDM/CAN_RX/TIM1_CH4/ANAT _CTLI11/ANAT_CTL011	VC0_O/ SWDIO
45	33	PA12	USART1_RTS/USBDP/CAN_TX/TIM1_ETR/ANAT _CTLI12/ANAT_CTL012	VC0_O/ SWCLK
46	34	SWDIO	TEST_TDI	PA13
47	35	Vss_2		
48	36	Vdd_2		
49	37	SWCLK	TEST_TCK	PA14
50	38	PA15	SPI3_NSS/I2S3_WS/QSPI_BK2_nCS/ANAT_CTLI13 /ANAT_CTL013	TIM2_CH1_ET R/SPI1_NSS/ TIM1B_ETR



LQFP 64	LQFP 48	Main function (after reset)	Default AFIO_MAPR=0	Remap AFIO_MAPR Define
51		PC10	UART4_TX/SDIO_D2/QSPI_BK1_IO2/ANAT_CTLI18/ANAT_CTLI18	USART3_TX/ <b>TIM1B_CH1</b>
52		PC11	UART4_RX/SDIO_D3/QSPI_BK1_IO3/ANAT_CTLI19/ANAT_CTLI19	USART3_RX/ <b>TIM1B_CH2</b>
53		PC12	UART5_TX/SDIO_CK/QSPI_CLK/ANAT_CTLI20/A/NAT_CTLI20	USART3_CK/ <b>TIM1B_CH3</b>
54		PD2	TIM2A_ETR/UART5_RX/SDIO_CMD/QSPI_BK1_nCS	<b>TIM1B_CH4</b>
55	39	PB3	SPI3_SCK/I2S3_CK/SCAN_EN/SMB_RST_L/FB_SE/ANAT_CODE1/USBPHYT_OUTCLKSEL	TRACESWO/TI_M2_CH2/SPI1_SCK/ <b>TIM1B_BKIN</b>
56	40	PB4	SPI3_MISO/SCAN_RST/SMB_TEST_H/FB_SIO_I/A/NAT_CODE2/USBPHYT_SE	TIM2A_CH1/SPI1_MISO
57	41	PB5	I2C1_SMBAI/SPI3_MOSI/I2S3_SD/SCAN_IN0/SMB_TEST_DONE/FB_SIO_O/ANAT_CODE3/USBPHYT_VI	TIM2A_CH2/SPI1_MOSI
58	42	PB6	I2C1_SCL/TIM2B_CH1/SCAN_IN1/SMB_FAIL_H/A/NAT_CODE4/USBPHYT_OE	USART1_TX/ <b>TIM1B_CH1N</b>
59	43	PB7	I2C1_SDA/TIM2B_CH2/SCAN_IN2/ANAT_CODE5/USBPHYT_PD	USART1_RX/ <b>TIM1B_CH2N</b>
60	44	BOOT0		
61	45	PB8	TIM2B_CH3/SDIO_D4/QSPI_BK2_IO0/SCAN_IN3/ANAT_CTLI0/ANAT_CTLI0/USBPHYT_DPPU	I2C1_SCL/CAN_RX// <b>TIM1B_CH3N</b>



LQFP 64	LQFP 48	Main function (after reset)	Default AFIO_MAPR=0	Remap AFIO_MAPR Define
62	46	PB9	TIM2B_CH4/SDIO_D5/QSPI_BK2_IO1/SCAN_IN4/ ANAT_CTLI1//ANAT_CTL01/USBPHYT_DMPU	I2C1_SDA/CA N_TX/
63	47	Vss_3		
64	48	Vdd_3		

Table 4-2 引脚 ST MD 功能说明表



#### 4.3-2 PINMD 引脚复用(AFMD=1)

LQFP		GPIO_AFR[i+3:i]									
64	48	0	1	2	3	4	5	6	7	8	9
1	1	VBAT									
2	2	PC13									
3	3	PC14	OSC32_IN								
4	4	PC15	OSC32_OUT								
5	5	OSC_IN	PDO								
6	6	OSC_OUT	PD1								
7	7	NRST									
8		PC0									
9		PC1									EVENTOUT
10		PC2									EVENTOUT
11		PC3									EVENTOUT
12	8	VSSA									EVENTOUT
13	9	VDDA									
14	10	PA0	TIM1A_ETR	LPT_EXT	TIM2_CH1_ETR		I2S3_WS/SPI3_NSS	SPI1_NSS	USART2_CTS		EVENTOUT
15	11	PA1		LPT_GATE	TIM2_CH2		I2S3_CK/SPI3_SCK	SPI1_SCK	USART2 RTS		EVENTOUT
16	12	PA2	TIM1A_CH3	LPT_TOG	TIM2_CH3	TXD_LP	SPI3_MISO	SPI1_MISO	USART2_TX	VC1_O	EVENTOUT
17	13	PA3	TIM1A_CH4	LPT_TOGN	TIM2_CH4	RXD_LP	I2S3_SD/SPI3_MOSI	SPI1_MOSI	USART2_RX	VC0_O	EVENTOUT
18		VSS_4									
19		VDD_4									
20	14	PA4	TIM1A_BK1N	TIM1A_CH1			SPI1_NSS		USART2_CK	OP1O	EVENTOUT
21	15	PA5	TIM1A_CH1N	TIM1A_CH2			I2C2_SCL	SPI1_SCK			EVENTOUT
22	16	PA6	TIM1_BKIN	TIM1A_BK1N	TIM2A_CH1	I2C2_SDA	SPI1_MISO				EVENTOUT
23	17	PA7	TIM1_CH1N	TIM1A_CH1N	TIM2A_CH2		SPI1_MOSI				EVENTOUT
24		PC4		LPT_EXT			QSPI_BK1_nCS		OP3O		
25		PC5		LPT_GATE			QSPI_CLK		OP2O		
26	18	PB0	TIM1_CH2N	TIM1A_CH2N	TIM2A_CH3	TXD_LP		QSPI_BK1_I00	UART4_TX		EVENTOUT
27	19	PB1	TIM1_CH3N	TIM1A_CH3N	TIM2A_CH4	RXD_LP		QSPI_BK1_I01	UART4_RX		EVENTOUT
28	20	PB2/BOOT1								USBPHYT_PONRST	EVENTOUT
29	21	PB10	TIM1A_CH	LPT_TOG	TIM2_CH3	I2C2_SCL	MCO_O	QSPI_BK2_I	USART3_TX	USBPHYT_	EVENTOUT



LQFP		GPIO_AFR[i+3:i]									
64	48	0	1	2	3	4	5	6	7	8	9
			1					O2		BCKPD	
30	22	PB11	TIM1A_CH2	LPT_TOGN	TIM2_CH4	I2C2_SDA		QSPI_BK2_I_O3	USART3_RX	USBPHYT_RCV	EVENTOUT
31	23	VSS_1									EVENTOUT
32	24	VDD_1									
33	25	PB12	TIM1_BKIN			I2C2_SMBA	I2S2_WS/SPI2_NSS		USART3_CK	USBPHYT_VOP	EVENTOUT
34	26	PB13	TIM1_CH1_N				I2S2_CK/SP_I2_SCK	QSPI_BK2_nCS	USART3_CS	USBPHYT_VON	EVENTOUT
35	27	PB14	TIM1_CH2_N				SPI2_MISO	QSPI_BK2_I_O0	USART3_RT	USBPHYT_CLK48M	EVENTOUT
36	28	PB15	TIM1_CH3_N				I2S2_SD/SP_I2_MOSI	QSPI_BK2_I_O1		USBPHYT_CLK_RDY	EVENTOUT
37		PC6	TIM1A_CH1	PCA_ECI	TIM2A_CH1	SDIO_D6	I2S2_MCK	QSPI_BK2_I_O2	UART4_TX		EVENTOUT
38		PC7	TIM1A_CH2	PCA_CH0	TIM2A_CH2	SDIO_D7	I2S3_MCK	QSPI_BK2_I_O3	UART4_RX		EVENTOUT
39		PC8	TIM1A_CH3	PCA_CH1	TIM2A_CH3	SDIO_D0		QSPI_BK1_I_O0			EVENTOUT
40		PC9	TIM1A_CH4	PCA_CH2	TIM2A_CH4	SDIO_D1		QSPI_BK1_I_O1			EVENTOUT
41	29	PA8	TIM1_CH1	PCA_CH3	TIM2C_CH1			QSPI_BK1_I_O2	USART1_CK	VC1_O	EVENTOUT
42	30	PA9	TIM1_CH2	PCA_CH4	TIM2C_CH2	SDIO_D2	I2S2_MCK	QSPI_BK1_I_O3	USART1_TX	VC1_O	EVENTOUT
43	31	PA10	TIM1_CH3		TIM2C_CH3	SDIO_D3	I2S3_MCK	QSPI_CLK	USART1_RX	VC0_O	EVENTOUT
44	32	PA11 (SWDIO)	TIM1_CH4		TIM2C_CH4	SDIO_CLK	CAN_RX	QSPI_BK1_nCS	USART1_CS	USBBDM	EVENTOUT
45	33	PA12 (SWCLK)	TIM1_ETR			SDIO_CMD	CAN_TX	MCO_O	USART1_RT	USBDP	EVENTOUT
46	34	PA13/ SWDIO/ JTMS			RXD_LP	I2C1_SDA	CAN_RX		UART5_RX		EVENTOUT
47	35	VSS_2									
48	36	VDD_2									
49	37	PA14/ SWCLK/ JTCK			TXD_LP	I2C1_SCL	CAN_TX		UART5_TX	MCO_O	EVENTOUT
50	38	PA15/JTDI	TIM1B_ETR		TIM2_CH1-ETR	SDIO_D2	I2S3_WS/SPI3_NSS	QSPI_BK2_nCS		VC0_O	EVENTOUT
51		PC10	TIM1B_CH1			SDIO_D3	UART4_TX	QSPI_BK1_I_O2	USART3_TX		EVENTOUT
52		PC11	TIM1B_CH2	PCA_ECI		SDIO_CLK	UART4_RX	QSPI_BK1_I_O3	USART3_RX		EVENTOUT
53		PC12	TIM1B_CH3	PCA_CH0		SDIO_CMD	UART5_TX	QSPI_CLK	USART3_CK		EVENTOUT
54		PD2	TIM1B_CH4	PCA_CH1	TIM2A_ETR		UART5_RX	QSPI_BK1_I			EVENTOUT



LQFP		GPIO_AFR[i+3:i]									
64	48	0	1	2	3	4	5	6	7	8	9
							nCS				
55	39	PB3/ JTDO	TIM1B_BKI N	TRACESWO	TIM2_CH2	I2C1_SCL	I2S3_CK/SP I3_SCK			USBPHYT_ OUTCLKSEL	EVENTOUT
56	40	PB4/ JNTRST	TIM1B_ETR		TIM2A_CH 1	I2C1_SDA	SPI3_MISO			USBPHYT_S E	EVENTOUT
57	41	PB5	TIM1B_CH1		TIM2A_CH 2	I2C1_SMBA I	I2S3_SD/SP I3_MOSI			USBPHYT_VI	EVENTOUT
58	42	PB6	TIM1B_CH1 N		TIM2B_CH1	I2C1_SCL			USART1_TX	USBPHYT_OE	EVENTOUT
59	43	PB7	TIM1B_CH2 N	PCA_CH2	TIM2B_CH2	I2C1_SDA			USART1_RX	USBPHYT_P D	EVENTOUT
60	44	BOOT0									
61	45	PB8	TIM1B_CH3 N	PCA_CH3	TIM2B_CH3	I2C1_SCL	CAN_RX	QSPI_BK2_I O0	SDIO_D4	USBPHYT_DPPU	EVENTOUT
62	46	PB9		PCA_CH4	TIM2B_CH4	I2C1_SDA	CAN_TX	QSPI_BK2_I O1	SDIO_D5	USBPHYT_DMPU	EVENTOUT
63	47	VSS_3									
64	48	VDD_3									

Table 4-3 引脚 PIN MD 复用功能说明表

Note : SJTAG/SWD 复用功能重映射 選擇由 User CONFIG 決定



#### 4.4 模块与引脚信号说明

模块功能	引脚名称	说明
Power	VDD	电源
	AVDD	电源
	VCAP	LDO 内核供电 (仅限内部电路使用, 外部连接电容至少 1uf)
Ground	VSS	接地
	AVSS	接地
GPIO (x=0~15)	PAx,PBx,PCx,PDx	PAx,PBx,PCx,PDx 通用数字输入/输出引脚 <b>PA[12:8] : 5V 容忍</b> <b>PB[15:2] : 5V 容忍</b> <b>PC[12:6] : 5V 容忍</b> <b>PD2 : 5V 容忍</b>
NRST	NRST	复位输入端口, 低有效, 芯片复位
EVENTOUT	EVENTOUT	Cortex-M3 CPU 事件输出
ADC	ADC123_IN0~ ADC123_IN3	ADC1,2,3 输入通道 0~15
	ADC12_IN4~ADC12_IN9	
	ADC123_IN10~	
	ADC123_IN13	
	ADC12_IN14~IN15	
	EXVREF	ADC 外部参考电压
	OPx_INN OPx_INP OPx_O	OPA 负端输入 OPA 负端输入 OPA 输出
VC X=0,1	VCxN0~VCxN11	选择 VC0 ,VC1 负端输入
	VCxP0~VCxP11	选择 VC0,VC1 正端输入
	VCx_O	VC0,VC1 比较输出
LVD	LVDIN0	电压侦测输入选择 0
	LVDIN1	电压侦测输入选择 1
	LVDIN2	电压侦测输入选择 2
	LVD_O	电压侦测输出
ISP	BOOT0	当复位时 BOOT0 管脚为高电平, 芯片工作于 ISP 编程模式, 可通过 ISP 协议对 FLASH 进行编程。



模块功能	引脚名称	说明
		当复位时 BOOT0 管脚为低电平，芯片工作于用户模式，芯片执行 FLASH 内的程序代码，可通过 SWD 协议对 FLASH 进行编程。
WKUP	WKUP_0~WKUP_5	PAWKUP 外部唤醒脚位
LPUART	TXD_LP	LPUART 数据发送端
	RXD_LP	LPUART 数据接收端
USART x=1,2,3	CTS_x	USARTx CTS
UART Y=4,5	RTS_x	USARTx RTS
	CK_x	USARTx 同步时钟
	TXD_x	USARTx ,USARTy 数据发送端
	RXD_x	USARTx, USARTy 数据接收端
SPI x=1	MISO_x	SPI 模块主机输入从机输出数据信号
	MOSI_x	SPI 模块主机输出从机输入数据信号
	SCK_x	SPI 模块时钟信号
	NSS_x	SPI 片选择致能
I2C x=0,1	SDA_x	I2C 模块数据信号
	SCL_x	I2C 模块时钟信号
通用定时器 TIMx X=2,3,4,5	Tx_CH1,2,3,4	Timer x 的捕获输入/比较输出/PWM 输出 Ch1,2,3,4
	Tx_ETR	Timerx 的外部计数输入信号
可编程计数阵列 PCA	PCA_ECI	外部时钟输入信号
	PCA_CH0~PCA_CH4	捕获输入/比较输出/PWM 输出 0~4
高级定时器 Advanced TIMX X=1,8,12,	TIMX_CH1,2,3,4	TIMX PWM 输出 channel 1/2/3/4
	TIMX_CH1N,2N,3N,4N	TIMX PWM 输出 反相 channel 1N/2N/3N/4N
	TIMX_BKIN	TIMX 刹车信号输入
	TIMX_ETR	外部触发输入
低功耗定时器 LPTimer	LP_ETR	LP Timer 的外部计数输入信号
	LP_GATE	LP Timer 的门控信号
	LP_TOG	比较输出正端



模块功能	引脚名称	说明
	LP_TOGN	比较输出负端
SPI/I2S x=2,3	MISO_x/I2S	SPI 模块主机输入从机输出数据信号
	MOSI_x/I2S_SD	SPI 模块主机输出从机输入数据信号
	SCK_x/I2S_CK	SPI 模块时钟信号
	NSS_x/I2Sx_WS	SPI 片选择致能
SDIO	SDIO_CLK	Host-to-card clock signal
	SDIO_CMD	Bidirectional command and response signal
	SDIO_D0~D7	Bidirectional data signal (1-bit, 4-bit, or 8-bit MMC Cards; 1-bit or 4-bit in SD cards)
USB	USBDP,USBDM	USB D+ and D- Signal
	USBPHYT_XXX	USB PHY test mode pin
低功耗定时器 LPTimer	LP_ETR	LP Timer 的外部计数输入信号
	LP_GATE	LP Timer 的门控信号
	LP_TOG	比较输出正端
	LP_TOGN	比较输出负端
QSPI	BK1_IO0~IO3	QSPI 模块主机(从机) 输入/输出数据信号
	BK2_IO0~IO3	QSPI 模块主机(从机) 输入/输出数据信号
	Q_CLK	SPI 模块时钟信号
	BK1_nCS,BK2_nCS	SPI 片选择致能
JTAG/SWD	TCK/SWCLK	I 测试时钟输入/序列時鐘輸入
	TMS_SWDIO	IO 测试模式选择/序列資料輸入輸出
	TDO_TRACEWO	O 测试数据输出/ 序列資料輸出引腳 (CPU 除錯介面可通過 SWO 引腳輸出一些除錯 資訊)
	TDI	I 测试数据输入
	NTRST	I 系統復位訊號
TRACE	TRACECLK	O 跟踪调适同步在输出
	TRACEDATA0~3	O 跟踪调适数据输出

Table 4-4 模块与引脚信号说明表



## 【5】 系统和存储器概要

### 5.1 系统架构

主要的系统构成:

- 8 个 AHB 总线系统 Master:
  - Cortex®-M3 内核 ICode 总线(I-bus),DCode 总线(D-bus), 和系统总线(S-bus)
  - PDMA1, PDMA2 总线控制器
  - USB 总线控制器
  - SDIO 总线控制器
- 15 个 AHB 总线 Slave
  - 内部 SRAM 6 個 Slave : 32KSRAMX2,64KBSRAMX4
  - 内部 Flash controller
  - 内部 OTP controller
  - QSPI XIP and QSPI 接口
  - AHB to APB1 Bridge, 包含所有 APB 接口的外设
  - AHB to APB2 Bridge, 包含所有 APB 接口的外设
  - USB 接口
  - RCC 模块
  - CRC,TRAN
  - AES 等 AHB 接口模块

系统的模块框图如图 [Figure3\\_1](#) 所示:

#### ICode 总线

该总线将 Cortex™-M3 内核的指令总线与闪存指令接口相连接。指令预取在此总线上完成。

#### DCode 总线

该总线将 Cortex™-M3 内核的 DCode 总线与存储器的数据接口相连接(常量加载和调试访问)。

#### 系统总线

此总线连接 Cortex®-M3 内核的系统总线(外设总线)到总线矩阵, 总线矩阵协调着内核和 DMA 间的访问。

#### DMA 总线

此总线将 DMA 的 AHB 主控接口与总线矩阵相联, 总线矩阵协调着 CPU 的 DCode 和 DMA 到 SRAM、闪存和外设的访问。

#### AHB/APB 桥(APB)

两个 AHB/APB 桥在 AHB 和 2 个 APB 总线间提供同步连接



## 5.2 总线矩阵

总线矩阵协调内核系统总线和 DMA 與 PM 主控总线之间的访问仲裁，仲裁利用轮换算法总线矩阵架构如 Figure 5-1，

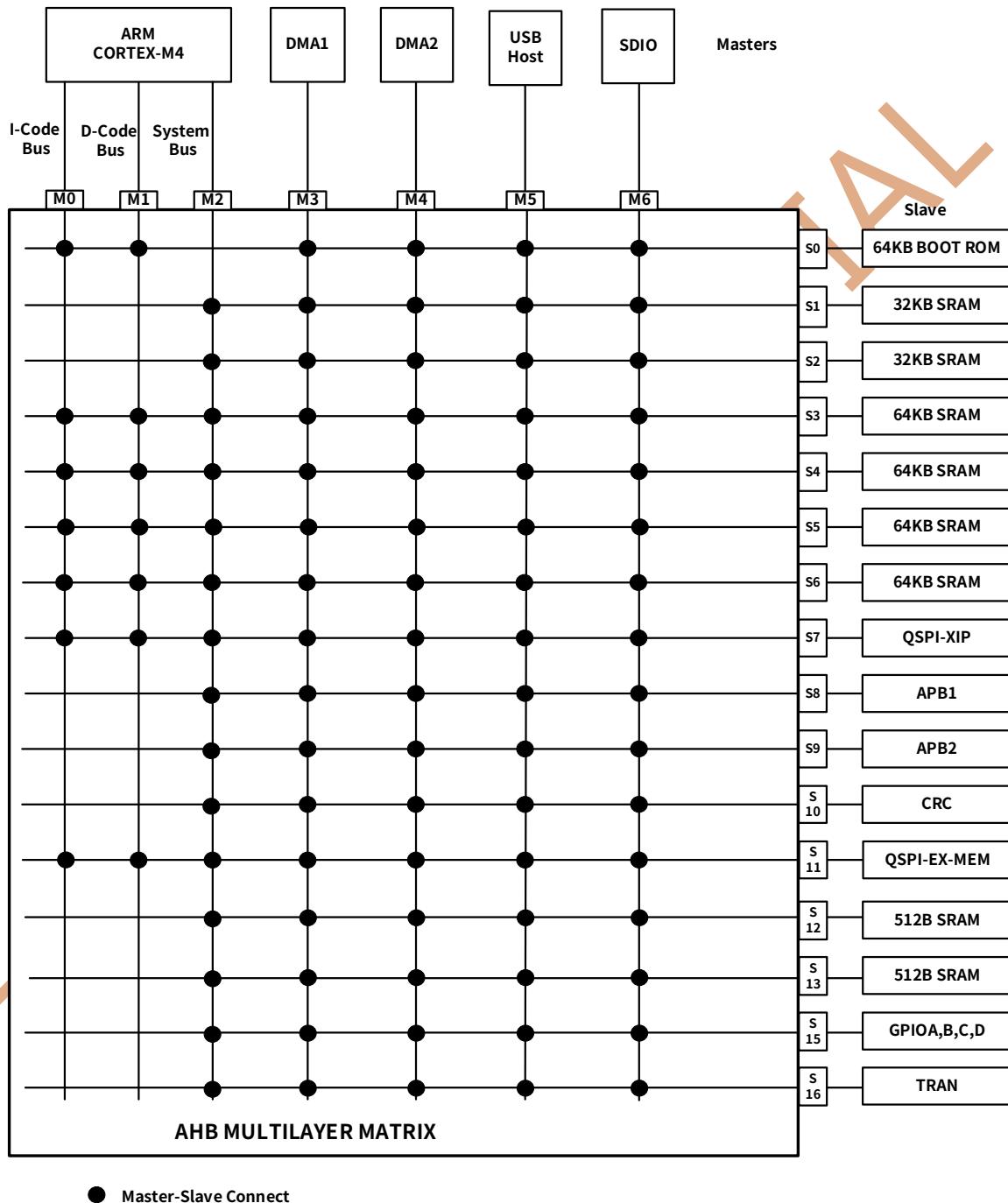


Figure 5-1 总线矩阵图



### 5.3 存储器映射

系统的地址空间总共包含程序存储空间、数据存储空间、周边模块寄存器、I/O 端口等。

数据使用小端点格式，就是数据的高字节保存在内存的高地址中，而数据的低字节保存在内存的低地址中。整个系统地址空间的划分如下图，图 5-2 所示：

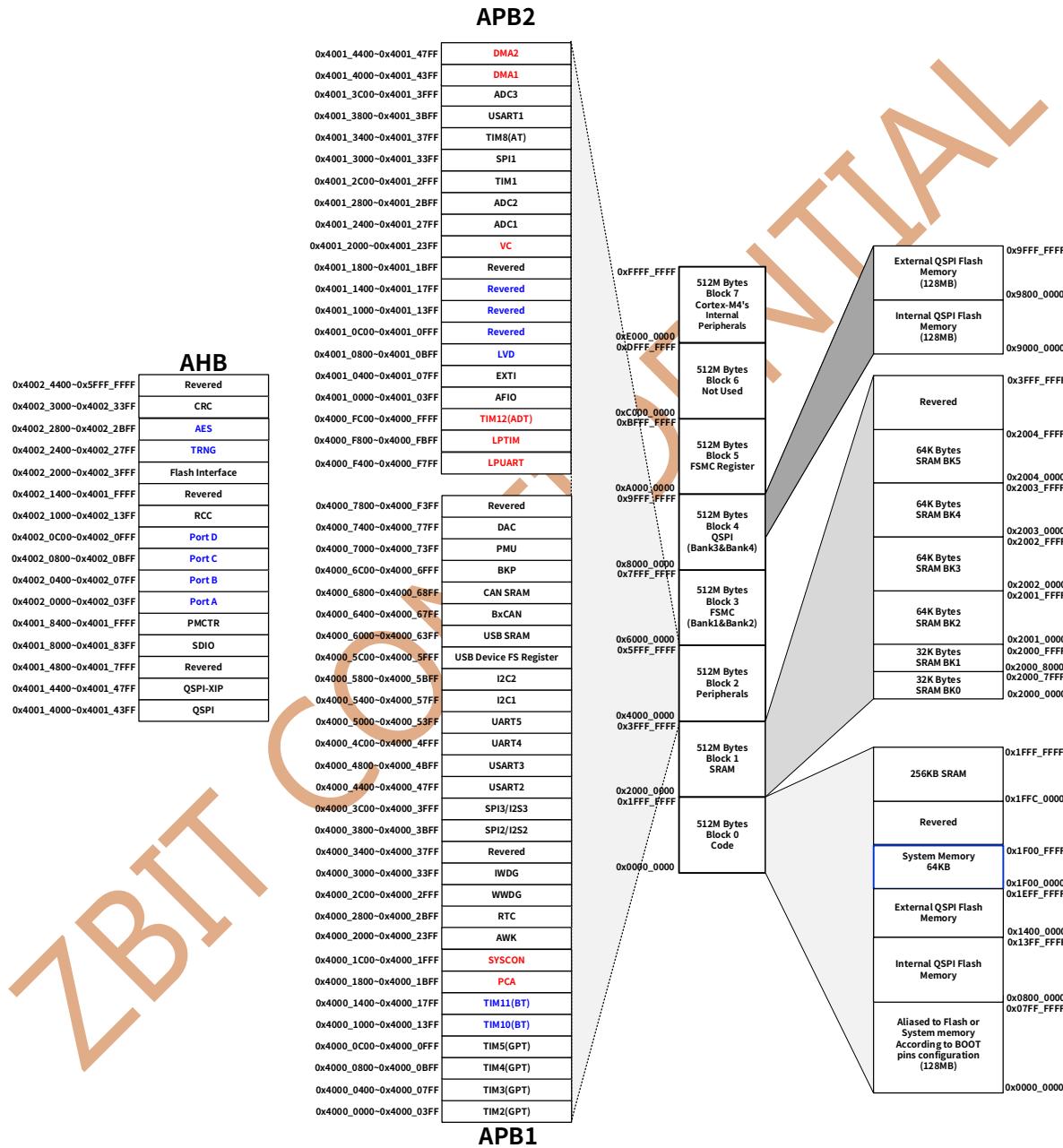


Figure 5-2 储存器映像图



#### 5.4 存储空间和模块地址

下列表给出了 ZB32F103 chip 内部包含的各模块的地址空间和边界信息。

边界地址	外设	Bus	寄存器映像
0x4002 3000 - 0x4002 33FF	CRC	AHB	
0x4002 2800 - 0x4002 2BFF	AES		
0x4002 2400 - 0x4002 27FF	TRNG		
0x4002 1000 - 0x4002 13FF	RCC		
0x4002 0400 - 0x4002 07FF	Revered		
0x4002 0000 - 0x4002 03FF	Revered		
0x4001_8400~0x4001_FFFF	PMCTR		
0x4001 8000 - 0x4001 83FF	SDIO		
0x4001 4000 - 0x4001 47FF	QSPI-XIP		
0x4001_4000~0x4001_43FF	QSPI		
0x4001 0000 - 0x4001 3FFF	AHB3PB2 bridge		
0x4000 6400 - 0x4000 FFFF	AHB3PB1 bridge		
0x4000 5C00 - 0x4000 63FF	USB		
0x4000 0000 - 0x4000 5BFF	AHB3PB1 bridge		
0x2000 0000 - 0x2000 FFFF	32KB SRAM		
0x1FFC 0000 - 0x1FFF FFFF	256KB SRAM		
0x1F00 0000 - 0x1F00 FFFF	64KB Boot ROM		
0x1400_0000-0x1EFF_FFFF	QSPI-内存映像模(128MB)	APB1	
0x0800_0000-0x13FF_FFFF	QSPI-XIP 内存映像模(128MB)		
0x4002 0C00 - 0x4002 0FFF	GPIO Port D		
0x4002 0800 - 0x4002 0BFF	GPIO Port C		
0x4002 0400 - 0x4002 07FF	GPIO Port B		
0x4002 0000 - 0x4002 03FF	GPIO Port A		
0x4000 7400 - 0x4000 77FF	DAC	APB1	
0x4000 7000 - 0x4000 73FF	PMU		



边界地址	外设	Bus	寄存器映像
0x4000 6C00 - 0x4000 6FFF	BKP		
0x4000 6400 - 0x4000 67FF	CANBUS		
0x4000 6200 - 0x4000 63FF	USB/CAN 共享的 512 字节 SRAM2		
0x4000 6000 - 0x4000 61FF	USB/CAN 共享的 512 字节 SRAM1		
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART5		
0x4000 4C00 - 0x4000 4FFF	UART4		
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 3C00 - 0x4000 3FFF	SPI3/I2S		
0x4000 3800 - 0x4000 3BFF	SPI2/I2S		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 4FF	RTC		
0x4000 2000 - 0x4000 23FF	AWK		
0x4000 1C00 - 0x4000 1FFF	SYSCON		
0x4000 1800 - 0x4000 12FF	PCA		
0x4000 1400 - 0x4000 17FF	TIM11 timer (TIM7)		
0x4000 1000 - 0x4000 13FF	TIM10 timer (TIM6)		
0x4000 0C00 - 0x4000 0FFF	TIM2C timer		
0x4000 0800 - 0x4000 0BFF	TIM2B timer		
0x4000 0400 - 0x4000 07FF	TIM2A timer		
0x4000 0000 - 0x4000 03FF	TIM2 timer		



边界地址	外设	Bus	寄存器映像
0x4001 4400 - 0x4001 47FF	DMA2		
0x4001 4000 - 0x4001 43FF	DMA1		
0x4001 3C00 - 0x4001 3FFF	ADC3		
0x4001 3800 - 0x4001 3BFF	USART1		
0x4001 3400 - 0x4001 37FF	TIM1A timer		
0x4001 3000 - 0x4001 33FF	SPI1		
0x4001 2C00 - 0x4001 2FFF	TIM1 timer		
0x4001 2800 - 0x4001 2BFF	ADC2		
0x4001 2400 - 0x4001 27FF	ADC1		
0x4001 2000 - 0x4001 23FF	VC(VC1,VC2)		
0x4001 0800 - 0x4001 0BFF	LVD		
0x4001 0400 - 0x4001 07FF	EXTI		
0x4001 0000 - 0x4001 03FF	AFIO		
0x4000 FC00 - 0x4001 FFFF	TIM1B timer		
0x4000 F800 - 0x4001 FBFF	LPTIM timer		
0x4000 F400 - 0x4001 F7FF	LPUART	APB2	

Table 5-1 地址对应表

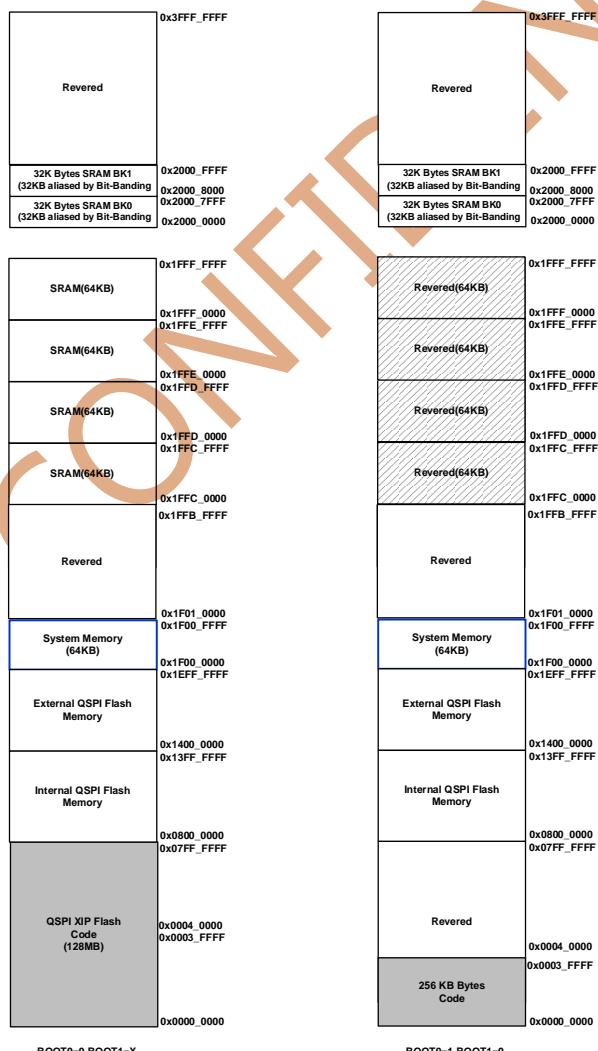


## 5.5 BOOT PIN 地址映射

L032 當發生 POR, NRST, WDT, LVR 重置時，會 Latch BOOT0, BOOT1PIN 狀態將儲存器作地址重新映射，如下圖所示([Timing Refer 11.3-8.2](#))：

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Boot ROM	系统存储器(Boot ROM) to Internal QSPI (XIP)
0	1	Embedded SRAM	Embedded SRAM is selected as boot space
1	1	Boot ROM	系统存储器(Boot ROM) to Internal SRAM (256KB)

Table 5-2BOOT 启动模式



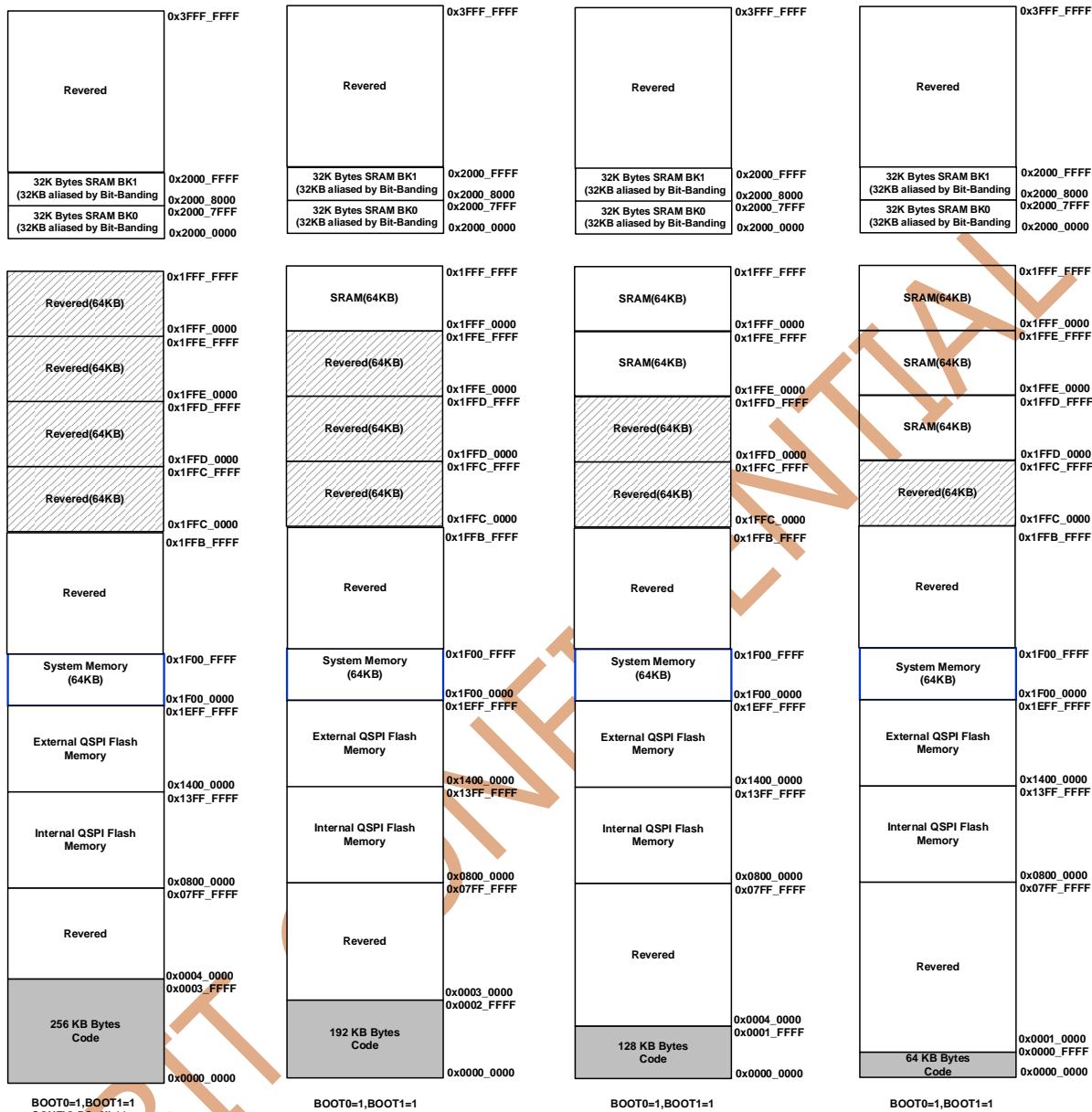


Figure 5-3 BOOT PIN 地址映射图



## 【6】 电源管理和工作模式

### 6.1 电源

的工作电压(VDD)为1.8~3.6V。通过内置的电压调节器提供所需的1.2V/1.8V电源。

当主电源VDD掉电后，通过VBAT脚为实时时钟(RTC)和备份寄存器提供电源。

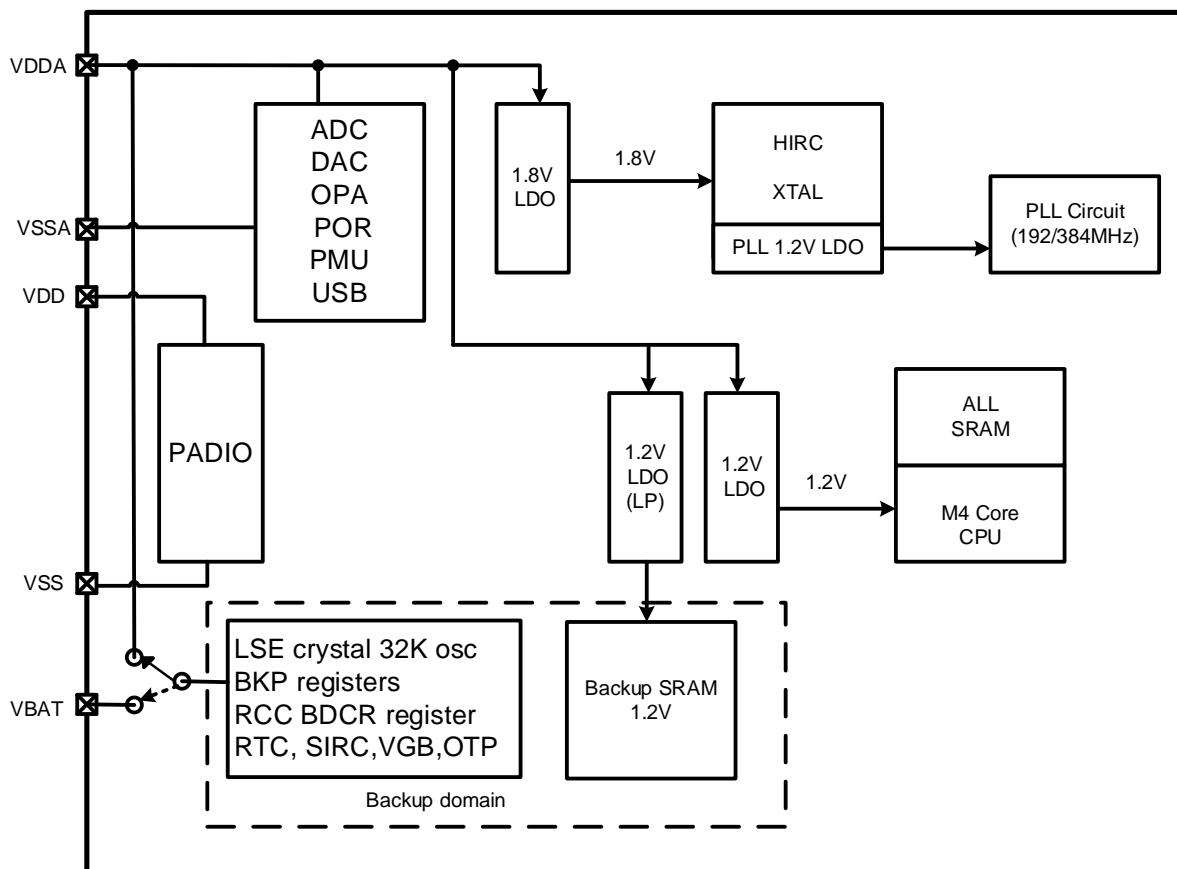


Figure 6-1 电源框图

#### 6.1-1 独立的A/D,OPA转换器供电和参考电压

为了提高ADC转换OPA1倍率的精确度和ADC/OPA使用一个独立的电源供电，过滤和屏蔽来自印刷电路板上的毛刺干扰

#### 6.1-2 电池备份区域

使用电池或其他电源连接到VBAT脚上，当VDD断电时，可以保存备份寄存器的内容和维持RTC的功能。

VBAT脚也为RTC、LSE振荡器和PC13至PC15供电，这保证当主要电源被切断时RTC能继续工作。切换到VBAT供电由复位模块中的掉电复位功能控制。



如果应用中没有使用外部电池，VBAT 必须连接到 VDD 引脚上

ZBIT CONFIDENTIAL

---

---



## 6.2 工作模式

**ZB32F103** 的电源管理模块负责管理本产品各种工作模式之间的切换，以及控制各工作模式下的各功能模块的工作状态。本产品的工作电压(VDD)为 1.8~3.6V。本产品有如下几个工作模式：

1. **运行模式(Active Mode)**: CPU 运行，周边功能模块运行。
2. **休眠模式(Sleep Mode)**: 在睡眠模式，只有 CPU 停止，所有外设处于工作状态并可在发生中断/事件时唤醒 CPU。
3. **停机模式(Stop Mode)** : 在保持 SRAM 和寄存器内容不丢失的情况下，停机模式可以达到最低的电能消耗。在停机模式下，停止所有内部 1.2V 部分的供电，PLL、HSI 的 RC 振荡器和 HSE 晶体振荡器被关闭，调压器可以被置于普通模式或低功耗模式。可以通过任一配置成 EXTI 的信号把微控制器从停机模式中唤醒，EXTI 信号可以是 16 个外部 I/O 口之一、PWD 的输出、RTC 闹钟或 USB 的唤醒信号。
4. **待机模式(Standby mode)** :  
在待机模式下可以达到最低的电能消耗。内部的电压调压器被关闭，因此所有内部 1.8V 部分的供电被切断；PLL、HSI 的 RC 振荡器和 HSE 晶体振荡器也被关闭；进入待机模式后，SRAM 和寄存器的内容将消失，但后备寄存器的内容仍然保留，待机电路仍工作。  
从待机模式退出的条件是：NRST 上的外部复位信号、IWDG 复位、WKUP 引脚上的一个上升边沿或 RTC 的闹钟到时。在进入停机或待机模式时，RTC、IWDG 和对应的时钟不会被停止
5. **关机模式(power down mode)** : 所有供电被切断 RTC、IWDG 和对应的时钟会被停止，仅保持 I/O 状态(关机模式前设定的状态)，关机模式退出的条件是 NRST 上的外部复位信号，WKUP 引脚上的一个上升边沿。启动时间较长。



## 6.2-1 运行模式(Active Mode)

系统在电源上电复位后，或从各低功耗模式唤醒后，微控制器 MCU 处于运行状态，各模块的运行状态如 Figure 6-1 运行模式下可运行模块一览所示。当 MCU 不需继续运行时，可以利用多种低功耗模

式来节能。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

运行模式(Active Mode)				
MCU Core	Timer	Serial Interface	Analog	Clock
Cortex-M3	TIM10,TIM11	USART1,2,3	ADC	XTAL32
SWD	TIM1,TIM1A,TIM1B	SPI/I2S	RNG	XTAH
Flash	TIM2,TIM2A,TIM2B,TIM2C	I2C1,2	OPA0,1	HIRC
SRAM	PCA	LPUART	Reset	SIRC
DMA1,DMA2	IWDG		POR/PDR	PLL
HMUL	WWDG		BandGap	RTC
CRC	LPTIM	GPIO	Temp. Sensor	CLK_TRIM
TRAN	AWK	UART4,5	LVD	
AES		USB	VC0,1	
		CAN	PIN Reset	
		SDIO	DAC0,1	

Table 6-1 运行模式下的菜单

几种降低运行模式下芯片功耗的方法：

1. 在运行模式下，通过对预分频寄存器(RCC\_HCLKDIV, RCC\_PCLKDIV)进行编程，可以降低任意一个系统时钟(HCLK, PCLK)的速度。进入睡眠模式前，也可以利用预分频器来降低外设的时钟。
2. 在运行模式下，关闭不使用外设的时钟(RCC\_HCLKEN, RCC\_PCLKEN)来减少功耗。
3. 使用深度休眠模式代替休眠模式，因为本产品的唤醒时间极短(~3us)，亦可满足系统的实时响应的需求。



## 6.2-2 低功耗模式

在系统或电源复位以后，微控制器处于运行状态。当 CPU 不需继续运行时，可以利用多种低功耗模式来节省功耗，例如等待某个外部事件时。用户需要根据最低电源消耗、最快速启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

ZB32F103 有三种低功耗模式：

- (1) 睡眠模式(Cortex™-M3 内核停止，所有外设包括 Cortex-M3 核心的外设，如 NVIC、系统时钟(SysTick)等仍在运行)
- (2) 停止模式(所有的时钟都已停止)
- (3) 待机模式(1.2V/1.8V 电源关闭)
- (4) 关机模式(1.2V/1.8V 电源关闭,VDD )

此外，在运行模式下，可以通过以下方式中的一种降低功耗：

- 降低系统时钟：

在运行模式下，通过对预分频寄存器进行编程，可以降低任意一个系统时钟(SYSCLK、HCLK、PCLK1、PCLK2)的速度。进入睡眠模式前，也可以利用预分频器来降低外设的时钟。  
详见第 6.3.2 节：时钟配置寄存器(RCC\_CFGR)。

- 关闭 APB 和 AHB 总线上未被使用的外设时钟：

在运行模式下，任何时候都可以通过停止为外设和内存提供时钟(HCLK 和 PCLKx)来减少功耗。

为了在睡眠模式下更多地减少功耗，可在执行 WFI 或 WFE 指令前关闭所有外设的时钟。

通过设置 AHB 外设时钟使能寄存器 (RCC\_AHBENR)、APB2 外设时钟使能寄存器 (RCC\_APB2ENR) 和 APB1 外设时钟使能寄存器(RCC\_APB1ENR)来开关各个外设模块的时钟。



模式	进入	唤醒	对 LDO 区域时钟的影响	对 VDD 区域时钟的影响	LDO	GPIO
睡眠 (SLEEP-NOW 或 SLEEP-ON-EXIT)	WFI WFE	任一中断唤醒事件	CPU 时钟关，对其他时钟和 ADC 时钟无影响	无	开	保持在运行模式时的状态
停机	PDDS 和 LPDS 位 +SLEEPDEEP 位 (M3.SCR) +WFI 或 WFE	任一外部中断 (在外部中断寄存器中设置)	关闭所有 LDO 区域的时钟	HSI 和 HSE 的振荡器关闭	开启或处于低功耗模式(依据电源控制寄存器 (PWR_CR) 的设定) (僅保持 SRAM 和寄存器内容不丢失)	保持在运行模式时的状态
待机	PDDS 位 +SLEEPDEEP (M3.SCR) 位 +WFI 或 WFE	(1)WKUP 引脚的上升沿 (2)RTC 闹钟件 (3)NRST 引脚上的外部复位 (4)IWDG 复位 (5)AWK		HSI 和 HSE 的振荡器关闭	关 (Latch DFF)	除了 (1) 复位引脚 (2) TAMPER 引脚 (3) 被使能的唤醒引脚 其他的 I/O 引脚处于高阻态
关机		(1)WKUP 引脚的上升沿 (2)NRST 引脚上的外部复位		HSI 和 HSE 的振荡器关闭和 VDD logic Power off Keep Register Latch Up Power		除了 (1) 复位引脚 (2) TAMPER 引脚 (3) 被使能的唤醒引脚 其他的 I/O 引脚处于高阻态

Table 6-2 低功耗模式表



### 6.2-2.1 休眠模式(Sleep Mode)

使用 WFI 指令可以进入休眠模式，休眠模式下，CPU 停止运行，但系统时钟、NVIC 中断处理以及非 HCLK 驱动的周边功能模块仍都可以工作。系统进入休眠状态，不会改变端口状态，在进入休眠前请根据需要更改 IO 的状态为休眠模式下的状态。

如何进入休眠模式：

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

**SLEEP-NOW:** 如果 SLEEPONEXIT=0，当 WFI 或 WFE 被执行时，微控制器立即进入睡眠模式。

**SLEEP-ON-EXIT:** 如果 SLEEPONEXIT=1，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入睡眠模式。关于如何进入睡眠模式，更多的细节参考 Table 6-4.

SLEEP-NOW 模式	说明
进入	在以下条件下执行 WFI(等待中断)或 WFE(等待事件)指令： – SLEEPDEEP = 0 和 – SLEEPONEXIT = 0 参考 Cortex-M3 系统控制寄存器。
退出	如果执行 WFI 进入睡眠模式： 中断：参考中断向量表( <a href="#">Table3-2</a> ) 如果执行 WFE 进入睡眠模式： 唤醒事件：参考唤醒事件管理(第 9.2.3 节)
唤醒延时	无

Table 6-3 SLEEP-NOW 模式表

如何退出休眠模式：

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

如果执行 WFE 指令进入睡眠模式，则一旦发生唤醒事件时，微处理器都将从睡眠模式退出。唤醒事件可以通过下述方式产生：

- 在外设控制寄存器中使能一个中断，而不是在 NVIC(嵌套向量中断控制器)中使能，并且在外设的 NVIC 中断通道挂起位(在 NVIC 中断清除挂起寄存器中)必须被清除。
  - 配置一个外部或内部的 EXIT 线为事件模式。当 MCU 从 WFE 中唤醒后，因为与事件线对应的挂起位未被设置，不必清除外设的中断挂起位或外设的 NVIC 中断通道挂起位。
- 该模式唤醒所需的时间最短，因为没有时间损失在中断的进入或退出上。
- 关于如何退出睡眠模式，更多的细节参考



SLEEP-ON_EXIT 模式	说明
进入	在以下条件下执行 WFI 指令： – SLEEPDEEP = 0 和 – SLEEPONEXIT = 1 参考 Cortex™-M3 系统控制寄存器
退出	中断：参考中断向量表( <a href="#">Table3-2</a> )
唤醒延时	无

Table 6-4 SLEEP-ON-EXIT 模式

### 6.2-2.2 停机模式(Stop Mode)

停止模式是在 Cortex™-M3 的深睡眠模式基础上结合了外设的时钟控制机制，在停止模式下电压调节器可运行在正常或低功耗模式。此时在 1.8V 供电区域的所有时钟都被停止，PLL、HSI 和 HSE RC 振荡器的功能被禁止，SRAM 和寄存器内容被保留下。

在停止模式下，所有的 I/O 引脚都保持它们在运行模式时的状态。

#### 进入停机模式

关于如何进入停机模式，详见 [Table 6-6](#)。

电源控制寄存器(PWR\_CR)在停机模式下，通过设置的 LPDS 位使内部调节器进入低功耗模式，能够降低更多的功耗。

如果正在进行闪存编程，直到对内存访问完成，系统才进入停机模式。

如果正在进行对 APB 的访问，直到对 APB 访问完成，系统才进入停机模式。

通过对独立的控制位进行编程，可选择以下功能：

- 独立看门狗(IWDG)：可通过写入看门狗的键寄存器或硬件选择来启动 IWDG。一旦启动了独立看门狗，除了系统复位，它不能再被停止。详见 17.3 节。
- 实时时钟(RTC)：通过备份域控制寄存器 (RCC\_BDCR)的 RTCEN 位来设置。
- 内部低速 RC 振荡器(SIRC RC)：通过控制/状态寄存器 (RCC\_CSR)的 SIRCENN 位来设置。
- 外部 32.768kHz 振荡器(LSE)：通过备份域控制寄存器 (RCC\_BDCR)的 LXTEN 位设置。

在停机模式下，如果在进入该模式前 ADC 和 DAC 没有被关闭，那么这些外设仍然消耗电流。通过设置寄存器 ADCX\_CR2 的 ADON 位和寄存器 DAC\_CR 的 ENx 位为 0 可关闭这 2 个外设。

#### 退出停机模式

关于如何退出停机模式，详见下表。



当一个中断或唤醒事件导致退出停止模式时，HSI RC 振荡器被选为系统时钟。

当电压调节器处于低功耗模式下，当系统从停止模式退出时，将会有一段额外的启动延时。如果在停止模式期间保持内部调节器开启，则退出启动时间会缩短，但相应的功耗会增加。

停止模式	说明
进入	<p>在以下条件下执行 WFI(等待中断)或 WFE(等待事件)指令：</p> <ul style="list-style-type: none"><li>- 设置 Cortex-M3 系统控制寄存器中的 SLEEPDEEP 位</li><li>- 清除电源控制寄存器(PWR_CR)中的 PDDS 位</li><li>- 通过设置 PWR_CR 中 LPDS 位选择电压调节器的模式</li></ul> <p>注：为了进入停止模式，所有的外部中断的请求位(挂起寄存器(EXTI_PR))和 RTC 的闹钟标志都必须被清除，否则停止模式的进入流程将会被跳过，程序继续运行。</p>
退出	<p>如果执行 WFI 进入停止模式：</p> <p>设置任一外部中断线为中断模式(在 NVIC 中必须使能相应的外部中断向量)。参考中断向量表(<a href="#">Table3-2</a>)</p> <p>如果执行 WFE 进入停止模式：</p> <p>设置任一外部中断线为事件模式。参见唤醒事件管理(第 9.2.3 节)。</p>
唤醒延时	HSI RC 唤醒时间 + 电压调节器从低功耗唤醒的时间。

Table 6-5 停止模式表



### 6.2-2.3 待机模式(Standby mode)

待机模式可实现系统的最低功耗。该模式是在 Cortex-M3 深睡眠模式时关闭电压调节器。整个 1.2V 供电区域被断电。PLL、HSI 和 HSE 振荡器也被断电。SRAM 和寄存器内容丢失。只有备份的寄存器和待机电路维持供电(见图 6-1)。

#### 进入待机模式

关于如何进入待机模式，详见表 12。

可以通过设置独立的控制位，选择以下待机模式的功能：

- 独立看门狗(IWDG): 可通过写入看门狗的键寄存器或硬件选择来启动 IWDG。一旦启动了独立看门狗，除了系统复位，它不能再被停止。详见 17.3 节。
- 实时时钟(RTC): 通过备用区域控制寄存器(RCC\_BDCR)的 RTCEN 位来设置。
- 内部低速 RC 振荡器(SIRC): 通过控制/状态寄存器(RCC\_CR)的 SIRCEN 位来设置。
- 外部 32.768kHz 振荡器(LXT): 通过备用区域控制寄存器(RCC\_CR)的 LXTON 位设置。

待机模式	说明
进入	在以下条件下执行 WFI(等待中断)或 WFE(等待事件)指令： - 设置 Cortex-M3 系统控制寄存器中的 SLEEPDEEP 位 - 设置电源控制寄存器(PWR_CR)中的 PDDE 位 - 清除电源控制/状态寄存器(PWR_CSR)中的 WUF 位
退出	WKUP 引脚的上升沿、RTC 闹钟事件的上升沿、NRST 引脚上外部复位、IWDG 复位
唤醒延时	复位阶段时电压调节器的启动。

Table 6-6 待机模式表

#### 待机模式下的输入/输出端口状态

在待机模式下，所有的 I/O 引脚处于高阻态，除了以下的引脚：

- 复位引脚(始终有效)
- 当被设置为防侵入或校准输出时的 TAMPER 引脚
- 被使能的唤醒引脚
- 被使能的唤醒引脚

#### 调试模式

默认情况下，如果在进行调试微处理器时，使微处理器进入停止或待机模式，将失去调试连接。这是因为 Cortex™-M3 的内核失去了时钟。然而，通过设置 DBGMCU\_CR 寄存器中的某些配置位，可以在使用低功耗模式下调试软件。更多的细节请参考第 29.16.1 节：低功耗模式的调试支持。



#### 6.2-2.4 关机模式(power down mode)

所有供电被切断 RTC、IWDG 和对应的时钟会被停止, I/O 引脚处于高阻态, 关机模式退出的条件是 NRST 上的外部复位信号, WKUP 引脚上的一个上升边沿.启动时间较长.

ZBT CONFIDENTIAL

---

---



### 6.3 PMU 寄存器列表

PMU 基址: 0x4000 7000

偏移地址	名称	描述	复位值
0x00	PWR_CR	电源控制寄存器	0x0000 0000
0x04	PWR_CSR	电源控制/状态寄存器	0x0000 0000
0x08	PWR_PSWK	掉电唤醒始能寄存器	0x0000 0000

Table 6-7 PMU 寄存器列表和复位值



## 6.4 PMU 寄存器說明

### 6.4-1 电源控制寄存器(PWR\_CR)

Address offset: 0x00 Reset value: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DBP	PLS			PVDE	CSBF	CWUF	PDDS
r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rc_w1	rc_w1	rw	rw

Bit	标记	功能描述	复位值	读写
31:9	Reserved		0x0	r
8	DBP	<p><b>DBP:</b> 取消后备区域的写保护</p> <p>在复位后，RTC 和后备寄存器处于被保护状态以防意外写入。</p> <p>设置这位允许写入这些寄存器。</p> <p>0: 禁止写入 RTC 和后备寄存器</p> <p>1: 允许写入 RTC 和后备寄存器</p>	0x0	rw
7:5	PLS[2:0]	<p><b>PVD电平选择</b></p> <p>这些位用于选择电源电压监测器的电压阀值</p> <p>000: 2.2V</p> <p>001: 2.3V</p> <p>010: 2.4V</p> <p>011: 2.5V</p> <p>100: 2.6V</p> <p>101: 2.7V</p> <p>110: 2.8V</p> <p>111: 2.9V</p>	0x0	rw
4	PVDE	<p>电源电压监测器(PVD)使能</p> <p>0: 禁止 PVD</p> <p>1: 开启 PVD</p>	0x0	rw



Bit	标记	功能描述	复位值	读写
3	CSBF	清除待机位 始终读出为 0 0: 无功效 1: 清除 SBF 待机位(写)	0x0	rc_w1
2	CWUF	清除唤醒位 始终读出为 0 0: 无功效 1: 2 个系统时钟周期后清除 WUF 唤醒位(写)	0x0	rc_w1
1	PDDS	PDDS: 掉电深睡眠 与LPDS位协同操作 0: 当CPU进入深睡眠时进入停机模式，调压器的状态由LPDS位控制。 1: CPU 进入深睡眠时进入待机模式。	0x0	rw
0	LPDS	LPDS: 深睡眠下的低功耗 PDDS=0 时，与 PDDS 位协同操作 0: 在停机模式下电压调压器开启 1: 在停机模式下电压调压器处于低功耗模式	0x0	rw



#### 6.4-2 掉电唤醒始能寄存器(PWR\_PDWK)

地址偏移: 0x08

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:8	PBWKEN[7:0]	PB WKEN[7] 0: PB7_WKUP 唤醒无效 1: PB7_WKUP唤醒使能 PB WKEN[6] 0: PB6_WKUP 唤醒无效 1: PB6_WKUP唤醒使能 PB WKEN[5] 0: PB5_WKUP 唤醒无效 1: PB5_WKUP唤醒使能 PB WKEN[4] 0: PB4_WKUP 唤醒无效 1: PB4_WKUP唤醒使能 PB WKEN[3] 0: PB3_WKUP 唤醒无效 1: PB3_WKUP唤醒使能 PB WKEN[2] 0: PB2_WKUP 唤醒无效 1: PB2_WKUP唤醒使能 PB WKEN[1] 0: PB1_WKUP 唤醒无效 1: PB1_WKUP唤醒使能 PB WKEN[0] Revered	0X0	R/W
7:0	PAWKEN[7:0]	PA WKEN[7] 0: PA15_WKUP 唤醒无效 1: PA15_WKUP唤醒使能 PA WKEN[6] 0: PA14_WKUP 唤醒无效 1: PA14_WKUP唤醒使能 PA WKEN[5] 0: PA13_WKUP 唤醒无效 1: PA13_WKUP唤醒使能	0X0	R/W



位	标记	功能描述	复位值	读写
		PA WKEN[4] 0: PA12_WKUP 唤醒无效 1: PA12_WKUP唤醒使能 PA WKEN[3] 0: PA11_WKUP 唤醒无效 1: PA11_WKUP唤醒使能 PA WKEN[2] 0: PA10_WKUP 唤醒无效 1: PA10_WKUP唤醒使能 PA WKEN[1] 0: PA9_WKUP 唤醒无效 1: PA9_WKUP唤醒使能 PA WKEN[0] 0: PA8_WKUP 唤醒无效 1: PA8_WKUP唤醒使能		

ZBT CONFIDENTIAL



#### 6.4-3 电源控制/状态寄存器(PWR\_CSR)

Address offset: 0x04 Reset value: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EWUP	Reserved				PVDO	SBF	WUF
r	r	r	r	r	r	r	rw	r	r	r	r	r	r	r	r

Bit	标记	功能描述	复位值	读写
31:9	Reserved		0x0	r
8	EWUP	ENUP: 使能 WKUP 引脚(PA0-WK) 0: WKUP 引脚为通用 I/O。WKUP 引脚上的事件不能将 CPU 从待机模式唤醒 1: WKUP 引脚用于将 CPU 从待机模式唤醒, WKUP 引脚被强置为输入下拉的配置(WKUP 引脚上的上升沿将系统从待机模式唤醒) 注: 在系统复位时清除这一位。	0x0	rw
7:3	Reserved	保留。始终读为 0。	0x0	r
3	Reserved	保留。始终读为 0。	0x0	r
2	PVDO	PVD 输出 当 PVD 被 PVDE 位使能后该位才有效 0: VDD/VDDA 高于由 PLS[2:0]选定的 PVD 阈值 1: VDD/VDDA 低于由 PLS[2:0]选定的 PVD 阈值 注: 在待机模式下 PVD 被停止。因此, 待机模式后或复位后, 直到设置 PVDE 位之前, 该位为 0。	0x0	r
1	SBF	待机标志(Standby flag) 该位由硬件设置, 并只能由 POR/PDR(上电/掉电复位)或设置	0x0	r



Bit	标记	功能描述	复位值	读写
		电源控制寄存器(PWR_CR)的 CSBF 位清除。 0: 系统不在待机模式 1: 系统进入待机模式		
0	WUF	唤醒标志(Wakeup flag) 该位由硬件设置，并只能由 POR/PDR(上电/掉电复位)或设置 电源控制寄存器(PWR_CR)的 CWUF 位清除。 0: 没有发生唤醒事件 1: 在 WKUP 引脚上发生唤醒事件或出现 RTC 闹钟事件。 注：当 WKUP 引脚已经是高电平时，在(通过设置 EWUP 位) 使能 WKUP 引脚时，会检测到一个额外的 事件。	0x0	r



## 【7】 备份寄存器(BKP)

### 7.1 BKP 简介

备份寄存器是 42 个 16 位的寄存器，可用来存储 84 个字节的用户应用程序数据。他们处在备份域里，当 VDD 电源被切断，他们仍然由 VBAT 维持供电。当系统在待机模式下被唤醒，或系统复位或电源复位时，他们也不会被复位。

此外，BKP 控制寄存器用来管理侵入检测和 RTC 校准功能。

复位后，对备份寄存器和 RTC 的访问被禁止，并且备份域被保护以防止可能存在的意外的写操作。执行以下操作可以使能对备份寄存器和 RTC 的访问。

- 通过设置寄存器 RCC\_APB1ENR 的 PWREN 和 BKPen 位来打开电源和后备接口的时钟
- 电源控制寄存器(PWR\_CR)的 DBP 位来使能对后备寄存器和 RTC 的访问。

### 7.2 BKP 特性

- 84 字节数据后备寄存器
- 用来管理防侵入检测并具有中断功能的状态/控制寄存器
- 用来存储 RTC 校验值的校验寄存器。
- 在 PC13 引脚(当该引脚不用于侵入检测时)上输出 RTC 校准时钟，RTC 铃声脉冲或者秒脉冲

### 7.3 BKP 功能描述

#### 7.3-1 侵入检测

当 TAMPER 引脚上的信号从 0 变成 1 或者从 1 变成 0(取决于备份控制寄存器 BKP\_CR 的 TPAL 位)，会产生一个侵入检测事件。侵入检测事件将所有数据备份寄存器内容清除。

然而为了避免丢失侵入事件，侵入检测信号是边沿检测的信号与侵入检测允许位的逻辑与，从而在侵入检测引脚被允许前发生的侵入事件也可以被检测到。

- 当 TPAL=0 时：如果在启动侵入检测 TAMPER 引脚前(通过设置 TPE 位)该引脚已经为高电平，一旦启动侵入检测功能，则会产生一个额外的侵入事件(尽管在 TPE 位置' 1' 后并没有出现上升沿)。
- 当 TPAL=1 时：如果在启动侵入检测引脚 TAMPER 前(通过设置 TPE 位)该引脚已经为低电平，一旦启动侵入检测功能，则会产生一个额外的侵入事件(尽管在 TPE 位置' 1' 后并没有出现下降沿)。

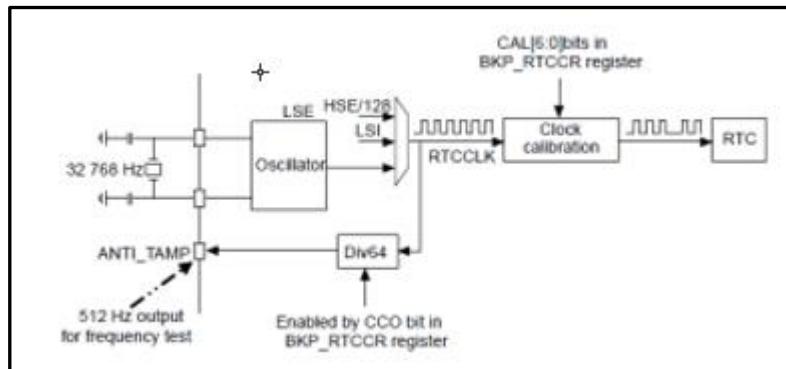
设置 BKP\_CSR 寄存器的 TPIE 位为' 1'，当检测到侵入事件时就会产生一个中断。

在一个侵入事件被检测到并被清除后，侵入检测引脚 TAMPER 应该被禁止。然后，在再次写入备份数据寄存器前重新用 TPE 位启动侵入检测功能。这样，可以阻止软件在侵入检测引脚上仍然有侵入事件时对备份数据寄存器进行写操作。这相当于对侵入引脚 TAMPER 进行电平检测。

注：当 VDD 电源断开时，侵入检测功能仍然有效。为了避免不必要的复位数据备份寄存器，TAMPER 引脚应该在片外连接到正确的电平。



### 7.3-2 RTC 校准



对于 RTC 时钟校准电路，它每  $2^{20}$  个时钟周期去除 0 到 127 个周期。次数 取决于已加载到 BKP 的 RTC 时钟校准寄存器的七个最低有效位中的值。由于 RTC 时钟校准 即使设备断电，校准值也不会丢失 VBAT 引脚连接了电池。

每个校准步骤的效果是每 1 048 576 (220) 个实际

振荡器周期减去 1 个振荡器周期。也就是说，

校准寄存器中每个校准步骤的调整量为  $0.954(1000000/220)$  ppm。因此，振荡器时钟可以从 0 降低到 121 ppm.



#### 7.4 BKP 寄存器列表

BKP 基址: 0x4000 6C00

偏移地址	名称	描述	复位值
0x00	Revered	Revered	0x0000 0000
0x04	BKP_DR1	备份数据寄存器1	0x0000 0000
0x08	BKP_DR2	备份数据寄存器2	0x0000 0000
0x0C	BKP_DR3	备份数据寄存器3	0x0000 0000
0x010	BKP_DR4	备份数据寄存器4	0x0000 0000
0x014	BKP_DR5	备份数据寄存器5	0x0000 0000
0x018	BKP_DR6	备份数据寄存器6	0x0000 0000
0x01C	BKP_DR7	备份数据寄存器7	0x0000 0000
0x020	BKP_DR8	备份数据寄存器8	0x0000 0000
0x024	BKP_DR9	备份数据寄存器9	0x0000 0000
0x028	BKP_DR10	备份数据寄存器10	0x0000 0000
0x02C	BKP_RTCCR	RTC时钟校准寄存器	0x0000 0000
0x030	RTC_CR	备份控制寄存器	0x0000 0000
0x034	RTC_CSR	备份控制/状态寄存器	0x0000 0000
0x038	Revered	Revered	0xFFFF FFFF
0x03C	Revered	Revered	0xFFFF FFFF
0x040	BKP_DR11	备份数据寄存器11	0x0000 0000
0x044	BKP_DR12	备份数据寄存器12	0x0000 0000
0x048	BKP_DR13	备份数据寄存器13	0x0000 0000
0x04C	BKP_DR14	备份数据寄存器14	0x0000 0000
0x050	BKP_DR15	备份数据寄存器15	0x0000 0000
0x054	BKP_DR16	备份数据寄存器16	0x0000 0000
0x058	BKP_DR17	备份数据寄存器17	0x0000 0000
0x05C	BKP_DR18	备份数据寄存器18	0x0000 0000
0x060	BKP_DR19	备份数据寄存器19	0x0000 0000
0x064	BKP_DR20	备份数据寄存器20	0x0000 0000
0x068	BKP_DR21	备份数据寄存器21	0x0000 0000



0x06C	BKP_DR22	备份数据寄存器22	0x0000 0000
0x070	BKP_DR23	备份数据寄存器23	0x0000 0000
0x074	BKP_DR24	备份数据寄存器24	0x0000 0000
0x078	BKP_DR25	备份数据寄存器25	0x0000 0000
0x07C	BKP_DR26	备份数据寄存器26	0x0000 0000
0x080	BKP_DR27	备份数据寄存器27	0x0000 0000
0x084	BKP_DR28	备份数据寄存器28	0x0000 0000
0x088	BKP_DR29	备份数据寄存器29	0x0000 0000
0x08C	BKP_DR30	备份数据寄存器30	0x0000 0000
0x090	BKP_DR31	备份数据寄存器31	0x0000 0000
0x094	BKP_DR32	备份数据寄存器32	0x0000 0000
0x098	BKP_DR33	备份数据寄存器33	0x0000 0000
0x09C	BKP_DR34	备份数据寄存器34	0x0000 0000
0x0A0	BKP_DR35	备份数据寄存器35	0x0000 0000
0x0A4	BKP_DR36	备份数据寄存器36	0x0000 0000
0x0A8	BKP_DR37	备份数据寄存器37	0x0000 0000
0x0AC	BKP_DR38	备份数据寄存器38	0x0000 0000
0x0B0	BKP_DR39	备份数据寄存器39	0x0000 0000
0x0B4	BKP_DR40	备份数据寄存器40	0x0000 0000
0x0B8	BKP_DR41	备份数据寄存器41	0x0000 0000
0x0BC	BKP_DR42	备份数据寄存器42	0x0000 0000

Table 7-1 BKP 寄存器列表和复位值



## 7.5 BKP 寄存器說明

### 7.5-1 备份数据寄存器 x(BKP\_DRx) (x = 1 ... 10)

地址偏移: 0x04 到 0x28, 0x40 到 0xBC 复位值: 0x0000 0000

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

Bit	标记	功能描述	复位值	读写
31:16	Reserved	-	-	-
15:0	D[15:0]	备份数据 这些位可以被用来写入用户数据。 注意: BKP_DRx 寄存器不会被系统复位、电源复位、从待机模式唤醒所复位。 它们可以由备份域复位来复位或(如果侵入检测引脚 TAMPER 功能被开启时)由侵入引脚事件复位。	0x0	RW



### 7.5-2 RTC 时钟校准寄存器(BKP\_RTCCR)

地址偏移: 0x2C 复位值: 0x0000 0000

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

Bit	标记	功能描述	复位值	读写
31:10	Reserved	-	-	-
9	ASOS	闹钟或秒输出选择(Alarm or second output selection)  当设置了 ASOE 位, ASOS 位可用于选择在 TAMPER 引脚上输出的是 RTC 秒脉冲还是闹钟脉冲信号。  0: 输出 RTC 闹钟脉冲 1: 输出秒脉冲  注: 该位只能被后备区的复位所清除	0	RW
8	ASOE	允许输出闹钟或秒脉冲(Alarm or second output enable)  根据 ASOS 位的设置, 该位允许 RTC 闹钟或秒脉冲输出到 TAMPER 引脚上。  输出脉冲的宽度为一个 RTC 时钟的周期。设置了 ASOE 位时不能开启 TAMPER 的功能。  注: 该位只能被后备区的复位所清除	0	RW
7	CCO	校准时钟输出(Calibration clock output)  0: 无影响 1: 此位置 1 可以在侵入检测引脚输出经 64 分频后的 RTC 时钟。当 CCO 位置 1 时, 必须关闭侵入检测功能以避免检测到无用的侵入信号。	0	RW



Bit	标记	功能描述	复位值	读写
		注：当 VDD 供电断开时，该位被清除。		
6:0	CAL[6:0]	校准值(Calibration value) 位 6:0 校准值表示在每 220 个时钟脉冲内将有多少个时钟脉冲被跳过。这可以用来对 RTC 进行校准，以 1000000/220ppm 的比例减慢时钟。 RTC 时钟可以被减慢 0~121ppm。	0	RW

ZBT CONFIDENTIAL



### 7.5-3 备份控制寄存器(BKP\_CR)

偏移地址: 0x30 复位值: 0x0000 0000

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

Bit	标记	功能描述	复位值	读写
31:2	Reserved	-	-	-
1	TPAL	侵入检测 TAMPER 引脚有效电平(TAMPER pin active level) 0: 侵入检测 TAMPER 引脚上的高电平会清除所有数据备份寄存器(如果 TPE 位为 1) 1: 侵入检测 TAMPER 引脚上的低电平会清除所有数据备份寄存器(如果 TPE 位为 1)	0	RW
0	TPE	启动侵入检测 TAMPER 引脚(TAMPER pin enable) 0: 侵入检测 TAMPER 引脚作为通用 IO 口使用 1: 开启侵入检测引脚作为侵入检测使用	0	RW



### 7.5-4 备份控制/状态寄存器(BKP\_CSR)

偏移地址: 0x34 复位值: 0x0000 0000

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

Bit	标记	功能描述	复位值	读写
31:10	Reserved	-	-	-
9	TIF	<p>TIF: 侵入中断标志(Tamper interrupt flag)  当检测到有侵入事件且 TPIE 位为 1 时，此位由硬件置 1。通过向 CTI 位写 1 来清除此标志位(同时也清除了中断)。如果 TPIE 位被清除，则此位也会被清除。  0: 无侵入中断  1: 产生侵入中断  注意：仅当系统复位或由待机模式唤醒后才复位该位</p>	0	R
8	TEF	<p>侵入事件标志(Tamper event flag)  当检测到侵入事件时此位由硬件置 1。通过向 CTE 位写 1 可清除此标志位  0: 无侵入事件  1: 检测到侵入事件  注：侵入事件会复位所有的 BKP_DRx 寄存器。只要 TEF 为 1，所有的 BKP_DRx 寄存器就一直保持复位状态。当此位被置 1 时，若对 BKP_DRx 进行写操作，写入的值不会被保存。</p>	0	R
7:3	Reserved	-	-	-
2	TPIE	<p>允许侵入 TAMPER 引脚中断(TAMPER pin interrupt enable)  0: 禁止侵入检测中断  1: 允许侵入检测中断(BKP_CR 寄存器的 TPE 位也必须被置 1)  注 1: 侵入中断无法将系统内核从低功耗模式唤醒。  注 2: 仅当系统复位或由待机模式唤醒后才复位该位。</p>	0	R/W



Bit	标记	功能描述	复位值	读写
1	CTI	CTI: 清除侵入检测中断(Clear tamper interrupt) 此位只能写入, 读出值为 0。 0: 无效 1: 清除侵入检测中断和 TIF 侵入检测中断标志	0	R/W
0	CTE	CTE: 清除侵入检测事件(Clear tamper event) 此位只能写入, 读出值为 0。 0: 无效 1: 清除 TEF 侵入检测事件标志(并复位侵入检测器)。	0	R/W

ZBT CONFIDENTIAL



## 【8】 系统复位与时钟(RCC)

### 8.1 复位

ZB32F103 支持三种复位形式，分别为系统复位、上电复位和备份区域复位。

#### 8.1-1 系统复位

除了时钟控制器的 RCC\_CSR 寄存器中的复位标志位和备份区域中的寄存器(见图 4)以外，系统复位将复位所有寄存器至它们的复位状态。

当发生以下任一事件时，产生一个系统复位：

- (1) 外部引脚复位(NRST PIN)，低电平复位
- (2) IWDG 复位
- (3) WWDG 复位
- (4) 软件复位(SW 复位)
- (5) 低功耗管理复位

##### 8.1-1.1 软件复位

通过将 Cortex™-M3 中断应用和复位控制寄存器中的 SYSRESETREQ 位置‘1’，可实现软件复位。请参考 Cortex™-M3 技术参考手册获得进一步信息。

##### 8.1-1.2 低功耗管理复位

在以下两种情况下可产生低功耗管理复位：

1. 在进入待机模式时产生低功耗管理复位：

通过将用户选择字节中的 nRST\_STDBY 位置‘1’ 将使能该复位。这时，即使执行了进入待机模式的过程，系统将被复位而不是进入待机模式。

2. 在进入停止模式时产生低功耗管理复位：

通过将用户选择字节中的 nRST\_STOP 位置‘1’ 将使能该复位。这时，即使执行了进入停机模式的过程，系统将被复位而不是进入停机模式。

#### 8.1-2 备份域复位

备份区域拥有两个专门的复位，它们只影响备份区域(见图 4)。

当以下事件中之一发生时，产生备份区域复位。

1. 软件复位，备份区域复位可由设置备份域控制寄存器 (RCC\_BDCR)(见 6.3.9 节)中的 BDRST 位产生。
2. 在 VDD 和 VBAT 两者掉电的前提下，VDD 或 VBAT 上电将引发备份区域复位。



### 8.1-3 电源复位

当以下事件中之一发生时，产生电源复位：

#### 1. 上电/掉电复位(POR/PDR 复位)

当芯片上电以及掉电时，如果电源低于一个阈值电压，内部会产生一个 POR 信号，当电源高于该阈值电压时，释放 POR 信号。POR 信号会把芯片的寄存器、控制信号复位。本产品有两个电压区域，VDD 区域以及 Vcore 区域，所以有两个 POR：VDD 区域的 POR 以及 Vcore 区域的 POR。

#### 2. 从待机模式中返回

电源复位将复位除了备份区域外的所有寄存器。(见图 6-1)

图中复位源将最终作用于 RESET 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000\_0004。

芯片内部的复位信号会在 NRST 引脚上输出，脉冲发生器保证每一个(外部或内部)复位源都能有至少 20 $\mu$ s 的脉冲延时；当 NRST 引脚被拉低产生外部复位时，它将产生复位脉冲。

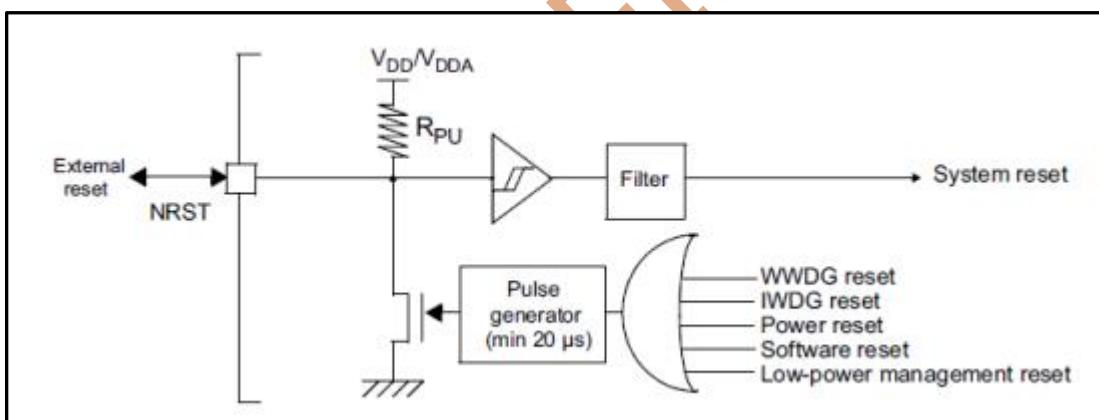


Figure 8-1 复位来源示意图



## 8.2 系统时钟

时钟控制模块主要控制系统时钟以及外设时钟，可以配置不同的时钟源作为系统时钟，可以配置不同的系统时钟分频，可以启动或禁用外设时钟。另外为了确保高精度，内部时钟都具有校准功能。本产品支持以下个不同的时钟源作为系统时钟：

- (1) 内部高速 RC 时钟 HIRC(HIRCmax=24Mhz, 4MHz[默认主频])
- (2) 外部低速晶振时钟 LXT(32.768KHz)
- (3) 内部低速 RC 时钟 SIRC(38.4KHz 与 32.768KHz 可配置)
- (4) 外部高速晶振时钟 HXT(4MHz~24MHz)
- (5) PLL 时钟 (提供 Max 时钟 : 192MHz)

这些设备有以下 2 种二级时钟源：

- 40kHz 低速内部 RC，可以用于驱动独立看门狗和通过程序选择驱动 RTC。RTC 用于从停机/待机模式下自动唤醒系统。
- 32.768kHz 低速外部晶体也可用来通过程序选择驱动 RTC(RTCCLK)。

当不被使用时，任一个时钟源都可被独立地启动或关闭，由此优化系统功耗



### 8.2-1 系统时钟模块图

ZB32F103 系统时钟控制模块图如下：

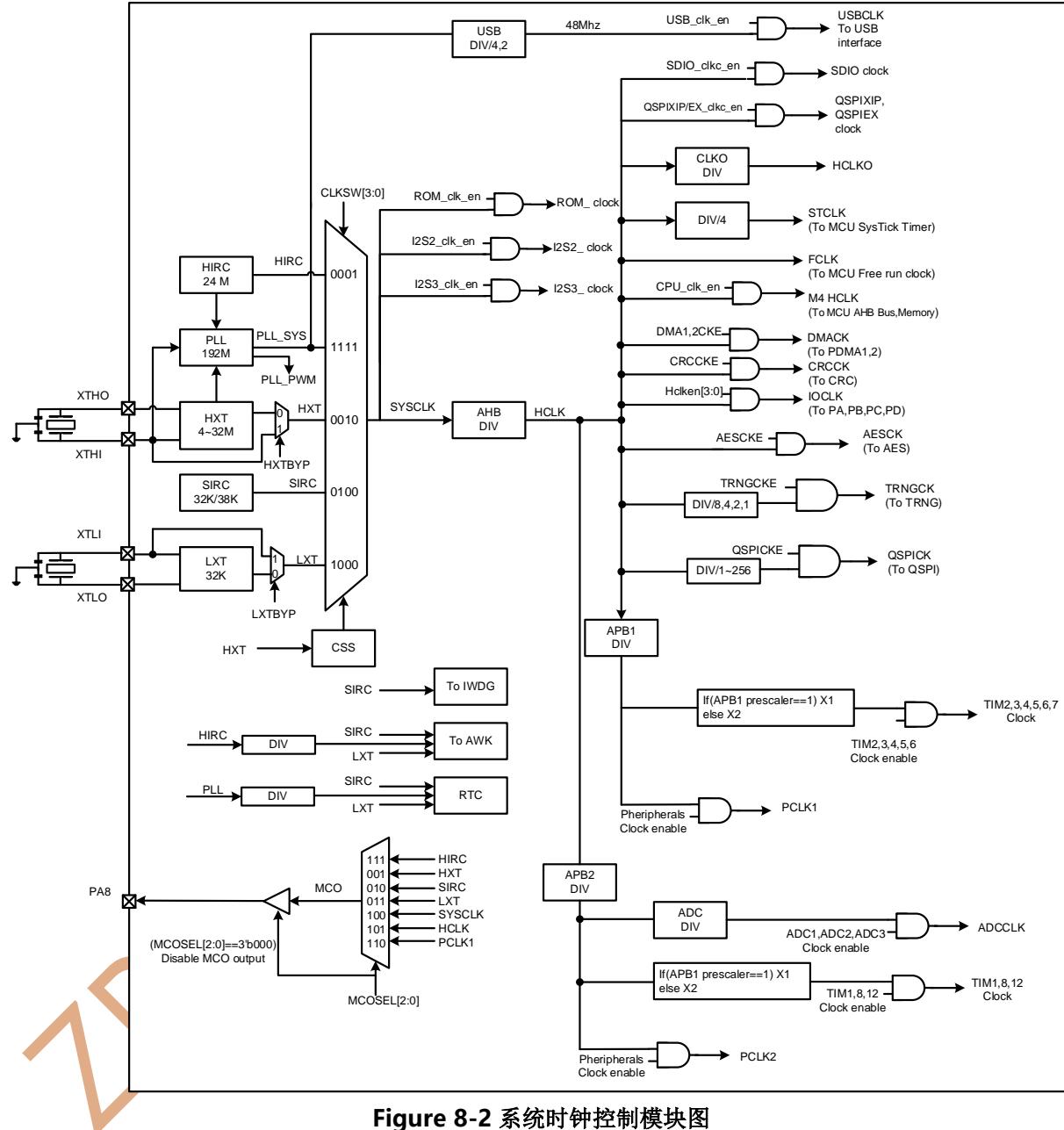


Figure 8-2 系统时钟控制模块图



### 8.2-2 内部高速 RC 时钟 (HIRC)

默认的系统时钟是内部高速 RC 时钟，在芯片上电或复位后即开始工作，通过寄存器 RCC\_HIRCCR[11:0]来配置内部高速时钟的频率，给出精确 4MHz、8MHz、16MHz、22.12MHz、24MHz 的频率值。因为内部高速时钟启动快，约 3us，为了让系统更为快速的响应外部中断，系统在从深度休眠模式被唤醒时，可以选择使用该时钟源作为系统时钟。

### 8.2-3 内部低速 RC 时钟 (SIRC)

内部低速 RC 时钟频率可配置成 38.4KHz、32.768KHz，在低速及对精度要求不高的应用场景下，可选择该时钟源作为系统时钟。

### 8.2-4 外部高速晶振时钟 (HXT)

外部高速晶振时钟需根据用户系统需求外接一个 4MHz~24MHz 的高速晶振。

外部晶振时钟可以选择两种输入方式：

- (1) HXT 外部晶体/陶瓷谐振器
- (2) HXT 用户外部时钟

为了减少时钟输出的失真和缩短启动稳定时间，晶体/陶瓷谐振器和负载电容器必须尽可能地靠近振荡器引脚。负载电容值必须根据所选择的振荡器来调整。

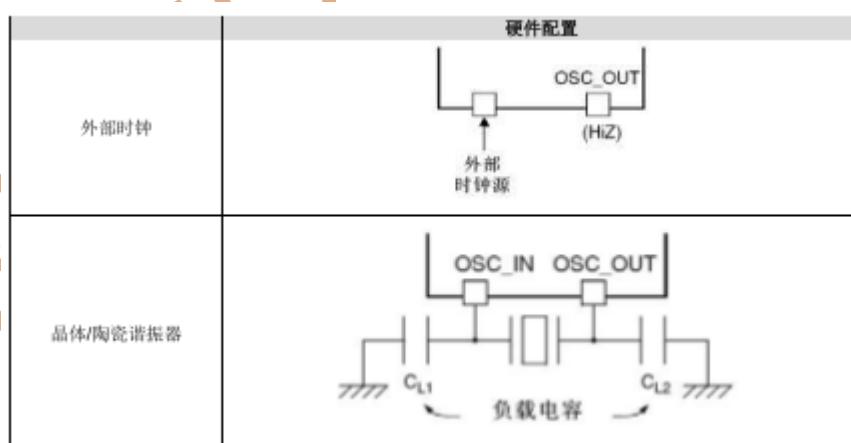


Figure 8-3 HXT/LXT 时钟源



### 8.2-5 外部低速晶振时钟 (LXT)

外部低速晶振时钟需外接一个 32.768KHz 的低功耗晶振，具有超高精度以及低功耗。超低功耗模式下工作的模块都可以选择此时钟源作为时钟信号。

ZBT CONFIDENTIAL



## 8.2-6 PLL 时钟

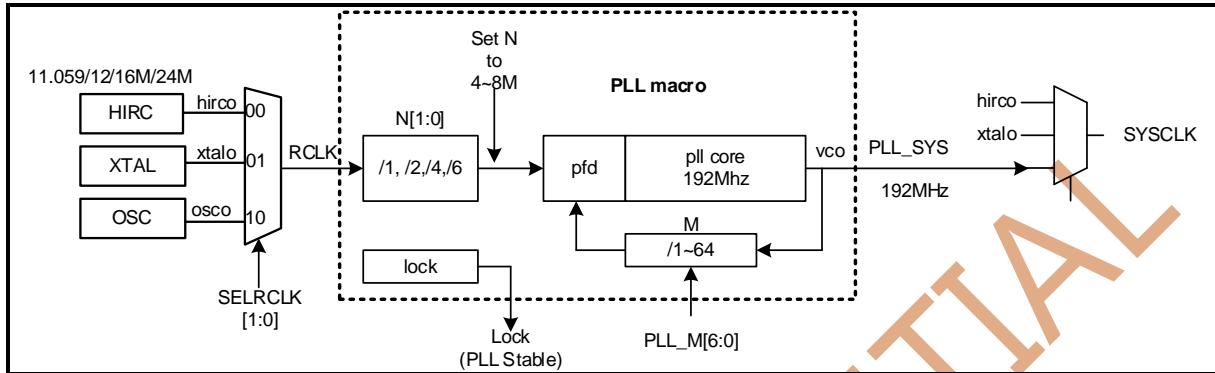


Figure 8-4 PLL 时钟源

内部 PLL 可以用来倍频 HIRC 的输出时钟或 XTAL 晶体输出时钟。参考图 7-4 和时钟控制寄存器。

PLL 的设置：

- PLL 时钟输出 (Set PLL=192Mhz) :  $PLL_O = (RCLK/N) * M$ , 可参考 Table7-2.
- 需 設定 N 值 Let  $RCLK/N = 4\sim8MHz$ .
- 参考时钟(RCLK)：范围 4~48 MHz.
- 当 Lock 为 High 时 : PLL 输出 稳定的时钟 (Duty ~50%) , 可切换时钟给 CPU 与外设使用.

RCLK (HIRC/XTAL/OSC)	PLL_N [1:0]	PLL_M[1*] [6:0]	PLL_O $=RCLK*(M/N)$
24MHz			
22.118MHz			
16MHz			
8 MHz			
4 MHz			

RCLK (HIRC/XTAL/OSC)	PLL_N [1:0]	M [6:0]	PLL_O $=RCLK*(M/N)$
24MHz	2' b11(6)	7' d32	128MHz



22.118MHz	2' b11(6)	7' d35	129.02 MHz
4MHz	2' b00(1)	7' d32	128MHz
24MHz	2' b11(6)	7' d48	192MHz
22.118MHz	2' b11(6)	7' d52	191.69 MHz

Table 8-1 PLL 参数表

Note: 1\* 建议设置 PLL\_M=48

### 8.2-7 系统时钟启动过程

上述四种时钟源都有一个启动稳定的时间，时钟源使能后都会等待一段稳定时间后，再把时钟切换给系统使用，芯片上电后，系统使用 4MHz 的内部高速时钟作为启动时钟，启动完成后用户可以根据自己的需要来修改高速时钟的频率以及切换时钟源。

外部低速晶振时钟需外接一个 32.768KHz 的低功耗晶振，具有超高精度以及低功耗。超低功耗模式下工作的模块都可以选择此时钟源作为时钟信号。

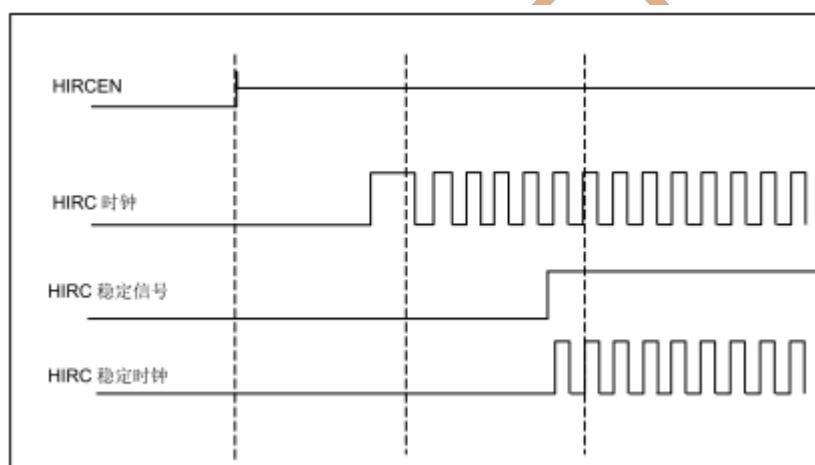


Figure 8-5 内部高速时钟启动示意图



## 8.2-8 系统时钟切换

时钟源的切换是由寄存器 `RCC_SYSCLKSEL[3:0]` 来控制。在双时钟模式下，当系统时钟从当前时钟切换到目标时钟时，必须按照一定的流程来实现，否则就会出现异常。

### 8.2-8.1 内部高速切换到外部低速

以从 HIRC(内部高速 RC 时钟)切换到 LXT(外部低速晶振时钟)为例，具体流程如下：

- (1) 通过 `RCC_LXTCR.LXTPORT` 位来配置要切换的时钟 LXT 使用的引脚为模拟引脚，或者通过对于引脚的 `GPIOx_AFR=0x0F` 来配置为模拟功能
- (2) 写 `RCC_LXTCR.LXTEN` 使能 LXT 时钟，
- (3) 等待寄存器 `RCC_LXTCR.LXTRDY` 位被硬件置“1”，
- (4) 写寄存器 `RCC_SYSCLKSEL.CLKSW[3:0]` 来切换时钟
- (5) 根据需要关闭 HIRC 时钟.

### 8.2-8.2 内部高速切换到外部高速

从 HIRC(内部高速 RC 时钟)切换到 HXT(外部高速晶振时钟)为例，具体流程如下：

- (1) 通过 `RCC_SYSCLKCR.HXTPORT` 位来配置要切换的时钟 HXT 使用的引脚为模拟引脚，或者通过对于引脚的 `GPIOx_AFR=0x0F` 来配置为模拟功能
- (2) 写寄存器 `RCC_SYSCLKCR.HXTEN` 位使能 HXT 时钟，
- (3) 等待寄存器 `RCC_HXTCR.HXTRDY` 位被硬件置“1”，
- (4) 写寄存器 `RCC_SYSCLKSEL.CLKSW[3:0]` 来切换时钟，
- (5) 根据需要关闭 HIRC 时钟

注意：使用外部高速 24M 晶振时，`RCC_HXTCR.HXTSTARTUP` 稳定时间控制位设置为 0x3，使用默认配置 0x2 稳定时间可能不够。

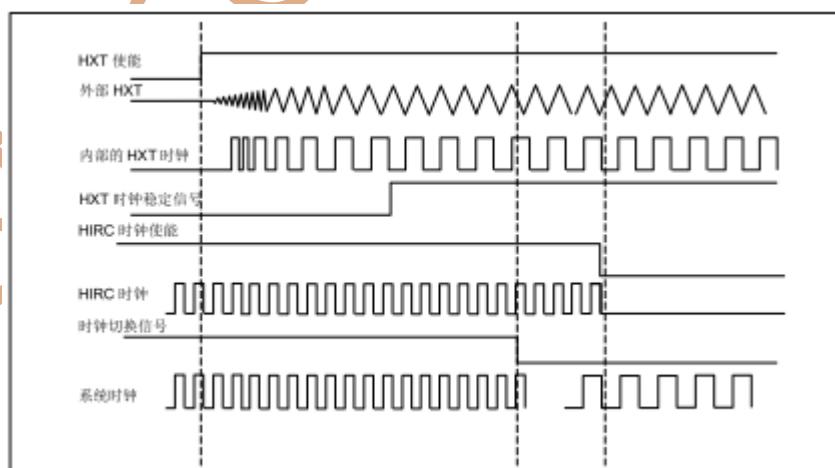


Figure 8-6 时钟切换示意图



### 8.2-8.3 内部低速切换到外部高速

从 SIRC(内部低速 RC 时钟)切换到 HXT(外部高速晶振时钟)为例，具体流程如下：

- (1) 通过 `RCC_SYSCLKCR.HXTPORT` 位来配置要切换的时钟 HXT 使用的引脚为模拟引脚，或者通过对于引脚的 `GPIOx_AFR=0x0F` 来配置为模拟功能
- (2) 写寄存器 `RCC_SYSCLKCR.HXTEN` 位使能 HXT 时钟
- (3) 等待寄存器 `RCC_HXTCR.HXTRDY` 位被硬件置“1”
- (4) 写寄存器 `RCC_SYSCLKSEL.CLKSW[3:0]` 来切换时钟
- (5) 根据需要关闭 SIRC 时钟

### 8.2-8.4 内部时钟切换到 PLL 时钟

从 内部时钟切换到 PLL 时钟为例，具体流程如下：

- (1) 选择 HIRC(24MHz) 为 RCLK : `RCC_PLLCR1.SELRCK=00`
- (2) 设置参数: `RCC_PLLCR1.PLL_N[1:0]=2' b11`, `RCC_PLLCR1.PLL_M[6:0]=48`
- (3) PLL 使能: `RCC_PLLCR1.PLLEN=1`
- (4) 查询 PLL 是否稳定：IF `RCC_PLLCR.PLLRDY=1`, 表示 PLL 时钟稳定.
- (5) 切换时钟: 写寄存器 `RCC_SYSCLKSEL.CLKSW[3:0] = '4' b1111` 来切换时钟.

### 8.2-9 系统时钟输出

微控制器允许输出时钟信号到外部 MCO 引脚。

有如下的 6 种信号可选为 MCO 时钟输出, MCO 时钟的选择由时钟输出控制寄存器(`RCC_MCOCR`)的 `MCOSEL[2:0]` 位决定

MCO	MCOSEL[2:0]
NO clock output	3' b000
HXT	3' b001
SIRC	3' b010
LXT	3' b011
<b>SYSCLK</b>	3' b100
HCLK_O	3' b101
HIRC	3' b111

微控制器系统时钟(SYSCLK)的选择:

SYSCLK	CLKSW[3:0]
HIRC	4' b0001



HXT	4' b0010
SIRC	4' b0100
LXT	4' b1000
PLL	4' b1111

### 8.2-10 系统时钟安全控制

当设定 CLKFAILEN 有效后，并且使能 CLKTRIM 的时钟监视功能后，当 HXT 时钟或 LXT 时钟停止后，系统时钟会切换到内部高速时钟。具体请参考 CLKTRIM 模块的监测功能。

### 8.2-11 IWDG 时钟

如果独立看门狗已经由硬件选项或软件启动，SIRC 振荡器将被强制在打开状态，并且不能被关闭。在 SIRC 振荡器稳定后，时钟供给给 IWDG。

### 8.2-12 RTC 时钟

RTCCLK 时钟源可以由 HXT 分频、LXT 或 SIRC 时钟提供。

### 8.2-13 AWK 时钟

AWKCLK 时钟源可以由 HXT 分频、LXT 或 SIRC 时钟提供。

### 8.2-14 低功耗模式

APP 外设时钟以及部分 AHB 外设时钟可以用软件禁止。睡眠模式停止 MCU 时钟，在 MCU 睡眠中存储器接口时钟(Flash 和 RAM 接口)被停止。

当配置了 SYSCON\_CFGR0.DBGDLSP\_DIS 后，那么 CPU 在相应的深度睡眠模式下也可以具有调试功能。



### 8.3 寄存器列表

这节详细描述了 RCC 控制模块的寄存器功能。

RCC 基址: **0x4002 1000**

偏移地址	名称	描述	复位值
0x00	RCC_CR	时钟控制寄存器	0x0000 0001
0x04	RCC_CFGR	时钟配置寄存器	0x0000 0000
0x08	RCC_CIR	时钟中断寄存器	0x0000 0100
0x0C	RCC_APB2RSTR	APB2 外设复位寄存器	0x0000 0000
0x10	RCC_APB1RSTR	APB1 外设复位寄存器	0x0000 0000
0x14	RCC_AHBENR	AHB外设时钟使能寄存器	0x0000 0000
0x18	RCC_APB2ENR	APB2 外设时钟使能寄存器	0x0000 0000
0x1C	RCC_APB1ENR	APB1 外设时钟使能寄存器	0x0000 00A0
0x20	RCC_BDCR	备份域控制寄存器	0x0000 0000
0x24	RCC_CSR	控制/状态寄存器	0x0000 0000
0x28	RCC_HIRCCR	内部高速 RC 振荡器控制寄存器	0x0000 1312
0x2C	RCC_HXTCR	外部高速晶体振荡器控制寄存器	0x0000 0027
0x30	RCC_SIRCCR	内部低速 RC 振荡器控制寄存器	0x0000 007F
0x34	RCC_LXTCR	外部低速晶体振荡器控制寄存器	0x0000 042F
0x38	Revered	Revered	0xxxxx xxxx
0x3C	RCC_STICKCR	SysTick Timer 周期校准寄存器	0x0100 9C3F
0x40	RCC_SWDIOCR	端子特殊功能选择寄存器	0x0000 0001
0x44	RCC_PERIRST	周边模块复位控制寄存器	0x0000 0000
0x48	RCC_RTCRST	RTC 控制寄存器	0x0000 0000
0x4C	RCC_MASK_INT_T	MASK中断时间控制寄存器	0x0000 0004
0x50	RCC_LPM	LPM控制寄存器	0x0000 0000
0x54	RCC_AHBRSTR	AHB外设复位控制1寄存器	0x0000 0000
0x58	RCC_PLLCR1	PLL控制寄存器1	0x0000 9200
0x5C	RCC_PLLCR2	PLL控制寄存器2	0x0000 0180
0x60	RCC_UNLOCK	寄存器写保护	0x0000 0000

Table 8-2 RCC 寄存器列表和复位值



## 8.4 寄存器说明

### 8.4-1 时钟控制寄存器(RCC\_CR)

地址偏移: 0x00

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
WKBYHRC	保留							CLKFAILEN	CSSON	HXT PORT	HXT BYP	LXTEN	PLLEN	SRCEN	HXTEN	HIRC EN
R/W	保留							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15	WKBYHRC	0: 从Deep Sleep 唤醒, system clock 来源怎么进怎么出。 1: 从Deep Sleep 唤醒是使用HIRC 开始唤醒, 硬件自动 enable HIRC, 并且system clock 自动切换到HIRC, 原时钟继续开启。	0	R/W
14:9	-	保留	0	-
8	CLKFAILEN	时钟失效检测使能控制 0: 时钟失效检测禁止 1: 时钟失效检测使能, 当检测到允许时钟失效, 自动切换系统时钟到HIRC	0	R/W
7	CSSON	<b>CSSON: 时钟安全系统使能 (Clock security system enable)</b> 由软件置'1'或清零以使能时钟监测器。 0: 时钟监测器关闭; 1: 如果外部4-24MHz振荡器就绪, 时钟监测器开启。	0	R? W
6	HXTPORT	OSCIN/OSCOUT 端子配置 0: GPIO 复用功能模式(AFR 决定其功能)。 1: HXT 端子模式(模拟功能)。	0	R/W
5	HXTBYP	外部高速时钟输入选择 0: HXT 内部振荡模块未被旁路, 与OSC_IN/OSC_OUT 相连 1: HXT 内部振荡模块被旁路, HXT 从端子OSCIN 直接输入	0	R/W



位	标记	功能描述	复位值	读写
4	LXTEN	外部 32K 晶振LXT 使能信号 0: 关闭 1: 使能 当系统时钟选择该时钟时, 不能关闭	0	R/W
3	PLLEN	PLL使能信号 0: 关闭 1: 使能 注意: 当系统进入 Deep Sleep 前须先将CPU时钟换成HIRC, 因高速时钟会自动关闭。	0	R/W
2	SIRCEN	内部低速时钟SIRC(32KHz) 使能信号。 0: 关闭 1: 使能 当系统时钟选择该时钟时, 不能关闭	0	R/W
1	HXTEN	外部 4M~24M 晶振 HXTOSC 使能信号。 0: 关闭 1: 使能 注意: 1. 当系统进入 Deep Sleep, 此高速时钟会自动关闭。 备注: 使用时, 与该晶振连接的两个外部端口必须设置成模拟端口 (配置 RCC_SYSCLKCR.HXTPORT 寄存器)。 2. 当 HXT 停止检出时, 此位会被硬件清 0 当系统时钟选择该时钟时, 不能关闭。	0	R/W
0	HIRCEN	内部高速时钟HIRC 使能信号。 0: 关闭 1: 使能 注意: 当系统进入 deep Sleep, 此高速时钟会自动关闭。 当 HXT 停止检出时, 如果系统时钟选择为 HXT, 且 CLKFAIL_EN 使能, HIRC_EN 会由硬件自动置 1。 当系统时钟选择该时钟时, 不能关闭	1	R/W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后, 才能写该寄存器。



#### 8.4-2 时钟配置寄存器(RCC\_CFGR)

地址偏移: 0x04 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		Reserved						MCO					Reserved	USBPRE	HIRCDIV [1:0]
r	r	r	r	r	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE		PPRE2			PPRE1			HPRE					SWS	SW	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw

位	标记	功能描述	复位值	读写
31:27	Reserved		0x0	r
26:24	MCO[2:0]	<b>MCO:</b> 微控制器时钟输出 (Microcontroller clock output) 000: No clock 001:HXT 010:SIRC 011:LXT 100: System clock (SYSCLK) selected 101: HCK clock selected 110: PCLK1 111: HIRCO 选择	0x0	rw
23	Reserved	Reserved	0x0	r
22	USBPRE	設定 USB clock 為 48 MHz k. 在 RCC_APB1ENR 寄存器中使能全速 OTG 时钟之前，必须保证该位已经有效 0: USB clock =PLL_SYS/4 1: USB clock =PLL_SYS/2 Note : This bit can't be reset if the USB clock is enabled.	0x0	rw
21:20	HIRCDIV[1:0]	HIRC 除频时钟 为HIRCO. 2' b00 :HIRCO= HIRC	0x0	rw



位	标记	功能描述	复位值	读写
		2' b01 : HIRCO=HIRC/2 2' b10: HIRCO=HIRC/3 2' b11: HIRCO= HIRC/6		
19:16	Reserved	Reserved	0x0	rw
15:14	ADCPRE	由软件置' 1' 或清' 0' to select the frequency of the clock to the ADCs. 00: PCLK2 divided by 2 01: PCLK2 divided by 4 10: PCLK2 divided by 6 11: PCLK2 divided by 8	0x0	rw
13:11	PPRE2	<b>PPRE2[2:0]:</b> 高速 APB 预分频(APB2) (APB high-speed prescaler (APB2)) 由软件置' 1' 或清' 0' 来控制高速 APB2 时钟(PCLK2)的预分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频	0x0	rw
10:8	PPRE1	<b>PPRE1[2:0]:</b> 低速 APB 预分频(APB1) (APB low-speed prescaler (APB1)) 由软件置' 1' 或清' 0' 来控制低速 APB1 时钟(PCLK1)的预分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频  警告：软件必须保证 APB1 时钟频率不超过 36MHz。	0x0	rw
7:4	HPRE	HPRE[3:0]: AHB 预分频 (AHB Prescaler) 由软件置' 1' 或清' 0' 来控制 AHB 时钟的预分频系数。 0xxx: SYSCLK not divided 1000: SYSCLK divided by 2 1001: SYSCLK divided by 4 1010: SYSCLK divided by 8 1011: SYSCLK divided by 16	0x0	rw



位	标记	功能描述	复位值	读写
		1100: SYSCLK divided by 64 1101: SYSCLK divided by 128 1110: SYSCLK divided by 256 1111: SYSCLK divided by 512		
3:0	CLKSWS	System Clock Source select Satus. 0001: HIRCO 选择 0010: HXT 选择 0100: SIRC 选择 1000: LXT 选择 1111: PLL 选择  注意: 当 HXT 停止检出时, 如果系统时钟选择为 HXT, 且 CLKFAIL_EN 使能, HIRC_EN 会由硬件自动置 1。系统时钟自动选择 HIRC。	0x0	r



#### 8.4-3 时钟中断寄存器(RCC\_CIR)

地址偏移: 0x08 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r	rw	rw	rw	rw	rw	rw	rw	w	rw	rw	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				PLL RDYIE	E	HXT RDYI	HIRC RDYI	LXT RDYIE	SIRC RDYI	CSSF	Reserved		PLL RDYF	HXT RDYF	HIRC RDYF
r	r	r	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位	标记	功能描述	复位值	读写
31:24	Reserved	保留, 始终读为 0。	0x0	r
23	CSSC	<b>CSSC:</b> 清除时钟安全系统中断 (Clock security system interrupt clear) 由软件置' 1' 来清除 CSSF 安全系统中断标志位 CSSF。 0: 无作用; 1: 清除 CSSF 安全系统中断标志位。	0x0	w
22:21	Reserved		0x0	r
20	PLLRDYC	<b>PLLRDYC:</b> 清除 PLL 就绪中断 (PLL ready interrupt clear) 由软件置' 1' 来清除 PLL 就绪中断标志位 PLLRDYF。 0: 无作用; 1: 清除 PLL 就绪中断标志位 PLLRDYF。	0x0	w
19	HXTRDYC	<b>HXTRDYC:</b> 清除外部高速 RC 振荡器就绪中断 0: 无作用; 1: 清除 HXT 就绪中断标志位 HXTRDYF。	0x0	w
18	HIRCRDYC	<b>HIRCRDYC:</b> 清除内部高速 RC 就绪中断 0: 无作用; 1: 清除 HIRC 就绪中断标志位 HIRCRDYF。	0x0	w
17	LXTRDYC	<b>LXTRDYC:</b> 清除外部低速 RC 振荡器 E 就绪中断 0: 无作用;	0x0	w



位	标记	功能描述	复位值	读写
		1: 清除 LXT 就绪中断标志位 LXTRDYF。		
16	SIRCRDYC	<b>SIRCRDYC:</b> 清除内部低速 RC 就绪中断 0: 无作用; 1: 清除 SIRC 就绪中断标志位 SIRCRDYF。	0x0	w
15:13	Reserved		0x0	r
12	PLLRDYIE	<b>PLLRDYIE:</b> PLL 就绪中断使能 (PLL ready interrupt enable) 由软件置'1' 或清'0' 来使能或关闭 PLL 就绪中断。 0: PLL 就绪中断关闭; 1: PLL 就绪中断使能。	0x0	rw
11	HXTRDYIE	<b>HXTRDYIE:</b> 外部高速 RC 振荡器就绪中断使能 由软件置'1' 或清'0' 来使能或关闭外部 3-25MHz 振荡器就绪中断。 0: HXT 就绪中断关闭; 1: HXT 就绪中断使能。	0x0	rw
10	HIRCRDYIE	<b>HIRCRDYIE:</b> 内部高速 RC 就绪中断使能 0: HIRC 就绪中断关闭; 1: HIRC 就绪中断使能。	0x0	rw
9	LXTRDYIE	<b>LXTRDYIE:</b> 外部低速 RC 振荡器就绪中断使能 由软件置'1' 或清'0' 来使能或关闭外部 32kHz RC 振荡器就绪中断。 0: LXT 就绪中断关闭; 1: LXT 就绪中断使能。	0x0	rw
8	SIRCRDYIE	<b>SIRCRDYIE:</b> 内部低速 RC 就绪中断使能 由软件置'1' 或清'0' 来使能或关闭内部 40kHz RC 振荡器就绪中断。 0: SIRC 就绪中断关闭; 1: SIRC 就绪中断使能。	0x0	rw
7	CSSF	<b>CSSF:</b> 时钟安全系统中断标志 (Clock security system interrupt flag) 在外部 4-25MHz 振荡器时钟出现故障时, 由硬件置'1'。 由软件通过置'1' CSSC 位来清除。 0: 无 HXT 时钟失效产生的安全系统中断; 1: HXT 时钟失效导致了时钟安全系统中断。	0x0	r
6:5	Reserved		0x0	r
4	PLLRDYF	<b>PLLRDYF:</b> PLL 就绪中断标志 (PLL ready interrupt flag) 在 PLL 就绪且 PLLRDYIE 位被置'1' 时, 由硬件置'1'。 由软件通过置'1' PLLRDYC 位来清除。 0: 无 PLL 上锁产生的时钟就绪中断; 1: PLL 上锁导致时钟就绪中断。	0x0	r



位	标记	功能描述	复位值	读写
3	HXTRDYF	<b>HXTRDYF:</b> 外部高速 RC 振荡器就绪中断标志 (HXT ready interrupt flag) 在外部低速时钟就绪且 HXTRDYIE 位被置' 1' 时, 由硬件置' 1'。 由软件通过置' 1' HXTRDYC 位来清除。 0: 无外部 4-24MHz 振荡器产生的时钟就绪中断; 1: 外部 4-24MHz 振荡器导致时钟就绪中断。	0x0	r
2	HIRCRDYF	<b>HIRCRDYF:</b> 内部高速 RC 就绪中断标志 在内部高速时钟就绪且 HIRCRDYIE 位被置' 1' 时, 由硬件置' 1'。 由软件通过置' 1' HIRCRDYC 位来清除。 0: 无内部 24MHz RC 振荡器产生的时钟就绪中断; 1: 内部 24MHz RC 振荡器导致时钟就绪中断。	0x0	r
1	LXTRDYF	<b>LXTRDYF:</b> 外部低速 RC 振荡器就绪中断标志 在外部低速时钟就绪且 LXTRDYIE 位被置' 1' 时, 由硬件置' 1'。 由软件通过置' 1' LXTRDYC 位来清除。 0: 无外部 32kHz 振荡器产生的时钟就绪中断; 1: 外部 32kHz 振荡器导致时钟就绪中断。	0x0	r
0	SIRCRDYF	<b>SIRCRDYF:</b> 内部低速 RC 就绪中断标志 (SIRC ready interrupt flag) 在内部低速时钟就绪且 SIRCRDYIE 位被置' 1' 时, 由硬件置' 1'。 由软件通过置' 1' SIRCRDYC 位来清除。 0: 无内部 32kHz RC 振荡器产生的时钟就绪中断; 1: 内部 32kHz RC 振荡器导致时钟就绪中断。	0x0	r



#### 8.4-4 APB2 外设复位寄存器(RCC\_APB2RSTR)

地址偏移: 0x0C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3RST	USART1R	TIM1ARS	SPI1RST	TIM1RST	ADC2RST	ADC1RST	Reserved				IOPDRST	IOPCRST	IOPBRST	IOPARST	AFIORST
rw	rw	rw	rw	rw	rw	rw	r	r	r	rw	rw	rw	rw	r	rw

位	标记	功能描述	复位值	位
31:16	Reserved		0x0	r
15	ADC3RST	由软件置'1'或清'0'. 0: No effect 1: Reset ADC 3 interface	0x0	rw
14	USART1RST	由软件置'1'或清'0'. 0: No effect 1: Reset USART1	0x0	rw
13	TIM1ARST	由软件置'1'或清'0'. 0: No effect 1: Reset TIM1A timer	0x0	rw
12	SPI1RST	由软件置'1'或清'0'. 0: No effect 1: Reset SPI1 timer	0x0	rw
11	TIM1RST	由软件置'1'或清'0'. 0: No effect 1: Reset TIM1 timer	0x0	rw
10	ADC2RST	由软件置'1'或清'0'. 0: No effect 1: Reset ADC 2 interface	0x0	rw



位	标记	功能描述	复位值	位
9	ADC1RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset ADC 1 interface	0x0	rw
8:7	Reserved		0x0	r
6	IOPERST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset IO port E	0x0	rw
5	IOPDRST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset IO port D	0x0	rw
4	IOPCRST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset IO port C	0x0	rw
3	IOPBRST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset IO port B	0x0	rw
2	IOPARST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset IO port A	0x0	rw
1	Reserved		0x0	r
0	AFIORST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset Alternate Function	0x0	rw



#### 8.4-5 APB1 外设复位寄存器(RCC\_APB1RSTR)

地址偏移: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Reserved	TIM2RST	TIM2ARS	0
r	r	r	rw	rw	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
w					w			w	w										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	TIM2BRS	TIM2CRS	TIM2CRS	TIM2CRS
SPI3RST	SPI2RST	Reserved		Reserved		Reserved		Reserved		TIM11RST	TIM10RST								
rw	rw	r	r	rw	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:30	Reserved		0x0	r
29	DACRST	由软件置'1' 或清'0'. 0: No effect 1: Reset DAC	0x0	rw
28	PMURST	由软件置'1' 或清'0'. 0: No effect 1: Reset PMU	0x0	rw
27	BKPRST	由软件置'1' 或清'0'. 0: No effect 1: Reset backup interface	0x0	rw
26	Reserved		0x0	r
25	CANRST	由软件置'1' 或清'0'. 0: No effect 1: Reset CAN	0x0	rw
24:23	Reserved		0x0	r
22	I2C2RST	由软件置'1' 或清'0'.	0x0	rw



位	标记	功能描述	复位值	读写
		0: No effect 1: Reset I2C2		
21	I2C1RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset I2C1	0x0	rw
20	UART5RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset UART5	0x0	rw
19	UART4RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset UART4	0x0	rw
18	USART3RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset USART3	0x0	rw
17	USART2RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset USART2	0x0	rw
16	Reserved		0x0	r
15	SPI3/I2S3RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset SPI3	0x0	rw
14	SPI2/I2S2RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset SPI2	0x0	rw
13:12	Reserved		0x0	r
11	WWDGRST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset window watchdog	0x0	rw
10:6	Reserved		0x0	r
5	TIM11RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset TIM11	0x0	rw
4	TIM10RST	由软件置' 1' 或清' 0' .	0x0	rw



位	标记	功能描述	复位值	读写
		0: No effect 1: Reset TIM10		
3	TIM2CRST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset TIM2C	0x0	rw
2	TIM2BRST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset TIM2B	0x0	rw
1	TIM2ARST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset TIM2A	0x0	rw
0	TIM2RST	由软件置' 1' 或清' 0' . 0: No effect 1: Reset TIM2	0x0	rw



#### 8.4-6 AHB 外设时钟使能寄存器(RCC\_AHBENR)

地址偏移: 0x14

复位值: 0x0000 021C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SDIOEN	QSPIX1PE	QSPIEN	AESEN	CRCEN	RNGEN	ROM	SRAM2E	SRAM1E	DMA2EN	DMA1EN
r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw

位	标记	功能描述	复位值	读写
31:11	Reserved		0x0	r
11	USBEN	USB Host 时钟使能/关闭. 0: SDIO clock disabled 1: SDIO clock enabled	0x0	rw
10	SDIOEN	SDIO 时钟使能/关闭. 0: SDIO clock disabled 1: SDIO clock enabled	0x0	rw
9	QSPXIEN	QSPIEX 时钟使能/关闭. 0: QSPI clock disabled 1: QSPI clock enabled	0x1	r/w
8	QSPIEXEN	QSPIEX 时钟使能/关闭. 0: QSPI clock disabled 1: QSPI clock enabled	0x0	rw
7	AESEN	AES 时钟使能/关闭. 0: AES clock disabled 1: AES clock enabled	0x0	rw
6	CRCEN	CRCEN 时钟使能/关闭. 0: CRC clock disabled 1: CRC clock enabled	0x0	rw
5	RNGEN	RNGEN 时钟使能/关闭	0x0	rw



位	标记	功能描述	复位值	读写
		0: TRNG clock disabled 1: TRNG clock enabled		
4	ROMEN	ROM 时钟使能/关闭。 0: ROM clock disabled during Sleep mode 1: ROM clock enabled during Sleep mode	0x1	rw
3	SRAM2EN	SRAM2 时钟使能/关闭。 0: SRAM interface clock disabled during Sleep mode. 1: SRAM interface clock enabled during Sleep mode	0x1	r
2	SRAM1EN	SRAM1 时钟使能/关闭。 0: SRAM interface clock disabled during Sleep mode. 1: SRAM interface clock enabled during Sleep mode	0x1	rw
1	DMA2EN	DMA1 时钟使能/关闭。 0: DMA2 clock disabled 1: DMA2 clock enabled	0x0	rw
0	DMA1EN	DMA1 时钟使能/关闭。 0: DMA1 clock disabled 1: DMA1 clock enabled	0x0	rw



#### 8.4-7 APB2 外设时钟使能寄存器(RCC\_APB2ENR)

地址偏移: 0x18

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-	-	-	-	-	-	LPUARTE	LPTIM	TIM1B
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3EN	USART1E	TIM1AEN	SPI1EN	TIM1EN	ADC2EN	ADC1EN	Reserved		IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFOEN	
rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	r	rw

位	标记	功能描述	复位值	读写
31:19	Revered		0x0	r
18	LPUARTEN	LPUART 时钟使能/关闭。 0: ADC3 interface clock disabled 1: ADC3 interface clock enabled	0x0	rw
17	LPTIM	LPTIM 时钟使能/关闭。 0: ADC3 interface clock disabled 1: ADC3 interface clock enabled	0x0	rw
16	TIM1B	TIM1B 时钟使能/关闭。 0: ADC3 interface clock disabled 1: ADC3 interface clock enabled	0x0	rw
15	ADC3EN	ADC3 时钟使能/关闭。 0: ADC3 interface clock disabled 1: ADC3 interface clock enabled	0x0	rw
14	USART1EN	USART1 时钟使能/关闭。 0: USART1 clock disabled 1: USART1 clock enabled	0x0	rw
13	TIM1AEN	TIM1A 时钟使能/关闭。 0: TIM1A timer clock disabled	0x0	rw



位	标记	功能描述	复位值	读写
		1: TIM1A timer clock enabled		
12	SPI1EN	SPI1 时钟使能/关闭. 0: SPI1 clock disabled 1: SPI1 clock enabled	0x0	rw
11	TIM1EN	TIM1 时钟使能/关闭. 0: TIM1 timer clock disabled 1: TIM1 timer clock enabled	0x0	rw
10	ADC2EN	ADC2 时钟使能/关闭. 0: ADC 2 interface disabled 1: ADC 2 interface clock enabled	0x0	rw
9	ADC1EN	ADC1 时钟使能/关闭. 0: ADC 1 interface disabled 1: ADC 1 interface clock enabled	0x0	rw
8:7	Reserved		0x0	r
6	IOPEEN	IO 端口 E 时钟使能/关闭. 0: IO port E clock disabled 1: IO port E clock enabled	0x0	rw
5	IOPDEN	IO 端口 D 时钟使能/关闭. 0: IO port D clock disabled 1: IO port D clock enabled	0x0	rw
4	IOPCEN	IO 端口 C 时钟使能/关闭. 0: IO port C clock disabled 1: IO port C clock enabled	0x0	rw
3	IOPBEN	IO 端口 B 时钟使能/关闭. 0: IO port B clock disabled 1: IO port B clock enabled	0x0	rw
2	IOPAEN	IO 端口 A 时钟使能/关闭. 0: IO port A clock disabled 1: IO port A clock enabled	0x0	rw
1	Reserved		0x0	r
0	AFIOEN	辅助功能 IO 时钟使能/关闭. 0: Alternate Function IO clock disabled 1: Alternate Function IO clock enabled	0x0	rw



#### 8.4-8 APB1 外设时钟使能寄存器(RCC\_APB1ENR)

地址偏移: 0x1C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
-	-	DACEN	PMUEN	BKPN	PCA	CANEN	Reserved	USBEN	I2C2EN	I2C1EN	UART5E	UART4E	USART3E	USART2E	Reserved
r	r	rw	rw	rw	r/w	rw	r	rw	rw	rw	rw	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3EN	SPI2EN	-	-	WWDG_E	-	-	-	-	-	TIM11EN	TIM10EN	TIM2CEN	TIM2BEN	TIM2AEN	TIM2EN
rw	rw	r	r	rw	r	r	r	r	r	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:30	Reserved		0x0	r
29	DACEN	时钟使能/关闭。 0: DAC interface clock disabled 1: DAC interface clock enable	0x0	rw
28	PMUEN	时钟使能/关闭。 0: PMU clock disabled 1: PMU clock enable	0x0	rw
27	BKPN	时钟使能/关闭。 0: Backup interface clock disabled 1: Backup interface clock enabled	0x0	rw
26	PCA	时钟使能/关闭。 0: PCA clock disabled 1: PCA clock enabled	0x0	r
25	CANEN	时钟使能/关闭。 0: CAN clock disabled 1: CAN clock enabled	0x0	rw
24	Reserved		0x0	r



位	标记	功能描述	复位值	读写
23	USBEN	时钟使能/关闭。 0: USB clock disabled 1: USB clock enabled	0x0	rw
22	I2C2EN	时钟使能/关闭。 0: I2C2 clock disabled 1: I2C2 clock enabled	0x0	rw
21	I2C1EN	时钟使能/关闭。 0: I2C1 clock disabled 1: I2C1 clock enabled	0x0	rw
20	UART5EN	时钟使能/关闭。 0: UART5 clock disabled 1: UART5 clock enabled	0x0	rw
19	UART4EN	时钟使能/关闭。 0: UART4 clock disabled 1: UART4 clock enabled	0x0	rw
18	USART3EN	时钟使能/关闭。 0: USART3 clock disabled 1: USART3 clock enabled	0x0	rw
17	USART2EN	时钟使能/关闭。 0: USART2 clock disabled 1: USART2 clock enabled	0x0	rw
16	Reserved		0x0	r
15	SPI3EN	时钟使能/关闭。 0: SPI3 clock disabled 1: SPI3 clock enabled	0x0	rw
14	SPI2EN	时钟使能/关闭。 0: SPI2 clock disabled 1: SPI2 clock enabled	0x0	rw
13:12	Reserved		0x0	r
11	WWDGGEN	时钟使能/关闭。 0: Window watchdog clock disabled 1: Window watchdog clock enabled	0x0	rw
10:6	Reserved		0x0	r



位	标记	功能描述	复位值	读写
5	TIM11EN	时钟使能/关闭。 0: TIM11 clock disabled 1: TIM11 clock enabled	0x0	rw
4	TIM10EN	时钟使能/关闭。 0: TIM10 clock disabled 1: TIM10 clock enabled	0x0	rw
3	TIM2CEN	时钟使能/关闭。 0: TIM2C clock disabled 1: TIM2C clock enabled	0x0	rw
2	TIM2BEN	时钟使能/关闭。 0: TIM2B clock disabled 1: TIM2B clock enabled	0x0	rw
1	TIM2AEN	时钟使能/关闭。 0: TIM2A clock disabled 1: TIM2A clock enabled	0x0	rw
0	TIM2EN	时钟使能/关闭。 0: TIM2 clock disabled 1: TIM2 clock enabled	0x0	rw



#### 8.4-9 备份域控制寄存器 (RCC\_BDCR)

地址偏移: 0x20

复位值: 0x0000 0000

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

注意：备份域控制寄存器中(RCC\_BDCR)的LSEON、LXTBYP、RTCSEL和RTCEN位处于备份域。因此，这些位在复位后处于写保护状态，只有在电源控制寄存器(PWR\_CR)中的DBP位置‘1’后才能对这些位进行改动。进一步信息请参考7.1节。这些位只能由备份域复位清除(见8.1.3节)。任何内部或外部复位都不会影响这些位。

位	标记	功能描述	复位值	读写
31:17	Reserved		0x0	r
16	BDRST	<b>BDRST:</b> 备份域软件复位 (Backup domain software reset) 由软件置‘1’或清‘0’ 0: 复位未激活； 1: 复位整个备份域。	0x0	rw
15	RTCEN	<b>RTCEN:</b> RTC 时钟使能 (RTC clock enable) 由软件置‘1’或清‘0’ 0: RTC 时钟关闭； 1: RTC 时钟开启。	0x0	rw
14:10	Reserved	保留，始终读为0。	0x0	r



位	标记	功能描述	复位值	读写
9:8	RTCSEL[1:0]	<p><b>RTCSEL[1:0]: RTC 时钟源选择 (RTC clock source selection)</b></p> <p>由软件设置来选择 RTC 时钟源。一旦 RTC 时钟源被选定，直到下次后备域被复位，它不能在被改变。可通过设置 BDRST 位来清除。</p> <p>00: 无时钟； 01: LXT 振荡器作为 RTC 时钟； 10: SIRC 振荡器作为 RTC 时钟； 11: HXT 振荡器在 128 分频后作为 RTC 时钟。</p>	0x0	rw
7:0	Reserved	保留，始终读为 0。	0x0	r



#### 8.4-10 控制/状态寄存器 (RCC\_CSR)

地址偏移: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PWRRSTF	PNRSTF	Reserved	RMVF	-	-	-	-	-	-	-	-
rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位	标记	功能描述	复位值	读写
31	LPWRRSTF	<p><b>LPWRRSTF:</b> 低功耗复位标志 (Low-power reset flag) 在低功耗管理复位发生时由硬件置' 1'；由软件通过写 RMVF 位清除。</p> <p>0: 无低功耗管理复位发生; 1: 发生低功耗管理复位。 关于低功耗管理复位的详细信息，请参考 6.1.1 节的“低功耗管理复位”。</p>	0x0	r/w
30	WWDGRSTF	<p><b>WWDGRSTF:</b> 窗口看门狗复位标志 (Window watchdog reset flag) 在窗口看门狗复位发生时由硬件置' 1'；由软件通过写 RMVF 位清除。</p> <p>0: 无窗口看门狗复位发生; 1: 发生窗口看门狗复位。</p>	0x0	r/w
29	IWDGRSTF	<p><b>IWDGRSTF:</b> 独立看门狗复位标志 (Independent watchdog reset flag) 在独立看门狗复位发生在 V<sub>DD</sub> 区域时由硬件置' 1'；由软件通过写 RMVF 位清除。</p>	0x0	r/w



位	标记	功能描述	复位值	读写
		0: 无独立看门狗复位发生; 1: 发生独立看门狗复位。		
28	SFTRSTF	<b>SFTRSTF:</b> 软件复位标志 (Software reset flag) 在软件复位发生时由硬件置'1'；由软件通过写 RMVF 位清除。 0: 无软件复位发生； 1: 发生软件复位。	0x0	r/w
27	PWRRSTF	<b>PWRRSTF:</b> 上电/掉电复位标志 (POR/PDR reset flag) 在上电/掉电复位发生时由硬件置'1'；由软件通过写 RMVF 位清除。 0: 无上电/掉电复位发生； 1: 发生上电/掉电复位。	0x1	r/w
26	PINRSTF	<b>PINRSTF:</b> NRST 引脚复位标志 (PIN reset flag) 在 NRST 引脚复位发生时由硬件置'1'；由软件通过写 RMVF 位清除。 0: 无 NRST 引脚复位发生； 1: 发生 NRST 引脚复位。	0x1	r/w
25	Reserved	保留, 读操作返回 0	0x0	r
24	RMVF	<b>RMVF:</b> 清除复位标志 (Remove reset flag) 由软件置'1' 来清除复位标志。 0: 无作用； 1: 清除复位标志。	0x0	rw
23:0	Reserved	保留, 读操作返回 0	0x0	r



#### 8.4-11 内部高速 RC 振荡器控制寄存器(RCC\_HIRCCR)

地址偏移: 0x28

复位值: 0x0000 1312

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

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

此寄存器不受软件复位控制

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:13	-	保留	0x0	-
12	HIRC RDY	HIRC 时钟稳定标志位。 0: 代表 HIRC 未稳定，不可以被内部电路使用。 1: 代表 HIRC 已经稳定，可以被内部电路使用。	1	RO
11:0	HIRCTR IM [11:0]	内部高速时钟频率调整 <b>注意：</b> 出厂时，频率校准值保存在 Flash 中， 用户需要将 Flash 值写入 RCC_HIRCCR.HIRCTRIM[11:0] 即可配置。 精准主频时钟。 <b>对于封装片 (Package)：</b> (1) 24M 校准值地址: 0x1800_00A0 (2) 22.12M 校准值地址: 0x1800_00A2 (3) 16M 校准值地址: 0x1800_00A4 (4) 8M 校准值地址: 0x1800_00A6 (5) 4M 校准值地址: 0x1800_00A8 <b>对于裸片 (KGD)：</b> (1) 24M 校准值地址: 0x1800_00C0 (2) 22.12M 校准值地址: 0x1800_00C2 (3) 16M 校准值地址: 0x1800_00C4 (4) 8M 校准值地址: 0x1800_00C6 (5) 4M 校准值地址: 0x1800_00C8	0x312	R/W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。



#### 8.4-12 外部高速晶体振荡器控制寄存器(RCC\_HXTCR)

地址偏移: 0x2C

复位值: 0x0000 0027

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
HXT RDY															
HXT STARTUP [1:0]															
RO R/W															
R/W															

位	标记	功能描述	复位值	读写
31:1 6	KEY	只有高位写 0xA69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:7	-	保留	0x0	-
6	HXTRDY	外部 4M~24M 晶振稳定标志位 0: 代表外部高速晶振时钟未稳定，不可以被内部电路使用。 1: 代表外部高速晶振时钟已经稳定，可以被内部电路使用。	0	RO
5:4	HXTSTARTU P [1:0]	外部 4M~24M 晶振稳定时间选择 00: 1024 个周期 01: 2048 个周期 10: 4096 个周期 11: 16384 个周期 使用高速晶振时钟时稳定时间需要设置为 11，否则稳定时间不够，可能导致系统时钟切换时或使用高速晶振时钟深度休眠唤醒时导致系统不稳定。	0x2	R/ W
3	Revered	-	0	-
2:0	HXTDRV[2:0]	外部 4M~24M 晶振驱动选择 000: 最小驱动 111: 最大驱动(推荐值)	0x7	R/ W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。



#### 8.4-13 内部低速 RC 振荡器控制寄存器(RCC\_SIRCCR)

地址偏移: 0x30

复位值: 0x0000 007F

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留			SIRC	SIRC		保	SIRCTRIM[8:0]									
			RDY	STARTUP[1:0]		留										
			RO	R/W		保	R/W									

此寄存器 LVD 和 Cortex-M3 软件复位以外的复位控制。

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:13	-	保留	0x0	-
12	SIRC RDY	内部低速时钟稳定标志位 0: 代表内部低速未稳定，不可以被内部电路使用。 1: 代表内部低速已经稳定，可以被内部电路使用。	0	RO
11:10	SIRC STARTUP [1:0]	内部低速时钟稳定时间选择 00: 4 个周期 01: 16 个周期 10: 64 个周期 11: 256 个周期	0x0	R/W
9	Revered	-	0	-
8:0	SIRCTRIM[8:0]	内部低速时钟频率调整 注意：出厂时，频率调整值保存在 Flash 中， 用户需要将 Flash 值写入 RCC_SIRCCR.SIRCTRIM[8:0]即可配置经过校准过的 38.4KHz/32.768KHz 内部低速时钟 对于封装片（Package） (1) 32.768KHz 校准值地址: 0x1800_00B0 (2) 38.4KHz 校准值地址: 0x1800_00B4	0x07 F	R/W



		对于裸片 ( <b>KGD</b> ) (1) 32.768KHz 校准值地址: 0x1800_00D0 (2) 38.4KHz 校准值地址: 0x1800_00D4		
--	--	---	--	--

注意: 只有 RCC\_UNLOCK 寄存器保护解除后, 才能写该寄存器。

ZBT CONFIDENTIAL



#### 8.4-14 外部低速晶体振荡器控制寄存器(RCC\_LXTCR)

地址偏移: 0x34

复位值: 0x0000 042F

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LXT	LXT	LXT	LXT	保留	LXT	LXT	STARTUP[1:0]	LXTDRV[3:0]			
				POR	AON	BYP	EN		RDY						
				T					RO						
				R/W	R/W	R/W	R/W								R/W

此寄存器在 RTC 域，只有 POR 才能 reset 掉这个寄存器。

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:12	-	保留	0x0	-
11	LXTPORT	X32K_IN/X32K_OUT function selection 0: GPIO 复用功能(AFR 决定其功能) 1: X32K 端子模式(模拟功能) 注意：该位写受 RCC_UNLOCK 保护	0	R/W
10	LXTAON	LXT 只能使能不能禁止控制 0: LXT_EN 允许禁止控制 1: LXT_~EN 只能使能不能禁止控制	1	R/W
9	LXTBYP	由软件设置和清除，该位仅在外部 32KHz 振荡器关闭的情况下写值。 0: LXT 振荡器未被旁路 1: LXT 振荡器被旁路 注：使用外部低速振荡时需要使能低速晶体振荡的使能位 LXT_EN	0	R/W
8:7	-	保留	0	-
6	LXTRDY	外部 32K 晶振稳定标志位 0: 代表外部 32K 晶振时钟未稳定，不可以被内部电路使用。 1: 代表外部 32K 晶振时钟已经稳定，可以被内部电路使用。	0	RO



位	标记	功能描述	复位值	读写
5:4	LXTSTARTUP [1:0]	外部 32.768KHz 晶振稳定时间选择 00: 1024 个周期 01: 2048 个周期 10: 4096 个周期 11: 16384 个周期  使用低速晶振时钟时稳定时间需要设置为 11，否则 稳定时间不够，可能导致系统时钟切换时或使用低速 晶振时钟深度休眠 唤醒时导致系统不稳定。	0x2	R/W
3:0	LXTDRV[3:0]	外部低速时钟驱动选择 0000: 最小驱动 1111: 最大驱动	0xF	R/W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。



#### 8.4-15 SysTick Timer 控制寄存器(RCC\_STICKCR)

地址偏移: 0x3C

复位值: 0x0100 9C3F

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

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

位	标记	功能描述	复位值	读写
31:2 6	-	保留	0x0	-
25	NOREF	SysTick 定时器是否使用外部参考时钟 0: HCLK/4 1: 使用内核时钟(HCLK) 注意: 1. 本寄存器与系统寄存器 SYST_CSR.CLKSOURCE 任意一个设置为 1 后使用内核时钟(HCLK) 2. 使用分频时钟HCLK/4 作为SysTick 时钟时, 参 考时钟频率不允许高于系统时钟HCLK	0	R/W
24	SKEW	10ms STCALIB 值是否准确 0: 准确 1: 不准确	1	R/W
23:0	STCALIB[23: 0]	SysTick 10ms 校准值, 此值为使用外部参考时钟 HCLK/4(4MHz) 的 10ms 校准值。	0x009 C3F	R/W



#### 8.4-16 SWDIO 端口控制寄存器(RCC\_SWDIOCR)

地址偏移: 0x40

复位值: 0x0000 0001

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

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

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:2	-	保留	0x0	
1:0	SWDPORT [1:0]	配置 PA13 和 PA14 的端子功能模式 2' b00: 周边模块功能模式 2' b01: SWD 端子功能 : PA13,PA14 2' b1x: SWD 端子功能 : PA11,PA12	2' b01	R/W

注意：该位写受 RCC\_UNLOCK 保护



#### 8.4-17 RTC 复位控制寄存器(RCC\_RTCRST)

地址偏移: 0x48

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
															RT C RS T
															R/ W

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15:5	-	保留, 必须保持复位值	0x0	-
0	RTC RST	RTC 模块复位 由软件置 1 或清 0 0: 复位未激活 1: 复位整个 RTC	0	R/W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后, 才能写该寄存器。



#### 8.4-18 MASK 中断时间控制寄存器(RCC\_MASK\_INT\_T)

地址偏移: 0x4C

复位值: 0x0000 0004

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

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

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:3	-	保留，必须保持复位值	0x0	-
2:0	MASK_INT_T [2:0]	进入 DEEPSLEEP 时, mask interrupt time 000: 1.6ms 001: 1.4ms 010: 1.2ms 011: 1.0ms 100: 0.8ms 101: 0.6ms 110: 0.4ms 111: 0.8ms	0x4	R/W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。



#### 8.4-19 LPM 控制寄存器(RCC\_LPM)

地址偏移: 0x50

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
														LP	LP
														M_	M_FW
														HA	ND
														LE	
														R/	R/
														W	W

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:2	-	保留，必须保持复位值	0x0	-
1	LPM_HA NDLE	1: LPM 由 CPU 控制 0: LPM 由硬件控制	0x0	R/W
0	LPM_FW	1: Vcore 进入 Low power mode on LPM_HANDLE=1 0: Vcore 处于 active mode on LPM_HANDLE=1	0x0	R/W

Note: 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。



#### 8.4-20 AHB 外设复位控制寄存器(RCC\_AHBRSTR)

地址偏移: 0x54

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
SDIORST															
QSPIRST															
USBRST															
TRNGRST															
AESRST															
DMA2RST															
R/W															

位	标记	功能描述	复位值	读写
31:7	保留	保留	0x0	--
6	SDIORST	SDIO 模块复位 0: 正常工作 1: 复位	0x0	R/W
5	QSPIRST	QSPI 模块复位 0: 正常工作 1: 复位	0x0	R/W
4	USBRST	USB 模块复位 0: 正常工作 1: 复位	0x0	R/W
3	TRNGRST	TRNG 模块复位 0: 正常工作 1: 复位	0x0	R/W
2	AESRST	AES 模块复位 0: 正常工作 1: 复位	0x0	R/W
1	DMA2RST	DMA2 模块复位 0: 正常工作 1: 复位	0x0	R/W



0	DMA1RST	DMA1 模块复位 0: 正常工作 1: 复位	0x0	R/W
---	---------	-------------------------------	-----	-----

ZBT CONFIDENTIAL



#### 8.4-21 PLL 控制寄存器 2(RCC\_PLLCR1)

地址偏移: 0x58

复位值: 0x0000 0027

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLTRIM[7:0]							-	PLLSYS[1:0]		PLLSTARTUP [1:0]		PLLRDY	Revered		
R/W							--	R/W		R/W		RW	R/W		

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0xA69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:8	PLLTRIM[7:0]	内部高速时钟频率调整 注意：出厂时，频率校准值保存在 Flash 中， 用户需要将 Flash 值写入 RCC_HIRCCR.HIRCTRIM[2:0] 即可配置。 精准主频时钟。 对于封装片 ( ): ● PLL 校准值地址: 0x1800_0092 对于裸片 (KGD): ● PLL 校准值地址: 0x1800_0090	0x92	R/W
7	CSSON	<b>CSSON:</b> 时钟安全系统使能 (Clock security system enable) 位19 由软件置‘1’或清零以使能时钟监测器。 0: 时钟监测器关闭； 1: 如果外部3-25MHz振荡器就绪，时钟监测器开启。	0	R/W
6:5	PLLSYS[1:0]	PLL_SYS 选择时钟 00: PLL_SYS=PLL_O/2 01: PLL_SYS=PLL_O/4 10: PLL_SYS=PLL_O/3 11:保留	0x0	R/W



4:3	PLLSTARTUP [1:0]	PLL 稳定时间选择 00: 1024 个周期 01: 2048 个周期 10: 4096 个周期 11: 16384 个周期	0x2	R/W
2	PLLRDY	PLL 80M~192M时钟稳定标志位 0: 代表PLL时钟未稳定, 不可以被内部电路使用。 1: 代表PLL时钟已经稳定, 可以被内部电路使用。	0	RO
1:0	Revered	-	0	r

Note: 只有 RCC\_UNLOCK 寄存器保护解除后, 才能写该寄存器。

ZBT CONFIDENTIAL



#### 8.4-22 PLL 控制寄存器 1(RCC\_PLLCR2)

地址偏移: 0x5C

复位值: 0x0000 0027

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

15~13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留	LDO_PLL_TR [1:0]		PLL_M[6:0]		PLL_N [1:0]		SELRCLK [1:0]								
	RW		RW		RW		RW								

位	标记	功能描述	复位值	读写								
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO								
15:13	Revered	Revered	--	--								
12:11	LDO_PLL_T R [1:0]	调整PLL 的 LDO电压,在1.2V使用ADC 调整。 11 : LDO 电压最低 10 : 01 : 00: LDO 电压最高	2' b1 1	R/W								
10:4	PLL_M[6:0]	PLL_M参考选择: 建议 PLL_M = XX <table border="1" style="width: 100%;"><tr> <td>RCLK (HIRC/ XTAL/OSC)</td><td>PLL_N [1:0]</td><td>M [6:0]</td><td>PLL_O =RCLK*(M/ N)</td></tr> <tr> <td>24MHz</td><td></td><td></td><td>384/192MH Z</td></tr> </table>	RCLK (HIRC/ XTAL/OSC)	PLL_N [1:0]	M [6:0]	PLL_O =RCLK*(M/ N)	24MHz			384/192MH Z	7' b01 00000 (7' dX X)	R/ W
RCLK (HIRC/ XTAL/OSC)	PLL_N [1:0]	M [6:0]	PLL_O =RCLK*(M/ N)									
24MHz			384/192MH Z									
3:2	PLL_N[1:0]	PLL_N参考选择: 00:RCLK/1 01: RCLK/2 10: RCLK/4 11: RCLK/6	2' b0 0	R/ W								
1:0	SELRCLK[1: 0]	PLL参考时钟(RCLK)选择: 00:HIRC 01:XTAL	0x0	R/ W								



		10:OSC(XIN)			
		11:Reverd			

Note: 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存

ZBT CONFIDENTIAL



#### 8.4-23 寄存器写保护控制寄存器(RCC\_UNLOCK)

地址偏移: 0x60

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:1	Key	只有高位写 0x3D5334C 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
0	UNLOCK	0: 寄存器写保护启用, 不能对受保护的寄存器写操作 1: 寄存器写保护禁止, 可以对受保护的寄存器写操作	0	R/W

Note : 写入 0x55AA6699,解除保护



## 【9】 系统控制(SYSCON)

本产品具有一组系统配置寄存器。系统配置控制器的主要用途如下：

1. 配置 GPIO 端子的中断产生模式
  2. 重映射 TIM10/11、PCA、TIM1,TIM1A,TIM1B,TIM2,TIM2A,TIM2B,TIM2C 的输入触发源
  3. SPI 从模式的 CS 输入重映射设定
  4. 系统级的 Deep Sleep 调试以及 Lockup 复位控制设定
  5. Boot 0,Boot1 PIN Latch 状态值
4. 系统级的 Deep Sleep 调试以及 Lockup 复位控制设定

### 9.1 寄存器列表

SYSCON Address : 0x4000 1C00

偏移地址	名称	描述	复位值
0x00	SYSCON_CFRG0	系统配置寄存器 0	0x0000 0000
0x04	SYSCON_PORTINTCR	端子的中断模式设定	0x0000 0000
0x08	SYSCON_PORTCR	端子控制寄存器	0x0000 0000
0x0C	SYSCON_PCACR	PCA 捕获通道来源选择	0x0000 0000
0x10	SYSCON_TIM1CR	TIM1 通道输入源选择	0x0000 0000
0x14	SYSCON_TIM2CR	TIM1A 通道输入源选择	0x0000 0000
0x18	SYSCON_TIM1ACR	TIM1B 通道输入源选择	0x0000 0000
0x1C	SYSCON_TIM1BCR	TIM2 通道输入源选择	0x0000 0000
0x20	SYSCON_TIM2ACR	TIM2A 通道输入源选择	0x0000 0000
0x24	SYSCON_TIM2BCR	TIM2B 通道输入源选择	0x0000 0000
0x28	SYSCON_TIM2CCR	TIM2C 通道输入源选择	0x0000 0000
0x50	SYSCON_UNLOCK	SYSCON 寄存器写保护	0x0000 0000
0x3C0	SYSCON_BOOT	BOOT 状态暂存器	0x0000 0000

Table 9-1 SYSCON 寄存器列表和复位值



## 9.2 寄存器说明

### 9.2-1 系统配置寄存器 0(SYSCON\_CFGR0)

地址偏移: 0x00

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										TREEDLY [2:0]	VD12_IRE	VDS_SEL	DBG	LOC KUP	
R/W										R/W	R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:16	Key	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15:7	-	保留	0x0	-
6:4	TREEDLY[2:0]	VD12 Inrush pulse option: TREEDLY[1:0] 2' b00 : Pulse Start at 7ns 2' b01 : Pulse Start at 8ns 2' b10 : Pulse Start at 9ns 2' b11 : Pulse Start at 10 ns TREEDLY[2] 0 : Pulse End at 12ns 1 : Pulse End at 11ns	0x0	R/W
3	VD12_IREN	1: LDO Inrush 補償. 0: LDO 1.2v Normal operation	0	R/W
2	VDS_SEL	进入Deep Sleep 模式,选择V12的电压 0: V12=0.85V 1:V12=0.7V	0	R/W
1:0	Revered	保留	0	-

Note :只有 SYSCON\_UNLOCK 寄存器保护解除后, 才能写该寄存器。



### 9.2-2 端子 Deep Sleep 中断模式控制寄存器(SYSCON\_PORTINTCR)

地址偏移: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													RET1NEN	PADDLSP	PADINTS
													R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:16	Key	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:3	-	保留	0x0	-
2	RET1NEN	SRAM Suspend mode 0: 进入Deep Sleep 模式 , RET1N=1 1: 进入Deep Sleep 模式 , RET1N=0	0x0	R/W
1	PADDLSPCON	0: 当进入Deep Sleep 后, PAD 的中断产生模式自动切换到 Deep Sleep 中断产生模式(没有Debounce 功能) 1: 当进入DeepSleep 后, PAD 的中断产生模式不会自动切换, 由SYSCON_PORTINTCR.PADINTSEL 位决定中断产生模式。	0	R/W
0	PADINTSEL	端口中断模式选择 0: ACTIVE/Sleep 中断产生模式 1: Deep Sleep 中断产生模式	0	R/W

Note:

- 选用 Deep Sleep 中断产生模式时, GPIO 端子的 Debounce 功能需要关闭, 抗干扰能力很差, 一般推荐用户在芯片需要使用端子中断唤醒 Deep Sleep 模式才选择此模式
- 只有 SYSCON\_UNLOCK 寄存器保护解除后, 才能写该寄存器。



### 9.2-3 端子控制寄存器(SYSCON\_PORTCR)

地址偏移: 0x08

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPTIM_GATE_E_SEL[1:0]	TIM11_GATE_SEL[2:0]				TIM10_GATE_SEL[1:0]	SPI3SSN_SEL[3:0]				SPI2SSN_SEL[3:0]					
R/W															

位	标记	功能描述	复位值	读写
31:15	-	保留	-	-
15:14	LPTIM_GATE_SEL[1:0]	Low Power Timer 门控输入信号来源选择 00: LPTIM_GATE 01: USART1_RXD 10: USART2_RXD 11: LPUART_RXD	0x0	R/W
13:11	TIM10_GATE_SEL[1:0]	Timer10 门控输入信号来源选择 000: TIM11_GATE 001: USART1_RXD 010: USART2_RXD 011: LPUART_RXD 100 : USART3_RXD 101: UART4_RXD 110: UART5_RXD 111: Revered	0x0	R/W
10:8	TIM11_GATE_SEL[1:0]	Timer11门控输入信号来源选择 000: TIM10_GATE 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100 : USART3_RXD 101: UART4_RXD 110: UART5_RXD 111: Revered	0x0	R/W



位	标记	功能描述	复位值	读写																																		
7:4	SPI3SSN_SEL[3:0]	<p>SPI3/I2S3 Slave SSN 信号来源选择.</p> <table border="1"><tr><td>SPI3SSN_SEL[3:0]</td><td>SSN</td></tr><tr><td>0000</td><td>固定高电平</td></tr><tr><td>0001</td><td>PA03</td></tr><tr><td>0010</td><td>PA04</td></tr><tr><td>0011</td><td>PA06</td></tr><tr><td>0100</td><td>PA08</td></tr><tr><td>0101</td><td>PA09</td></tr><tr><td>0110</td><td>PA12</td></tr><tr><td>0111</td><td>PA15</td></tr><tr><td>1000</td><td>PB01</td></tr><tr><td>1001</td><td>PB02</td></tr><tr><td>1010</td><td>PB05</td></tr><tr><td>1011</td><td>PB06</td></tr><tr><td>1100</td><td>PB09</td></tr><tr><td>1101</td><td>PB10</td></tr><tr><td>1110</td><td>PB12</td></tr><tr><td>1111</td><td>PB14</td></tr></table>	SPI3SSN_SEL[3:0]	SSN	0000	固定高电平	0001	PA03	0010	PA04	0011	PA06	0100	PA08	0101	PA09	0110	PA12	0111	PA15	1000	PB01	1001	PB02	1010	PB05	1011	PB06	1100	PB09	1101	PB10	1110	PB12	1111	PB14	0x0	R/W
SPI3SSN_SEL[3:0]	SSN																																					
0000	固定高电平																																					
0001	PA03																																					
0010	PA04																																					
0011	PA06																																					
0100	PA08																																					
0101	PA09																																					
0110	PA12																																					
0111	PA15																																					
1000	PB01																																					
1001	PB02																																					
1010	PB05																																					
1011	PB06																																					
1100	PB09																																					
1101	PB10																																					
1110	PB12																																					
1111	PB14																																					
3:0	SPI2SSN_SEL[3:0]	<p>SPI2/I2S2 Slave SSN 信号来源选择.</p> <table border="1"><tr><td>SPI2SSN_SEL[3:0]</td><td>SSN</td></tr><tr><td>0000</td><td>固定高电平</td></tr><tr><td>0001</td><td>PA03</td></tr><tr><td>0010</td><td>PA04</td></tr><tr><td>0011</td><td>PA06</td></tr><tr><td>0100</td><td>PA08</td></tr><tr><td>0101</td><td>PA09</td></tr><tr><td>0110</td><td>PA12</td></tr><tr><td>0111</td><td>PA15</td></tr><tr><td>1000</td><td>PB01</td></tr><tr><td>1001</td><td>PB02</td></tr><tr><td>1010</td><td>PB05</td></tr><tr><td>1011</td><td>PB06</td></tr><tr><td>1100</td><td>PB09</td></tr><tr><td>1101</td><td>PB10</td></tr><tr><td>1110</td><td>PB12</td></tr><tr><td>1111</td><td>PB14</td></tr></table>	SPI2SSN_SEL[3:0]	SSN	0000	固定高电平	0001	PA03	0010	PA04	0011	PA06	0100	PA08	0101	PA09	0110	PA12	0111	PA15	1000	PB01	1001	PB02	1010	PB05	1011	PB06	1100	PB09	1101	PB10	1110	PB12	1111	PB14	0x0	R/W
SPI2SSN_SEL[3:0]	SSN																																					
0000	固定高电平																																					
0001	PA03																																					
0010	PA04																																					
0011	PA06																																					
0100	PA08																																					
0101	PA09																																					
0110	PA12																																					
0111	PA15																																					
1000	PB01																																					
1001	PB02																																					
1010	PB05																																					
1011	PB06																																					
1100	PB09																																					
1101	PB10																																					
1110	PB12																																					
1111	PB14																																					



### 9.2-4 PCA 捕获通道控制寄存器(SYSCON\_PCACR)

地址偏移: 0x0C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留	PCA_CAP4_S EL[2:0]			PCA_CAP3_S EL[2:0]			PCA_CAP2_S EL[2:0]			PCA_CAP1_S EL[2:0]			PCA_CAP0_S EL[2:0]		
保 留	R/W														

位	标记	功能描述	复位值	读写
31:10	-	保留	0x0	-
14:12	PCA_CAP4_SEL[1:0]	PCA 捕获通道 4 信号来源选择 000: PCA_CH4 001: USART1_RXD 010: USART2_RXD 011: LPUART_RXD 100: USART3_RXD 101: UART4_RXD 110: UART4_RXD 111: Revered	0x0	R/W
11:9	PCA_CAP3_SEL[1:0]	PCA 捕获通道 3 信号来源选择 000: PCA_CH3 001: USART1_RXD 010: USART2_RXD 011: LPUART_RXD 100: USART3_RXD 101: UART4_RXD 110: UART4_RXD 111: Revered	0x0	R/W
8:6	PCA_CAP2_SEL[1:0]	PCA 捕获通道 2 信号来源选择 000: PCA_CH2 001: USART1_RXD 010: USART2_RXD 011: LPUART_RXD 100: USART3_RXD 101: UART4_RXD 110: UART4_RXD 111: Revered	0x0	R/W



5:3	PCA_CAP1_SEL[1:0]	PCA 捕获通道 1 信号来源选择 000: PCA_CH1 001: USART1_RXD 010: USART2_RXD 011: LPUART_RXD 100: USART3_RXD 101: UART4_RXD 110: UART4_RXD 111: Revered	0x0	R/W
2:0	PCA_CAP0_SEL[1:0]	PCA 捕获通道 0 信号来源选择 00: PCA_CH0 001: USART1_RXD 010: USART2_RXD 011: LPUART_RXD 100: USART3_RXD 101: UART4_RXD 110: UART4_RXD 111: Revered	0x0	R/W

### 9.2-5 TIM1,8,12 通道输入源选择(SYSCON\_TIMxCR,X=1,1A,1B)

SYSCON\_TIM1CR : 地址偏移: 0x10

复位值: 0x0000 0000

SYSCON\_TIM1ACR : 地址偏移: 0x18

复位值: 0x0000 0000

SYSCON\_TIM1BCR : 地址偏移: 0x1C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	TIMXETR_SEL[3:0]
保留									CLOCK	TIMX	TIMX					
保留									DSLP	BRK OUT	BRK OUT					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIMXCH4IN_SEL [2:0]				保留	TIMXCH3IN_SEL [2:0]				TIMXCH2IN_SEL [2:0]				TIMXCH1IN_SEL [2:0]	
保留	R/ W	R/ W	R/ W	保 留	R/ W	R/ W	R/ W	保 留	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W

位	标记	功能描述	复位值	读写



位	标记	功能描述	复位值	读写																																
31:23	-	保留	0x0	-																																
22	TIMXCLKFAILBRKEN	系统时钟停止检出时TIMX Break 使能 0: 无效 1: 使能	0	R/W																																
21	TIMXDSLPBRKEN	Deep Sleep 模式时TIMX Break 使能 0: 无效 1: 使能	0	R/W																																
20	TIMXBRKOUTCFG	0: ocxp/ocxnp 在 break 模式的输出由 TIMX控制 1: ocxp/ocxnp 在 break 模式同时输出 0	0	R/W																																
19:16	TIMXETR_SEL[3:0]	TIMX ETR 信号来源选择  <table border="1"><tr><td>0000</td><td>固定低电平</td></tr><tr><td>0001</td><td>PA1</td></tr><tr><td>0010</td><td>PA2</td></tr><tr><td>0011</td><td>PA3</td></tr><tr><td>0100</td><td>PB4</td></tr><tr><td>0101</td><td>PB5</td></tr><tr><td>0110</td><td>PA00</td></tr><tr><td>0111</td><td>PD00</td></tr><tr><td>1000</td><td>PC5</td></tr><tr><td>1001</td><td>PC6</td></tr><tr><td>1010</td><td>PC7</td></tr><tr><td>1011</td><td>PD01</td></tr><tr><td>1100</td><td>PA03</td></tr><tr><td>1101</td><td>PA04</td></tr><tr><td>1110</td><td>PA05</td></tr><tr><td>1111</td><td>PA06</td></tr></table>	0000	固定低电平	0001	PA1	0010	PA2	0011	PA3	0100	PB4	0101	PB5	0110	PA00	0111	PD00	1000	PC5	1001	PC6	1010	PC7	1011	PD01	1100	PA03	1101	PA04	1110	PA05	1111	PA06	0x0	R/W
0000	固定低电平																																			
0001	PA1																																			
0010	PA2																																			
0011	PA3																																			
0100	PB4																																			
0101	PB5																																			
0110	PA00																																			
0111	PD00																																			
1000	PC5																																			
1001	PC6																																			
1010	PC7																																			
1011	PD01																																			
1100	PA03																																			
1101	PA04																																			
1110	PA05																																			
1111	PA06																																			
15	-	保留	0	-																																
14:12	TIMXCH4IN_SEL[2:0]	TIMX CH4 输入通道信号来源选择 000: TIMX_CH4 001: USART1_RXD(TIM1,TIM1A) : UART4_RXD(TIM1B) 010: USART2_RXD(TIM1,TIM1A) : UART5_RXD(TIM1B) 011: LPUART_RXD 100: SIRC 101: PA15 (可连接霍尔传感器) 110: PC14 (可连接霍尔传感器)	0x0	R/W																																



位	标记	功能描述	复位值	读写
		111: PA14 (可连接霍尔传感器)		
11	-	保留	0	-
10:8	TIMXCH3IN_SEL[2:0]	TIMX CH3 输入通道信号来源选择 000: TIMX_CH3 001: USART1_RXD(TIM1,TIM1A) : UART4_RXD(TIM1B) 010: USART2_RXD(TIM1,TIM1A) : UART5_RXD(TIM1B) 011: LPUART_RXD 100: SIRC 101: PD01 (可连接霍尔传感器) 110: PC15 (可连接霍尔传感器) 111: PD00 (可连接霍尔传感器)	0x0	R/W
7	-	保留	0	-
6:4	TIMXCH2IN_SEL[2:0]	TIMX CH2 输入通道信号来源选择 000: TIMX_CH2 001: USART1_RXD(TIM1,TIM1A) : UART4_RXD(TIM1B) 010: USART2_RXD(TIM1,TIM1A) : UART5_RXD(TIM1B) 011: LPUART_RXD 100: SIRC 101: PD07(可连接霍尔传感器) 110: PB03(可连接霍尔传感器) 111: PA00(可连接霍尔传感器)	0x0	R/W
3	-	保留	0	-



位	标记	功能描述	复位值	读写
2:0	TIMXCH1IN_SEL[2:0]	TIMX CH1 输入通道信号来源选择 000: TIMX_CH1 001: USART1_RXD(TIM1,TIM1A) : UART4_RXD(TIM1B) 010: USART2_RXD(TIM1,TIM1A) : UART5_RXD(TIM1B) 011: LPUART_RXD 100: SIRC 101: PD06(可连接霍尔传感器) 110: PA15(可连接霍尔传感器) 111: PA02(可连接霍尔传感器)	0x0	R/W

Note : TIMX : X=1,1A,1B



### 9.2-6 TIM2/3/4/5 通道输入源选择(SYSCON\_TIMXCR)

SYSCON\_TIM2CR : 地址偏移: 0x20 复位值: 0x0000 0000

SYSCON\_TIM2ACR : 地址偏移: 0x24 复位值: 0x0000 0000

SYSCON\_TIM2BCR : 地址偏移: 0x28 复位值: 0x0000 0000

SYSCON\_TIM2CCR : 地址偏移: 0x2C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														TIMXETR_SEL[3:0]	
保留														R/W	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIMXCH4IN_SEL[2:0]			保留	TIMXCH3IN_SEL[2:0]			保留	TIMXCH2IN_SEL[2:0]			保留	TIMXCH1IN_SEL[2:0]		
	R/ W	R/ W	R/ W		R/ W	保留	R/ W		R/ W	R/ W	R/ W		R/ W	R/ W	R/ W
	-	-	-		-	-	-		-	-	-		-	-	-

位	标记	功能描述	复位值	读写																																
31:20	-	保留	0x0	-																																
19:16	TIMXETR_SEL[3:0]	TIMX ETR 信号来源选择 <table border="1"> <tr><td>0000</td><td>固定低电平</td></tr> <tr><td>0001</td><td>PA1</td></tr> <tr><td>0010</td><td>PA2</td></tr> <tr><td>0011</td><td>PA3</td></tr> <tr><td>0100</td><td>PB4</td></tr> <tr><td>0101</td><td>PB5</td></tr> <tr><td>0110</td><td>PA00</td></tr> <tr><td>0111</td><td>PD00</td></tr> <tr><td>1000</td><td>PC5</td></tr> <tr><td>1001</td><td>PC6</td></tr> <tr><td>1010</td><td>PC7</td></tr> <tr><td>1011</td><td>PD01</td></tr> <tr><td>1100</td><td>PA03</td></tr> <tr><td>1101</td><td>PA04</td></tr> <tr><td>1110</td><td>PA05</td></tr> <tr><td>1111</td><td>PA06</td></tr> </table>	0000	固定低电平	0001	PA1	0010	PA2	0011	PA3	0100	PB4	0101	PB5	0110	PA00	0111	PD00	1000	PC5	1001	PC6	1010	PC7	1011	PD01	1100	PA03	1101	PA04	1110	PA05	1111	PA06	0x0	R/W
0000	固定低电平																																			
0001	PA1																																			
0010	PA2																																			
0011	PA3																																			
0100	PB4																																			
0101	PB5																																			
0110	PA00																																			
0111	PD00																																			
1000	PC5																																			
1001	PC6																																			
1010	PC7																																			
1011	PD01																																			
1100	PA03																																			
1101	PA04																																			
1110	PA05																																			
1111	PA06																																			
15	-	保留	0	-																																



位	标记	功能描述	复位值	读写
14:12	TIMXCH4IN_SEL[2:0]	TIMX CH4 输入通道信号来源选择 000: TIMX_CH4 001: USART1_RXD(TIM2,TIM2A) :UART4_RXD(TIM2B,TIM2C) 010: USART2_RXD (TIM2,TIM2A) UART5_RXD(TIM2B,TIM2C) 011: LPUART_RXD 100: SIRC 101: PB13 110: PC14 111: PB02	0x0	R/W
11	-	保留	0	-
10:8	TIMXCH3IN_SEL[2:0]	TIMX CH3 输入通道信号来源选择 000: TIMX_CH3 001: USART1_RXD(TIM2,TIM2A) :UART4_RXD(TIM2B,TIM2C) 010: USART2_RXD (TIM2,TIM2A) UART5_RXD(TIM2B,TIM2C) 011: LPUART_RXD 100: SIRC 101: PB12 110: PC15 111: PB01	0x0	R/W
7	-	保留	0	-
6:4	TIMXCH2IN_SEL[2:0]	TIMX CH2 输入通道信号来源选择 000: TIMX_CH2 001: USART1_RXD(TIM2,TIM2A) :UART4_RXD(TIM2B,TIM2C) 010: USART2_RXD (TIM2,TIM2A) UART5_RXD(TIM2B,TIM2C) 011: LPUART_RXD	0x0	R/W



位	标记	功能描述	复位值	读写
		100: SIRC 101: PB11 110: PB03 111: PB00		
3	-	保留	0	-
2:0	TIMXCH1IN_SEL[2:0]	TIMX CH1 输入通道信号来源选择 000: TIMX_CH1 001: USART1_RXD(TIM2,TIM2A) 010: USART2_RXD (TIM2,TIM2A) 011: LPUART_RXD 100: SIRC 101: PB10 110: PA15 111: PA07	0x0	R/W

Note: TIMX : X=2,3,4,5



### 9.2-7 SYSCON 寄存器写保护(SYSCON\_UNLOCK)

地址偏移: 0x50

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
WO															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:1]															
WO															

位	标记	功能描述	复位值	读写
31: 1	KEY	只有高位写 0x3D5334C 时配置该寄存器才有效, 写其它值时无效。	0x0	-
0	UNLOCK	0: 保护有效 1: 解除保护	0	R/W

Note : 写 0x55AA6699, 解除保护



### 9.2-8 SYSCON BOOT 状态暂存器(SYSCON\_BOOT)

地址偏移: 0x3C0

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
WO															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:1]															
WO															

位	标记	功能描述	复位值	读写
31:2	保留	只有高位写 0x3D53		--
1	BOOT1	請参考5.5節	0	RO
0	BOOT0		0	RO



## 【10】中断和事件

在 ZB32F103 的 NVIC 接收多个中断源产生的中断请求，如下图所示：

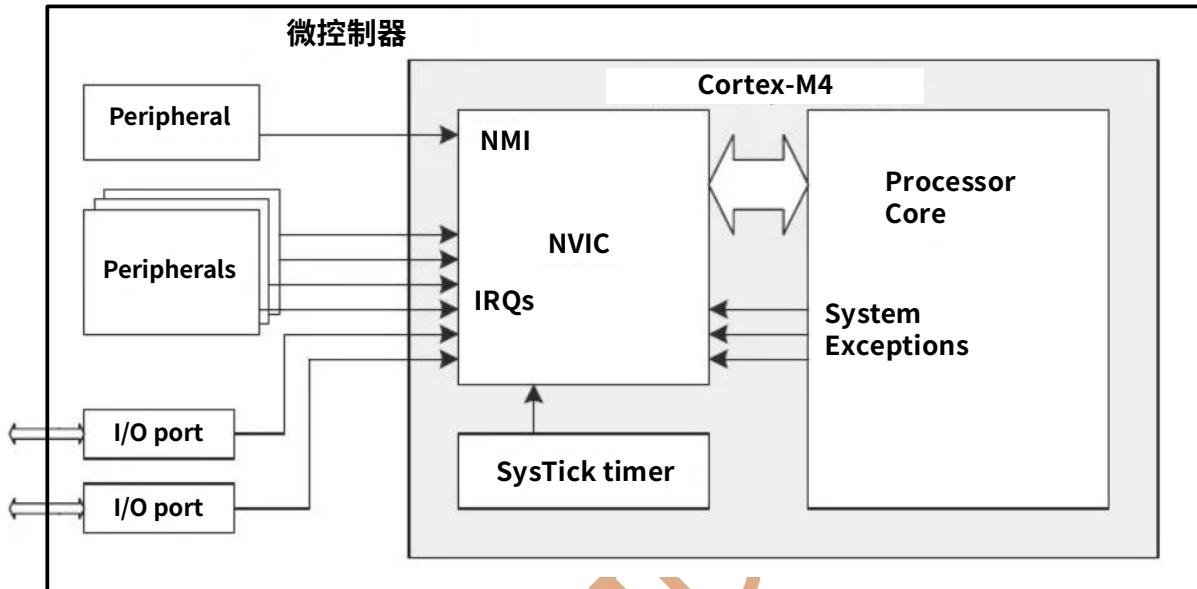


Figure 10-1 MCU 的各种异常源框图

### 10.1 嵌套向量中断控制器

特性

1. 68 个可屏蔽中断通道(不包含 16 个 Cortex™-M3 的中断)
2. 16 个可编程的优先等级(使用了 4 位中断优先级)
3. 低延时的异常和中断处理
4. 电源管理控制
5. 系统控制寄存器的实现
6. 支持嵌套和向量中断
7. 动态改变优先级

嵌套向量中断控制器(NVIC)和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。嵌套向量中断控制器管理着包括内核异常等中断。

#### 10.1-1 系统时钟(SysTick)校准值寄存器

系统时钟校准值固定为 9000，当系统时钟设定为 9MHz(HCLK/8 的大值)，产生 1ms 时间基准。



### 10.1.2 中断和异常向量表

下表為 ZB32F103 产品的向量表。

Table 10-1 中断向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000_0000
	-3	固定	Reset	复位	0x0000_0004
	-2	固定	NMI	不可屏蔽中断 RCC 时钟安全系统(CSS)联接到 NMI 向量	0x0000_0008
	-1	固定	硬件失效(HardFault)	所有类型的失效	0x0000_000C
	0	可设置	存储管理 (MemManage)	存储器管理	0x0000_0010
	1	可设置	总线错误(BusFault)	预取指失败, 存储器访问失败	0x0000_0014
	2	可设置	错误应用(UsageFault)	未定义的指令或非法状态	0x0000_0018
	-	-	-	保留	0x0000_001C ~0x0000_004
	3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C
	4	可设置	调试 监控 (DebugMonitor)	调试监控器	0x0000_0030
	-	-	-	保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDG	窗口定时器中断	0x0000_0040
1	8	可设置	PVD	连到 EXTI 的电源电压检测(PVD) 中断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	RTC	实时时钟(RTC)全局中断	0x0000_004C
4	11	可设置	QSPIXIP	QSPIXIP 全局中断	0x0000_0050
5	12	可设置	RCC	复位和时钟控制(RCC)中断	0x0000_0054



位置	优先级	优先级类型	名称	说明	地址
6	13	可设置	EXTI0	EXTI 线 0 中断	0x0000_0058
7	14	可设置	EXTI1	EXTI 线 1 中断	0x0000_005C
8	15	可设置	EXTI2	EXTI 线 2 中断	0x0000_0060
9	16	可设置	EXTI3	EXTI 线 3 中断	0x0000_0064
10	17	可设置	EXTI4	EXTI 线 4 中断	0x0000_0068
11	18	可设置	DMA1 通道 0	DMA1 通道 0 全局中断	0x0000_006C
12	19	可设置	DMA1 通道 1	DMA1 通道 1 全局中断	0x0000_0070
13	20	可设置	DMA1 通道 2	DMA1 通道 2 全局中断	0x0000_0074
14	21	可设置	DMA1 通道 3	DMA1 通道 3 全局中断	0x0000_0078
15	22	可设置	DMA1 通道 4	DMA1 通道 4 全局中断	0x0000_007C
16	23	可设置	DMA1 通道 5	DMA1 通道 5 全局中断	0x0000_0080
17	24	可设置	DMA1 通道 6	DMA1 通道 6 全局中断	0x0000_0084
18	25	可设置	ADC1_2	ADC1 和 ADC2 的全局中断	0x0000_0088
19	26	可设置	USB_HP_CAN_TX	USB 高优先级或 CAN 发送中断	0x0000_008C
20	27	可设置	USB_LP_CAN_RX0	USB 低优先级或 CAN 接收 0 中断	0x0000_0090
21	28	可设置	CAN_RX1	CAN 接收 1 中断	0x0000_0094
22	29	可设置	CAN_SCE	CAN SCE 中断	0x0000_0098
23	30	可设置	EXTI9_5	EXTI 线[9:5]中断	0x0000_009C
24	31	可设置	TIM1_BRK	TIM1 刹车中断	0x0000_00A0
25	32	可设置	TIM1_UP	TIM1 更新中断	0x0000_00A4
26	33	可设置	TIM1_TRG_COM	TIM1 触发和通信中断	0x0000_00A8
27	34	可设置	TIM1_CC	TIM1 捕获比较中断	0x0000_00AC
28	35	可设置	TIM2	TIM2 全局中断	0x0000_00B0
29	36	可设置	TIM2A	TIM2A 全局中断	0x0000_00B4
30	37	可设置	TIM2B	TIM2B 全局中断	0x0000_00B8
31	38	可设置	I2C1_EV	I2C1 事件中断	0x0000_00BC
32	39	可设置	I2C1_ER	I2C1 错误中断	0x0000_00C0
33	40	可设置	I2C2_EV	I2C2 事件中断	0x0000_00C4
34	41	可设置	I2C2_ER	I2C2 错误中断	0x0000_00C8
35	42	可设置	SPI1	SPI1 全局中断	0x0000_00CC



位置	优先级	优先级类型	名称	说明	地址
36	43	可设置	SPI2	SPI2 全局中断	0x0000_00D0
37	44	可设置	USART1	USART1 全局中断	0x0000_00D4
38	45	可设置	USART2	USART2 全局中断	0x0000_00D8
39	46	可设置	USART3	USART3 全局中断	0x0000_00DC
40	47	可设置	EXTI15_10	EXTI 线[15:10]中断	0x0000_00E0
41	48	可设置	RTCAlarm	连到 EXTI 的 RTC 阔钟中断	0x0000_00E4
42	49	可设置	USB 唤醒	连到 EXTI 的从 USB 待机唤醒中断	0x0000_00E8
43	50	可设置	TIM1A_BRK	TIM1A 刹车中断	0x0000_00EC
44	51	可设置	TIM1A_UP	TIM1A 更新中断	0x0000_00F0
45	52	可设置	TIM1A_TRG_COM	TIM1A 触发和通信中断	0x0000_00F4
46	53	可设置	TIM1A_CC	TIM1A 捕获比较中断	0x0000_00F8
47	54	可设置	ADC3	ADC3 全局中断	0x0000_00FC
48	55	可设置	FSMC	FSMC 全局中断	0x0000_0100
49	56	可设置	SDIO	SDIO 全局中断	0x0000_0104
50	57	可设置	TIM2C	TIM2C 全局中断	0x0000_0108
51	58	可设置	SPI3	SPI3 全局中断	0x0000_010C
52	59	可设置	UART4	UART4 全局中断	0x0000_0110
53	60	可设置	UART5	UART5 全局中断	0x0000_0114
54	61	可设置	TIM10	TIM10 全局中断	0x0000_0118
55	62	可设置	TIM11	TIM11 全局中断	0x0000_011C
56	63	可设置	DMA2 通道 0	DMA2 通道 0 全局中断	0x0000_0120
57	64	可设置	DMA2 通道 1	DMA2 通道 1 全局中断	0x0000_0124
58	65	可设置	DMA2 通道 2	DMA2 通道 2 全局中断	0x0000_0128
59	66	可设置	DMA2 通道 3	DMA2 通道 3 全局中断	0x0000_012C
60	67	可设置	DMA2 通道 4	DMA2 通道 4 全局中断	0x0000_0130
61	68	可设置	DMA2 通道 5	DMA2 通道 5 全局中断	0x0000_0134
62	69	可设置	DMA2 通道 6	DMA2 通道 6 全局中断	0x0000_0138
63	70	可设置	DMA2 通道 7	DMA2 通道 7 全局中断	0x0000_013C
64	71	可设置	DMA2 通道 8	DMA2 通道 8 全局中断	0x0000_0140
65	72	可设置	DMA2 通道 9	DMA2 通道 9 全局中断	0x0000_0144



位置	优先级	优先级类型	名称	说明	地址
66	73	可设置	DMA2 通道 10	DMA2 通道 10 全局中断	0x0000_0148
67	74	可设置	DMA2 通道 11	DMA2 通道 11 全局中断	0x0000_014C
68	75	可设置	DMA2 通道 12	DMA2 通道 12 全局中断	0x0000_0150
69	76	可设置	DMA2 通道 13	DMA2 通道 13 全局中断	0x0000_0154
70	77	可设置	DMA2 通道 14	DMA2 通道 14 全局中断	0x0000_0158
71	78	可设置	DMA2 通道 15	DMA2 通道 15 全局中断	0x0000_015C
72	79	可设置	DMA1 通道 7	DMA1 通道 7 全局中断	0x0000_0160
73	80	可设置	DMA1 通道 8	DMA1 通道 8 全局中断	0x0000_0164
74	81	可设置	DMA1 通道 9	DMA1 通道 9 全局中断	0x0000_0168
75	82	可设置	DMA1 通道 10	DMA1 通道 10 全局中断	0x0000_016C
76	83	可设置	DMA1 通道 11	DMA1 通道 11 全局中断	0x0000_0170
77	84	可设置	DMA1 通道 12	DMA1 通道 12 全局中断	0x0000_0174
78	85	可设置	DMA1 通道 13	DMA1 通道 13 全局中断	0x0000_0178
79	86	可设置	DMA1 通道 14	DMA1 通道 14 全局中断	0x0000_017C
80	87	可设置	QSPIEX	QSPIEX 全局中断	0x0000_0180
81	88	可设置	TIM1B_BRK	TIM1B 刹车中断	0x0000_0184
82	89	可设置	TIM1B_UP	TIM1B 更新中断	0x0000_0188
83	90	可设置	TIM1B_TRG_COM	TIM1B 触发和通信中断	0x0000_018C
84	91	可设置	TIM1B_CC	TIM1B 捕获比较中断	0x0000_0190
85	92	可设置	PCA	PCA 全局中断	0x0000_0194
86	93	可设置	LPTIM	LPTIM 全局中断	0x0000_0198
87	94	可设置	LPUART	LPUART 中断	0x0000_019C
88	95	可设置	LVD	LVD 中断	0x0000_01A0
89	96	可设置	TRNG	TRNG 中断	0x0000_01A4
90	97	可设置	AES	AES 中断	0x0000_01A8



## 10.2 外部中断/事件控制器(EXTI)

外部中断/事件控制器(EXTI)管理了控制器的 25 个中断/事件线。每个中断/事件线都对应有一个边沿检测器，可以实现输入信号的上升沿检测和下降沿的检测。EXTI 可以实现对每个中断/事件线进行单独配置，可以单独配置为中断或者事件，以及触发事件的属性。

### 10.2-1 主要特性

EXTI 控制器的主要特性如下：

- 每个中断/事件都有独立的触发和屏蔽
- 每个中断线都有专用的状态位
- 支持多达 20 个软件的中断/事件请求
- 检测脉冲宽度低于 APB2 时钟宽度的外部信号。参见数据手册中电气特性部分的相关参数。



### 10.2-2 框图

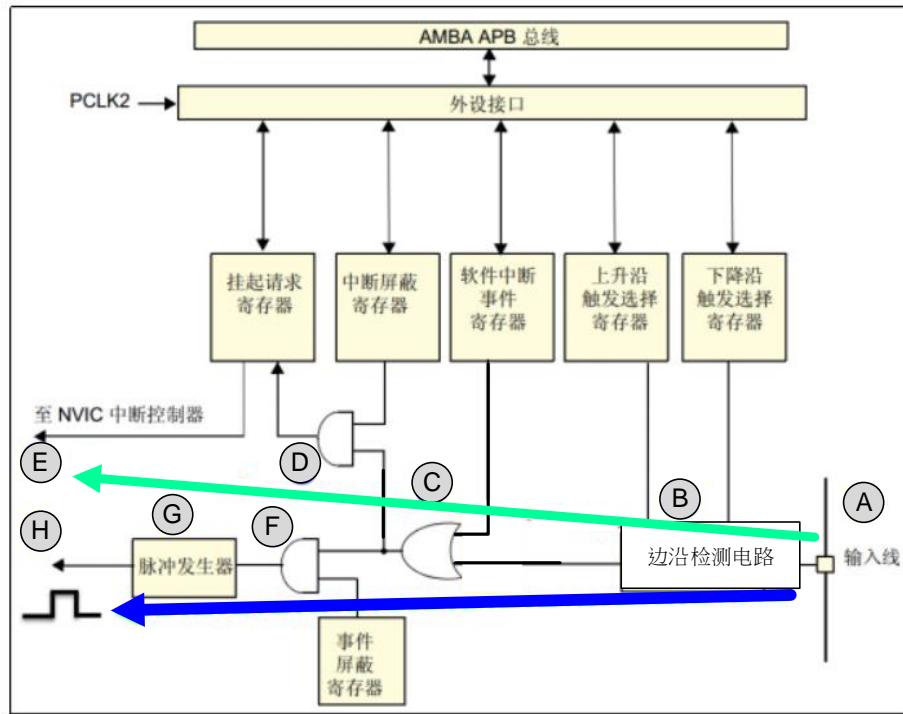


Figure 10-2 EXTI 功能框图

EXTI 可分为两大部分功能，一个是产生中断，另一个是产生事件，这两个功能从硬件上就有所不同。首先我们来看图 17-1 中红色虚线指示的电路流程。它是一个产生中断的线路，最终信号流入到 NVIC 控制器内。

**编号 A** 是输入线，EXTI 控制器有 25 个中断/事件输入线，这些输入线可以通过寄存器设置为任意一个 GPIO，也可以是一些外设的事件，这部分内容我们将在后面专门讲解。输入线一般是存在电平变化的信号。

**编号 B** 是一个边沿检测电路，它会根据上升沿触发选择寄存器(EXTI\_RTSR)和下降沿触发选择寄存器(EXTI\_FTSR)对应位的设置来控制信号触发。边沿检测电路以输入线作为信号输入端，如果检测到有边沿跳变就输出有效信号 A 给编号 C 电路，否则输出无效信号 0。而 EXTI\_RTSR 和 EXTI\_FTSR 两个寄存器可以控制控制器需要检测哪些类型的电平跳变过程，可以是只有上升沿触发、只有下降沿触发或者上升沿和下降沿都触发。

**编号 C** 电路实际就是一个或门电路，它一个输入来自编号 B 电路，另外一输入来自软件中断事件寄存器(EXTI\_SWIER)。EXTI\_SWIER 允许我们通过程序控制就可以启动中断/事件线，这在某些地方非常有用。我们知道或门的作用就是有 1 就为 1，所以这两个输入随便一个有有效信号 1 就可以输出 1 给编号 D 和编号 F 电路。



**编号 D** 电路是一个与门电路，它一个输入编号 C 电路，另外一个输入来自中断屏蔽寄存器(EXTI\_IMR)。与门电路要求输入都为 1 才输出 1，导致的结果如果 EXTI\_IMR 设置为 0 时，那不管编号 3 电路的输出信号是 1 还是 0，最终编号 4 电路输出的信号都为 0；如果 EXTI\_IMR 设置为 1 时，最终编号 D 电路输出的信号才由编号 C 电路的输出信号决定，这样我们可以简单的控制 EXTI\_IMR 来实现是否产生中断的目的。编号 D 电路输出的信号会被保存到挂起寄存器(EXTI\_PR)内，如果确定编号 D 电路输出为 1 就会把 EXTI\_PR 对应位置 1。

**编号 E** 是将 EXTI\_PR 寄存器内容输出到 NVIC 内，从而实现系统中断事件控制。

接下来我们来看看绿色虚线指示的电路流程。它是一个产生事件的线路，最终输出一个脉冲信号。

产生事件线路是在编号 C 电路之后与中断线路有所不同，之前电路都是共用的。编号 F 电路是一个与门，它一个输入编号 C 电路，另外一个输入来自事件屏蔽寄存器(EXTI\_EMR)。如果 EXTI\_EMR 设置为 0 时，那不管编号 C 电路的输出信号是 1 还是 0，最终编号 6 电路输出的信号都为 0；如果 EXTI\_EMR 设置为 1 时，最终编号 F 电路输出的信号才由编号 C 电路的输出信号决定，这样我们可以简单的控制 EXTI\_EMR 来实现是否产生事件的目的。

**编号 G** 是一个脉冲发生器电路，当它的输入端，即编号 F 电路的输出端，是一个有效信号 1 时就会产生一个脉冲；如果输入端是无效信号就不会输出脉冲。

**编号 H** 是一个脉冲信号，就是产生事件的线路最终的产物，这个脉冲信号可以给其他外设电路使用，比如定时器 TIM、模拟数字转换器 ADC 等等。

产生中断线路目的是把输入信号输入到 NVIC，进一步会运行中断服务函数，实现功能，这样是软件级的。而产生事件线路目的就是传输一个脉冲信号给其他外设使用，并且是电路级别的信号传输，属于硬件级的。

注: EXTI 是在 APB2 总线上的，在编程时候需要注意到这点。

### 10.2.3 唤醒事件管理

ZB32F103 可以处理外部或内部事件来唤醒内核(WFE)。唤醒事件可以通过下述配置产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时在 Cortex-M3 的系统控制寄存器中使能 SEVONPEND 位。当 CPU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位(在 NVIC 中断清除挂起寄存器中)。
- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

在互联型产品中，以太网唤醒事件同样具有 WFE 唤醒功能。

使用外部 I/O 端口作为唤醒事件，请参见 10.2.4 节的功能说明



#### 10.2.4 功能说明

要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写‘1’允许中断请求。当外部中断线上发生了期待的边沿时，将产生一个中断请求，对应的挂起位也随之被置‘1’。在挂起寄存器的对应位写‘1’，将清除该中断请求。如果需要产生事件，必须先配置好并使能事件线。根据需要的边沿检测通过设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写‘1’允许事件请求。当事件线上发生了需要的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置‘1’。通过在软件中断/事件寄存器写‘1’，也可以通过软件产生中断/事件请求。

##### 硬件中断选择

通过下面的过程来配置 25 个线路做为中断源：

- 配置 25 个中断线的屏蔽位(EXTI\_IMR)
- 配置所选中断线的触发选择位(EXTI\_RTSR 和 EXTI\_FTSR);
- 配置对应到外部中断控制器(EXTI)的 NVIC 中断通道的使能和屏蔽位，使得 25 个中断线中的请求可以被正确地响应。

##### 硬件事件选择

通过下面的过程，可以配置 25 个线路为事件源

- 配置 25 个事件线的屏蔽位(EXTI\_EMR)
  - 配置事件线的触发选择位(EXTI\_RTSR 和 EXTI\_FTSR) 软件中断/事件的选择
- 25 个线路可以被配置成软件中断/事件线。下面是产生软件中断的过程：
- 配置 25 个中断/事件线屏蔽位(EXTI\_IMR, EXTI\_EMR)

设置软件中断寄存器的请求位(EXTI\_SWIER)



### 10.2-5 外部中断/事件线路映像

通用 I/O 端口以下图的方式连接到 16 个外部中断/事件线上：

如下图 外部中断通用 I/O 映像

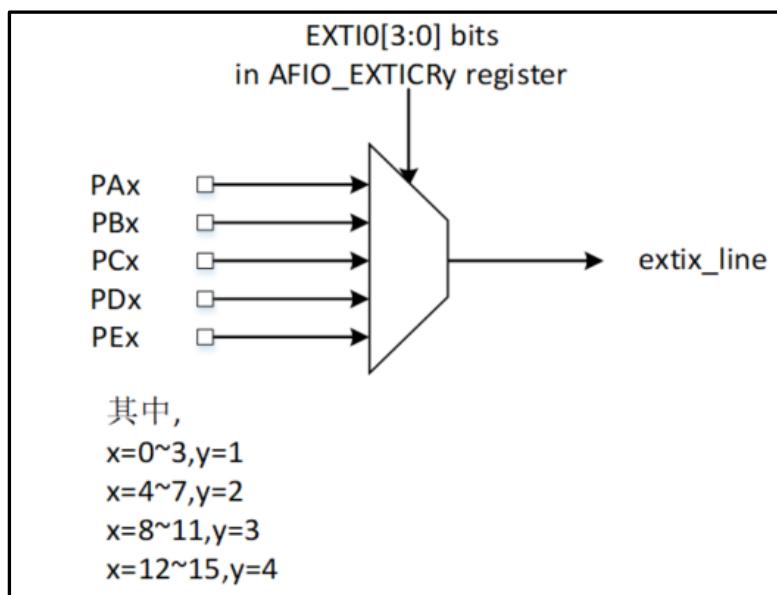


Figure 10-3 外部中断通用 I/O 映像图

Note: 1. To configure the AFIO\_EXTICRx for the mapping of external interrupt/event lines onto GPIOs, the AFIO clock should first be enabled.

All the 24 exti lines are connected as follow table:

EXTI lines	Line source
exti line 0~15	GPIOs
exti line 16	PVD output
exti line 17	RTC Alarm event
exti line 18	USB Wakeup event
exti line 19	USART1 Wakeup event
exti line 20	USART2 Wakeup event
exti line 21	AWK Wakeup event
exti line 22	UART4 Wakeup event
exti line 23	UART5 Wakeup event
exti line 24	LPUART Wakeup event

Table 10-2 其他 EXTI 线的连接表



### 10.3 EXTI 寄存器列表

EXTI 基地址: 0x4001 0400

Table 10-3 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	NVIC_ISER	中断屏蔽寄存器(EXTI_IMR)	0x0000 0000
0x04	NVIC_ICER	事件屏蔽寄存器(EXTI_EMR)	0x0000 0000
0x08	NVIC_ISPR	上升沿触发选择寄存器(EXTI_RTSR)	0x0000 0000
0x0C	NVIC_ICPR	下降沿触发选择寄存器(EXTI_FTSR)	0x0000 0000
0x10	NVIC_IPR0	软件中断事件寄存器(EXTI_SWIER)	0x0000 0000
0x14	NVIC_IPR1	挂起寄存器(EXTI_PR)	0x0000 0000



## 10.4 寄存器说明

### 10.4-1 中断屏蔽寄存器(EXTI\_IMR)

地址偏移: 0x00

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:25	Reserved	保留，必须始终保持为复位状态(0)。	0x0000 0000	ro
24:0	MRx	<b>MRx:</b> 线 x 上的中断屏蔽 (Interrupt Mask on line x) 0: 屏蔽来自线 x 上的 中断请求； 1: 开放来自线 x 上的 中断请求。 。	0x0	rw



#### 10.4-2 事件屏蔽寄存器(EXTI\_EMR)

地址偏移: 0x04

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:25	Reserved	Reserved, must be kept cleared.	0x000 0000	ro
24:0	MRx	<b>MRx:</b> 线 x 上的事件屏蔽 (Event Mask on line x) 0: 屏蔽来自线x上的事件请求; 1: 开放来自线x上的事件请求	0x0	rw



#### 10.4.3 上升沿触发选择寄存器(EXTI\_RTSR)

地址偏移: 0x08 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TR 24	TR 23	TR 22	TR 21	TR 20	TR 19	TR 18	TR 17
ro	ro	ro	ro	ro	ro	ro	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR 15	TR 14	TR 13	TR 12	TR. 11	TR 10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:25	Reserved	保留, 必须始终保持为复位状态(0)。	0x0000 0000	ro
24:0	TRx	<b>TRx:</b> 线 x 上的上升沿触发事件配置位 (Rising trigger event configuration bit of line x) 0: 禁止输入线 x 上的上升沿触发(中断和事件) 1: 允许输入线 x 上的上升沿触发(中断和事件)	0x0	rw

注意: 外部唤醒线是边沿触发的, 这些线上不能出现毛刺信号。

在写 **EXTI\_RTSR** 寄存器时, 在外部中断线上的上升沿信号不能被识别, 挂起位也不会被置位。

在同一中断线上, 可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。



#### 10.4.4 下降沿触发选择寄存器(EXTI\_FTSR)

地址偏移: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							TF2 3	TF2 3	TF2 2	TF2 1	TF2 0	TF1 9	TF1 8	TF1 7	TF1 6
ro	ro	ro	ro	ro	ro	ro	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TF1 5	TF1 4	TF1 3	TF1 2	TF1 1	TF1 0	TF 9	TF8	TF7	TF6	TF5	TF4	TF3	TF2	TF1	TF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:25	Reserved	保留，必须始终保持为复位状态(0)。	0x0000 0000	ro
24:0	TFx	<b>TFx:</b> 线 x 上的下降沿触发事件配置位 (Falling trigger event configuration bit of line x) 0: 禁止输入线 x 上的下降沿触发(中断和事件) 1: 允许输入线 x 上的下降沿触发(中断和事件)	0x0	rw



#### 10.4.5 软件中断事件寄存器(EXTI\_SWIER)

地址偏移: 0x10 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:24	Reserved	保留，必须始终保持为复位状态(0)。	0x0000 0000	ro
24:0	SWIERx	<b>SWIERx:</b> 线 x 上的软件中断 (Software interrupt on line x)  当该位为' 0' 时，写' 1' 将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则此时将产生一个中断。  注：通过清除 EXTI_PR 的对应位(写入' 1' )，可以清除该位为' 0' 。	0x0	rw



#### 10.4.6 挂起寄存器(EXTI\_PR)

地址偏移: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							PR 24	PR 23	PR 22	PR 21	PR 20	PR 19	PR 18	PR 17	PR 16
ro	rc_ w1														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rc_ w1															

位	标记	功能描述	复位值	读写
31:25	Reserved	保留，必须始终保持为复位状态(0)。	0x0000 0000	ro
24:0	PRx	<p><b>PRx:</b> 挂起位 (Pending bit)</p> <p>0: 没有发生触发请求</p> <p>1: 发生了选择的触发请求</p> <p>当在外部中断线上发生了选择的边沿事件，该位被置‘1’。在该位中写入‘1’可以清除它，也可以通过改变边沿检测的极性清除。</p>	0x0	rc_w1



## 【11】通用和复用功能 I/O(GPIO 和 AFIO)

### 11.1 GPIO 简介

通用输入/输出口用于芯片和外部进行数据传输，共有 4 组 GPIO：GPIOA、GPIOB、GPIOC 和 GPIOD。4 组 GPIO 的功能基本相同，可以通过配置将 GPIO 映像到对应芯片引脚，每个引脚可以被独立配置为数字输入或者输出口，也可以被配置为模拟输入。另外，还可以配置成外部中断、片上外设输入/输出等复用功能。同一时刻一个引脚仅可被映射一个复用功能，通过端口复用功能寄存器(AFIO)配置。

每个端口都可以配置成内部上拉(pull up)/下拉(pull down)的输入/输出，高阻输入(floating input)，推挽输出(push-pull output)，开漏输出(open drain output)，增强驱动能力输出。芯片复位后端口复位为高阻输入，目的是防止芯片在异常复位时，导致外部器件产生异常动作。为了避免高阻输入而产生的漏电，用户要在芯片启动之后对端口进行相应配置(配置成内部拉高输入或者输出)。端口被配成模拟端口后，数字功能被隔离，不能输出数字“1”和“0”，CPU 读取端口的结果为“0”。指定端口都可以提供外部中断，并且每个中断都可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发或者任意边沿触发，支持边沿模式下的输入消抖。支持在工作模式/睡眠模式/深度睡眠模式下产生中断。

### 11.2 GPIO 主要特性

1. 输出状态：带有上拉或下拉的推挽输出或开漏输出
2. 从数据寄存器或外设(复用功能输出)输出数据
3. 可配的每个 I/O 口的速度
4. 输入状态：浮空、上拉/下拉、模拟输入
5. 从数据寄存器或外设输入数据(复用功能输出)
6. 输出置位/复位寄存器
7. 模拟功能引脚/调试引脚/数字通用引脚/数字功能引脚复用
8. 允许 GPIO 口和外设引脚的高灵活性复用



### 11.3 GPIO 功能描述

每个 GPIO 端口有两个 32 位配置寄存器(GPIOx\_CRL, GPIOx\_CRH), 两个 32 位数据寄存器(GPIOx\_IDR 和 GPIOx\_ODR), 一个 32 位置位/复位寄存器(GPIOx\_BSRR), 一个 16 位复位寄存器(GPIOx\_BRR)和一个 32 位锁定寄存器(GPIOx\_LCKR)。

根据数据手册中列出的每个 I/O 端口的特定硬件特征, GPIO 端口的每个位可以由软件分别配置成多种模式。:

1. 浮空输入
2. 上拉输入
3. 下拉输入
4. 模拟输入
5. 具有上拉或下拉能力的开漏输出
6. 具有上拉或下拉能力的推挽输出
7. 复用功能且具有上拉或下拉能力的推挽输出
8. 复用功能且具有上拉或下拉能力的开漏输出

每个 I/O 端口位可以自由编程, I/O 端口寄存器可按 32 位字访问。GPIOx\_ODSET, GPIOx\_ODCLR 寄存器允许对 GPIOx\_ODR 进行位读/改写操作。这种情况下, 可以避免在读和更改访问之间产生中断而发生的异常。

图 11-1 给出了一个标准 I/O 端口的基本结构。

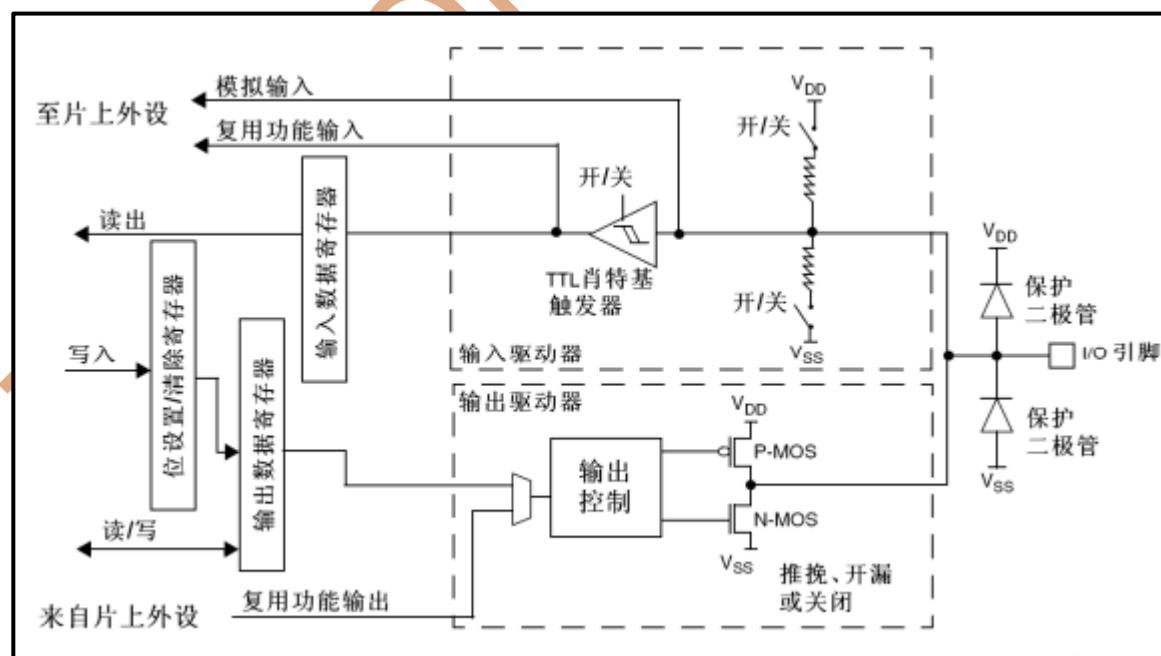


Figure 11-1 标准 I/O 端口的位基本结构

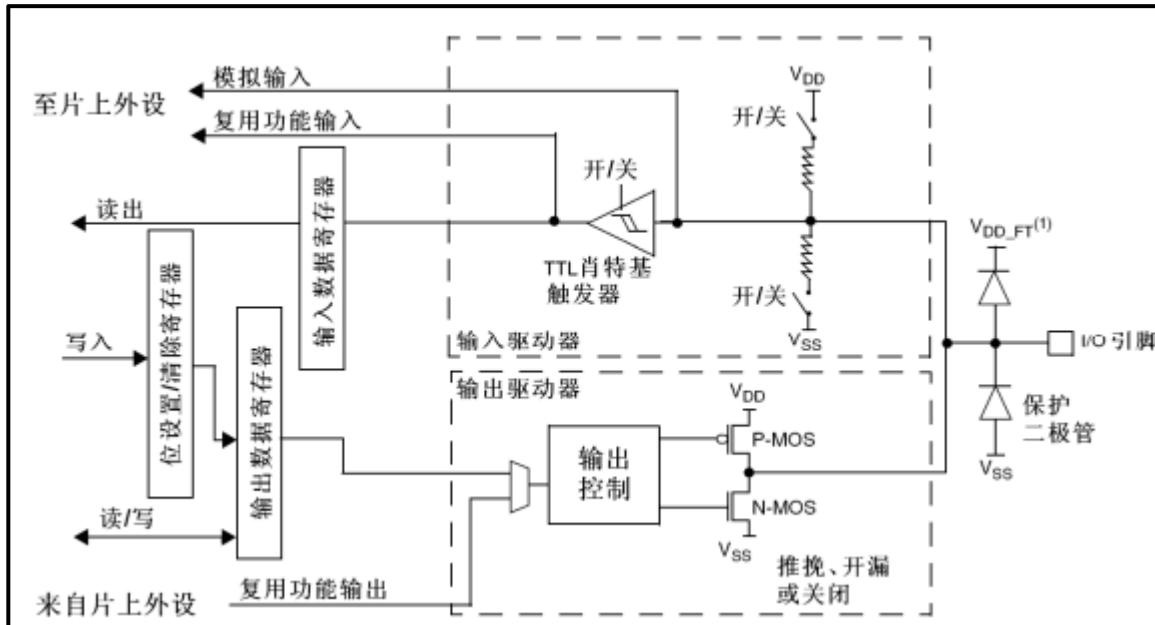


Figure 11-2 5 伏兼容 I/O 端口位的位基本结构

Note : (1) VDD\_FT 对 5 伏容忍 I/O 脚是特殊的, 它与 VDD 不同

表 11-1 给出了 GPIO 口和外设引脚的复用功能映射

Table 11-1 端口位配置表

配置模式		CNF1	CNF0	MODE1	MODE0	PxODR 寄存器
通用输出	推挽(Push-Pull)	0	0	00: Revered 01: 大输出速度为 10MHz 10: 大输出速度为 2MHz 11: 大输出速度为 50MHz		0 或 1
	开漏(Open-Drain)		1			0 或 1
复用功能输出	推挽(Push-Pull)	1	0			不使用
	开漏(Open-Drain)		1			不使用
输入	模拟输入	0	0	00		不使用
	浮空输入		1			不使用
	下拉输入	1	0			0



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

上拉输入

1

ZBIT CONFIDENTIAL



### 11.3-1 通用 I/O(GPIO)

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成浮空输入模式( $CNFx[1:0]=012$ ,  $MODEx[1:0]=00b$ )。

复位后，JTAG 引脚被置于输入上拉或下拉模式：

- PA15: JTDI 置于上拉模式
- PA14: JTCK 置于下拉模式
- PA13: JTMS 置于上拉模式
- PB4: JNTRST 置于上拉模式

当作为输出配置时，写到输出数据寄存器上的值( $GPIOx_ODR$ )输出到相应的 I/O 引脚。可以以推挽模式或开漏模式(当输出 0 时，只有 N-MOS 被打开)使用输出驱动器。

输入数据寄存器( $GPIOx_IDR$ )在每个 APB2 时钟周期捕捉 I/O 引脚上的数据。

所有 GPIO 引脚有一个内部弱上拉和弱下拉，当配置为输入时，它们可以被激活也可以被断开。

### 11.3-2 单独的位设置或位清除

当对  $GPIOx_ODR$  的个别位编程时，软件不需要禁止中断：在单次 APB2 写操作里，可以只更改一个或多个位。

这是通过对“置位/复位寄存器”( $GPIOx_BSRR$ , 复位是  $GPIOx_BRR$ )中想要更改的位写‘1’来实现的。没被选择的位将不被更改。

### 11.3-3 外部中断/唤醒线

所有端口都有外部中断能力。为了使用外部中断线，端口必须配置成输入模式。更多的关于外部中断的信息，参考：

- 外部中断/事件控制器(EXTI): [link](#)
- 唤醒事件管理: [link](#)

### 11.3-4 复用功能(AF)

使用默认复用功能前必须对端口位配置寄存器编程。

- 对于复用的输入功能，端口必须配置成输入模式(浮空、上拉或下拉)且输入引脚必须由外部驱动。

注意：也可以通过软件来模拟复用功能输入引脚，这种模拟可以通过对 GPIO 控制器编程来实现。此时，端口应当被设置为复用功能输出模式。显然，这时相应的引脚不再由外部驱动，而是通过 GPIO 控制器由软件来驱动。

- 对于复用输出功能，端口必须配置成复用功能输出模式(推挽或开漏)。
- 对于双向复用功能，端口位必须配置复用功能输出模式(推挽或开漏)。这时，输入驱动器被配置成浮空输入模式。



如果把端口配置成复用输出功能，则引脚和输出寄存器断开，并和片上外设的输出信号连接。

如果软件把一个 GPIO 脚配置成复用输出功能，但是外设没有被激活，它的输出将不确定。

ZBT CONFIDENTIAL



### 11.3-5 软件重新映射 I/O 复用功能

为了使不同器件封装的外设 I/O 功能的数量达到最优，可以把一些复用功能重新映射到其他一些脚上。这可以通过软件配置相应的寄存器来完成(参考 AFIO 寄存器描述)。这时，复用功能就不再映射到它们的原始引脚上了。

### 11.3-6 输出配置

当 I/O 端口被配置为输出时：

- 输出缓冲器被激活
  - 开漏模式：输出寄存器上的‘0’激活 N-MOS，而输出寄存器上的‘1’将端口置于高阻状态(P-MOS 从不被激活)。
  - 推挽模式：输出寄存器上的‘0’激活 N-MOS，而输出寄存器上的‘1’将激活 P-MOS。
- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 出现在 I/O 脚上的数据在每个 APB2 时钟被采样到输入数据寄存器
- 在开漏模式时，对输入数据寄存器的读访问可得到 I/O 状态
- 在推挽式模式时，对输出数据寄存器的读访问得到最后一次写的值。

下图给出了 I/O 端口位的输出配置。

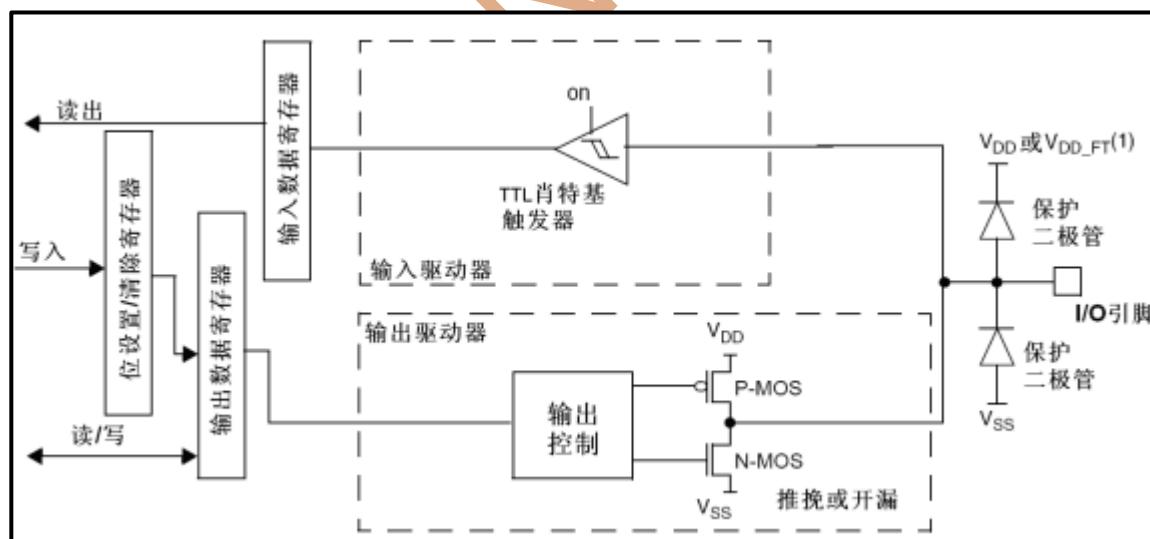


Figure 11-3 输出配置

Note: (1) VDD\_FT 对 5 伏兼容 I/O 脚是特殊的，它与 VDD 不同



### 11.3-7 输入配置

当 I/O 端口配置为输入时：

- 输出缓冲器被禁止
- 施密特触发输入被激活
- 根据输入配置(上拉, 下拉或浮动)的不同, 弱上拉和下拉电阻被连接
- 出现在 I/O 脚上的数据在每个 APB2 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态

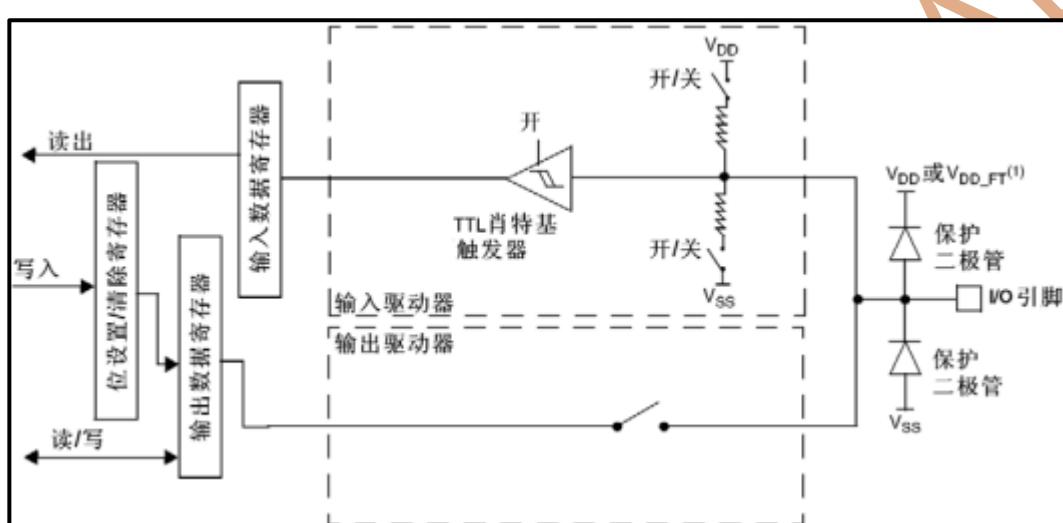


Figure 11-4 输出配置

Note: (1) V<sub>DD\_FT</sub> 对 5 伏兼容 I/O 脚是特殊的, 它与 V<sub>DD</sub> 不同

### 11.3-8 GPIO 锁定机制

锁定机制允许冻结 I/O 配置。当在一个端口位上执行了锁定(LOCK)程序，在下一次复位之前，将不能再更改端口位的配置。



### 11.3-9 复用功能配置

当 I/O 端口被配置为复用功能时：

- 在开漏或推挽式配置中，输出缓冲器被打开
- 内置外设的信号驱动输出缓冲器(复用功能输出)
- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 在每个 APB2 时钟周期，出现在 I/O 脚上的数据被采样到输入数据寄存器
- 开漏模式时，读输入数据寄存器时可得到 I/O 口状态
- 在推挽模式时，读输出数据寄存器时可得到最后一次写的值

8.4 节-AFIO 寄存器描述。

下图示出了 I/O 端口位的复用功能配置。详见

一组复用功能 I/O 寄存器允许用户把一些复用功能重新映象到不同的引脚。

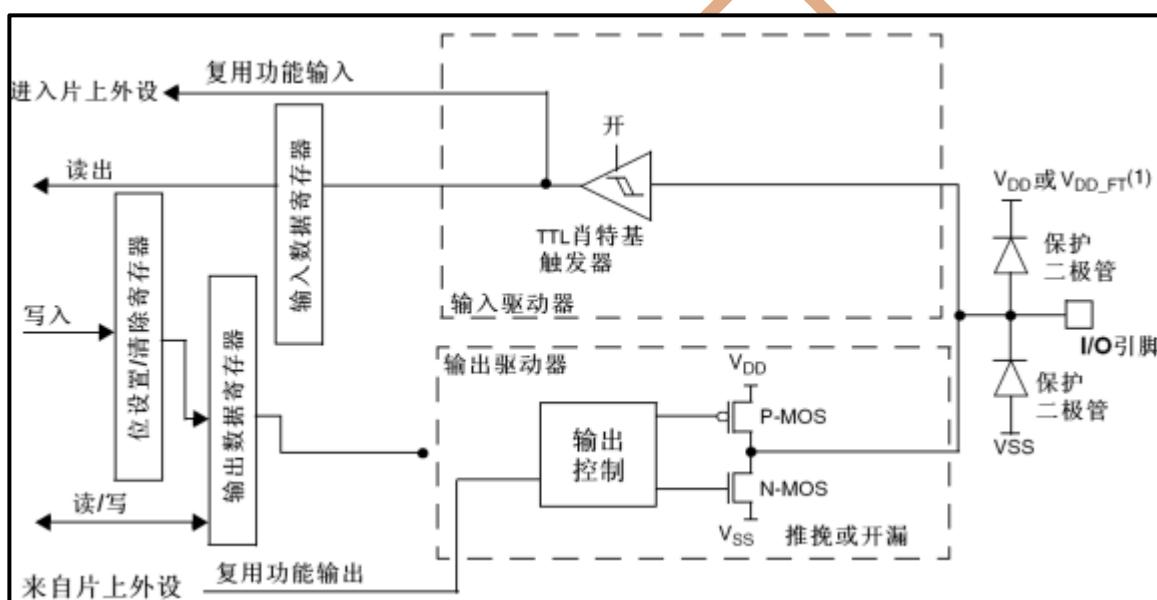


Figure 11-5 复用功能配置

Note: (1) V<sub>DD\_FT</sub> 对 5 伏兼容 I/O 脚是特殊的，它与 V<sub>DD</sub> 不同

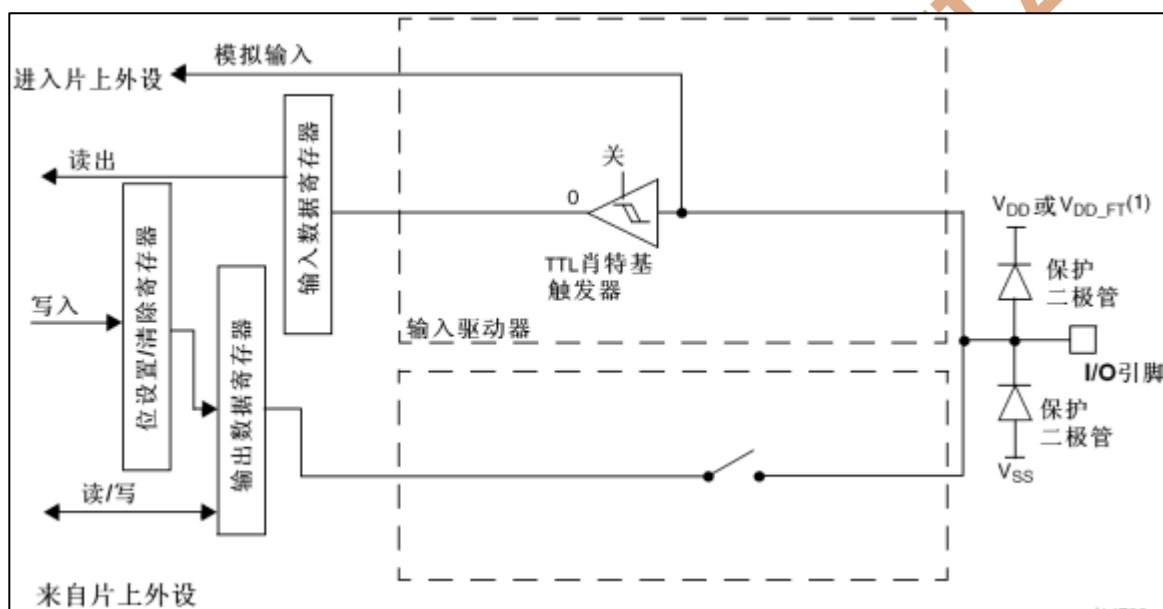


### 11.3-10 模拟输入配置

当 I/O 端口被配置为模拟输入配置时：

- 输出缓冲器被禁止；
- 禁止施密特触发输入，实现了每个模拟 I/O 引脚上的零消耗。施密特触发输出值被强置为' 0' ；
- 弱上拉和下拉电阻被禁止；
- 读取输入数据寄存器时数值为' 0' 。

下图示出了 I/O 端口位的高阻抗模拟输入配置：



Note: (1) VDD\_FT 对 5 伏兼容 I/O 脚是特殊的，它与 VDD 不同



### 11.3-11 外设的 GPIO 配置

#### 11.3-11.1 TIM1,8,12 的引脚配置

TIM1/TIM1A 引脚	配置	GPIO 配置
TIM1/8_CHx	输入捕获通道 x	浮空输入
	输出比较通道 x	推挽复用输出
TIM1/8_CHxN	互补输出通道 x	推挽复用输出
TIM1/8_BKIN	刹车输入	浮空输入
TIM1/8_ETR	外部触发时钟输入	浮空输入

Table 11-2 高级定时器 TIM1/TIM/TIM1B 引脚配置表

#### 11.3-11.2 TIM2,3,4,5 的引脚配置

TIM2/3/4/5 引脚	配置	GPIO 配置
TIM2/3/4/5_CHx	输入捕获通道 x	浮空输入
	输出比较通道 x	推挽复用输出
TIM2/3/4/5_ETR	外部触发时钟输入	浮空输入

Table 11-3 通用定时器 TIM2/3/4/5 引脚配置表

#### 11.3-11.1 PCA 的引脚配置

TIM2/3/4/5 引脚	配置	GPIO 配置
PCA_CHx	输入捕获通道 x	浮空输入
	输出比较通道 x	推挽复用输出
PCA_ECI	外部时钟输入信号	浮空输入

Table 11-4 通用定时器 TIM2/3/4/5 引脚配置表



### 11.3-11.2 USART 的引脚配置

USART 引脚	配置	GPIO 配置
USARTx_TX	全双工模式	推挽复用输出
	半双工同步模式	推挽复用输出
USARTx_RX	全双工模式	浮空输入或带上拉输入
	半双工同步模式	未用，可作为通用 I/O
USARTx_CK	同步模式	推挽复用输出
USARTx_RTS	硬件流量控制	推挽复用输出
USARTx_CTS	硬件流量控制	浮空输入或带上拉输入

Table 11-5USART 引脚配置表

### 11.3-11.3 SPI 的引脚配置

SPI 引脚	配置	GPIO 配置
SPIx_SCK	主模式	推挽复用输出
	从模式	浮空输入
SPIx_MOSI	全双工模式/主模式	推挽复用输出
	全双工模式/从模式	浮空输入或带上拉输入
	简单的双向数据线/主模式	推挽复用输出
	简单的双向数据线/从模式	未用，可作为通用 I/O
SPIx_MISO	全双工模式/主模式	浮空输入或带上拉输入
	全双工模式/从模式	推挽复用输出
	简单的双向数据线/主模式	未用，可作为通用 I/O
	简单的双向数据线/从模式	推挽复用输出
SPIx_NSS	硬件主/从模式	浮空输入或带上拉输入或带下拉输入
	硬件主模式/NSS 输出使能	推挽复用输出
	软件模式	未用，可作为通用 I/O



Table 11-6 SPI 引脚配置表

ZBT CONFIDENTIAL



### 11.3-11.4 I2S 的引脚配置

I2S 引脚	配置	GPIO 配置
I2Sx_WS	主模式	推挽复用输出
	从模式	浮空输入
I2Sx_CK	主模式	推挽复用输出
	从模式	浮空输入
I2Sx_SD	发送器	推挽复用输出
	接收器	浮空输入或带上拉输入或带下拉输入
I2Sx_MCK	主模式	推挽复用输出
	从模式	未用，可作为通用 I/O

Table 11-7 I2S 引脚配置表

### 11.3-11.5 CAN 的引脚配置

BxCAN 引脚	GPIO 配置
CAN_TX	推挽复用输出
CAN_RX	浮空输入或带上拉输入

Table 11-8 CAN 引脚配置表

### 11.3-11.6 USB 的引脚配

USB 引脚	GPIO 配置



USB\_DM / USB\_DP

一旦使能了 USB 模块，这些引脚会自动连接到内部 USB 收发器

Table 11-9 USB 引脚配置表

ZBT CONFIDENTIAL



### 11.3-11.7 SDIO 的引脚配置

SDIO 引脚	GPIO 配置
SDIO_CK	推挽复用输出
SDIO_CMD	推挽复用输出
SDIO[D7:D0]	推挽复用输出

Table 11-10 SDIO 引脚配置表

### 11.3-11.8 ADC/DAC 的引脚配置

ADC/DAC 引脚	GPIO 配置
ADC/DAC	模拟输入

Table 11-11 SDIO 引脚配置表

### 11.3-11.9 其它 I/O 功能的引脚配置

引脚	复用功能	GPIO 配置
TAMPER-RTC	RTC 输出	当配置 BKP_CR 和 BKP_RTCCR 寄存器时，由硬件强制设置
	侵入事件输入	
MCO	时钟输出	推挽复用输出
EXTI 输入线	外部中断输入	浮空输入或带上拉输入或带下拉输入

Table 11-12 SDIO 引脚配置表



#### 11.4 复用功能 I/O 和调试配置(AFIo)

为了优化 64 PIN 的外设数目，可以把一些复用功能重新映射到其他引脚上。设置复用重映射和调试 I/O 配置寄存器(AFIo\_MAPR/GPIOx\_AFR)实现引脚的重新映射。这时，复用功能不再映射到它们的原始分配上。

[Note] AFIo\_MAPR: ST MODE(AFMOD=0) , GPIOx\_AFR : PIN Mode Select (AFMOD=1)

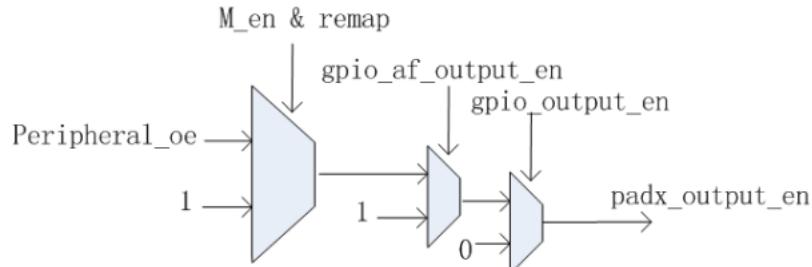


Figure 11-7 PAD output enable logic

Note : M\_en : 外設置能位原訊號

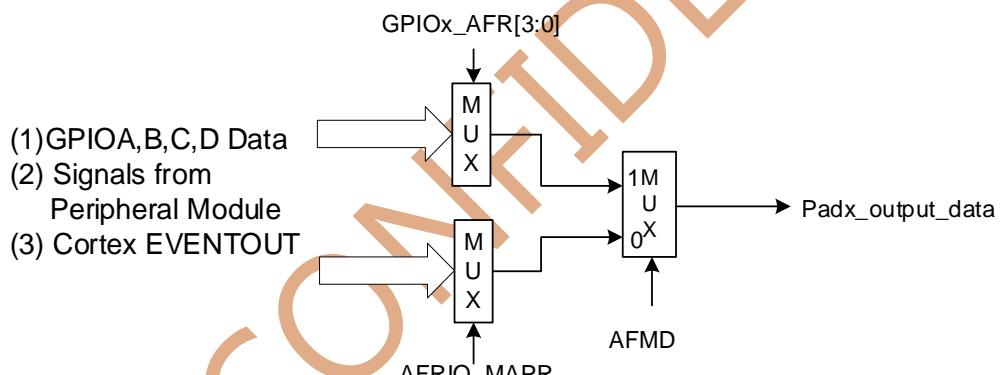


Figure 11-8 PAD output data logic

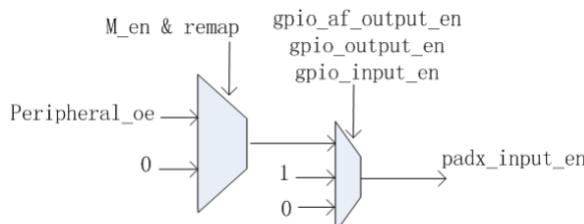


Figure 11-9 PAD input enable logic

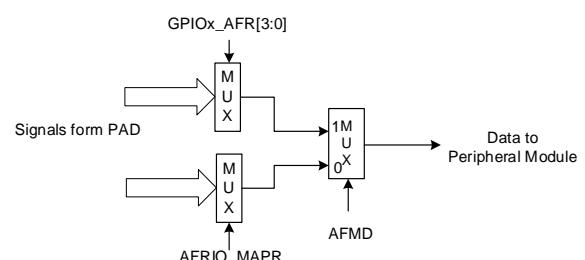


Figure 11-10 PAD input data logic



#### 11.4-1 使用 OSC32\_IN/OSC32\_OUT 作为 GPIO 端口 PC14/PC15

当 LSE 振荡器关闭时，LSE 振荡器引脚 OSC32\_IN/OSC32\_OUT 可以分别用做 GPIO 的 PC14/PC15，LSE 功能始终优先于通用 I/O 口的功能。

Note：1. 当关闭 1.8V 电压区(进入待机模式)或后备区域使用 VBAT 供电(不再有 VDD 供电)时，不能使用 PC14/PC15 的 GPIO 口功能；

2. 参见第 4.1.2 节有关 I/O 口使用的限制

#### 11.4-2 使用 OSC\_IN/OSC\_OUT 引脚作为 GPIO 端口 PD0/PD1

外部振荡器引脚 OSC\_IN/OSC\_OUT 可以用做 GPIO 的 PD0/PD1，通过设置复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR)实现。这个重映射只适用于 36、48 和 64 脚的封装。

#### 11.4-3 CAN 复用功能重映射

CAN 信号可以被映射到端口 A、端口 B 或端口 D 上，如下表所示。对于端口 D，在 36、48 和 64 脚的封装上没有重映射功能。

复用功能 <sup>(1)</sup>	CAN_REMAP[1:0] = " 0 0"	CAN_REMAP[1:0] = " 10" <sup>(2)</sup>	CAN_REMAP[1:0] = " 11" <sup>(3)</sup>
CAN1_RX 或 AN_RX	PA11	PB8	PD0
CAN1_TX 或 AN_TX	PA12	PB9	PD1

Table 11-13 CAN 引脚配置表

#### 11.4-4 JTAG/SWD 复用功能重映射

调试接口信号被映射到 GPIO 端口上，如下表所示。

复用功能		GPIO 端口	
JTAG	SWD	<u>SWDPORT=2' b01</u>	<u>SWDPORT=2' b1x</u>
JTMS	SWDIO	PA13	PA11
JTCK/	SWCLK	PA14	PA12
JTDI			PA15



JTDO	TRACESWO	PB3
JNTRST		PB4

Table 11-14 调试接口信号引脚配置表

ZBT CONFIDENTIAL



为了在调试期间可以使用更多 GPIOs，通过设置复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR)的 SWJ\_CFG[2:0]位，可以改变上述重映像配置。参见下表。

SWJ_CFG [2:0]	SWD_EN	可能的调试端口	SWJ I/O 引脚分配				
			PA13/ JTMS/ SWDIO	PA14/ JTCK/ SWCLK	PA15/ JTDI	PB3/ JTDO/ TRACESWO	PB4/ NJTRST
000	1	SWD	I/O 不可用	I/O 不可用	I/O 可用	I/O 可用	I/O 可用
001	1	SWD, TRACESWO(跟踪)	I/O 不可用	I/O 不可用	I/O 可用	I/O 不可用	I/O 可用
010	1	SWD	I/O 不可用	I/O 不可用	I/O 可用	I/O 可用	I/O 可用
100	0	关闭 JTAG- DP, 关闭 SW- DP	I/O 可用	I/O 可用	I/O 可用	I/O 不可用	I/O 可用
其它		禁用					

Table 11-15 调试端口映像表

#### 11.4-5 ADC 复用功能重映射 1

参阅复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR)。

复用功能	ADC1_ETRGINJ_REMAP = 0	ADC1_ETRGINJ_REMAP = 1
ADC1 外部触发注入转换	ADC1 外部触发注入转换与 EXTI15 相连	ADC1 外部触发注入转换与 TIM1A_CH4 相连

Table 11-16 ADC1 外部触发注入转换复用功能重映射表

复用功能	ADC1_ETRGREG_REMAP = 0	ADC1_ETRGREG_REMAP = 1
ADC1 外部触发规则转换	ADC1 外部触发规则转换与 EXTI11 相连	ADC1 外部触发规则转换与 TIM1A_TRGO 相连

Table 11-17 ADC1 外部触发注入转换复用功能重映射表



复用功能	<b>ADC1_ETRGINJ_REMAP = 0</b>	<b>ADC1_ETRGINJ_REMAP = 1</b>
ADC2 外部触发注入转换	ADC2 外部触发注入转换与 EXTI15 相连	ADC2 外部触发注入转换与 TIM1A_CH4 相连

Table 11-18 ADC2 外部触发注入转换复用功能重映射表

复用功能	<b>ADC1_ETRGREG_REMAP = 0</b>	<b>ADC1_ETRGREG_REMAP = 1</b>
ADC2 外部触发规则转换	ADC2 外部触发规则转换与 EXTI11 相连	ADC2 外部触发规则转换与 TIM1A_TRGO 相连

Table 11-19 ADC2 外部触发注入转换复用功能重映射表

#### 11.4-6 定时器复用功能重映射

定时器 4 的通道 1 到通道 4 可以从端口 B 重映射到端口 D。其他定时器的重映射列在表 42~表 44。

参见复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR)。

复用功能	<b>TIM2CCH4_IREMAP = 0</b>	<b>TIM2CCH4_IREMAP = 1</b>
TIM2C_CH4	TIM2C 的通道 4 连至 PA3	SIRC 内部时钟连至 TIM2C_CH4 的输入作为校准使用

Table 11-20 TIM2C 复用功能重映像表

复用功能	<b>TIM2B_REMAP = 0</b>
TIM2B_CH1	PB6
TIM2B_CH2	PB7
TIM2B_CH3	PB8
TIM2B_CH4	PB9

Table 11-21 TIM2B 复用功能重映像表

复用功能	<b>TIM2A_REMAP[1:0] = 00</b>	<b>TIM2A_REMAP[1:0] = 10</b>	<b>TIM2A_REMAP[1:0] = 11</b>
------	------------------------------	------------------------------	------------------------------



	(没有重映像)	(部分重映像)	(完全重映像) <sup>(1)</sup>
TIM2A_CH1	PA6	PB4	PC6
TIM2A_CH2	PA7	PB5	PC7
TIM2A_CH3	PB0		PC8
TIM2A_CH4	PB1		PC9

Table 11-22 TIM2A 复用功能重映像表

Note : (1) 重映像只适用于 64



复用功能	<b>TIM2_REMAP[1:0] = 00</b> (没有重映像)	<b>TIM2_REMAP[1:0] = 01</b> (部分重映像)	<b>TIM2_REMAP[1:0] = 10</b> (部分重映像) <sup>(1)</sup>	<b>TIM2_REMAP[1:0] = 11</b> (完全重映像) <sup>(1)</sup>
TIM2_CH1_ETR <sup>(2)</sup>	PA0	PA15	PA0	PA15
TIM2_CH2	PA1	PB3	PA1	PB3
TIM2_CH3		PA2		PB10
TIM2_CH4		PA3		PB11

Table 11-23 TIM2 复用功能重映像表

Note:

1. 重映像不适用于 36 脚的封装
2. TIM2\_CH1 和 TIM2\_ETR 共用一个引脚，但不能同时使用(因此在此使用这样的标记：TIM2\_CH1\_ETR) 表 44 TIM1 复用功能重映像

复用功能映像	<b>TIM1_REMAP[1:0] = 00</b> (没有重映像)	<b>TIM1_REMAP[1:0] = 01</b> (部分重映像)
TIM1_ETR		PA12
TIM1_CH1		PA8
TIM1_CH2		PA9
TIM1_CH3		PA10
TIM1_CH4		PA11
TIM1_BKIN	PB12 <sup>(2)</sup>	PA6
TIM1_CH1N	PB13 <sup>(2)</sup>	PA7
TIM1_CH2N	PB14 <sup>(2)</sup>	PB0
TIM1_CH3N	PB15 <sup>(2)</sup>	PB1

Table 11-24 TIM1 复用功能重映像表

Note : 2 重映像不适用于 36 脚的封装



#### 11.4-7 USART 复用功能重映射

参见复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR) 表

复用功能	USART3_REMAP[1:0] = 00 (没有重映像)	USART3_REMAP[1:0] = 01 (部分重映像) <sup>(1)</sup>
USART3_TX	PB10	PC10
USART3_RX	PB11	PC11
USART3_CK	PB12	PC12
USART3_CTS	PB13	
USART3_RTS	PB14	

Table 11-25 USART3 复用功能重映像表

Note : (1) 重映像只适用于 64 PIN

复用功能	USART2_REMAP = 0
USART2_CTS	PA0
USART2_RTS	PA1
USART2_TX	PA2
USART2_RX	PA3
USART2_CK	PA4

Table 11-26 USART2 复用功能重映像表

复用功能	USART1_REMAP = 0	USART1_REMAP = 1
USART1_TX	PA9	PB6
USART1_RX	PA10	PB7

Table 11-27 USART1 复用功能重映像表



#### 11.4-8 I2C1 复用功能重映射

参见复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR)

复用功能	I2C1_REMAP = 0	I2C1_REMAP = 1 <sup>(1)</sup>
I2C1_SCL	PB6	PB8
I2C1_SDA	PB7	PB9

Table 11-28 I2C1 复用功能重映像表

Note : (1) 重映像不适用于 36 脚封装

#### 11.4-9 SPI 1 复用功能重映射

参见复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR)

复用功能	SPI1_REMAP = 0	SPI1_REMAP = 1
SPI1_NSS	PA4	PA15
SPI1_SCK	PA5	PB3
SPI1_MISO	PA6	PB4
SPI1_MOSI	PA7	PB5

Table 11-29 SPI1 复用功能重映像表

#### 11.4-10 SPI3 复用功能重映射

参见复用重映射和调试 I/O 配置寄存器(AFIO\_MAPR), 这个重映射只适用于互联型产品

复用功能	SPI3_REMAP = 0	SPI3_REMAP = 1
SPI3_NSS	PA15	PA4
SPI3_SCK	PB3	PC10
SPI3_MISO	PB4	PC11
SPI3_MOSI	PB5	PC12

Table 11-30 SPI3 复用功能重映像表



### 11.5 GPIOX 和 AFIO 寄存器列表

x=A、B、C、D

GPIOx 基地址

GPIOx	基地地址	描述
AFIO	0x4001 0000	AFIO 基移地址
GPIOA	0x4002 0000	GPIOA 基移地址
GPIOB	0x4002 0400	GPIOB 基移地址
GPIOC	0x4002 0800	GPIOC 基移地址
GPIOD	0x4002 0C00	GPIOD 基移地址

Table 11-31 GPIOxAFIO 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	GPIOx_CRL	端口配置低寄存器	0x0000 0000
0x04	GPIOx_CRH	端口配置高寄存器	0x0000 0000
0x08	GPIOx_IDR	端口输入数据寄存器	0x0000 0000
0x0C	GPIOx_ODR	端口输出数据寄存器	0x0000 0000
0x10	GPIOx_BSRR	端口位设置/清除寄存器	0x0000 0000
0x14	GPIOx_BRR	端口位清除寄存器	0x0000 0000
0x18	GPIOx_LCKR	端口配置锁定寄存器	0x0000 0000
0x1C	GPIOx_AFRL(1*)	复用功能寄存器(1*)	0x0000 0000
0x20	GPIOx_AFRH(2*)	复用功能寄存器(2*)	0x0000 0000

偏移地址	名称	描述	复位值
0x00	AFIO_EVCR	事件控制寄存器	0x0000 0000
0x04	AFIO_MAPR	复用重映射和调试I/O配置寄存器	0x0000 0000
0x08	AFIO_EXTICR1	外部中断配置寄存器1	0x0000 0000
0x0C	AFIO_EXTICR2	外部中断配置寄存器2	0x0000 0000
0x10	AFIO_EXTICR3	外部中断配置寄存器3	0x0000 0000
0x14	AFIO_EXTICR4	外部中断配置寄存器4	0x0000 0000

Note : 0: 表示逻辑值 0; 1: 表示逻辑值 1; X: 表示不确定



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBIT CONFIDENTIAL



## 11.6 GPIO 寄存器说明

### 11.6-1 GPIO 端口置低寄存器(GPIOx\_CRL)(x = A..D)

偏移地址: 0x00 复位值: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]	CNF3[1:0]		MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CNFy[1:0]: Port x configuration bits (y = 0 .. 7)	CNFy[1:0]: 端口 x 配置位(y = 0...7) (Port x configuration bits)		软件通过这些位配置相应的 I/O 端口, 请参考表 17 端口位配置表。		在输入模式(MODE[1:0]=00):		00: 模拟输入模式		01: 浮空输入模式(复位后的状态)		10: 上拉/下拉输入模式		11: 保留		在输出模式(MODE[1:0]>00):	
MODEy[1:0]: Port x mode bits (y = 0 .. 7)	MODEy[1:0]: 端口 x 的模式位(y = 0...7) (Port x mode bits)		软件通过这些位配置相应的 I/O 端口, 请参考表 17 端口位配置表。		00: 输入模式(复位后的状态)		01: 输出模式, 大速度 10MHz		10: 输出模式, 大速度 2MHz		11: 输出模式, 大速度 50MHz					

位	标记	功能描述	复位值	读写
31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2	CNFy[1:0]: Port x configuration bits (y = 0 .. 7)	<b>CNFy[1:0]:</b> 端口 x 配置位(y = 0...7) (Port x configuration bits)  软件通过这些位配置相应的 I/O 端口, 请参考表 17 端口位配置表。  在输入模式(MODE[1:0]=00): 00: 模拟输入模式 01: 浮空输入模式(复位后的状态) 10: 上拉/下拉输入模式 11: 保留  在输出模式(MODE[1:0]>00): 00: 通用推挽输出模式 01: 通用开漏输出模式 10: 复用功能推挽输出模式 11: 复用功能开漏输出模式	0x4	rw
29:28, 25:24, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0	MODEy[1:0]: Port x mode bits (y = 0 .. 7)	<b>MODEy[1:0]:</b> 端口 x 的模式位(y = 0...7) (Port x mode bits)  软件通过这些位配置相应的 I/O 端口, 请参考表 17 端口位配置表。  00: 输入模式(复位后的状态) 01: 输出模式, 大速度 10MHz 10: 输出模式, 大速度 2MHz 11: 输出模式, 大速度 50MHz	0x4	rw



### 11.6-2 端口配置高寄存器(GPIOx\_CRH) (x = A..D)

偏移地址: 0x04

复位值: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1: 0]	MODE15[1: 0]	CNF14[1: 0]	MODE14[1: 0]	CNF13[1: 0]	MODE13[1: 0]	CNF12[1: 0]	MODE12[1: 0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1: 0]	MODE11[1: 0]	CNF10[1: 0]	MODE10[1: 0]	CNF9[1:0]	MODE9[1:0]	CNF8[1:0]	MODE8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2	CNFy[1:0]: Port x configuration bits (y= 8 .. 15)	<b>CNFy[1:0]:</b> 端口 x 配置位(y = 8...15) (Port x configuration bits) 软件通过这些位配置相应的 I/O 端口, 请参考表 17 端口位配置表。 在输入模式(MODE[1:0]=00): 00: 模拟输入模式 01: 浮空输入模式(复位后的状态) 10: 上拉/下拉输入模式 11: 保留 在输出模式(MODE[1:0]>00): 00: 通用推挽输出模式 01: 通用开漏输出模式 10: 复用功能推挽输出模式 11: 复用功能开漏输出模式	0x4	rw
29:28, 25:24, 21:20, 17:16,	MODEy[1:0]: Port x mode bits (y= 8 .. 15)	<b>MODEy[1:0]:</b> 端口 x 的模式位(y = 8...15) (Port x mode bits) 软件通过这些位配置相应的 I/O 端口, 请参考表 17 端口位配置表。	0x4	rw



13:12, 9:8, 5:4, 1:0		00: 输入模式(复位后的状态) 01: 输出模式, 大速度 10MHz 10: 输出模式, 大速度 2MHz 11: 输出模式, 大速度 50MHz		
-------------------------------	--	--	--	--

ZBT CONFIDENTIAL



### 11.6-3 端口输入数据寄存器(GPIOx\_IDR)(x = A..D)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR	IDR	IDR	IDR	IDR	IDR	ID									
15	14	13	12	11	10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位	标记	功能描述	复位值	读写
31:16	Reserved	must be kept at reset value.	0x0000	
15:0	IDRy: Port input data (y= 0 .. 15)	TIDRy[15:0]: 端口输入数据(y = 0...15) (Port input data) 这些位为只读并只能以字(16 位)的形式读出。读出的值为对应 I/O 口的状态。	0xxxxx	r



#### 11.6-4 Port output data register (GPIOx\_ODR) (x = A..D)

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD	OD	OD	OD	OD	OD	OD	OD	OD	OD	OD	OD	OD	OD	OD	OD
R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:16	Reserved	must be kept at reset value.	0x0000	
15:0	ODRy: Port output data (y= 0 .. 15)	<b>ODRy[15:0]:</b> 端口输出数据(y = 0...15) (Port output data) 这些位可读可写并只能以字(16 位)的形式操作。注: 对 GPIOx_BSRR(x = A...E), 可以分别地对各个 ODR 位进行独立的设置/清除。	0x0000	rw



### 11.6-5 Port bit set/reset register (GPIOx\_BSRR) (x = A..D)

偏移地址: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR1	BR1	BR1	BR1	BR1	BR1	BR									
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS1	BS1	BS1	BS1	BS1	BS1	BS									
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位	标记	功能描述	复位值	读写
31:16	BRy: Port x Reset bit y (y= 0 .. 15)	<b>BRy:</b> 清除端口 x 的位 y (y = 0...15) (Port x Reset bit y) 这些位只能写入并只能以字(16 位)的形式操作。 0: 对对应的 ODRy 位不产生影响 1: 清除对应的 ODRy 位为 0 注: 如果同时设置了 BSy 和 BRy 的对应位, BSy 位起作用。	0x0000	w
15:0	BSy: Port x Set bit y (y= 0 .. 15)	<b>BSy:</b> 设置端口 x 的位 y (y = 0...15) (Port x Set bit y) 这些位只能写入并只能以字(16 位)的形式操作。 0: 对对应的 ODRy 位不产生影响 1: 设置对应的 ODRy 位为 1	0x0000	w



### 11.6-6 Port bit reset register (GPIOx\_BRR) (x = A..D)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR1	BR1	BR1	BR1	BR1	BR1	BR									
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位	标记	功能描述	复位值	读写
31:16	Reserved	must be kept at reset value.	0x0000	
15:0	BRy: Port x Reset bit y (y= 0 .. 15)	<b>BRy:</b> 清除端口 x 的位 y (y = 0..15) (Port x Reset bit y) 这些位只能写入并只能以字(16 位) 的形式操作。 0: 对对应的 ODRy 位不产生影响 1: 清除对应的 ODRy 位为 0	0x0000	w



### 11.6-7 Port configuration lock register (GPIOx\_LCKR) (x = A..D)

偏移地址: 0x18

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															LC
															KK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK	LCK	LCK	LCK	LCK	LCK	LC									
15	14	13	12	11	10	K9	K8	K7	K6	K5	K4	K3	K2	K1	K0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:17	Reserved	must be kept at reset value.	0x0000	
16	LCKK: Lock key	<p>LCKK: 锁键 (Lock key)            该位可随时读出, 它只可通过锁键写入序列修改。            0: 端口配置锁键位激活            1: 端口配置锁键位被激活, 下次系统复位前            GPIOx_LCKR 寄存器被锁住。            锁键的写入序列:            写 1 -&gt; 写 0 -&gt; 写 1 -&gt; 读 0 -&gt; 读 1            后一个读可省略, 但可以用来确认锁键已被激活。            注: 在操作锁键的写入序列时, 不能改变 LCK[15:0] 的值。            操作锁键写入序列中的任何错误将不能激活锁键。</p>	0x0	rw
15:0	LCKy: Port x Lock bit y (y = 0 .. 15)	<p>LCKy: 端口 x 的锁位 y (y = 0...15) (Port x Lock bit y)            这些位可读可写但只能在 LCKK 位为 0 时写入。            0: 不锁定端口的配置            1: 锁定端口的配置</p>	0x0000	rw



### 11.7 AFIO 寄存器描述

对寄存器 *AFIO\_EVCR*, *AFIO\_MAPR* 和 *AFIO\_EXTICRX* 进行读写操作前, 应当首先打开 *AFIO* 的时钟。

参考第 6.3.7 节 *APB2 外设时钟使能寄存器(RCC\_APB2ENR)*。

必须以字(32 位)的方式操作这些外设寄存器

#### 11.7.1 事件控制寄存器(*AFIO\_EVCR*)

偏移地址: 0x00 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:9	Reserved	Reserved, must be kept cleared.	0x00 0000	ro
8	AFMD	复用模式選擇。 1 : PIN Mode 复用模式 0 : 相容 ST 复用模式	0	rw
7	EVOE	EVOE: 允许事件输出 (Event output enable) 该位可由软件读写。当设置该位后, Cortex 的 EVENTOUT 将连接到由 PORT[2:0]和 PIN[3:0]选定的 I/O 口	0x0	rw
6:4	PORT	PORT[2:0]: 端口选择 (Port selection) 选择用于输出 Cortex 的 EVENTOUT 信号的端口: 000: 选择 PA 001: 选择 PB 010: 选择 PC	0x0	rw



位	标记	功能描述	复位值	读写
		011: 选择 PD 100: Revered		
3:0	PIN	PIN[3:0]: 引脚选择 Pin selection (x = A .. D) 选择用于输出 Cortex 的 EVENTOUT 信号的引脚 0000: Px0 selected 0001: Px1 selected 0010: Px2 selected 0011: Px3 selected ... 1111: Px15 selected	0x0	rw



### 11.7-2 复用重映射和调试 I/O 配置寄存器(AFI0\_MAPR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OP_REMAP		Reserved		SWJ_CNF		Reserved		ADC2_ETRG_REG_R_EMAP	ADC2_ETRG_REG_R_EMAP	ADC1_ETRG_REG_REMAP	ADC1_ETRG_REG_INJ_R_EMAP	ADC1_ETRG_REG_EMAP	TIM2_CCH4_IRE_MAP		
rw	rw	rw	ro	ro	r w	r w	rw	ro	ro	ro	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01_REMAP	CAN_REMAP	TIM2B_REMAP	TIM2A_REMAP	TIM2_REMAP	TIM1_REMAP	USART3_REMAP	USART2_REMAP	USART1_REMAP	I2C1_REMAP	SPI1_REMAP					
rw	rw	r w	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:30	Reserved	Reserved, must be kept cleared.	0x0	-
30	TIM1B_REMAP	<p><b>TIM1B_REMAP: PCA重映射 (PCA remapping)</b>          该位可由软件设置          0: 没有重映射 (ETR/PA15, BKIN/PB3)          1: 重映射 (ETR/PA15, BKIN/PB3,          CH1/PC10, CH2/PC11, CH3/PC12, CH4/PD2,          CH1N/PB6, CH2N/PB7, CH3N/PB8)</p>	0	rw
29	TIM2ITR1_R_EMAP	<p><b>TIM2ITR1_REMAP: TIM2内部触发1重映射 (TIM2 internal trigger 1 remapping)</b>          该位可由软件置'1'或置'0'。它控制TIM2_ITR1的内部重映射。          0: 为了校准的需要, 在内部连接TIM2_ITR1至以太网的PTP输出;</p>	0	rw



位	标记	功能描述	复位值	读写
		1: 为了校准的需要，在内部连接TIM2_ITR1至全速USB OTG的SOF(帧开始)输出。		
28	SPI3_REMAP	<p>SPI3_REMAP: SPI3重映射 (SPI3 remapping)</p> <p>该位可由软件置'1'或置'0'。它控制SPI3的NSS、SCK、MISO、MOSI在GPIO端口的复用功能。</p> <p>0: 没有重映射 (NSS/PA15、SCK/PB3、MISO/PB4、MOSI/PB5);</p> <p>1: 重映射 (NSS/PA4、SCK/PC10、MISO/PC11、MOSI/PC12)。</p>	0	rw
27	PCA_REMAP	<p>PCA_REMAP: PCA重映射 (PCA remapping)</p> <p>该位可由软件设置</p> <p>0: 没有重映射 (PCA_ECI /PC4)</p> <p>1: 重映射 (PCA_ECI/PC4,PCA_CH0/PC5,CH1/PB0,CH2/PB1,CH3/PC2,CH4/PB10)</p>	0x0	rw
26:24	SWJ_CNF	<p>SWD 访问 Cortex 的调试端口,</p> <p>SWJ_CFG != 100&amp;SWDIO_EN = 1</p> <p>: PA13 and PA14 作为 SWDIO. 调试端口</p> <p>SWDIO_EN = 0</p> <p>: PA13 and PA14 作为 GPIO 端口</p> <p>SWDIO_EN <a href="#">请参考RCC_SWIOCR</a></p>	0x0	rw
23:22	LPUART_REMAP	<p>LPUART_REMAP[1:0] LPUART重映射 (LPUART remapping)</p> <p>该位可由软件设置</p> <p>00: 没有重映射</p> <p>01: 未用组合</p> <p>10: 重映射 (TXD_LP/PA6,RXD_LP/PA7)</p> <p>11: 重映射(TXD_LP/PB0,RXD_LP/PB1)</p>	0x0	rw
21	LPTIM_REMAP	<p>LPTIM_REMAP: LPTIM重映射 (LPTIM remapping)</p> <p>该位可由软件设置</p> <p>0: 没有重映射 (LPT_EXT/PC4,LPT_GATE/PC5)</p> <p>1: 重映射 (LPT_EXT/PC4,LPT_GATE/PC5, LPT_LOG/PA4,LPT_LOGN/PA5)</p>	0x0	rw
20	ADC2_ETRG_REG_REMAP	ADC2_ETRGREG_REMAP: ADC2规则转换外部触发重映射 (ADC 2 external trigger regular conversion remapping)	0x0	rw



位	标记	功能描述	复位值	读写
		该位可由软件设置 0: ADC2规则转换外部触发与EXTI11相连; 1: ADC2规则转换外部触发与TIM1A TRGO相连。		
19	ADC2_ETRGI NJ_REMAP	ADC2_ETRGINJ_REMAP: ADC2注入转换外部触发重映射 (ADC 2 external trigger injected conversion remapping) 该位可由软件设置 0: ADC2注入转换外部触发与EXTI15相连; 1: ADC2注入转换外部触发与TIM1A CH4通道4相连。	0x0	rw
18	ADC1_ETRG REG_REMAP	ADC1_ETRGREG_REMAP: ADC1规则转换外部触发重映射 (ADC 1 external trigger regular conversion remapping) 该位可由软件设置 0: ADC1规则转换外部触发与EXTI11相连; 1: ADC1规则转换外部触发与TIM1A TRGO相连。	0x0	rw
17	ADC1_ETRGI NJ_REMAP	ADC1_ETRGINJ_REMAP: ADC2注入转换外部触发重映射 (ADC1 external trigger injected conversion remapping) 该位可由软件设置 0: ADC1注入转换外部触发与EXTI15相连; 1: ADC1注入转换外部触发与TIM1A CH4通道4相连。	0x0	rw
16	TIM2CCH4_I REMAP	TIM2CCH4_I_REMAP: TIM2C通道4内部重映射 (TIM2C channel4 internal remap) 该位可由软件置'1'或置'0'。它控制TIM2C通道4内部映像。 0: TIM2C_CH4与PA3相连; 1: 当该SIRC内部振荡器与TIM2C_CH4相连，目的是对SIRC进行校准。	0x0	rw
15	PD01_REMA P	PD01_REMAP: 端口D0/端口D1映像到OSC_IN/OSC_OUT (Port D0/Port D1 mapping on OSC_IN/OSC_OUT) 该位可由软件置'1'或置'0'。它控制PD0和PD1的GPIO功能映像。当不使用主振荡器HXT时(系统运行于内部的8MHz阻容振荡器)，PD0和PD1可以映像到OSC_IN和OSC_OUT引脚。 0: 不进行PD0和PD1的重映像; 1: PD0映像到OSC_IN, PD1映像到OSC_OUT。	0x0	rw
14:13	CAN_REMA P	CAN_REMAP[1:0]: CAN复用功能重映像 (CAN alternate function remapping) 这些位可由软件置'1'或置'0'，在只有单个CAN接口的产品上控制复用功能CAN_RX和CAN_TX的重映像。 00: CAN_RX映像到PA11, CAN_TX映像到PA12; 01: 未用组合;	0x0	rw



位	标记	功能描述	复位值	读写
		10: CAN_RX映像到PB8, CAN_TX映像到PB9(不能用于36脚的封装); 11: CAN_RX映像到PD0, CAN_TX映像到PD1。		
12	TIM2B_REMAP	TIM2B_REMAP: 定时器4的重映像 (TIM2B remapping) 该位可由软件置'1' 或置'0'，控制将TIM2B的通道1-4映射到GPIO端口上。 0: 没有重映像。 (TIM2B_CH1/PB6, TIM2B_CH2/PB7, TIM2B_CH3/PB8, TIM2B_CH4/PB9); 1: 完全映像 (TIM2B_CH1/PD12, TIM2B_CH2/PD13, TIM2B_CH3/PD14, TIM2B_CH4/PD15)。	0x0	rw
11:10	TIM2A_REMAP	TIM2A_REMAP[1:0]: 定时器3的重映像 (TIM2A remapping) 这些位可由软件置'1' 或置'0'，控制定时器3的通道1至4在GPIO端口的映像。 00: 没有重映像 (CH1/PA6, CH2/PA7, CH3/PB0, CH4/PB1); 01: 未用组合; 10: 部分映像 (CH1/PB4, CH2/PB5, CH3/PB0, CH4/PB1); 11: 完全映像 (CH1/PC6, CH2/PC7, CH3/PC8, CH4/PC9)。 注：重映像不影响在PD2上的TIM2A ETR。	0x0	rw
9:8	TIM2_REMAP	TIM2_REMAP[1:0]: 定时器2的重映像 (TIM2 remapping) 位9:8 这些位可由软件置'1' 或置'0'，控制定时器2的通道1至4和外部触发(ETR)在GPIO端口的映像。 00: 没有重映像 (CH1/ETR/PA0, CH2/PA1, CH3/PA2, CH4/PA3); 01: 部分映像 (CH1/ETR/PA15, CH2/PB3, CH3/PA2, CH4/PA3); 10: 部分映像 (CH1/ETR/PA0, CH2/PA1, CH3/PB10, CH4/PB11); 11: 完全映像 (CH1/ETR/PA15, CH2/PB3, CH3/PB10, CH4/PB11)。	0x0	rw
7:6	TIM1_REMAP	TIM1_REMAP[1:0]: 定时器1的重映像 (TIM1 remapping) 这些位可由软件置'1' 或置'0'，控制定时器1的通道1至4、1N至3N、外部触发(ETR)和刹车输入(BKIN)在GPIO	0x0	rw



位	标记	功能描述	复位值	读写
		<p>端口的映像。</p> <p>00: 没有重映像 (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15);</p> <p>01: 部分映像 (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1);</p> <p>10: 未用组合;</p> <p>11: 完全映像 (ETR/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BKIN/PE15, CH1N/PE8, CH2N/PE10, CH3N/PE12)。</p>		
5:4	USART3_REMAP	<p>USART3_REMAP[1:0]: USART3的重映像 (USART3 remapping)</p> <p>这些位可由软件置'1'或置'0'，控制USART3的CTS、RTS、CK、TX和RX复用功能在GPIO端口的映像。</p> <p>00: 没有重映像 (TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14);</p> <p>01: 部分映像 (TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14);</p> <p>10: 未用组合;</p> <p>11: 完全映像 (TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12)。</p>	0x0	rw
3	USART2_REMAP	<p>USART2_REMAP: USART2的重映像 (USART2 remapping)</p> <p>这些位可由软件置'1'或置'0'，控制USART2的CTS、RTS、CK、TX和RX复用功能在GPIO端口的映像。</p> <p>0: 没有重映像 (CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4);</p> <p>1: 重映像 (CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7);</p>	0x0	rw
2	USART1_REMAP	<p>USART1_REMAP: USART1的重映像 (USART1 remapping)</p> <p>该位可由软件置'1'或置'0'，控制USART1的TX和RX复用功能在GPIO端口的映像。</p> <p>0: 没有重映像 (TX/PA9, RX/PA10);</p> <p>1: 重映像</p>	0x0	rw



位	标记	功能描述	复位值	读写
		(TX/PB6, RX/PB7)。		
1	I2C1_REMAP	I2C1_REMAP: I2C1的重映像 (I2C1 remapping) 该位可由软件置'1' 或置'0'，控制I2C1的SCL和SDA复用功能在GPIO端口的映像。 0: 没有重映像(SCL/PB6, SDA/PB7); 1: 重映像(SCL/PB8, SDA/PB9)。	0x0	rw
0	SPI1_REMAP	SPI1_REMAP: SPI1的重映像 该位可由软件置'1' 或置'0'，控制SPI1的NSS、SCK、MISO和MOSI复用功能在GPIO端口的映像。 0: 没有重映像 (NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7); 1: 重映像 (NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5)。	0x0	rw

ZBT CONFIDENTIAL



### 11.7.3 外部中断配置寄存器 1(AFIO\_EXTICR1)

偏移地址: 0x08 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
ro															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3				EXTI2				EXTI1				EXTI0			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:16	Reserved	Reserved, must be kept cleared.	0x0000	ro
15:0	EXTIx (x=3~0)	EEXTIx[3:0]: EXTIx配置(x = 0 ... 3) (EXTIx configuration)  位15:0  这些位可由软件读写, 用于选择EXTIx外部中断的输入源。  参看 <a href="#">10.2.5节</a> 。  0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin	0x0	rw



#### 11.7-4 外部中断配置寄存器 2(AFIO\_EXTICR2)

偏移地址: 0x0C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
ro															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7				EXTI6				EXTI5				EXTI4			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:16	Reserved	Reserved, must be kept cleared.	0x0000	ro
15:0	EXTIx (x=7~4)	EXTIx[3:0]: EXTIx配置(x = 4 ... 7) (EXTIx configuration) 这些位可由软件读写, 用于选择EXTIx外部中断的输入源。 参看 <a href="#">10.2.5节</a> 。  0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin	0x0	rw



### 11.7-5 外部中断配置寄存器 3(AFIO\_EXTICR3)

偏移地址: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
ro															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11				EXTI10				EXTI9				EXTI8			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:16	Reserved	Reserved, must be kept cleared.	0x0000	ro
15:0	EXTIx (x=11~8)	EXTIx[3:0]: EXTIx配置(x = 8 ... 11) (EXTI x configuration) 这些位可由软件读写, 用于选择EXTIx外部中断的输入源。 参看 <a href="#">10.2.5节</a> 。  0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin	0x0	rw



### 11.7-6 外部中断配置寄存器 4(AFIO\_EXTICR4)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
ro															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15				EXTI14				EXTI13				EXTI12			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:16	Reserved	Reserved, must be kept cleared.	0x0000	ro
15:0	EXTIx (x=15~12)	EXTIx[3:0]: EXTIx配置(x = 12 ... 15) (EXTIx configuration) 这些位可由软件读写, 用于选择EXTIx外部中断的输入源。 参看 <a href="#">10.2.5节</a> 。  0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin	0x0	rw



### 11.7-7 GPIOA 端口复用功能寄存器(GPIOA\_AFR)

边界地址:0x4001 0800

偏移地址: 0x1C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PAAFR7[3:0]				PAAFR6[3:0]				PAAFR5[3:0]				PAAFR4[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAAFR3[3:0]				PAAFR2[3:0]				PAAFR1[3:0]				PAAFR0[3:0]			
R/W				R/W				R/W				R/W			

偏移地址: 0x20 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PAAFR15[3:0]				PAAFR14[3:0]				PAAFR13[3:0]				PAAFR12[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAAFR11[3:0]				PAAFR10[3:0]				PAAFR9[3:0]				PxAFR8[3:0]			
R/W				R/W				R/W				R/W			

Note : AFMD=1 GPIOA\_AFR 複用選擇有效用

#### 11.7-7.1 PORTA 端口复用配置

请参考 Table4-2



### 11.7-8 GPIOB 端口复用功能寄存器(GPIOB\_AFR )

边界地址:0x4001 0C00

偏移地址: 0x1C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBAFR7[3:0]				PBAFR6[3:0]				PBAFR5[3:0]				PBAFR4[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBAFR3[3:0]				PBAFR2[3:0]				PBAFR1[3:0]				PBAFR0[3:0]			
R/W				R/W				R/W				R/W			

偏移地址: 0x20 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBAFR15[3:0]				PBAFR14[3:0]				PBAFR13[3:0]				PBAFR12[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBAFR11[3:0]				PBAFR10[3:0]				PBAFR9[3:0]				PBAFR8[3:0]			
R/W				R/W				R/W				R/W			

#### 11.7-8.1 PORTB 端口复用配置

请参考 Table4-2



### 11.7-9 GPIOC 端口复用功能寄存器(GPIOC\_AFR )

边界地址:0x4001 1000

偏移地址: 0x1C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCAFR7[3:0]				PCAFR6[3:0]				PCAFR5[3:0]				PCAFR4[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCAFR3[3:0]				PCAFR2[3:0]				PCAFR1[3:0]				PCAFR0[3:0]			
R/W				R/W				R/W				R/W			

偏移地址: 0x20 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCAFR15[3:0]				PCAFR14[3:0]				PCAFR13[3:0]				PCAFR12[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCAFR11[3:0]				PCAFR10[3:0]				PCAFR9[3:0]				PCAFR8[3:0]			
R/W				R/W				R/W				R/W			

#### 11.7-9.1 PORTC 端口复用配置

请参考 Table4-2



### 11.7-10 GPIOD 端口复用功能寄存器(GPIOA\_AFR )

边界地址:0x4001 1400

偏移地址: 0x1C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDAFR7[3:0]				PDAFR6[3:0]				PDAFR5[3:0]				PDAFR4[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDAFR3[3:0]				PDAFR2[3:0]				PDAFR1[3:0]				PDAFR0[3:0]			
R/W				R/W				R/W				R/W			

偏移地址: 0x20 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDAFR15[3:0]				PDAFR14[3:0]				PDAFR13[3:0]				PDAFR12[3:0]			
R/W				R/W				R/W				R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDAFR11[3:0]				PDAFR10[3:0]				PDAFR9[3:0]				PDAFR8[3:0]			
R/W				R/W				R/W				R/W			

#### 11.7-10.1 PORTD 端口复用配置

请参考 Table4-2



## 【12】循环冗余校验计算单元(CRC)

### 12.1 概述

循环冗余校验(CRC)计算单元是根据固定的生成多项式得到任意字节数据的 CRC 计算结果。在应用中，CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。以下示意了 CRC 算法在数据传输中的一个最典型的应用：

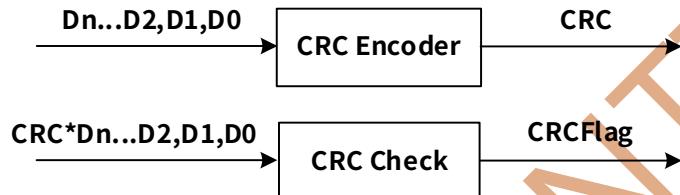


Figure 12-1 CRC 应用示意图

### 12.2 功能描述

本模块算法遵从 ISO/IEC13239 的定义，采用 16 /32 位长度的 CRC，计算多项式为：

- (1)  $x^{16} + x^{12} + x^5 + x$
- (2)  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

计算初始值为 0xFFFF。

本模块功能包括：

- CRC 编码和 CRC 校验
- 3 种位宽访问方式：8 位、16 位、32 位
- 8 位位宽下输入数据示例为 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16 位位宽下输入数据示例为 0x1100, 0x3322, 0x5544, 0x7766
- 32 位位宽下输入数据示例为 0x33221100, 0x77665544

#### 12.2-1 CRC 编码模式

编码模式可以对原始数据编码以计算其 CRC 值，操作流程如下所示：

- Step1：向 CRC\_RESULT.RESULT 写入 0xFFFF，初始化 CRC 计算。
- Step2：将待编码的原始数据按 8 位/16 位/32 位的组织方式，依次写入 CRC\_DATA 寄存器。
- Step3：读取 CRC\_RESULT.RESULT，即为 CRC 值。



### 12.2-2 CRC 检验模式

检验模式可以检验已编码的数据是否被篡改，操作流程如下所示

Step1：向 CRC\_RESULT.RESULT 写入 0xFFFF，初始化 CRC 计算。

Step2：将已编码的数据按 8 位/16 位/32 位的组织方式，依次写入 CRC\_DATA 寄存器。

注：按 8 位组织方式写 CRC 值到 CRC\_DATA 寄存器时，应先写入低 8 位，后写入高 8 位。

Step3：读取 CRC\_CR.FLAG，以判定 CRC 校验是否成功。

ZBT CONFIDENTIAL



### 12.3 CRC 寄存器列表

基地址: 0x 4002 3000

偏移地址	名称	描述	复位值
0x04	CRC_RESULT	CRC 结果寄存器, 计算完成后对该寄存器读取即获得结果。	0x0000 0000
0x08	CRC_CR	CRC 控制寄存器.	0x0000 0000
0x80-0xFF	CRC_DATA	CRC 数据寄存器, 用于输入需要运算的数据。	0x0000 0000

Table 12-1 CRC 寄存器列表和复位值



## 12.4 寄存器说明

### 12.4-1 CRC 结果寄存器(CRC\_RESULT)

地址偏移: 0x04 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	1 7	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	--------	----

RESULT[31:16]

R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

RESULT[15:0]

R/W

位	标记	功能描述	复位值	读写
31:0	RESULT [31:0]	本寄存器用于每次 CRC 计算结果的更新和保存。运算后，读取本寄存器将得到 16/32 位的 CRC 编码结果。  注意：根据标准规定，运算完成后，16/32 位的 CRC 编码值是运算寄存器取反后的结果，因此本寄存器[31:0]的读取将得到本寄存器当前[31:0]的取反值。	0x0000	R/W



### 12.4.2 CRC 控制寄存器(CRC\_CR)

地址偏移: 0x08 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	1	16
Revered															
Revered															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Revered															
Revered															

位	标记	功能描述	复位值	读写
31:2	-	保留	0x0	-
1	FLAG	校验结果标志; 0: 当前校验错误, 1: 当前校验正确。 注意: 寄存器[1]是一个只读位, 对其进行写操作不会产生影响。进行CRC校验时, 应该在所有的数据和16/32位CRC编码输入数据寄存器之后读取本位, 如果为1则表明校验成功。	0x0	RO
0	SEL	CRC polynomial codes select 1 : CRC16 → (x <sup>16</sup> + x <sup>12</sup> + x <sup>5</sup> + 1) 0 : CRC32 →(x <sup>32</sup> + x <sup>26</sup> + x <sup>23</sup> + x <sup>22</sup> + x <sup>16</sup> + x <sup>12</sup> + x <sup>11</sup> + x <sup>10</sup> + x <sup>8</sup> + x <sup>7</sup> + x <sup>5</sup> + x <sup>4</sup> + x <sup>2</sup> + x + 1)	0x0	R/W



### 12.4-3 CRC 数据寄存器(CRC\_DATA)

地址偏移: 0x80-0xFF 复位值: 0x0000 0000

31	30	29	28		27	26	25	24	23	22	21	20	19	1	17
CRC_DATA[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	CRC_DATA	本寄存器用于输入需要运算的数据。 注意：本寄存器的地址是一个范围(0x80-0xFF)，对该范围内的任何一个地址进行操作都会认为是对本寄存器进行操作。这样定义的目的就是为了方便软件可以用STM 指令对本寄存器进行连续的 32 位数据写入操作，以加快运算速度。本寄存器支持 8/16/32 位的输入方式。	0x0	R/W



## 【13】高级控制定时器 TIM1X(TIM1/1A/1B)

### 13.1 Advanced Timer 简介

高级控制定时器(TIM1/8/12)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 Advanced Timer 主要特性

Advanced Timer 的功能包括：

- 16 位向上、向下、向上/向下自动装载计数器
- 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请求/DMA 请求：
  - 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

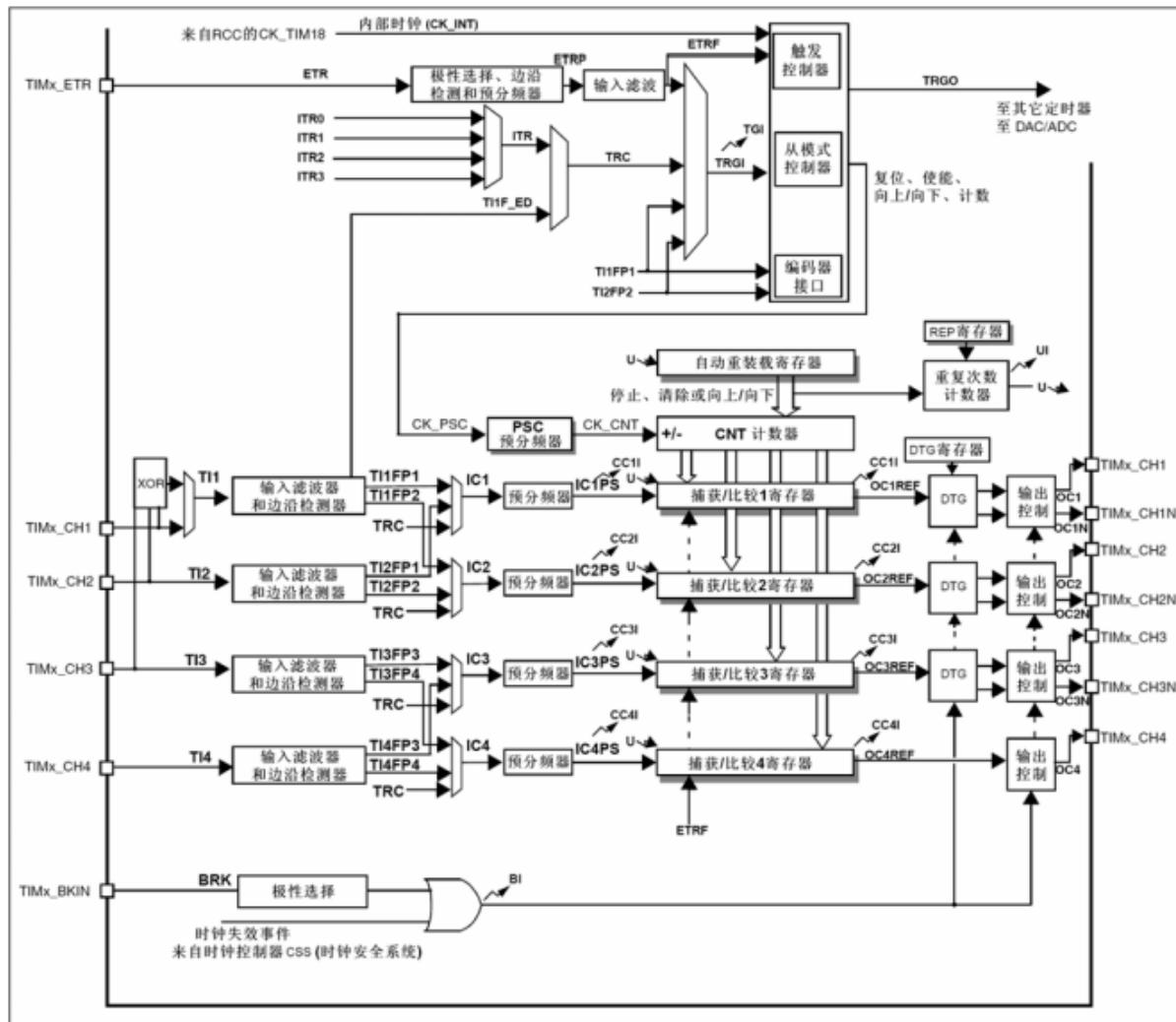


Figure 13-1 高级控制定时器框图

Note :



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 Advanced Timer 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：(Note : TIM1X\* → TIM1 , TIM1A ,TIM1B)

- 计数器寄存器(TIM1X\*\_CNT)
- 预分频器寄存器(TIM1X\*\_PSC)
- 自动装载寄存器(TIM1X\*\_ARR)
- 重复次数寄存器(TIM1X\*\_RCR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器根据在 TIM1X\*\_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 TIM1X\*\_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 TIM1X\*\_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 TIM1X\*\_CR1 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

#### 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 TIM1X\*\_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一次更新事件到来时被采用。

图 13-2 和图 13-3 给出了在预分频器运行时，更改计数器参数的例子。

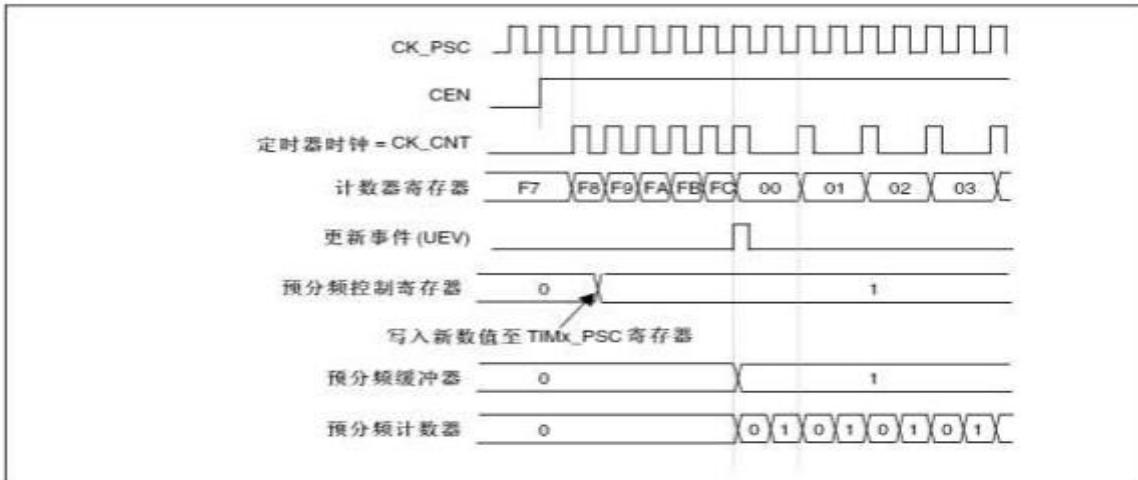


Figure 13-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

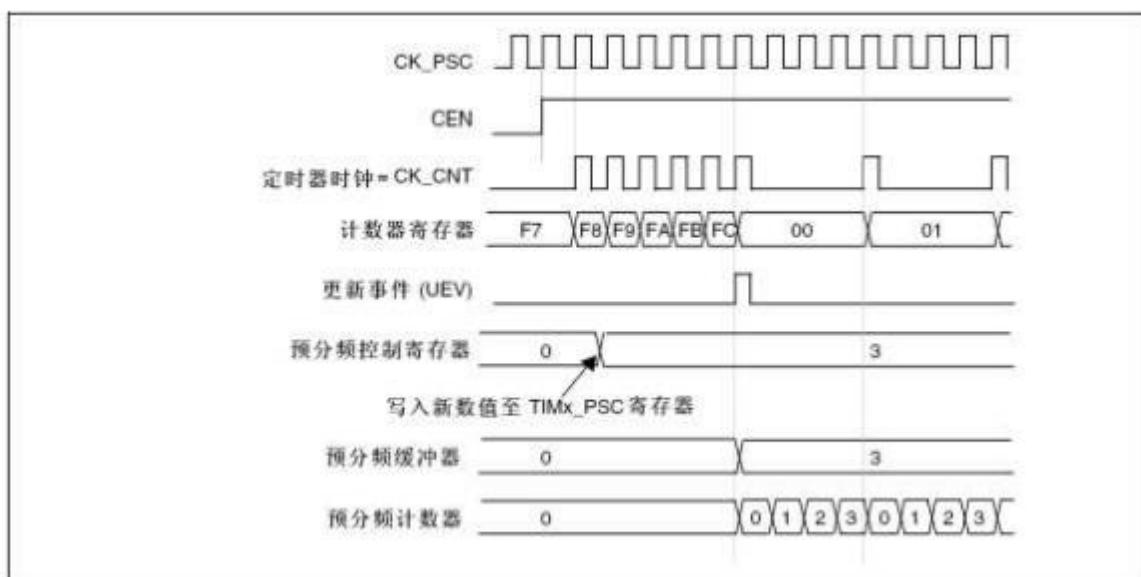


Figure 13-3 当预分频器的参数从 1 变到 4 时，计数器的时序图

### 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(TIM1X\*\_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(TIM1X\*\_RCR)时，产生更新事件(UEV)；否则每次计数器溢出时产生更新事件。

在 TIM1X\*\_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 `TIM1X*_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 `TIM1X*_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)。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(`TIM1X*_SR` 寄存器中的 `UIF` 位)。

- 重复计数器被重新加载为 `TIM1X*_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(`TIM1X*_ARR`)。
- 预分频器的缓冲区被置入预装载寄存器的值(`TIM1X*_PSC` 寄存器的内容)。

下图给出一些例子，当 `TIM1X*_ARR=0x36` 时计数器在不同时钟频率下的动作。

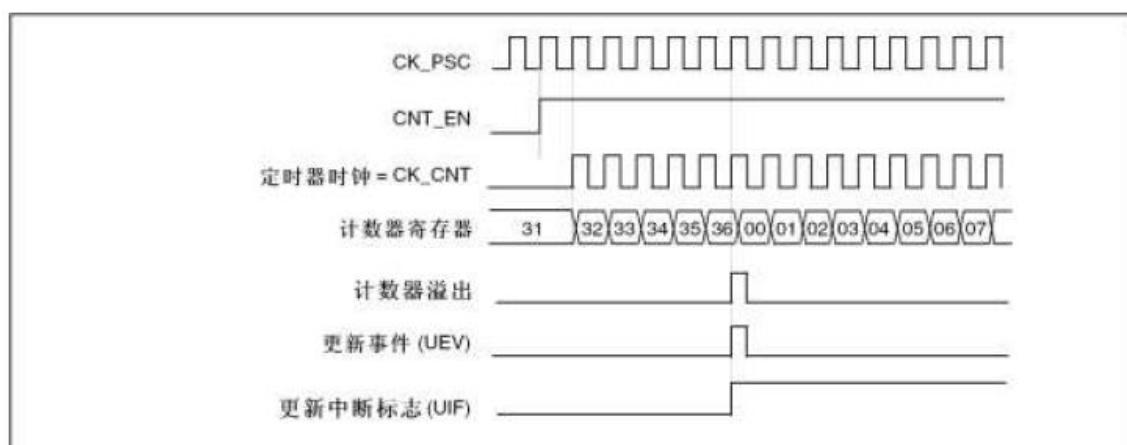


Figure 13-4 计数器时序图：内部时钟分频因子为 1

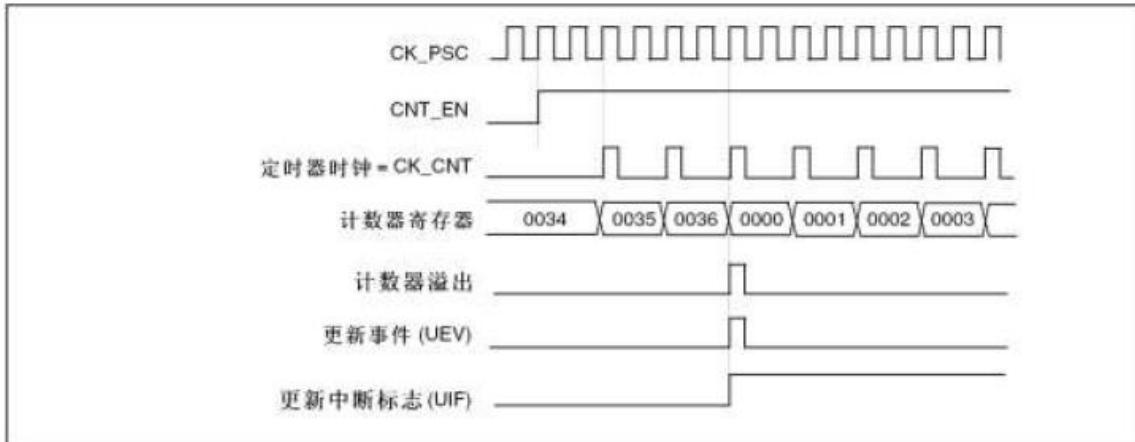


Figure 13-5 计数器时序图：内部时钟分频因子为 2

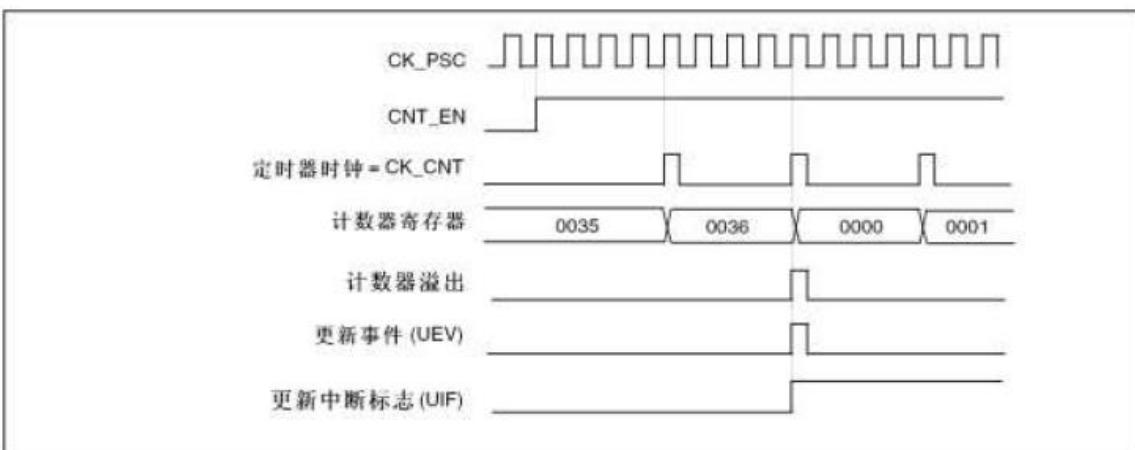


Figure 13-6 计数器时序图：内部时钟分频因子为 4

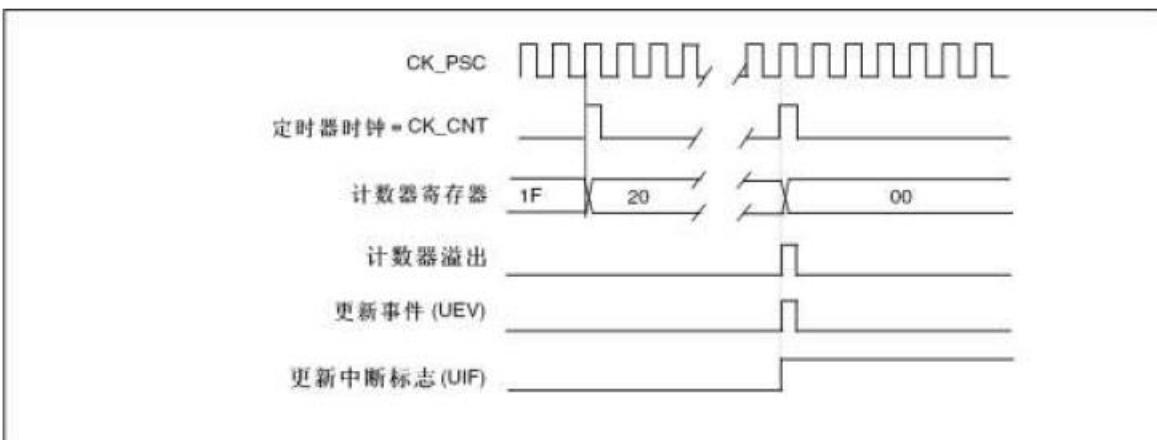


Figure 13-7 计数器时序图：内部时钟分频因子为 N

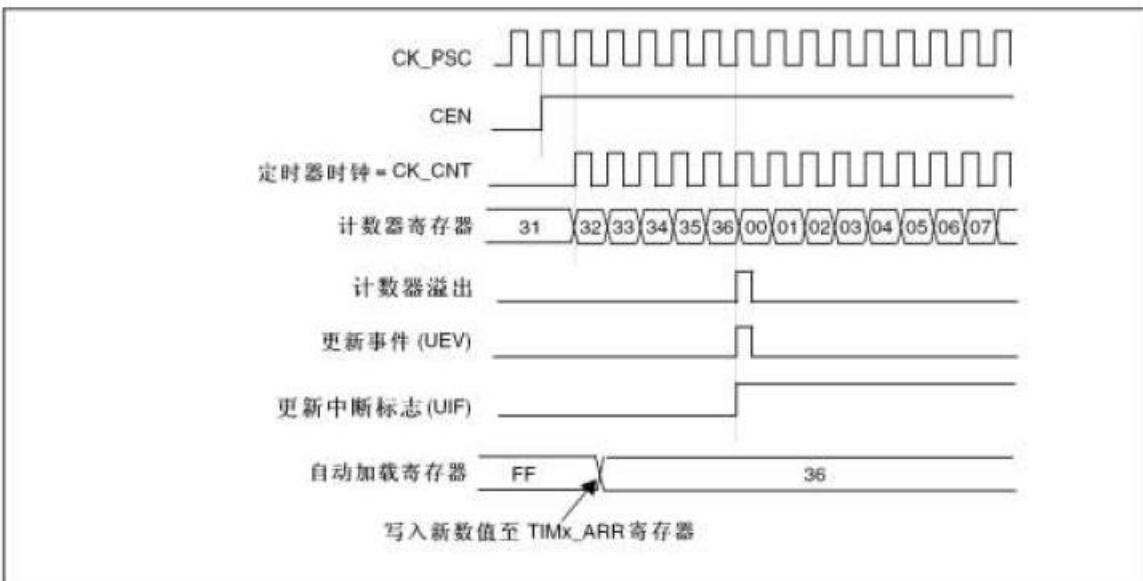


Figure 13-8 计数器时序图：当 ARPE=0 时的更新事件(TIM1X\*\_ARR 没有预装入)

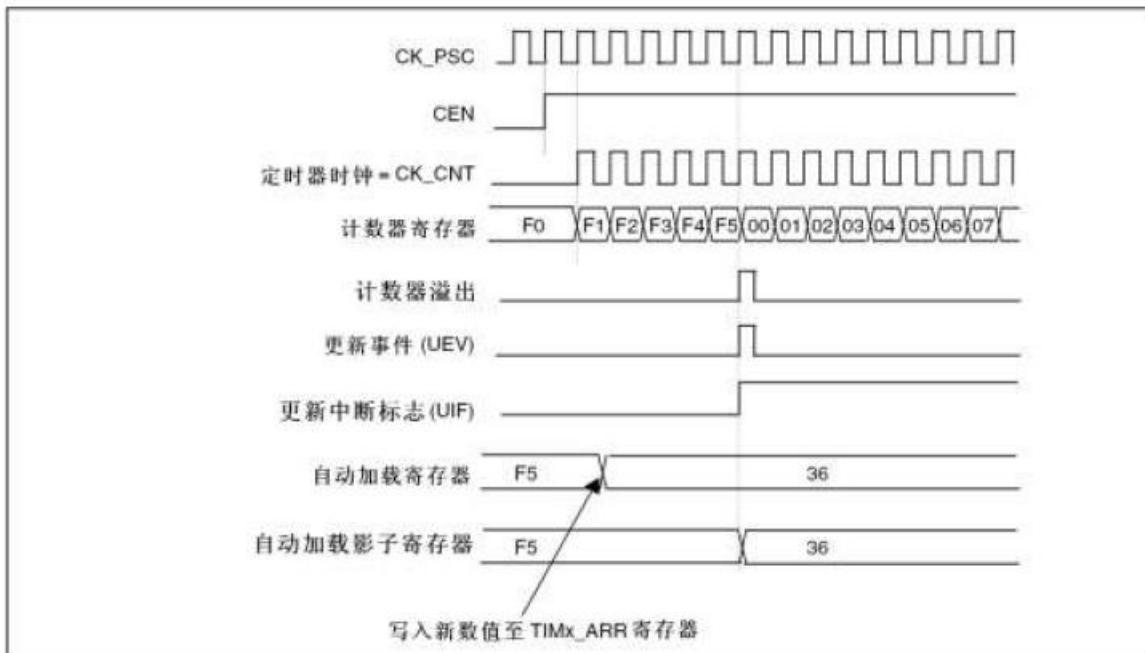


Figure 13-9 计数器时序图：当 ARPE=1 时的更新事件(预装入了 TIM1X\*\_ARR)



### 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(TIM1X\*\_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(TIM1X\*\_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产生更新事件。

在 TIM1X\*\_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 TIM1X\*\_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 TIM1X\*\_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)，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(TIM1X\*\_SR 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 TIM1X\*\_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(TIM1X\*\_PSC 寄存器的值)
- 当前的自动加载寄存器被更新为预装载值(TIM1X\*\_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 TIM1X\*\_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

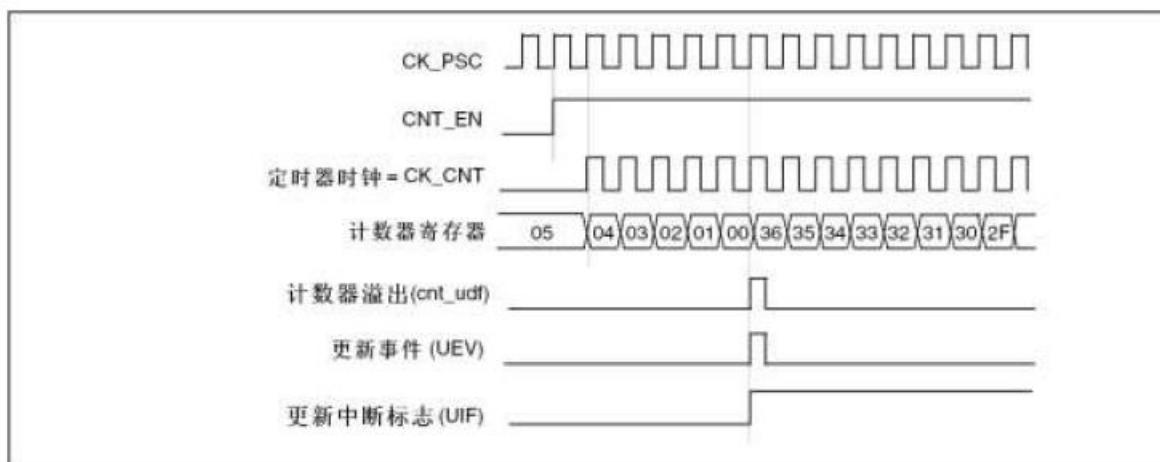


Figure 13-10 计数器时序图：内部时钟分频因子为 1

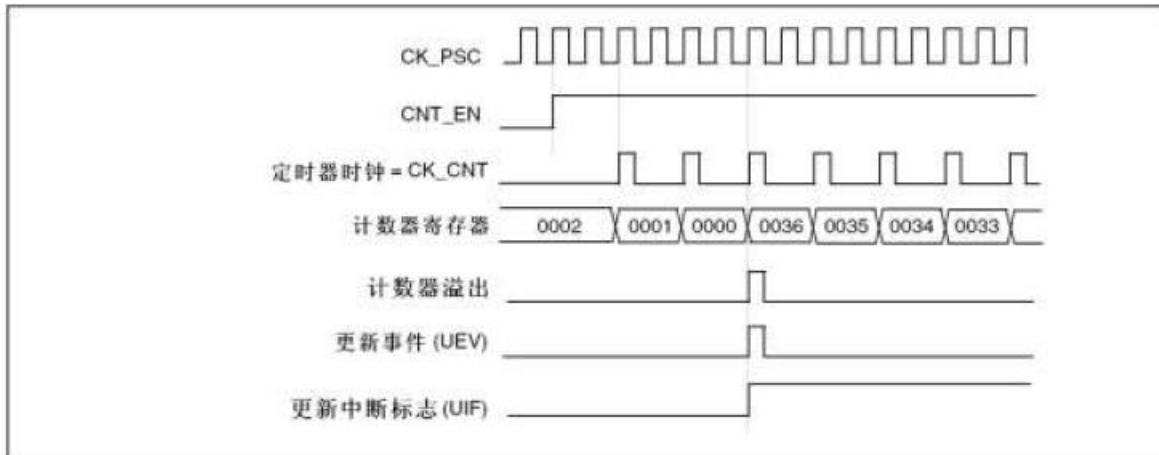


Figure 13-11 计数器时序图：内部时钟分频因子为 2

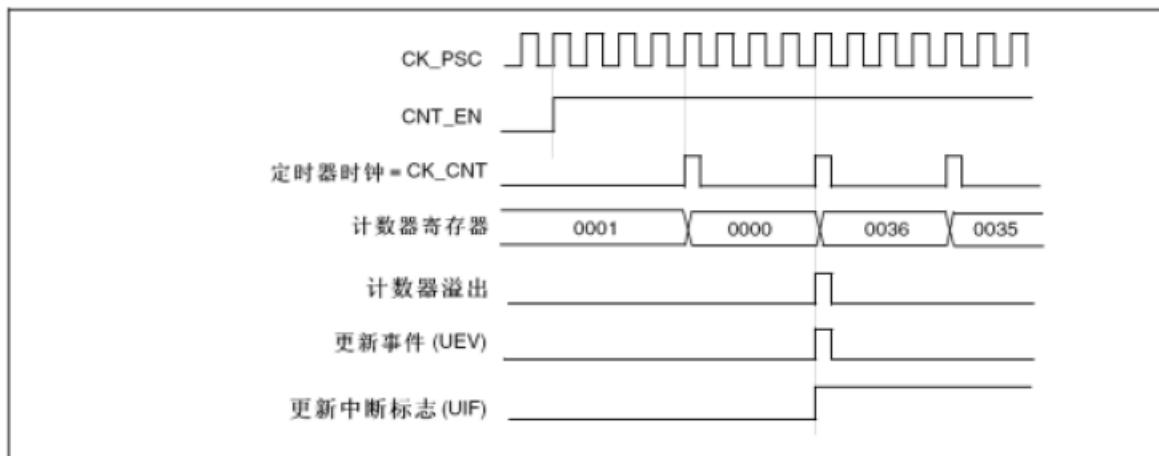


Figure 13-12 计数器时序图：内部时钟分频因子为 4

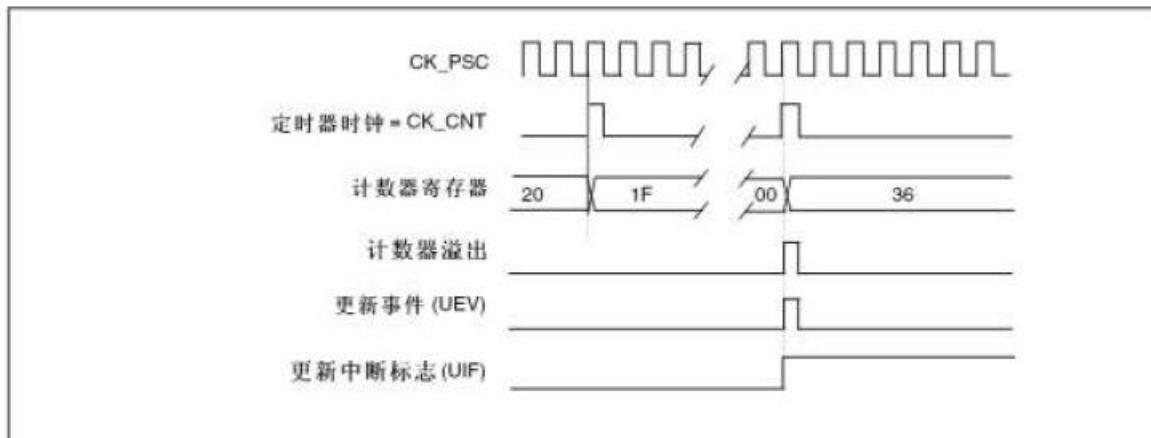


Figure 13-13 计数器时序图：内部时钟分频因子为 N

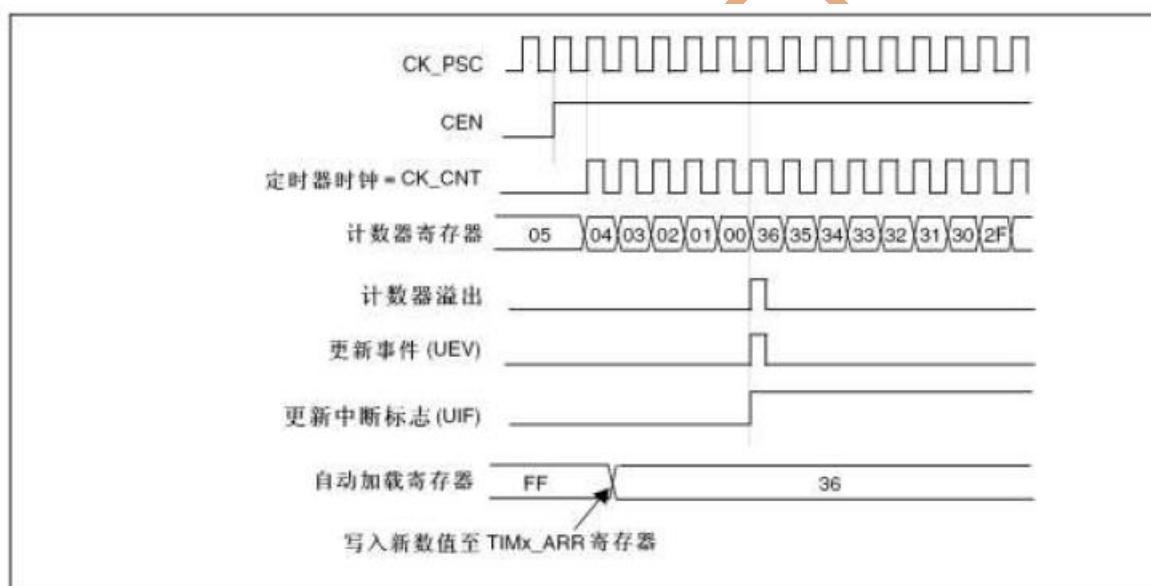


Figure 13-14 计数器时序图：当没有使用重复计数器时的更新事件



### 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(**TIM1X\*\_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 **TIM1X\*\_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 **TIM1X\*\_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 **TIM1X\*\_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 **TIM1X\*\_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)，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(**TIM1X\*\_SR** 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 **TIM1X\*\_RCR** 寄存器中的内容
- 预分频器的缓存器被加载为预装载(**TIM1X\*\_PSC** 寄存器)的值
- 当前的自动加载寄存器被更新为预装载值(**TIM1X\*\_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：

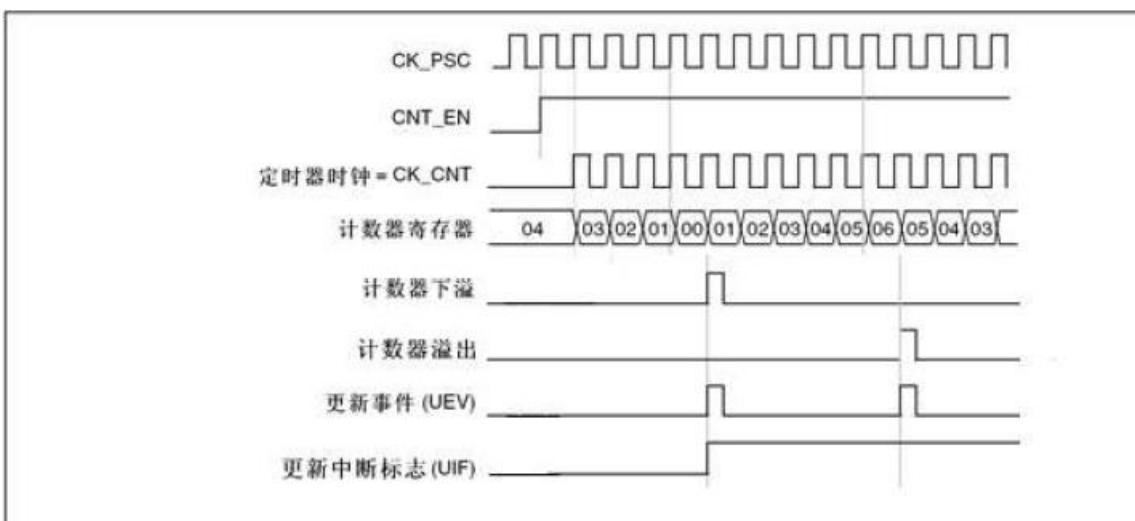


Figure 13-15 计数器时序图：内部时钟分频因子为 1, TIM1X\*\_ARR=0x6

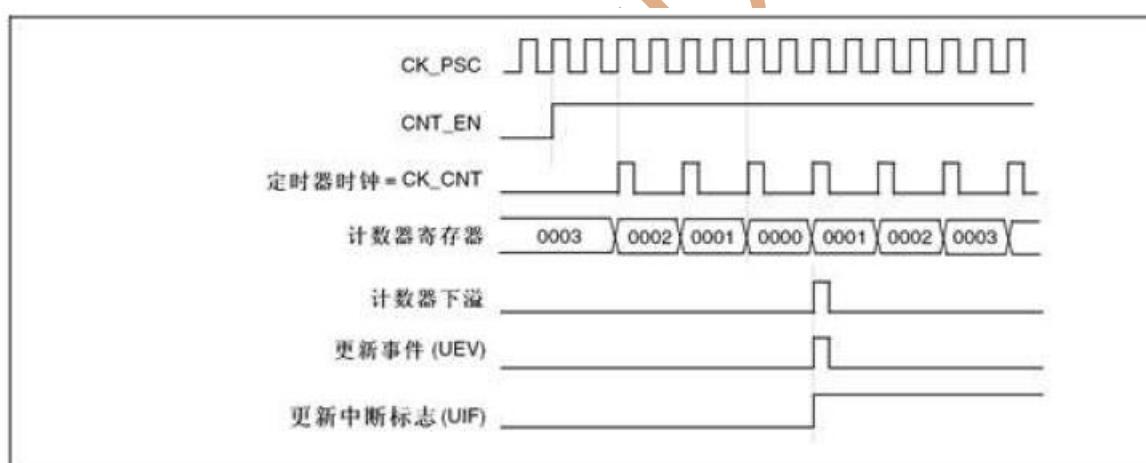


Figure 13-16 计数器时序图：内部时钟分频因子为 2

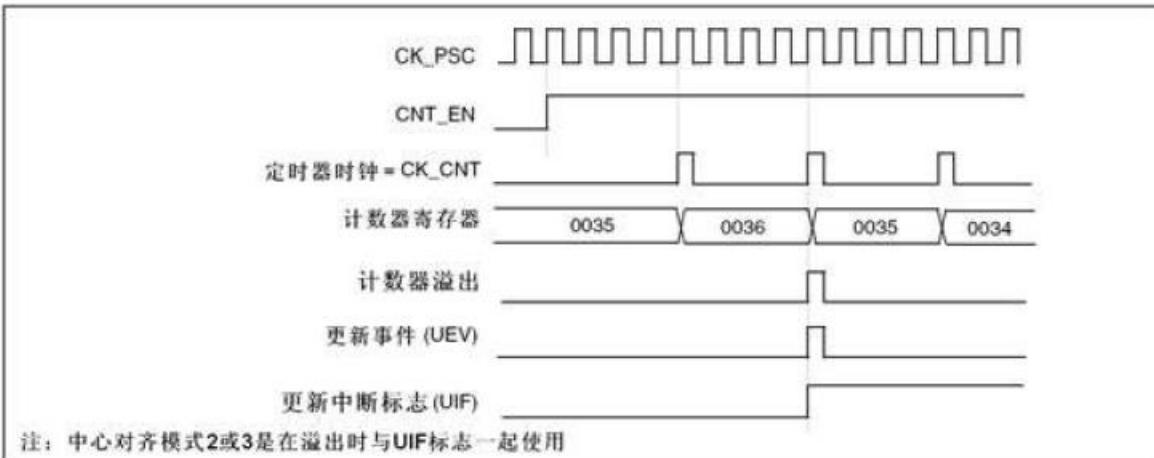


Figure 13-17 计数器时序图：内部时钟分频因子为 4, TIM1X\*\_ARR=0x36

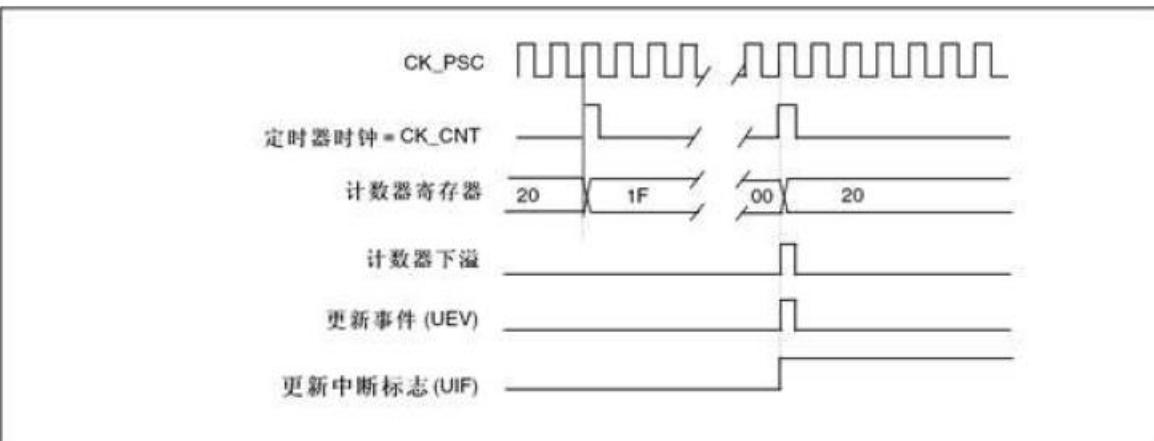


Figure 13-18 计数器时序图：内部时钟分频因子为 N

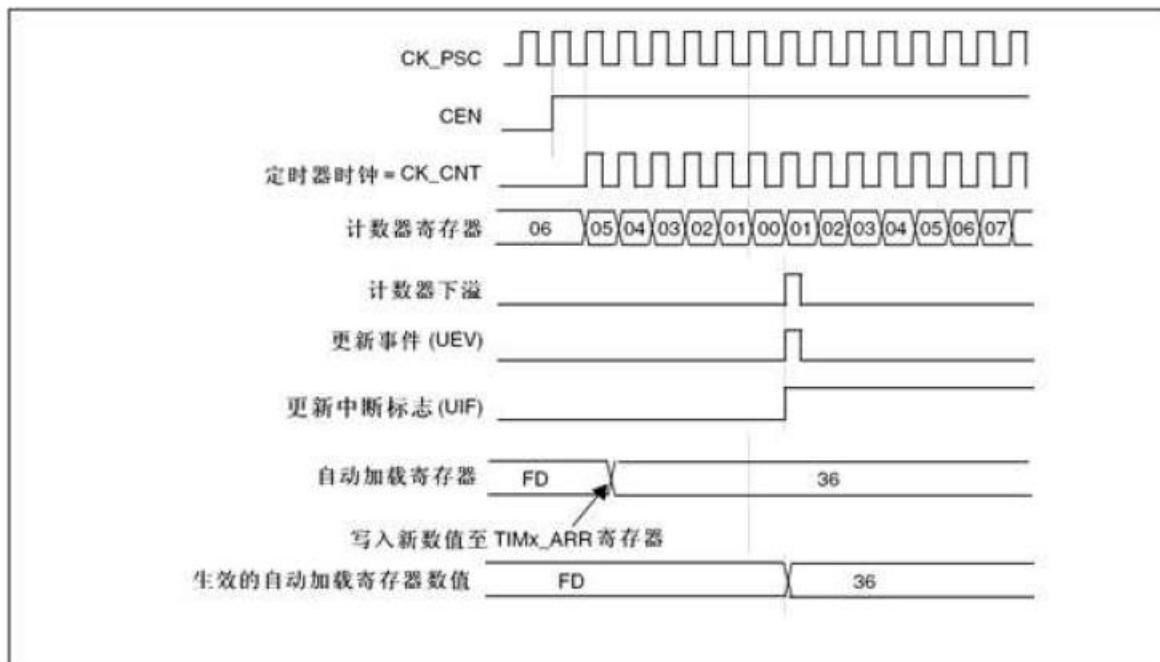


Figure 13-19 计数器时序图: ARPE=1 时的更新事件(计数器下溢)

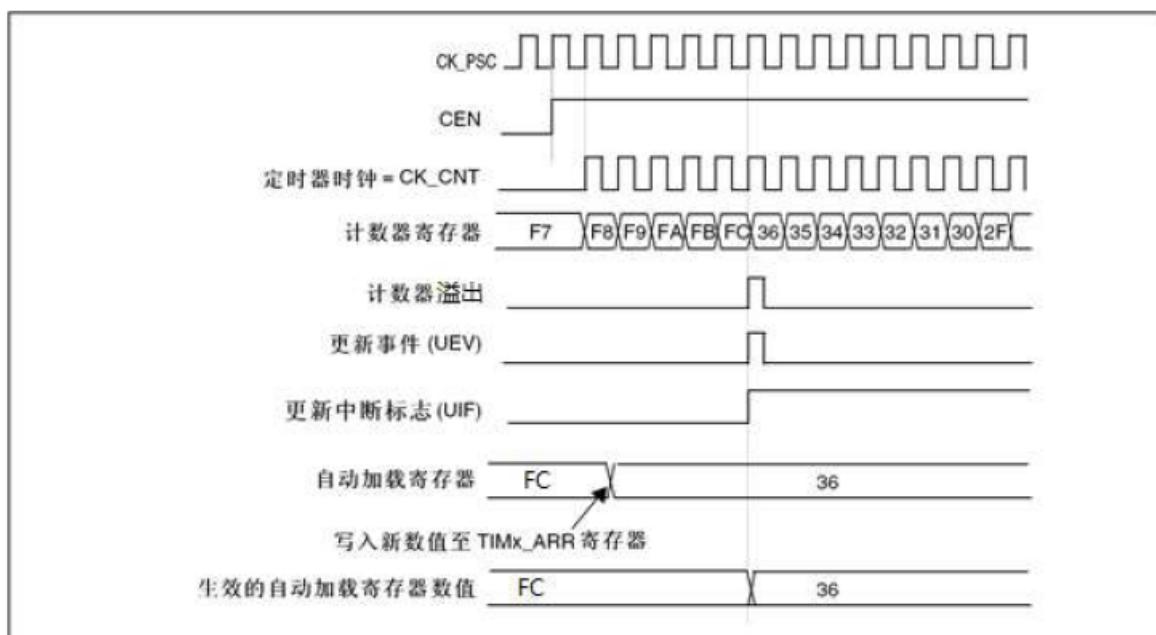


Figure 13-20 计数器时序图: ARPE=1 时的更新事件(计数器溢出)



### 13.3.3 重复计数器

14.3.1 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(TIM1X\*\_ARR 自动重载入寄存器，TIM1X\*\_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 TIM1X\*\_CCRx)，N 是 TIM1X\*\_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 TIM1X\*\_RCR 寄存器的值定义(参看图 13-21)。当更新事件由软件产生(通过设置 TIM1X\*\_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 TIM1X\*\_RCR 寄存器中的内容被重载入到重复计数器。

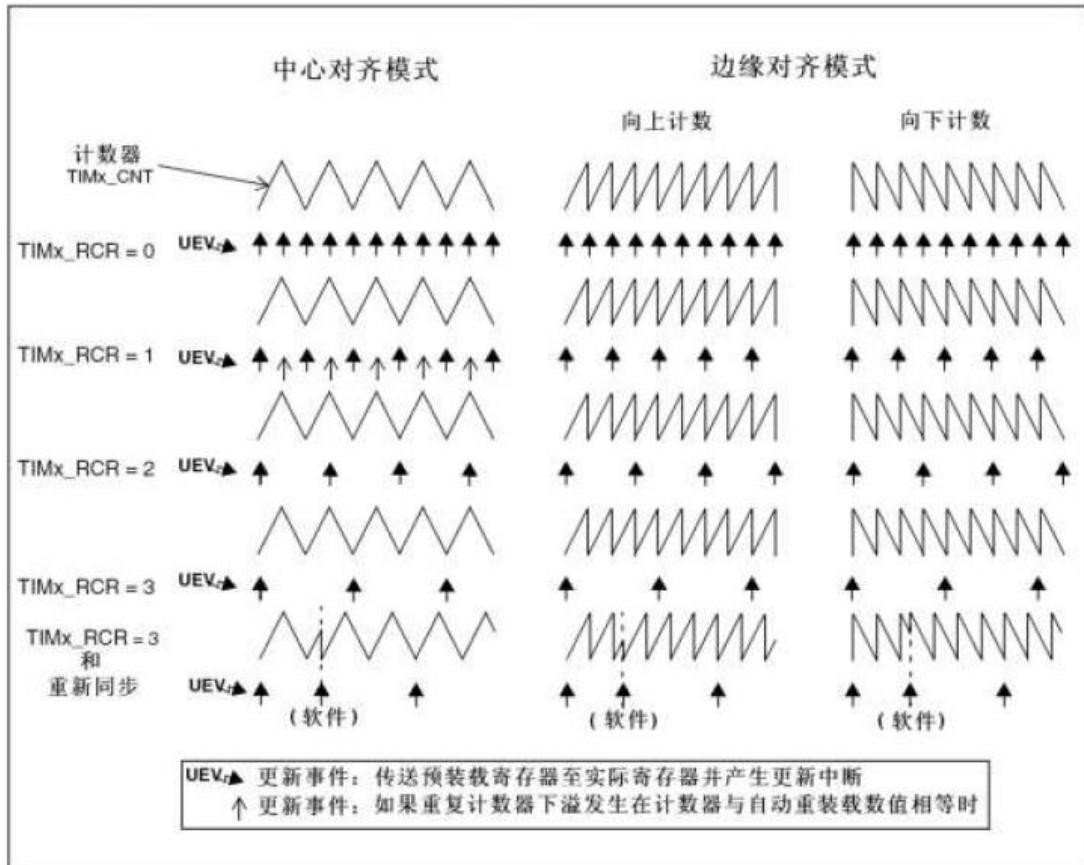


Figure 13-21 不同模式下更新速率的例子，及 TIM1X\*\_RCR 的寄存器设置

#### 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 TIM1X\* 而作为另一个定时器 TIM2 的预分频器，详见 14.3.15.1 节。
- 内部时钟源(CK\_INT)

如果禁止了从模式控制器(SMS=000)，则 CEN、DIR(TIM1X\*\_CR1 寄存器)和 UG 位(TIM1X\*\_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。

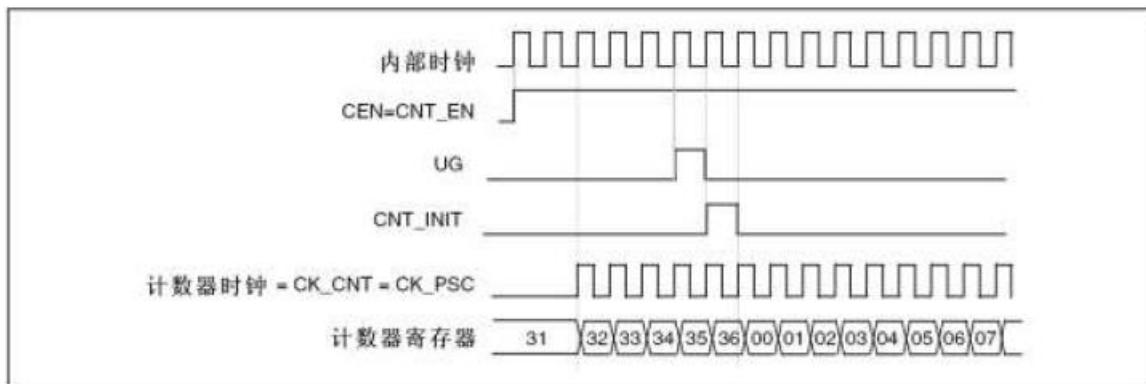


Figure 13-22 一般模式下的控制电路，内部时钟分频因子为 1



### 13.3-4.1 外部时钟源模式 1

当 TIM1X\*\_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沿计数。

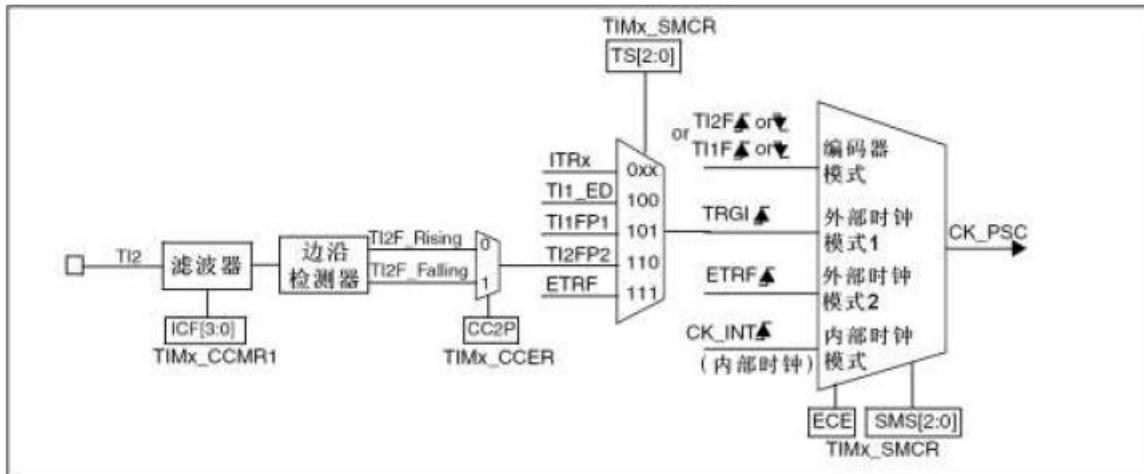


Figure 13-23 TI2 外部时钟连接例子

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

配置 TIM1X\*\_CCMR1 寄存器 CC2S=01，配置信道 2 检测 TI2 输入的上升沿

配置 TIM1X\*\_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 TIM1X\*\_CCER 寄存器的 CC2P=0，选定上升沿极性

配置 TIM1X\*\_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1

配置 TIM1X\*\_SMCR 寄存器中的 TS=110，选定 TI2 作为触发输入源

设置 TIM1X\*\_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。

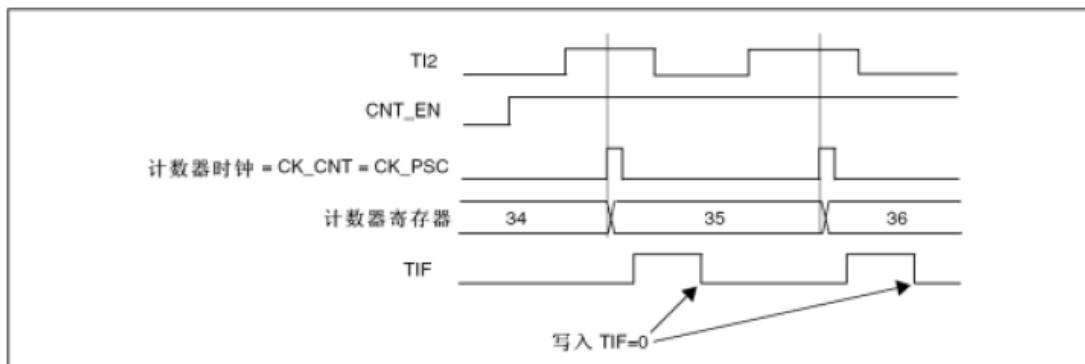


Figure 13-24 外部时钟模式 1 下的控制电路

### 13.3-4.2 外部时钟源模式 2

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

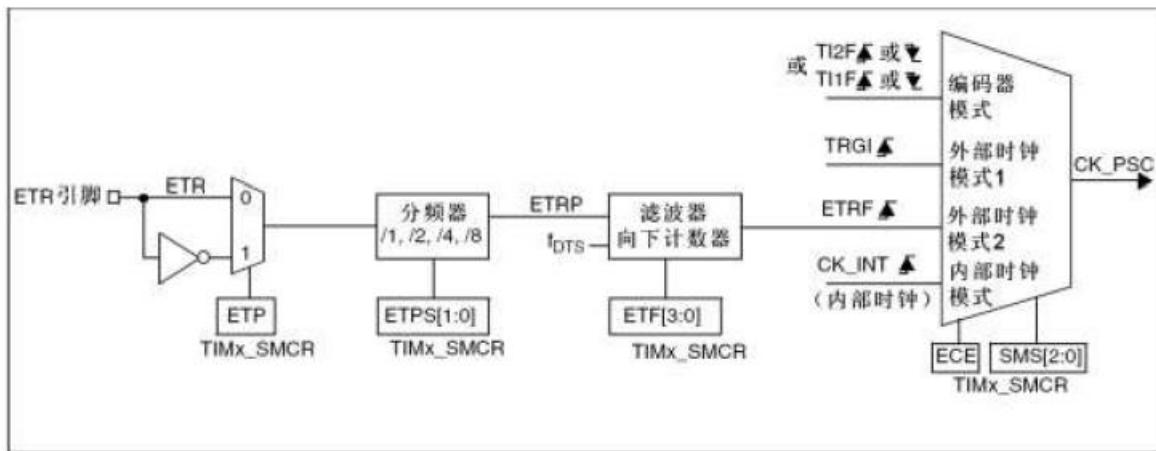


Figure 13-25 外部触发输入框图

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：

本例中不需要滤波器，置 TIM1X\*\_SMCR 寄存器中的 ETF[3:0]=0000；

设置预分频器，置 TIM1X\*\_SMCR 寄存器中的 ETPS[1:0]=01；

选择 ETR 的上升沿检测，置 TIM1X\*\_SMCR 寄存器中的 ETP=0；

开启外部时钟模式 2，写 TIM1X\*\_SMCR 寄存器中的 ECE=1；

启动计数器，写 TIM1X\*\_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。

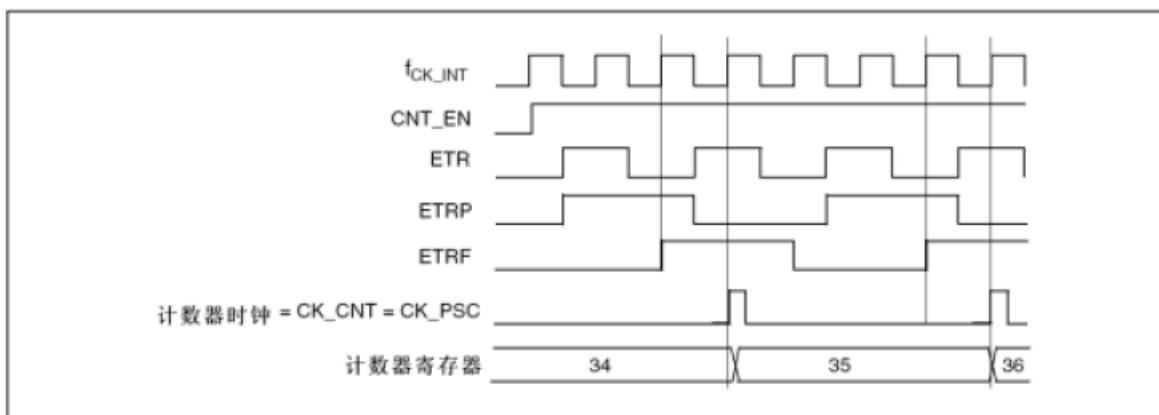


Figure 13-26 外部时钟模式 2 下的控制电路

ZBT CONFIDENTIAL



### 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 13-27 至图 13-30 是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号(TIxFPx)，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(ICxPS)。

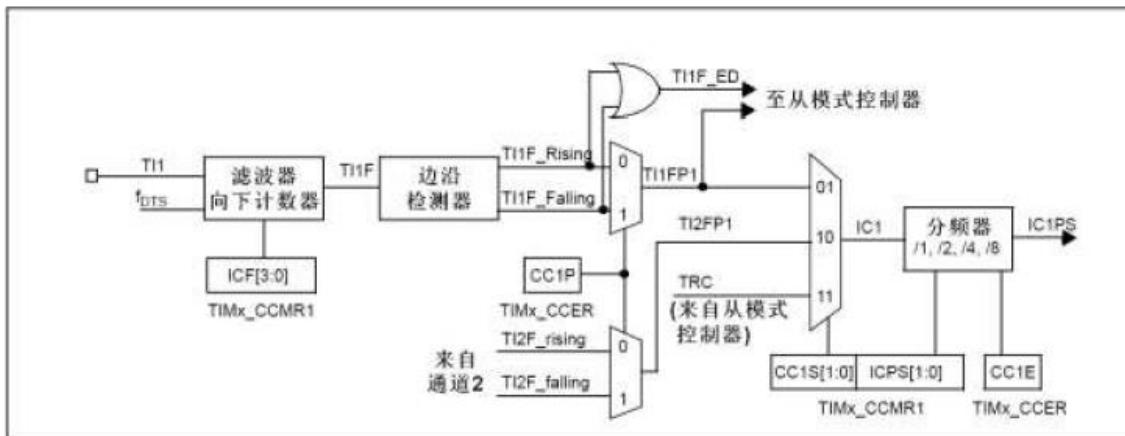


Figure 13-27 捕获/比较通道(如：通道 1 输入部分)

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

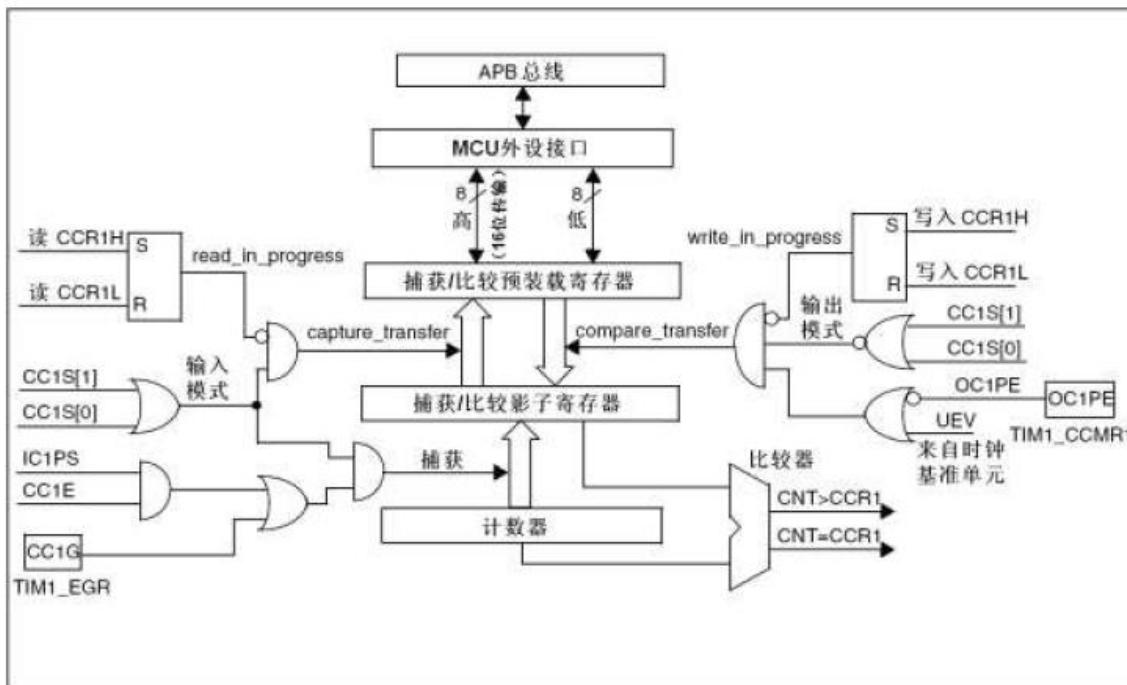


Figure 13-28 捕获/比较通道 1 的主电路

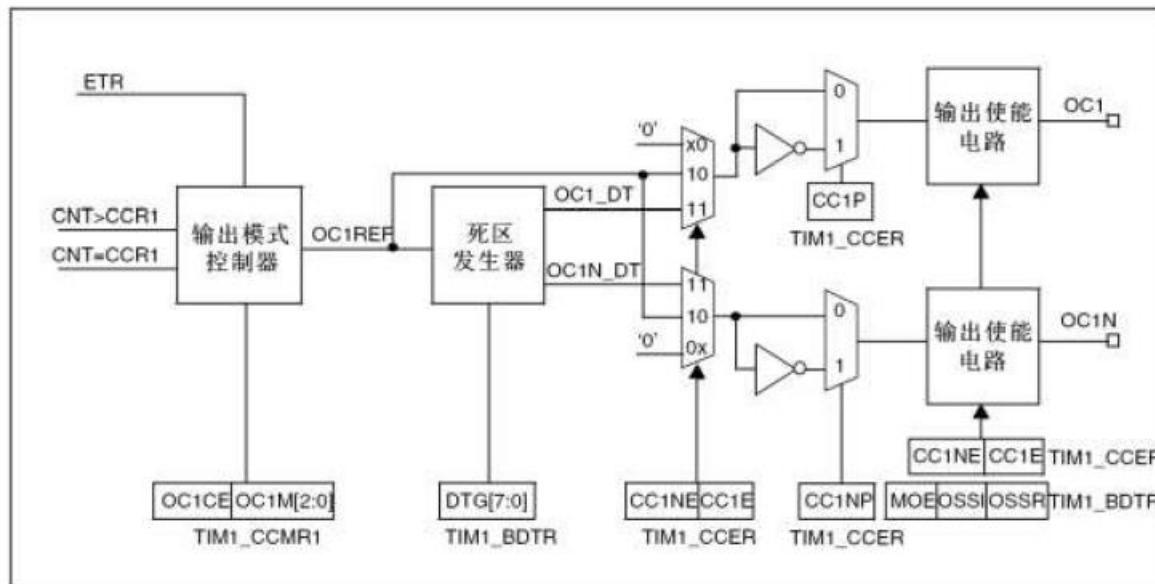


Figure 13-29 捕获/比较通道的输出部分(通道 1 至 3)

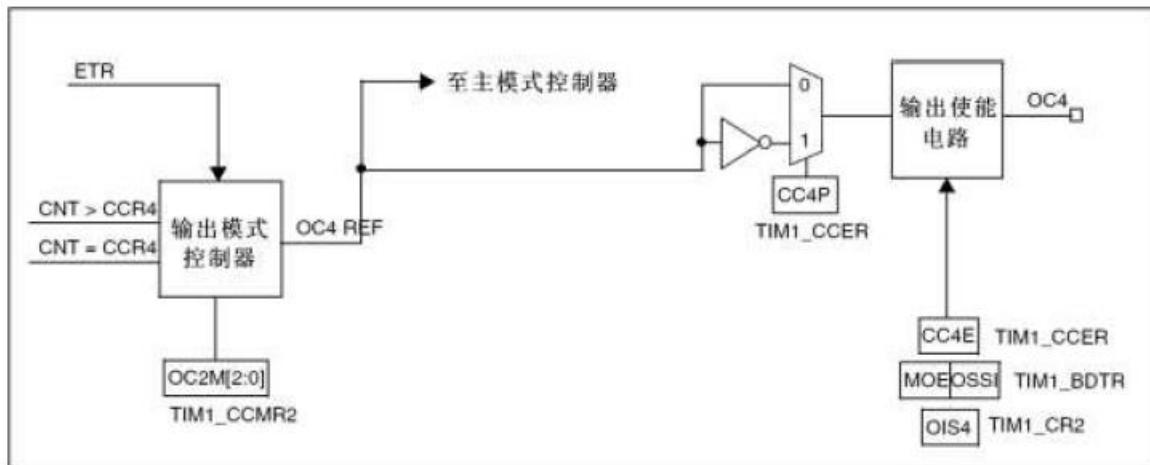


Figure 13-30 捕获/比较通道的输出部分(通道 4)

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<sub>x</sub> 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 (TIM1X\*\_CCR<sub>x</sub>) 中。当发生捕获事件时，相应的 CC<sub>x</sub>IF 标志 (TIM1X\*\_SR 寄存器) 被置 1，如果开放了中断。

如果发生捕获事件时 CC<sub>x</sub>IF 标志已经为高，那么重複捕获标志 CC<sub>x</sub>OF (TIM1X\*\_SR 寄存器) 被置 1。写 CC<sub>x</sub>IF=0 可清除 CC<sub>x</sub>IF，或读取存储在 TIM1X\*\_CCR<sub>x</sub> 寄存器中的捕获数据也可清除 CC<sub>x</sub>IF。写 CC<sub>x</sub>OF=0 可清除 CC<sub>x</sub>OF。

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 TIM1X\*\_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIM1X\*\_CCR1 必须连接到 TI1 输入，所以写入 TIM1X\*\_CCMR1 寄存器中的 CC1S=01，只要 CC1S 不为' 00'，信道被配置为输入，并且 TIM1X\*\_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<sub>x</sub> 时，输入滤波器控制位是 TIM1X\*\_CCMR<sub>x</sub> 寄存器中的 IC<sub>x</sub>F 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  $f_{DTS}$  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 `TIM1X*_CCMR1` 寄存器中写入 `IC1F=0011`。

- 选择 TI1 通道的有效转换边沿，在 `TIM1X*_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 `TIM1X*_CCMR1` 寄存器的 `IC1PS=00`)。
- 设置 `TIM1X*_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 `TIM1X*_DIER` 寄存器中的 `CC1IE` 位允许相关中断请求，通过设置 `TIM1_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 `TIM1X*_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 `TIM1X*_EGR` 寄存器中相应的 `CCxG` 位，可以通过软件产生输入捕获中断/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 IC<sub>x</sub> 信号被映射至同一个 TI<sub>x</sub> 输入。
  - 这 2 个 IC<sub>x</sub> 信号为边沿有效，但是极性相反。
  - 其中一个 TI<sub>x</sub>FP 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(TIM1X\*\_CCR1 寄存器)和占空比(TIM1X\*\_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 TIM1X\*\_CCR1 的有效输入：置 TIM1X\*\_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 TIM1X\*\_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
  - 选择 TIM1X\*\_CCR2 的有效输入：置 TIM1X\*\_CCMR1 寄存器的 CC2S=10(选中 TI1)。
  - 选择 TI1FP2 的有效极性(捕获数据到 TIM1X\*\_CCR2)：置 CC2P=1(下降沿有效)。
  - 选择有效的触发输入信号：置 TIM1X\*\_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 TIM1X\*\_SMCR 中的 SMS=100。
  - 使能捕获：置 TIM1X\*\_CCER 寄存器中 CC1E=1 且 CC2E=1。

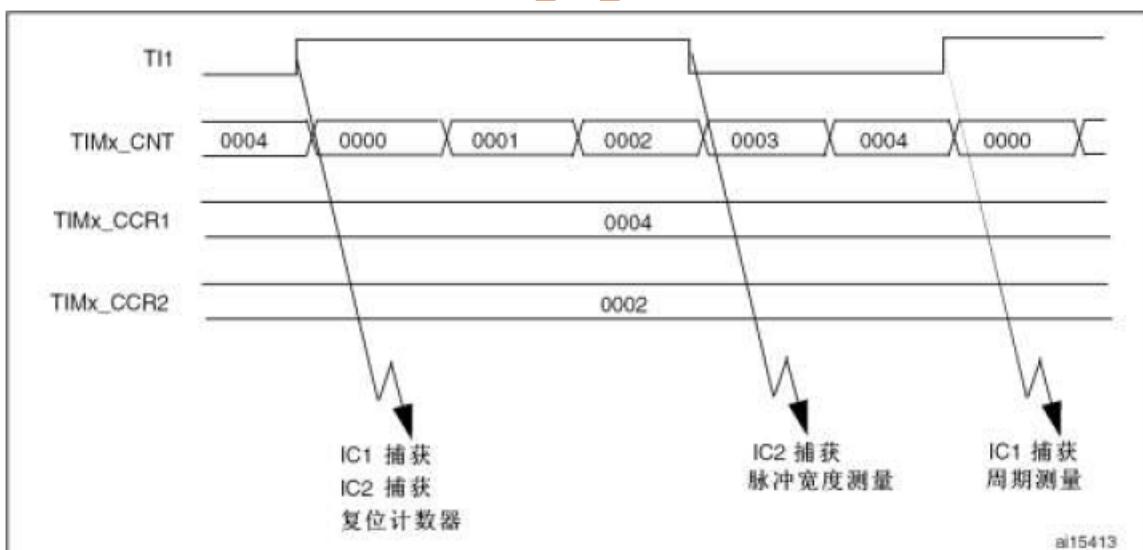


Figure 13-31 PWM 输入模式时序

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 TIM1X\*\_CH1/TIM1X\*\_CH2 信号。



### 13.3.8 强置输出模式

在输出模式( $\text{TIM1X}^*\_\text{CCMRx}$  寄存器中  $\text{CCxS}=00$ )下，输出比较信号( $\text{OCxREF}$  和相应的  $\text{OCx}/\text{OCxN}$ )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  $\text{TIM1X}^*\_\text{CCMRx}$  寄存器中相应的  $\text{OCxM}=101$ ，即可强置输出比较信号( $\text{OCxREF}/\text{OCx}$ )为有效状态。

这样  $\text{OCxREF}$  被强置为高电平( $\text{OCxREF}$  始终为高电平有效)，同时  $\text{OCx}$  得到  $\text{CCxP}$  极性相反的信号。

例如： $\text{CCxP}=0$ ( $\text{OCx}$  高电平有效)，则  $\text{OCx}$  被强置为高电平。

置  $\text{TIM1X}^*\_\text{CCMRx}$  寄存器中的  $\text{OCxM}=100$ ，可强置  $\text{OCxREF}$  信号为低。

该模式下，在  $\text{TIM1X}^*\_\text{CCRx}$  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结束。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( $\text{TIM1X}^*\_\text{CCMRx}$  寄存器中的  $\text{OCxM}$  位)和输出极性( $\text{TIM1X}^*\_\text{CCER}$  寄存器中的  $\text{CCxP}$  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( $\text{OCxM}=000$ )、被设置成有效电平( $\text{OCxM}=001$ )、被设置成无效电平( $\text{OCxM}=010$ )或进行翻转( $\text{OCxM}=011$ )。

- 设置中断状态寄存器中的标志位( $\text{TIM1X}^*\_\text{SR}$  寄存器中的  $\text{CCxIF}$  位)。
- 若设置了相应的中断屏蔽( $\text{TIM1X}^*\_\text{DIER}$  寄存器中的  $\text{CCxIE}$  位)，则产生一个中断。
- 若设置了相应的使能位( $\text{TIM1}_\text{DIER}$  寄存器中的  $\text{CCxE}$  位， $\text{TIM1}_\text{CR2}$  寄存器中的  $\text{CCDS}$  位选择 DMA 请求功能)，则产生一个 DMA 请求。

$\text{TIM1X}^*\_\text{CCMRx}$  中的  $\text{OCxPE}$  位选择  $\text{TIM1X}^*\_\text{CCRx}$  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  $\text{UEV}$  对  $\text{OCxREF}$  和  $\text{OCx}$  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  $\text{TIM1X}^*\_\text{ARR}$  和  $\text{TIM1X}^*\_\text{CCRx}$  寄存器中。
- (3) 如果要产生一个中断请求，设置  $\text{CCxIE}$  位。
- (4) 选择输出模式，例如：
  - 要求计数器与  $\text{CCRx}$  匹配时翻转  $\text{OCx}$  的输出引脚，设置  $\text{OCxM}=011$
  - 置  $\text{OCxPE} = 0$  禁用预装载寄存器
  - 置  $\text{CCxP} = 0$  选择极性为高电平有效
  - 置  $\text{CCxE} = 1$  使能输出



(5) 设置 TIM1X\*\_CR1 寄存器的 CEN 位启动计数器

TIM1X\*\_CCR<sub>x</sub> 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 TIM1X\*\_CCR<sub>x</sub> 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。

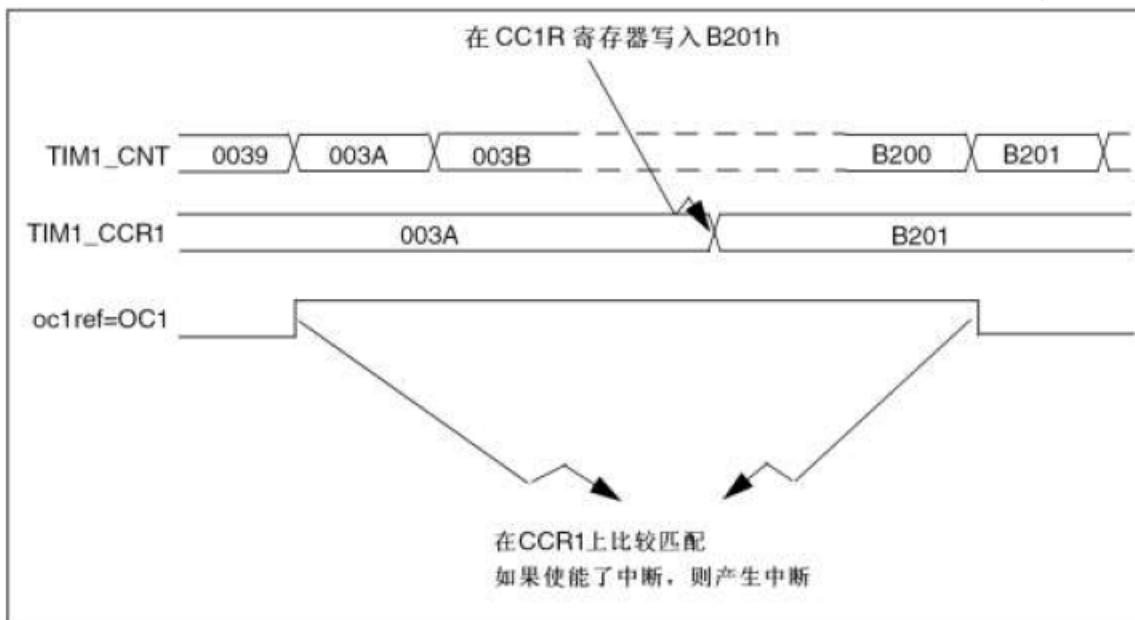


Figure 13-32 输出比较模式，翻转 OC1

ZBIT CO



### 13.3-10 PWM 模式

脉冲宽度调制模式可以产生一个由 `TIM1X*_ARR` 寄存器确定频率、由 `TIM1X*_CCRx` 寄存器确定占空比的信号。在 `TIM1X*_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 `TIM1X*_CCMRx` 寄存器的 `OCxPE` 位使能相应的预装载寄存器，最后还要设置 `TIM1X*_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 `TIM1X*_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 `TIM1X*_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(`TIM1X*_CCER` 和 `TIM1X*_BDTR` 寄存器中)`CCxE`、`CCxNE`、`MOE`、`OSSI` 和 `OSSR` 位的组合控制。详见 `TIM1X*_CCER` 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下，`TIM1X*_CNT` 和 `TIM1X*_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 `TIM1X*_CCRx≤TIM1X*_CNT` 或者 `TIM1X*_CNT≤TIM1X*_CCRx`。根据 `TIM1X*_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 `TIM1X*_CR1` 寄存器中的 `DIR` 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 `TIM1X*_CNT<TIM1X*_CCRx` 时，PWM 参考信号 `OCxREF` 为高，

否则为低。如果 `TIM1X*_CCRx` 中的比较值大于自动重装载值(`TIM1X*_ARR`)，则 `OCxREF` 保持为' 1'。如果比较值为 0，则 `OCxREF` 保持为' 0'。下图为 `TIM1X*_ARR=8` 时边沿对齐的 PWM 波形实例。

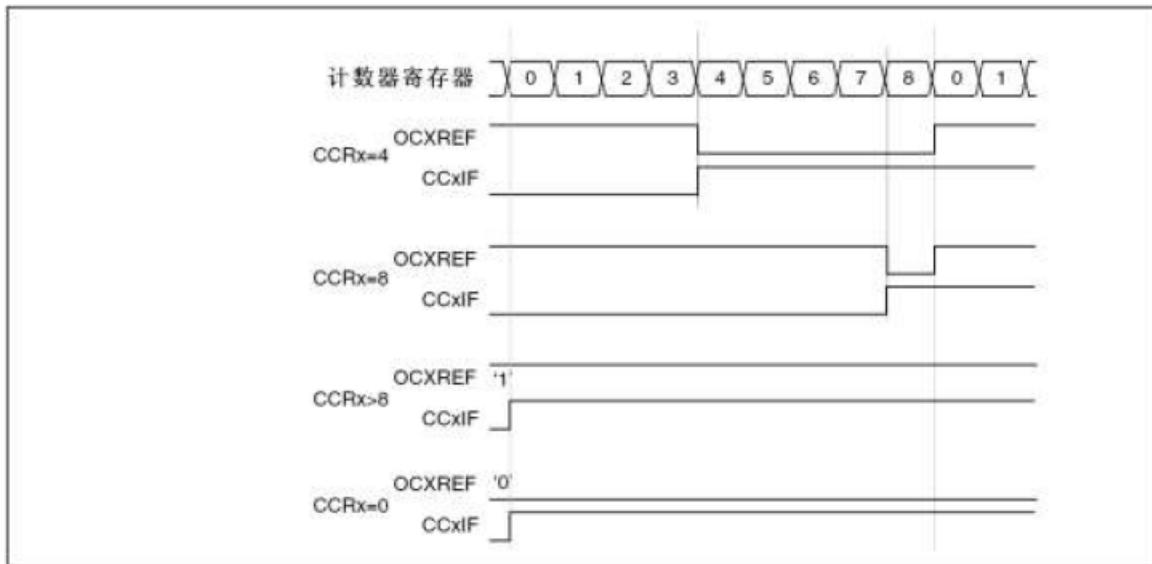


Figure 13-33 边沿对齐的 PWM 波形(ARR=8)

向下计数的配置

当 TIM1X\*\_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 TIM1X\*\_CNT>TIM1X\*\_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIM1X\*\_CCRx 中的比较值大于 TIM1X\*\_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  $\text{TIM1X\_CR1}$  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  $\text{OCxREF}/\text{OCx}$  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。 $\text{TIM1X\_CR1}$  寄存器中的计数方向位(DIR)由硬件更新, 不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子

- $\text{TIM1X\_ARR}=8$
- PWM 模式 1
- $\text{TIM1X\_CR1}$  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。

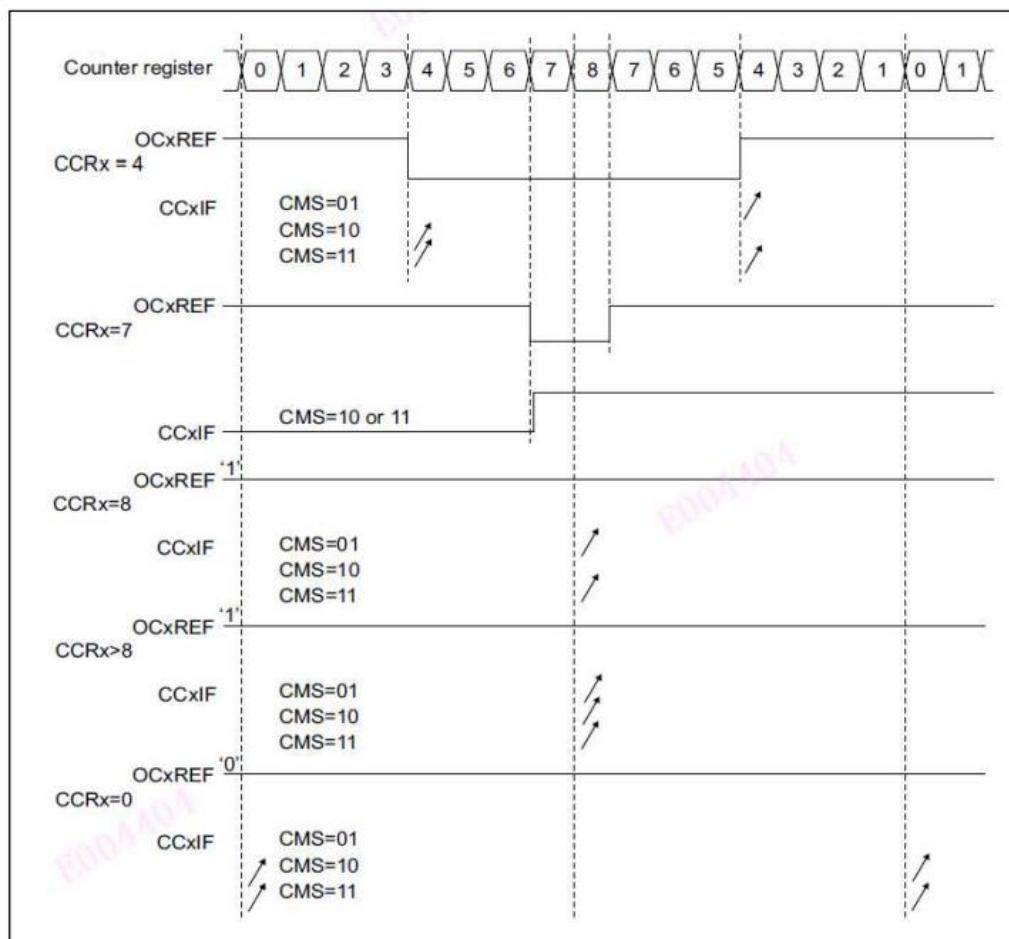


Figure 13-34 中央对齐的 PWM 波形(APR=8)



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。这就意味着计数器向上还是向下计数取决于 **TIM1X\*\_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(**TIM1X\*\_CNT>TIM1X\*\_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 **TIM1X\*\_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 **TIM1X\*\_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(TIM1X\*)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 TIM1X\*\_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：TIM1X\*\_CCER 寄存器的 CCxE 和 CCxNE 位，TIM1X\*\_BDTR 和 TIM1X\*\_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见表 12-4

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)

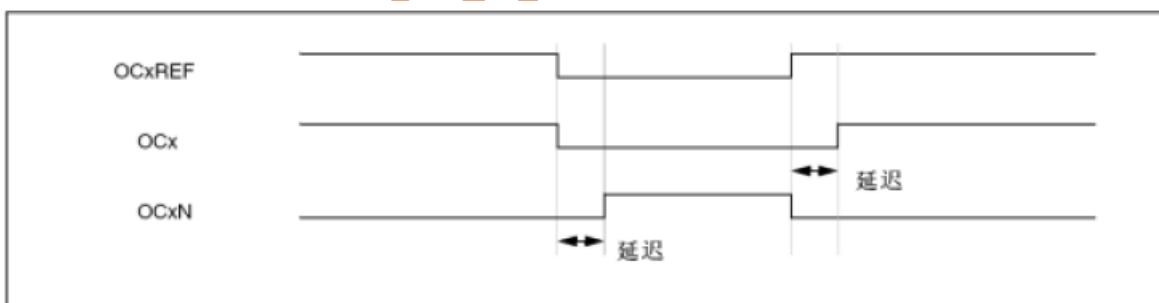


Figure 13-35 带死区插入的互补输出

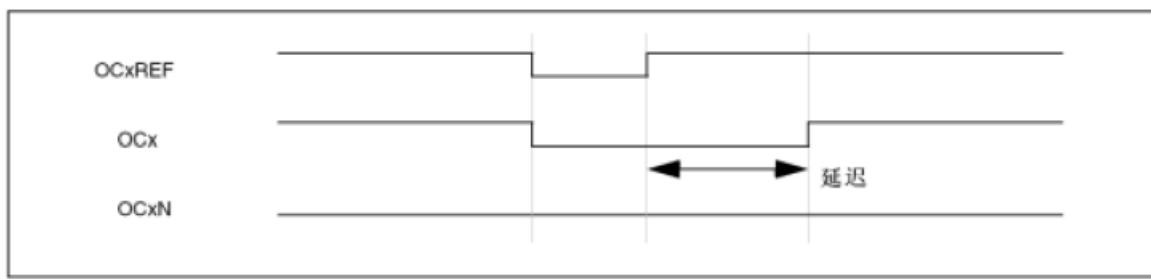


Figure 13-36 死区波形延迟大于负脉冲

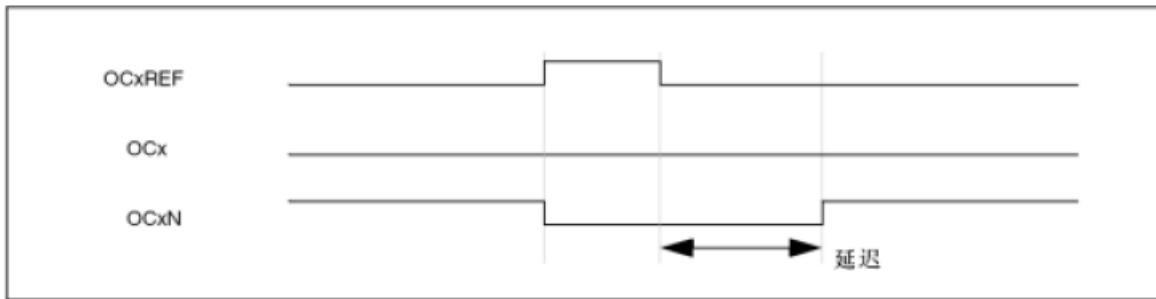


Figure 13-37 死区波形延迟大于正脉冲

每一个通道的死区延时都是相同的，是由 TIM1X\*\_BDTR 寄存器中的 DTG 位编程配置。详见 TIM1X\* 刹车和死区寄存器(TIM1X\*\_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 TIM1X\*\_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(TIM1X\*\_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，

TIM1X\*\_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 12-4 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。

系统复位后，刹车电路被禁止，MOE 位为低。设置 TIM1X\*\_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 TIM1X\*\_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 TIM1X\*\_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 OSSI=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 TIM1X\*\_DIER 寄存器中的 BIE 位，当刹车状态标志(TIM1X\*\_SR 寄存器中的 BIF 位)为‘1’时，则产生一个中断。如果设置了 TIM1\_DIER 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIM1X\*\_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 TIM1X\*\_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 TIM1X\*\_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 0 节 TIM1X\* 刹车和死区寄存器(TIM1X\*\_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。

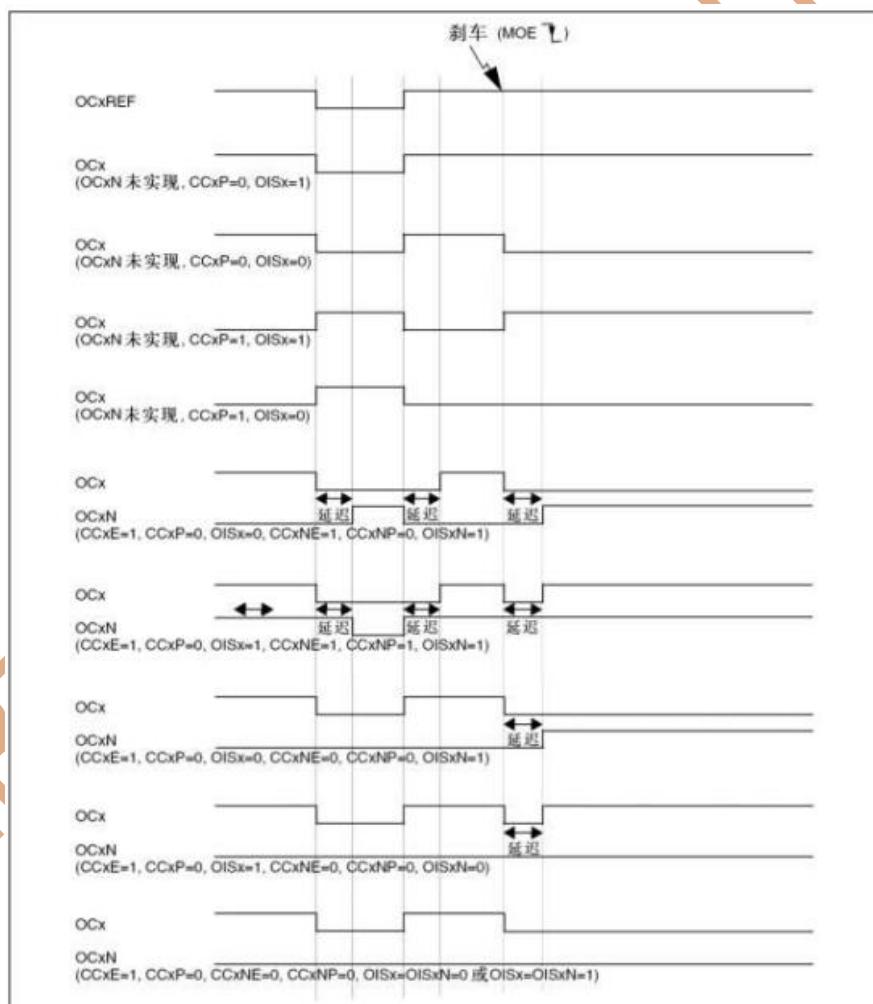


Figure 13-38 响应刹车的输出



### 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 `TIM1X*_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：`TIM1X*_SMCR` 寄存器中的 `ETPS[1:0]=00`。
- 2) 必须禁止外部时钟模式 2：`TIM1X*_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 `TIM1X*` 被置于 `PWM` 模式。

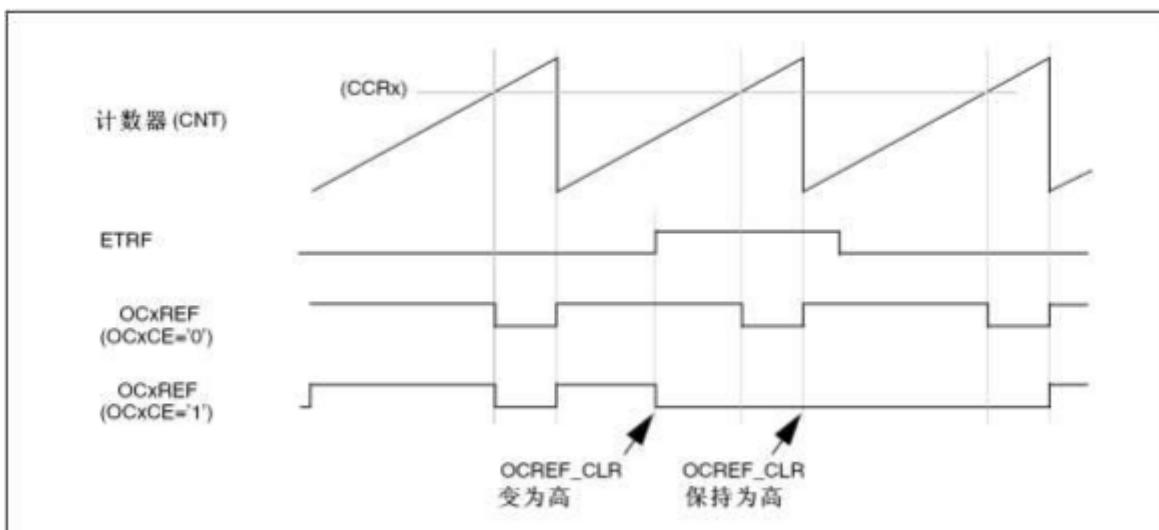
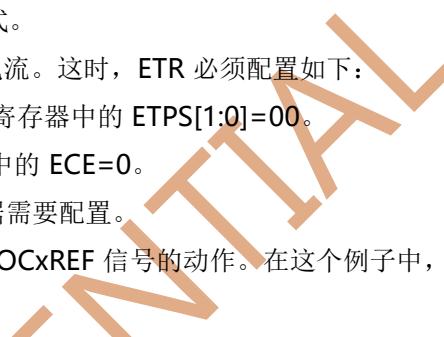


Figure 13-39 清除 TIM1X\* 的 OCxREF



### 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骤配置，并在同一个时刻同时修改更改所有信道的配置。COM 可以通过设置 TIM1X\*\_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(TIM1X\*\_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIM1X\*\_DIER 寄存器的 COMIE 位，则产生一个中断。；或者如果已设置了 TIM1\_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。

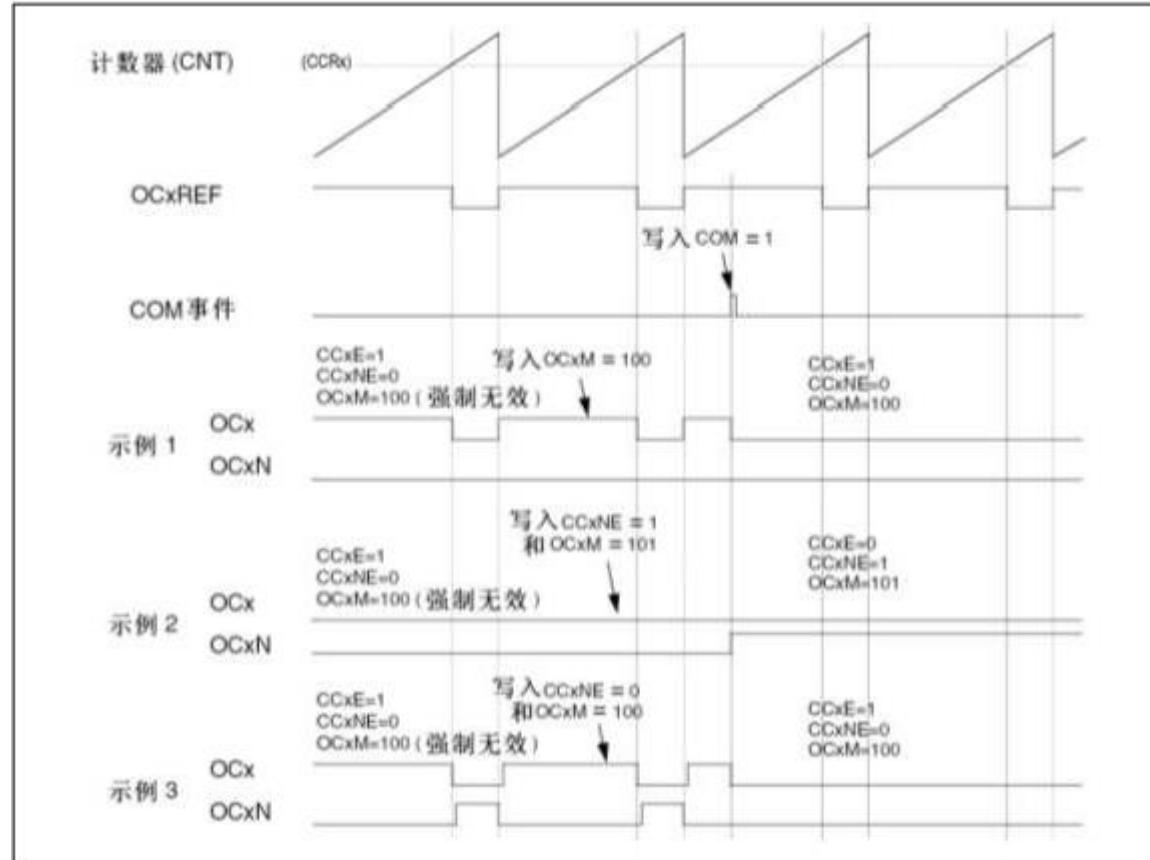


Figure 13-40 产生六步 PWM, 使用 COM 的例子(OSSR=1)



### 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编程控制的脉冲。

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 `TIM1X*_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  $CNT < CCRx \leq ARR$ (特别地， $0 < CCRx$ )
- 向下计数方式：计数器  $CNT > CCRx$

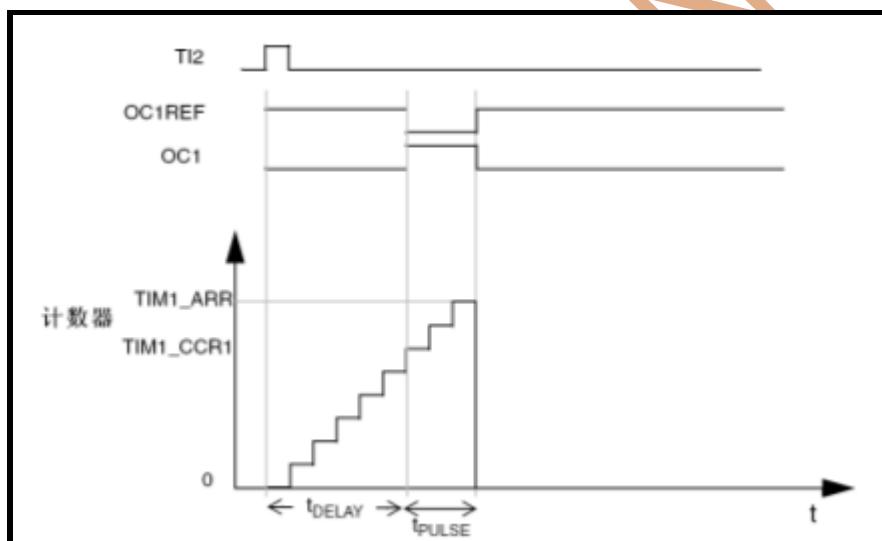


Figure 13-41 单脉冲模式的例子

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 `TIM1X*_CCMR1` 寄存器中的 `CC2S=01`，把 `TI2FP2` 映像到 `TI2`。
- 置 `TIM1X*_CCER` 寄存器中的 `CC2P=0`，使 `TI2FP2` 能够检测上升沿。
- 置 `TIM1X*_SMCR` 寄存器中的 `TS=110`, `TI2FP2` 作为从模式控制器的触发(`TRGI`)。
- 置 `TIM1X*_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 `TIM1X*_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(`TIM1X*_ARR - TIM1X*_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 **TIM1X\*\_CCMR1** 寄存器的 **OC1M=111**，进入 **PWM** 模式 2；根据需要有选择地使能预装载寄存器：置 **TIM1X\*\_CCMR1** 中的 **OC1PE=1** 和 **TIM1X\*\_CR1** 寄存器中的 **ARPE**；然后在 **TIM1X\*\_CCR1** 寄存器中填写比较值，在 **TIM1X\*\_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，**TIM1X\*\_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 **TIM1X\*\_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。  
**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 **tDELAY**。  
如果要以最小延时输出波形，可以设置 **TIM1X\*\_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

选择编码器接口的方法是：如果计数器只在 **TI2** 的边沿计数，则置 **TIM1X\*\_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 **TIM1X\*\_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 12-1，假定计数器已经启动(**TIM1X\*\_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 **TIM1X\*\_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 **TIM1X\*\_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 **TIM1X\*\_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。

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

Table 13-1 计数方向与编码器信号的关系

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' (TIM1X\*\_CCMR1 寄存器, IC1FP1 映射到 TI1)
- CC2S= '01' (TIM1X\*\_CCMR2 寄存器, IC2FP2 映射到 TI2)
- CC1P= '0' (TIM1X\*\_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- CC2P= '0' (TIM1X\*\_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)
- SMS= '011' (TIM1X\*\_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效)
- CEN= '1' (TIM1X\*\_CR1 寄存器, 计数器使能)

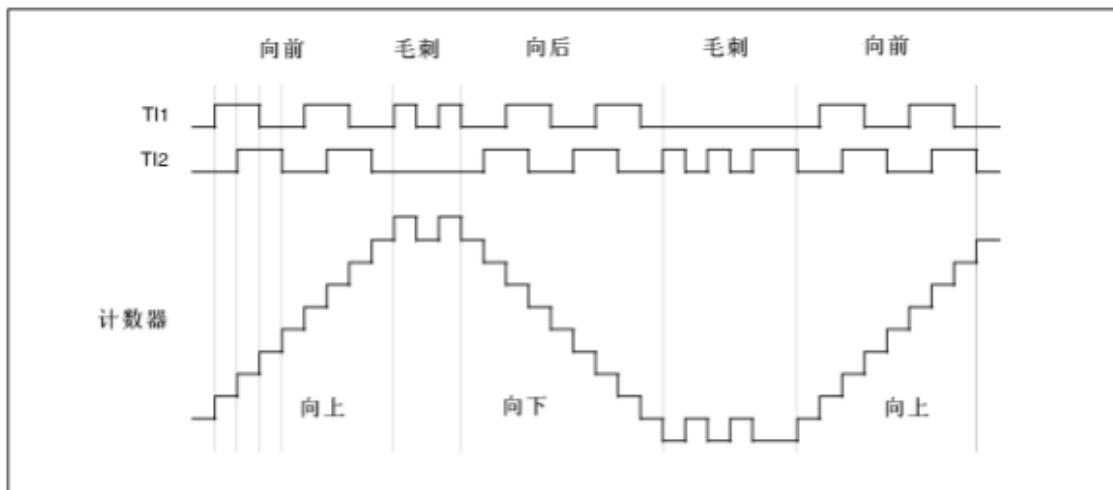


Figure 13-42 编码器模式下的计数器操作实例

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)

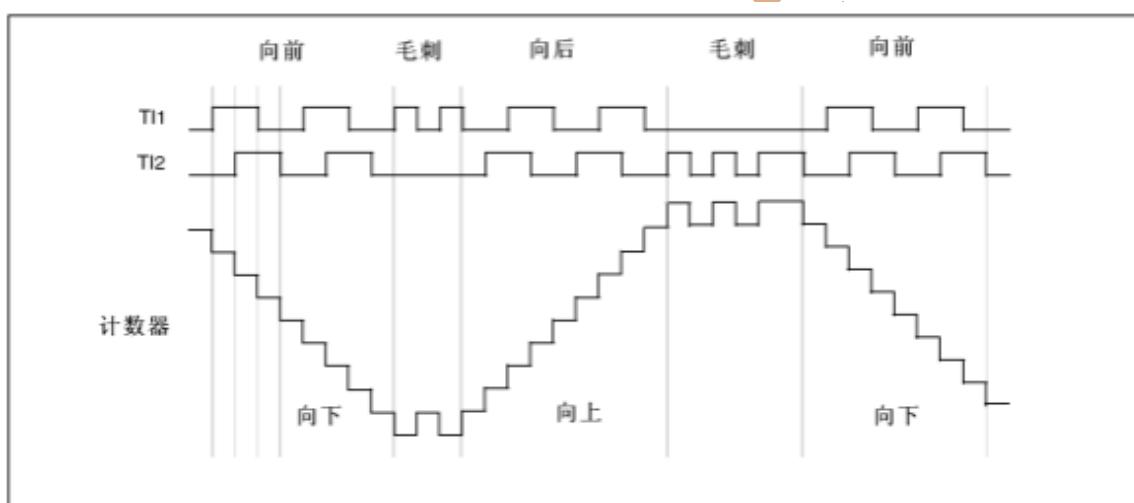


Figure 13-43 IC1FP1 反相的编码器接口模式实例

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)。它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

TIM1X\*\_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 TIM1X\*\_CH1、TIM1X\*\_CH2 和 TIM1X\*\_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 12.3.18 给出了此特性用于连接霍尔传感器的例子。

ZBT CONFIDENTIAL



### 13.3-18 与霍尔传感器的接口

使用高级控制定时器(TIM1X\*)产生 PWM 信号驱动马达时，可以用另一个通用 TIM2 定时器作为“接口定时器”来连接霍尔传感器，见图 13-44，3 个定时器输入脚(CC1、CC2、CC3)通过一个异或门连接到 TI1 输入通道(通过设置 TIM1X\*\_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置于复位模式，从输入是 TI1F\_ED。每当 3 个输入之一变化时，计数器从新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较信道 1 配置为捕获模式，捕获信号为 TRC(见图 13-27)。捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级定时器 TIM1X\* 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1X\*。

举例：霍尔输入连接到 TIM2X\* 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIM1X\* 的 PWM 配置。

- 置 TIM2X\*\_CR2 寄存器的 TI1S 位为‘1’，配置三个定时器输入逻辑或到 TI1 输入，
- 时基编程：置 TIM2X\*\_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一

个最大的计数器周期，它长于传感器上的两次变化的时间间隔。

- 设置信道 1 为捕获模式(选中 TRC)：置 TIM2X\*\_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIM2X\*\_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIM2X\*\_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1X\* 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIM1X\*\_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件 (TIM1X\*\_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一步的 PWM 控制位(CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个实例。

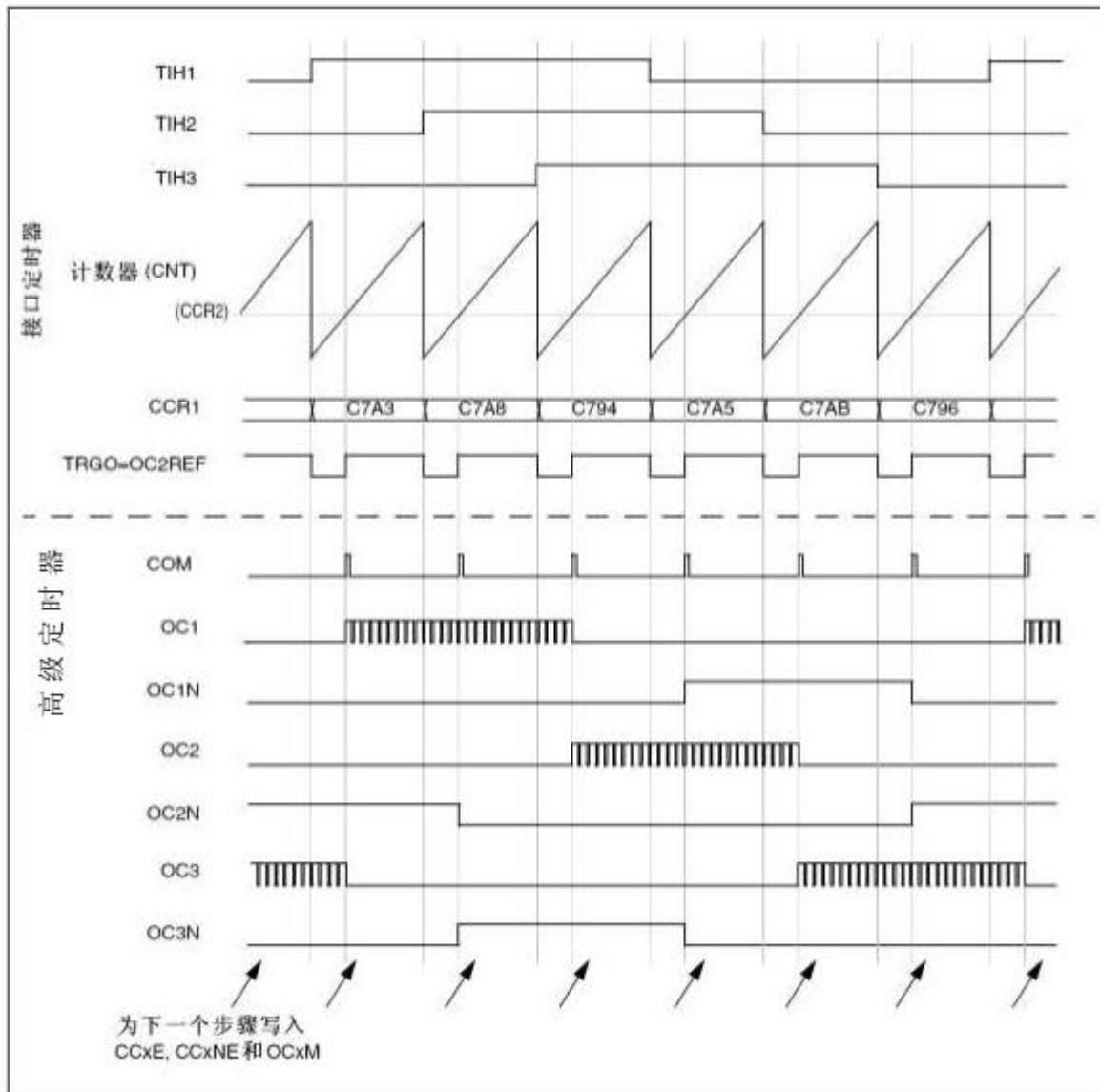


Figure 13-44 霍尔传感器接口的实例

Note : TIH1 : 可设定 TIM1XCH1IN\_SEL[2:0], 选择对应的输入脚位;

TIH2 : 可设定 TIM1XCH2IN\_SEL[2:0], 选择对应的输入脚位;

TIH3: 可设定 TIM1XCH3IN\_SEL[2:0], 选择对应的输入脚位;



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBT CONFIDENTIAL



### 13.3-19 TIM1X\* 定时器和外部触发的同步

TIM1X\* 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。

#### 13.3-19.1 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 TIM1X\*\_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(TIM1X\*\_ARR, TIM1X\*\_CCR<sub>x</sub>)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 TIM1X\*\_CCMR1 寄存器中 CC1S=01。置 TIM1X\*\_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- 置 TIM1X\*\_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM1X\*\_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM1X\*\_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(TIM1X\*\_SR 寄存器中的 TIF 位)被设置，根据 TIM1X\*\_DIER 寄存器中 TIE(中断使能)位的设置和 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 TIM1X\*\_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。

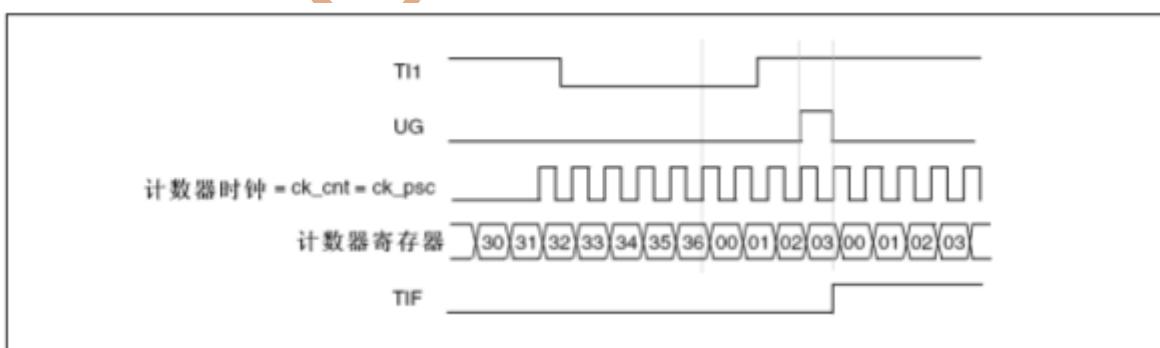


Figure 13-45 复位模式下的控制电路



### 13.3-19.2 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 TIM1X\*\_CCMR1 寄存器中 CC1S=01。置 TIM1X\*\_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIM1X\*\_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM1X\*\_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM1X\*\_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 TIM1X\*\_SR 中的 TIF 标置。

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。

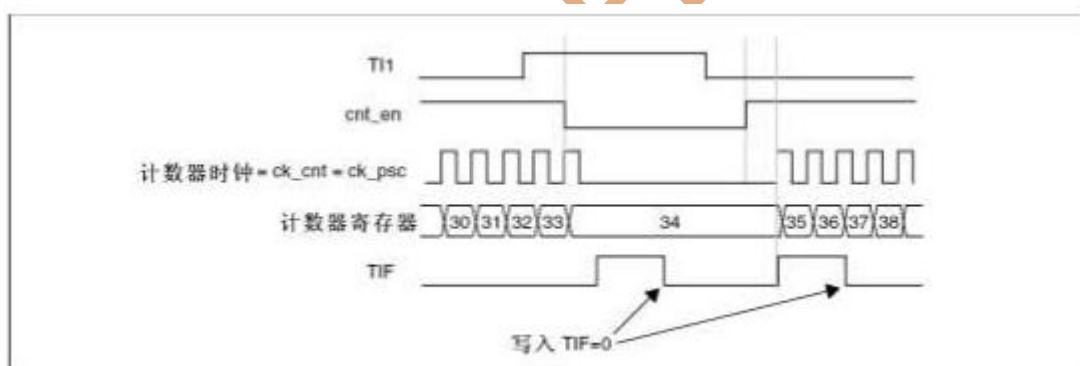


Figure 13-46 门控模式下的控制电路



### 13.3-19.3 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 TIM1X\*\_CCMR1 寄存器中 CC2S=01。置 TIM1X\*\_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- 置 TIM1X\*\_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM1X\*\_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。

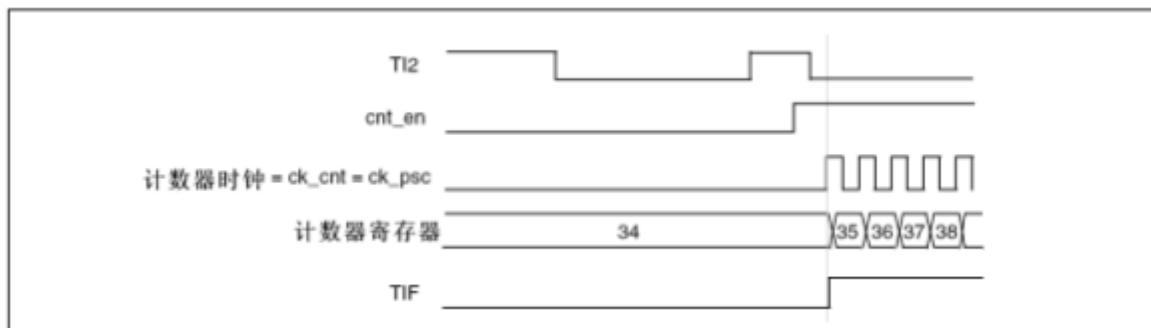


Figure 13- 1 触发器模式下的控制电路



### 13.3-19.4 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。不建议使用 TIM1X\*\_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：

通过 TIM1X\*\_SMCR 寄存器配置外部触发输入电路：

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

按如下配置信道 1，检测 TI 的上升沿：

- IC1F=0000: 没有滤波
- 触发操作中不使用捕获预分频器，不需要配置
- 置 TIM1X\*\_CCMR1 寄存器中 CC1S=01，选择输入捕获源
- 置 TIM1X\*\_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)
- 置 TIM1X\*\_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIM1X\*\_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。

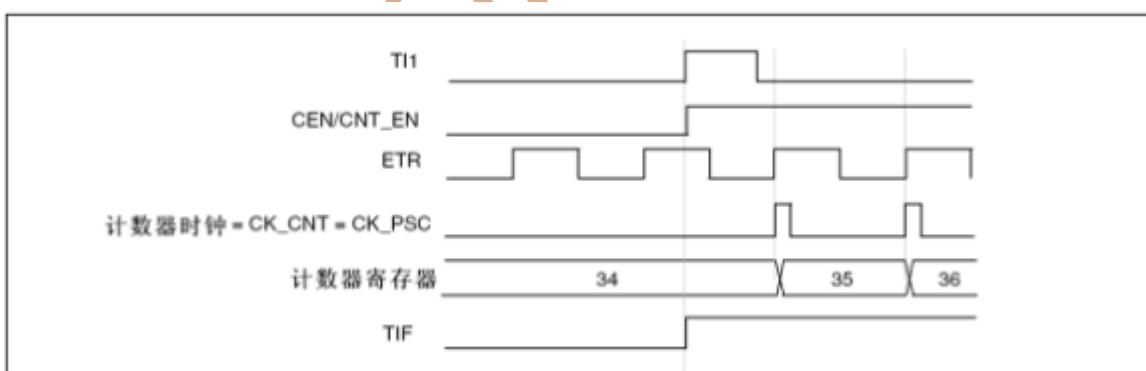


Figure 13-47 外部时钟模式 2 + 触发模式下的控制电路

### 13.3-20 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。详见下一章 13.3.15 定时器同步

### 13.3-21 调试模式

当微控制器进入调试模式时(Cortex-M0+核心停止)，根据 DBG 模块中 DBG\_TIM1X\*\_STOP 的设置，TIM1X\* 计数器可以或者继续正常操作，或者停止。



### 13.4 高级控制定时器 TIM1X(TIM1/1A/1B)寄存器列表

可以用字(32位)的方式操作这些外设寄存器:

**TIM1** 基地址 **0x4001 2C00**

**TIM1A** 基地址 **0x4001 3400**

**TIM1B** 基地址 **0x4000 F400**

偏移地址	名称	描述	复位值
0x00	TIM1X*_CR1	TIM1X* 控制寄存器 1	0x0000 0000
0x04	TIM1X*_CR2	TIM1X* 控制寄存器 2	0x0000 0000
0x08	TIM1X*_SMCR	TIM1X* 从模式控制寄存器	0x0000 0000
0x0C	TIM1X*_DIER	TIM1X* 中断使能寄存器	0x0000 0000
0x10	TIM1X*_SR	TIM1X* 状态寄存器	0x0000 0000
0x14	TIM1X*_EGR	TIM1X* 事件产生寄存器	0x0000 0000
0x18	TIM1X*_CCMR1	TIM1X* 捕获/比较模式寄存器 1	0x0000 0000
0x1C	TIM1X*_CCMR2	TIM1X* 捕获/比较模式寄存器 2	0x0000 0000
0x20	TIM1X*_CCER	TIM1X* 捕获/比较使能寄存器	0x0000 0000
0x24	TIM1X*_CNT	TIM1X* 计数器	0x0000 0000
0x28	TIM1X*_PSC	TIM1X* 预分频器	0x0000 0000
0x2C	TIM1X*_ARR	TIM1X* 自动重装载寄存器	0x0000 0000
0x30	TIM1X*_RCR	TIM1X* 重复计数寄存器	0x0000 0000
0x34	TIM1X*_CCR1	TIM1X* 捕获/比较寄存器 1	0x0000 0000
0x38	TIM1X*_CCR2	TIM1X* 捕获/比较寄存器 2	0x0000 0000
0x3C	TIM1X*_CCR3	TIM1X* 捕获/比较寄存器 3	0x0000 0000
0x40	TIM1X*_CCR4	TIM1X* 捕获/比较寄存器 4	0x0000 0000
0x44	TIM1X*_BDTR	TIM1X* 刹车和死区寄存器	0x0000 0000
0x48	TIM1X*_DCR	TIM1X* DMA控制寄存器	0x0000 0000
0x4C	TIM1X*_DMAR	TIM1X* 连续模式的DMA地址	0x0000 0000

Table 13-2 TIM1,TIM1A,TIM1B 寄存器列表和复位值

Note : TIM1X\* → TIM1 , TIM1A ,TIM1B



### 13.5 TIM1X\*寄存器说明

#### 13.5-1 TIM1X\* 控制寄存器 1(TIM1X\*\_CR1)

地址偏移: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKD[1:0]		AR	CMS[1:0]		DIR	OP	UR	UD	CE
								PE			M	S	IS	N	
						R/W		R/W	R/W		R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:10	-	保留, 始终读为 0。	0	-
9:8	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: tDTS = tCK_INT 01: tDTS = 2 x tCK_INT 10: tDTS = 4 x tCK_INT 11: 保留, 不要使用这个配置	0x0	R/W
7	ARPE	自动重装载预装载允许位(Auto-reload preload enable) 0: TIM1X*_ARR 寄存器没有缓冲; 1: TIM1X*_ARR 寄存器被装入缓冲器。	0	R/W
6:5	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(TIM1X*_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(TIM1X*_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(TIM1X*_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。	0x0	R/W



位	标记	功能描述	复位值	读写
4	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。	0	R/W
3	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。	0	R/W
2	URS	更新请求源(Update request source) 软件通过该位选择UEV 事件的源 0: 如果置 0, 则下述任一事件产生更新中断: - 计数器溢出/下溢 - 设置 UG 位 - 从模式控制器产生的更新 1: 如果置 1, 则只有计数器溢出/下溢才产生更新中断。	0	R/W
1	UDIS	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、CCRx) 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。	0	R/W
0	CEN	使能计数(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。	0	R/W



### 13.5.2 TIM1X\* 控制寄存器 2(TIM1X\*\_CR2)

地址偏移: 0x04 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OIS4	OIS3_N	OIS3	OIS2_N	OIS2	OIS1_N	OIS1	T1S	[1:0]	MMS	DIR	CCDS	CCUS	保留	CCP C
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:15	-	保留, 始终读为 0。	0x0	-
14	OIS4	输出空闲状态 4(OC4 输出)。参见 OIS1 位。	0	R/W
13	OIS3N	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。	0	R/W
12	OIS3	输出空闲状态 3(OC3 输出)。参见 OIS1 位。	0	R/W
11	OIS2N	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。	0	R/W
10	OIS2	输出空闲状态 2(OC2 输出)。参见 OIS1 位。	0	R/W
9	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(TIMX*_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。	0	R/W
8	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。	0	R/W



位	标记	功能描述	复位值	读写
7	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。	0	R/W
6:4	MMS [2:0]	主模式选择(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)。	0x0	R/W
3	CCDS	捕获/比较 DMA 请求选择 0:当CCX事件发生时会产生CCX DMA请求 1:当CCX更新事件发生时会产生CCX DMA请求	0	R/W
2	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。	0	R/W
1	-	保留, 始终读为 0。	0	-



位	标记	功能描述	复位值	读写
0	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。	0	R/W

ZBT CONFIDENTIAL



### 13.5.3 TIM1X\* 从模式控制寄存器(TIM1X\*\_SMCR)

地址偏移: 0x08 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]			MS M	TS[2:0]			保 留	SMS[2:0]			
R/ W	R/ W	R/ W	R/ W	R/ W	R/W		R/ W	R/W		R/ W	R/ W	R/ W	R/ W	R/ W	

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15	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。	0	R/W
14	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。	0	R/W
13:12	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 TIM1X*CLK 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;	0	R/W



位	标记	功能描述	复位值	读写
		10: ETRP 频率除以 4; 11: ETRP 频率除以 8。		
11:8	ETF[3:0]	<p>外部触发滤波(External trigger filter)</p> <p>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。</p> <p>0000: 无滤波器，以fDTS 采样 1000: 采样频率 fSAMPLING=fDTS/8, N=6</p> <p>0001: 采样频率fSAMPLING=fCK_INT, N=2 1001: 采样频率fSAMPLING=fDTS/8, N=8</p> <p>0010: 采样频率fSAMPLING=fCK_INT, N=4 1010: 采样频率 fSAMPLING=fDTS/16, N=5</p> <p>0011: 采样频率 fSAMPLING=fCK_INT, N=8 1011: 采样频率 fSAMPLING=fDTS/16, N=6</p> <p>0100: 采样频率fSAMPLING=fDTS/2, N=6 1100: 采样频率 fSAMPLING= fDTS /16, N=8</p> <p>0101: 采样频率fSAMPLING=fDTS/2, N=8 1101: 采样频率 fSAMPLING= fDTS /32, N=5</p> <p>0110: 采样频率fSAMPLING=fDTS/4, N=6 1110: 采样频率 fSAMPLING= fDTS /32, N=6</p> <p>0111: 采样频率fSAMPLING=fDTS/4, N=8 1111: 采样频率 fSAMPLING= fDTS /32, N=8</p>	0	R/W
7	MSM	<p>主/从模式(Master/slave mode) 0: 无作用;</p> <p>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。</p>	0	R/W
6:4	TS[2:0]	<p>触发选择(Trigger selection)</p> <p>这 3 位选择用于同步计数器的触发输入。</p> <p>000: 内部触发 0(ITR0) ; 100: TI1 的边沿检测器(TI1F_ED)</p> <p>001: 内部触发 1(ITR1) ; 101: 滤波后的定时器输入 1(TI1FP1)</p> <p>010: 内部触发 2(ITR2) ; 110: 滤波后的定时器输入 2(TI2FP2)</p> <p>011: 内部触发 3(ITR3) ; 111: 外部触发输入(ETRF)</p> <p>更多有关ITRx 的细节，参见 Table13- 1。</p>	0	R/W



位	标记	功能描述	复位值	读写
		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。		
3	-	保留，始终读为 0。	0	-
2:0	SMS[2:0]	<p>从模式选择(Slave mode selection)</p> <p>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)</p> <p>000: 关闭从模式 – 如果CEN=1，则预分频器直接由内部时钟驱动。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p>	0	R/W

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	Tim2_trgo	irq_timer10	irq_timer11	irq_pca
TIM1A	TIM2A_trgo	irq_timer10	irq_timer11	irq_pca
TIM1B	TIM2B_trgo	irq_timer10	irq_timer11	irq_pca

Table 13-3 TIM1X\* 内部触发连接



### 13.5.4 TIM1X\* 中断与 DMA 请求使能寄存器(TIM1X\*\_DIER)

地址偏移: 0x0C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	TDE	CO M DE	CC4 DE	CC3 DE	CC2 DE	CC1 DE	UD E	BIE	TIE	CO MIE	CC4 IE	CC3 IE	CC2 IE	CC1 IE	UIE
--	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:15	-	保留, 始终读为 0。	0	-
14	TDE	始能DMA请求(TDREQ) 0: 无DMA请求 1: 始能触发DMA请求(TDREQ)	0	R/W
13	COMDE	始能COM DMA请求(COMREQ) 0: 无DMA请求 1: 始能COM DMA请求(COMREQ)	0	R/W
12	CC4DE	始能捕获/比较4DMA请求 (CC4DREQ) 0: 无DMA请求 1: 始能捕获/比较4DMA请求 (CC4DREQ)	0	R/W
11	CC3DE	始能DMA请求(UDREQ) 0: 无DMA请求 1: 始能捕获/比较3DMA请求 (CC3DREQ)	0	R/W
10	CC2DE	始能DMA请求(UDREQ) 0: 无DMA请求 1: 始能捕获/比较2DMA请求 (CC2DREQ)	0	R/W



9	CC1DE	始能DMA请求(UDREQ) 0: 无DMA请求 1: 始能捕获/比较1DMA请求 (CC1DREQ)	0	R/W
8	UDE	始能DMA请求(UDREQ) 0: 无DMA请求 1: 始能更新DMA请求(UDREQ)	0	R/W
7	BIE	允许刹车中断(Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。	0	R/W
6	TIE	触发中断使能(Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。	0	R/W
5	COMIE	允许COM 中断(COM interrupt enable) 0: 禁止COM 中断; 1: 允许COM 中断。	0	R/W
4	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。	0	R/W
3	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。	0	R/W
2	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。	0	R/W
1	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。	0	R/W
0	UIE	允许更新中断(Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。	0	R/W



### 13.5.5 TIM1X\* 状态寄存器(TIM1X\*\_SR)

地址偏移: 0x10 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	--	--	CC4OF	CC3OF	CC2OF	CC1OF	--	BIF	TIF	CO MIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
--	R	R	RC W0	RC W0	RC W0	RC W0	R	RC W0	R	R	RC W0				

位	标记	功能描述	复位值	读写
31:13	-	保留, 始终读为 0。	0	-
12	CC4OF	捕获/比较 4 重复捕获标记(Capture/Compare 4 overcapture flag) 参见CC1OF 描述。	0	RC W0
11	CC3OF	捕获/比较 3 重复捕获标记(Capture/Compare 3 overcapture flag) 参见CC1OF 描述。	0	RC W0
10	CC2OF	捕获/比较 2 重复捕获标记(Capture/Compare 2 overcapture flag) 参见CC1OF 描述。	0	RC W0
9	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 TIM1X*_CCR1 寄存器时, CC1IF 的状态已经为'1'。	0	RC W0
8	-	保留, 始终读为 0。	0	-
7	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。	0	RC W0



6	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。	0	RC W0
5	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。	0	RC W0
4	CC4IF	捕获/比较 4 中断标记(Capture/Compare 4 interrupt flag) 参考CC1IF 描述。	0	RC W0
3	CC3IF	捕获/比较 3 中断标记(Capture/Compare 3 interrupt flag) 参考CC1IF 描述。	0	RC W0
2	CC2IF	捕获/比较 2 中断标记(Capture/Compare 2 interrupt flag) 参考CC1IF 描述。	0	RC W0
1	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 TIM1X*_CR1 寄存器的CMS 位)。它由软件清' 0' 。 0: 无匹配发生; 1: TIM1X*_CNT 的值与TIM1X*_CCR1 的值匹配。 当 TIM1X*_CCR1 的内容大于 TIM1X*_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TIM1X*_CCR1 清' 0' 。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIM1X*_CCR1(在 IC1 上检测到与所选极性相同的边沿)。	0	RC W0
		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' :		RC W0



0	UIF	-若TIM1X*_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TIM1X*_CR1 寄存器的URS=0、UDIS=0，当设置TIM1X*_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TIM1X*_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 13.5-3 TIM1X* 从模式控制寄存器(TIM1X*_SMCR)).	0	
---	-----	---	---	--

ZBT CONFIDENTIAL



### 13.5.6 TIM1X\* 事件产生寄存器(TIM1X\*\_EGR)

地址偏移: 0x14 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	--	--	--	--	--	--	--	BG	TG	CO M G	CC4 G	CC3 G	CC2 G	CC1 G	UG
--	--	--	--	--	--	--	--	WO	WO	WO	WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:8	-	保留, 始终读为 0。	0x0	-
7	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, 则产生相应的中断。	0	WO
6	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: TIM1X*_SR 寄存器的TIF=1, 若开启对应的中断, 则产生相应的中断。	0	WO
5	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。	0	WO
4	CC4G	产生捕获/比较 4 事件(Capture/Compare 4 generation) 参考CC1G 描述。	0	WO
3	CC3G	产生捕获/比较 3 事件(Capture/Compare 3 generation) 参考CC1G 描述。	0	WO
2	CC2G	产生捕获/比较 2 事件(Capture/Compare 2 generation) 参考CC1G 描述。	0	WO



1	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，则产生相应的中断。若信道 CC1 配置为输入： 当前的计数器值被捕获至TIM1X*_CCR1 寄存器；设置CC1IF=1，若开启对应的中断，则产生相应的中断。若CC1IF 已经为 1，则设置CC1OF=1。	0	WO
0	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TIM1X*_ARR 的值。	0	WO



### 13.5.7 TIM1X\* 捕获/比较模式寄存器 1 (TIM1X\*\_CCMR1)

地址偏移: 0x18 复位值: 0x0000 0000

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 CC<sub>x</sub>S 位定义。该寄存器其它位的作用在输入和输出模式下不同。OC<sub>xx</sub> 描述了信道在输出模式下的功能, IC<sub>xx</sub> 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:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15	OC2CE	输出比较 2 清 0 使能(Output Compare 2 clear enable)	0	R/W
14:12	OC2M[2:0]	输出比较 2 模式(Output Compare 2 mode)	0x0	R/W
11	OC2PE	输出比较 2 预装载使能(Output Compare 2 preload enable)	0	R/W
10	OC2FE	输出比较 2 快速使能(Output Compare 2 fast enable)	0	R/W
9:8	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 TIM1X*_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM1X*_CCER 寄存器的 CC2E=0)才是可写的。	0x0	R/W
7	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;	0	R/W



位	标记	功能描述	复位值	读写
		1: 一旦检测到ETRF 输入高电平, 清除OC1REF=0。		
6:4	OC1M[2:0] ]	<p>输出比较 1 模式(Output Compare 1 mode)</p> <p>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。</p> <p>000: 冻结。输出比较寄存器 TIM1X*_CCR1 与计数器 TIM1X*_CNT 间的比较对 OC1REF 不起作用;</p> <p>001: 匹配时设置通道 1 为有效电平。当计数器TIM1X*_CNT 的值与捕获 /比较寄存器 1(TIM1X*_CCR1)相同时, 强制OC1REF 为高。</p> <p>010: 匹配时设置通道 1 为无效电平。当计数器TIM1X*_CNT 的值与捕获 /比较寄存器 1(TIM1X*_CCR1)相同时, 强制OC1REF 为低。</p> <p>011: 翻转。当TIM1X*_CCR1=TIM1X*_CNT 时, 翻转OC1REF 的电平。</p> <p>100: 强制为无效电平。强制 OC1REF 为低。</p> <p>101: 强制为有效电平。强制 OC1REF 为高。</p> <p>110: PWM 模式 1— 在向上计数时, 一旦 TIM1X*_CNT&lt;TIM1X*_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 TIM1X*_CNT&gt;TIM1X*_CCR1 时通道 1 为无效电平 (OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111 : PWM 模式 2 — 在向上计数时, 一旦 TIM1X*_CNT&lt;TIM1*_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 TIM1X*_CNT&gt;TIM1X*_CCR1 时通道 1 为有效电平, 否则为无效电平。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIM1X*_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。</p> <p>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。</p>	0x0	R/W



位	标记	功能描述	复位值	读写
3	OC1PE	<p>输出比较 1 预装载使能(Output Compare 1 preload enable)</p> <p>0: 禁止 TIM1X*_CCR1 寄存器的预装载功能, 可随时写入 TIM1X*_CCR1 寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启 TIM1X*_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1X*_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIM1X*_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。</p> <p>注 2: 仅在单脉冲模式下(TIM1X*_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。</p>	0	R/W
2	OC1FE	<p>输出比较 1 快速使能(Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>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。</p> <p>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。</p> <p>OCFE 只在信道被配置成 PWM1 或 PWM2 模式时起作用。</p>	0	R/W
1:0	CC1S[1:0]	<p>捕获/比较 1 选择。(Capture/Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 信道被配置为输出;</p> <p>01: CC1 信道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 信道被配置为输入, IC1 映射在 TI2 上;</p> <p>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 TIM1X*_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC1S 仅在通道关闭时(TIM1X*_CCER 寄存器的 CC1E=0)才是可写的。</p>	0x0	R/W



输入捕获模式:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:12	IC2F[3:0]	输入捕获 2 滤波器(Input capture 2 filter)	0x0	R/W
11:10	IC2PSC[1:0]	输入/捕获 2 预分频器(Input capture 2 prescaler)	0x0	R/W
9:8	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TIM1X*_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM1X*_CCER 寄存器的CC2E=0)才是可写的。	0x0	R/W
7:4	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: 0000: 无滤波器, 以fDTS 采样 1000: 采样频率fSAMPLING=fDTS/8, N=6 0001: 采样频率fSAMPLING=fCK_INT, N=2 1001: 采样频率fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010: 采样频率 fSAMPLING=fDTS/16, N=5 0011: 采样频率 fSAMPLING=fCK_INT, N=8 1011: 采样频率fSAMPLING=fDTS/16, N=6 0100: 采样频率 fSAMPLING=fDTS/2, N=6 1100: 采样频率 fSAMPLING=fDTS/16, N=8 0101: 采样频率fSAMPLING= fDTS /2, N=8 1101: 采样频率fSAMPLING=fDTS/32, N=5 0110: 采样频率fSAMPLING= fDTS /4, N=6 1110: 采样频率fSAMPLING=fDTS/32, N=6 0111: 采样频率fSAMPLING= fDTS /4, N=8 1111: 采样频率fSAMPLING=fDTS/32, N=8	0x0	R/W



3:2	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(TIM1X*_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。	0x0	R/W
1:0	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TIM1X*_SMCR 寄存器的 TS 位选择)。 注：CC1S 仅在通道关闭时(TIM1X*_CCER 寄存器的CC1E=0)才是可写的。	0x0	R/W



### 13.5.8 TIM1X\* 捕获/比较模式寄存器 2(TIM1X\*\_CCMR2)

地址偏移: 0x1C 复位值: 0x0000 0000

参看以上 CCMR1 寄存器的描述。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]	OC4PE	OC4FE	CC4S[1:0]	OC3CE	OC3M[2:0]	OC3PE	OC3FE	IC3F[3:0]	IC3PSC[1:0]	CC3S[1:0]				
IC4F[3:0]	IC4PSC[1:0]				IC3F[3:0]	IC3PSC[1:0]									
R/W															

#### ● 输出比较模式:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15	OC4CE	输出比较 4 清 0 使能(Output Compare 4 clear enable)	0	R/W
14:12	OC4M[2:0]	输出比较 4 模式(Output Compare 4 mode)	0x0	R/W
11	OC4PE	输出比较 4 预装载使能(Output Compare 4 preload enable)	0	R/W
10	OC4FE	输出比较 4 快速使能(Output Compare 4 fast enable)	0	R/W
9:8	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 TIM1X*_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM1X*_CCER 寄存器的CC4E=0) 才是可写的。	0x0	R/W
7	OC3CE	输出比较 3 清' 0' 使能(Output Compare 3 clear enable)	0	R/W
6:4	OC3M[2:0]	输出比较 3 模式(Output Compare 3 mode)	0x0	R/W
3	OC3PE	输出比较 3 预装载使能(Output Compare 3 preload enable)	0	R/W



2	OC3FE	输出比较 3 快速使能(Output Compare 3 fast enable)	0	R/W
1:0	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 TIM1X*_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM1X*_CCER 寄存器的 CC3E=0)才是可写的。	0x0	R/W

● 输入捕获模式:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:12	IC4F[3:0]	输入捕获 4 滤波器(Input capture 4 filter)	0x0	R/W
11:10	IC4PSC[1:0]	输入/捕获 4 预分频器(Input capture 4 prescaler)	0x0	R/W
9:8	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 TIM1X*_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM1X*_CCER 寄存器的 CC4E=0)才是可写的。	0x0	R/W
7:4	IC3F[3:0]	输入捕获 3 滤波器(Input capture 3 filter)	0x0	R/W
3:2	IC3PSC[1:0]	输入/捕获 3 预分频器(Input capture 3 prescaler)	0x0	R/W



1:0	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TIM1X*_SMCR 寄存器的 TS 位选择)。注: CC3S 仅在通道关闭时(TIM1X*_CCER 寄存 器的CC3E=0)才是可 写的。	0x0	R/W
-----	-----------	--	-----	-----

ZBT CONFIDENTIAL



### 13.5.9 TIM1X\* 捕获/比较使能寄存器(TIM1X\*\_CCER)

偏移地址: 0x20 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	--	CC4	CC4	CC3	CC3	CC3	CC3	CC2	CC2	CC2	CC2	CC1	CC1	CC1	CC1
--	--	P	E	NP	NE	P	E	NP	NE	P	E	NP	NE	P	E

位	标记	功能描述	复位值	读写
31:14	-	保留, 始终读为 0。	0	-
13	CC4P	输入/捕获4 输出极性(Capture/Compare 4 output polarity) 参考CC1P 的描述。	0	R/W
12	CC4E	输入/捕获4 输出使能(Capture/Compare 4 output enable) 参考CC1E 的描述。	0	R/W
11	CC3NP	输入/捕获 3 互补输出极性(Capture/Compare 3 complementary output polarity)参考CC1NP 的描述。	0	R/W
10	CC3NE	输入/捕获 3 互补输出使能(Capture/Compare 3 complementary outputenable) 参考CC1NE 的描述。	0	R/W
9	CC3P	输入/捕获3 输出极性(Capture/Compare 3 output polarity) 参考CC1P 的描述。	0	R/W
8	CC3E	输入/捕获3 输出使能(Capture/Compare 3 output enable) 参考CC1E 的描述。	0	R/W
7	CC2NP	输入/捕获 2 互补输出极性(Capture/Compare 2 complementary output polarity)参考CC1NP 的描述。	0	R/W
6	CC2NE	输入/捕获 2 互补输出使能(Capture/Compare 2 complementary outputenable) 参考CC1NE 的描述。	0	R/W
5	CC2P	输入/捕获2 输出极性(Capture/Compare 2 output polarity) 参考CC1P 的描述。	0	R/W
4	CC2E	输入/捕获2 输出使能(Capture/Compare 2 output enable) 参考CC1E 的描述。	0	R/W



位	标记	功能描述	复位值	读写
3	CC1NP	<p>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(TIM1X*_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。</p>	0	R/W
2	CC1NE	<p>: 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。</p>	0	R/W
1	CC1P	<p>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(TIM1X*_BDTR 寄存器中的LOCK 位)设为 3 或 2, 则该位不能被修改。</p>	0	R/W
0	CC1E	<p>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TIM1X*_CCR1 寄存器。 0: 捕获禁止; 1: 捕获使能。</p>	0	R/W



Table 13-4 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	
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=1
		1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
0	X	0	0	0	输出禁止(与定时器断开)	
		0	0	1	异步地: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0;	
		0	1	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。	
		1	1	1	关闭状态(输出使能且为无效电平)	
		0	0	0	异步地: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1;	
		0	1	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	0		
		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。



### 13.5-10 TIM1X\* 计数器(TIM1X\*\_CNT)

偏移地址: 0x24 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:0	CNT[15:0]	计数器的值(Counter value)	0x0	R/W

### 13.5-11 TIM1X\* 预分频器(TIM1X\*\_PSC)

偏移地址: 0x28 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:0	PSC[15:0]	预分频器的值(Prescaler value) 计数器的时钟频率(CK_CNT)等于 fCK_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'。	0x0	R/W



### 13.5.12 TIM1X\* 自动重装载寄存器(TIM1X\*\_ARR)

偏移地址:0x2C 复位值:0x0000 0000

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

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:0	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 12.3.1 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。	0	R/W

### 13.5.13 TIM1X\* 重复计数寄存器(TIM1X\*\_RCR)

偏移地址: 0x30 复位值: 0x0000 0000



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

位	标记	功能描述	复位值	读写
31:8	-	保留, 始终读为 0。	0	-
7:0	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 TIM1_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;	0	R/W

ZBT CONN



### 13.5.14 TIM1X\* 捕获/比较寄存器 1(TIM1X\*\_CCR1)

偏移地址: 0x34 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。 捕获/比较通道 1 的值(Capture/Compare 1 value) 若 CC1 通道配置为输出: 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 TIM1_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。	0	-
15:0	CCR1[15:0]	当前捕获/比较寄存器参与同计数器 TIM1_CNT 的比较, 并在 OC1 端口上产生输出信号。 若 CC1 通道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。	0	R/W

ZBIT CO'



### 13.5-15 TIM1X\* 捕获/比较寄存器 2(TIM1X\*\_CCR2)

偏移地址: 0x38 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:0	CCR2[15:0]	捕获/比较通道 2 的值(Capture/Compare 2 value) 若 CC2 通道配置为输出: 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 TIM1_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 TIM1_CNT 的比较, 并在 OC2 端口上产生输出信号。 若 CC2 通道配置为输入: CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。	0	R/W

### 13.5-16 TIM1X\* 捕获/比较寄存器 3(TIM1X\*\_CCR3)

偏移地址: 0x3C 复位值: 0x0000 0000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:0	CCR3[15:0]	捕获/比较通道 3 的值(Capture/Compare 3 value) 若 CC3 通道配置为输出: 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 TIM1_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 TIM1_CNT 的比较, 并在 OC3 端口上产生输出信号。 若 CC3 通道配置为输入: CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。	0	R/W

### 13.5.17 TIM1X\* 捕获/比较寄存器 4(TIM1X\*\_CCR4)

偏移地址: 0x40 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:0	CCR4[15:0]	捕获/比较通道 4 的值(Capture/Compare 4 value) 若 CC4 通道配置为输出: 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 TIM1_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 TIM1_CNT 的比较, 并在 OC4 端口上产生输出信号。 若 CC4 通道配置为输入: CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。	0x0	R/W



### 13.5.18 TIM1X\* 刹车和死区寄存器(TIM1X\*\_BDTR)

偏移地址: 0x44 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]	DTG[7:0]								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W								

注: 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 TIM1\_BDTR 寄存器时对它们进行配置。

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15	MOE	主输出使能(Main output enable) 一旦刹车输入有效, 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 输出或强制为空闲状态; 1: 如果设置了相应的使能位(TIM1X*_CCER 寄存器的 CCxE、CCxNE 位), 则开启OC 和 OCN 输出。 有关OC/OCN 使能的细节, 参见 11.5.9 TIM 和 TIM1C 捕获/比较使能寄存器(TIM1X*_CCER)。	0	R/W
14	AOE	自动输出使能(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' (如果刹车输入无效)。 注: 一旦 LOCK 级别(TIM1X*_BDTR 寄存器中的 LOCK 位)设为' 1' , 则该位不能被修改。	0	R/W
13	BKP	刹车输入极性(Break polarity)0: 刹车输入低电平有效; 1: 刹车输入高电平有效。 注: 一旦 LOCK 级别(TIM1X*_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	0	R/W



位	标记	功能描述	复位值	读写
		能起作用。		
12	BKE	<p>刹车功能使能(Break enable) 0: 禁止刹车输入(BRK 及 CCS 时钟失效事件); 1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。 注: 当设置了 LOCK 级别 1 时(TIM1X*_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。</p>	0	R/W
11	OSSR	<p>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(13.5.9 节, TIM1,TIM1A 和 TIM1B 捕获/比较使能寄存器(TIM1X*_CCER)). 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(TIM1X*_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。</p>	0	R/W
10	OSSI	<p>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(13.5.9 节, TIM1,TIM1A 和 TIM1B 捕获/比较使能寄存器(TIM1X*_CCER)). 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(TIM1X*_BDTR 寄存器中的 LOCK 位)设</p>	0	R/W



位	标记	功能描述	复位值	读写
		为 2，则该位不能被修改。		
9:8	LOCK[1:0]	<p>锁定设置(Lock Configuration) 该位为防止软件错误而提供写保护。</p> <p>00: 锁定关闭，寄存器无写保护；</p> <p>01: 锁定级别 1，不能写入 TIM1X*_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIM1X*_CR2 寄存器的 OISx/OISxN 位；</p> <p>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 极性位(一旦相关通道通过 CCxS 位设为输出，CC 极性位是 TIM1X*_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位；</p> <p>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 TIM1X*_CCMRx 寄存器的 OCxM/OCxPE 位)；</p> <p>注：在系统复位后，只能写一次 LOCK 位，一旦写入 TIM1X*_BDTR 寄存器，则其内容冻结直至复位。</p>	0x0	R/W
7:0	DTG[7:0]	<p>死区发生器设置(Dead-time generator setup)</p> <p>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 =&gt; DT=DTG[7:0] × Tdtg， Tdtg = TDTS; DTG[7:5]=10x =&gt; DT=(64+DTG[5:0])× Tdtg, Tdtg = 2 × TDTS; DTG[7:5]=110 =&gt; DT=(32+DTG[4:0])× Tdtg, Tdtg = 8 × TDTS; DTG[7:5]=111 =&gt; DT=(32+DTG[4:0])× Tdtg, Tdtg = 16 × TDTS;</p> <p>例：若 TDTS = 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；</p>	0x0	R/W



位	标记	功能描述	复位值	读写
		64us 到 126us, 若步长时间为 2us; 注: 一旦 LOCK 级别(TIM1X*_BDTR 寄存器中的LOCK 位)设为 1、2 或 3, 则不能修改这些位。		

ZBT CONFIDENTIAL



### 13.5.19 TM1X\*DMA 控制寄存器(TIM1X\*\_DCR)

地址偏移: 0x48 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DBL[4:0]					保留				DBA[4:0]				
保留		rw	rw	rw	rw	rw	保留				rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:13	-	保留, 始终读为 0。	0x0	-
12:8	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。	0	R/W
7:5	保留	-	0x0	-



位	标记	功能描述	复位值	读写
4:0	DBA[4:0]	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: TIM1x_CR1, 00001: TIM1x_CR2, 00010: TIM1x_SMCR, .....	0x0	R/W

ZBT CONFIDENTIAL



### 13.5-20 TM1X\*连续模式的 DMA 地址(TIM1X\*\_DMAR)

地址偏移: 0x4C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
DMAB[31:16]														
rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw															

位	标记	功能描述	复位值	读写
31:0	DMAB[31:0]	DMAB[31:0]: DMA连续传送寄存器  对TIM1X*_DMAR寄存器的读或写  会导致对以下地址的寄存器的存取操作：  “TIM1X*_CR1 Address + (DBA + DMA index)x4”，其中：  “TIM1X*_CR1地址”是控制寄存器1的地址；  “DBA”是TIM1X*_DCR寄存器中定义的基地址；  “DMA指针”是由DMA自动控制的偏移量，它取决于 TIM1_DCR寄存器中定义的DBL。	0x0000	R/W



## 【14】通用定时器 TIM2X\*(TIM2/2A/2B/2C)

### 14.1 通用定时器 简介

通用定时器由一个通过可编程预分频器驱动的 16 位自动装载计数器构成。

它适用于多种场合，包括测量输入信号的脉冲长度(输入捕获)或者产生输出波形(输出比较和 PWM)。使用定时器预分频器和 RCC 时钟控制器预分频器，脉冲长度和波形周期可以在几个微秒到几个毫秒间调整。

每个定时器都是完全独立的，没有互相共享任何资源。它们可以一起同步操作。

### 14.2 通用定时器主要功能

通用 TIM2/2A/2B/2C 定时器功能包括：

- 16 位向上、向下、向上/向下自动装载计数器
- 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 1~65536 之间的任意数值
- 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
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断/DMA 请求：
  - 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
- 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

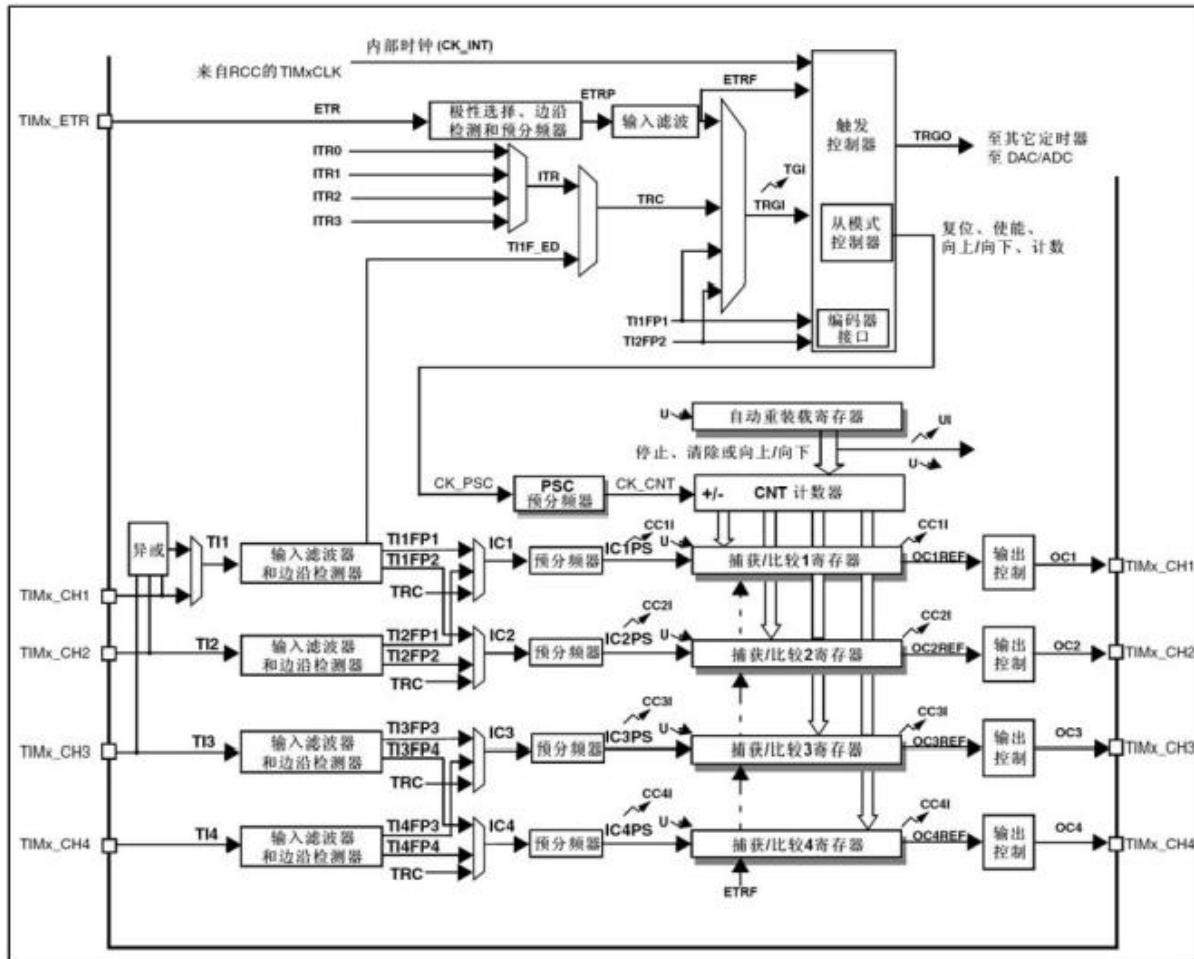


Figure 14-1 通用定时器框图

Note :



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

TIM2X\* : TIM2,TIM2A,TIM2B,TIM2C



### 14.3 TIM2X\* 功能描述

#### 14.3-1 时基单元

可编程通用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，在计数器运行时仍可以读写。时基单元包含：

- 计数器寄存器(TIM2X\*\_CNT)
- 预分频器寄存器(TIM2X\*\_PSC)
- 自动装载寄存器(TIM2X\*\_ARR)

自动装载寄存器是预先装载的，写或读自动重装载寄存器将访问预装载寄存器。根据在 **TIM2X\*\_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 **TIM2X\*\_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 **TIM2X\*\_CR1** 寄存器中的计数器使能位(**CEN**)时，**CK\_CNT** 才有效。(有关计数器使能的细节，请参见控制器的从模式描述)。

注：真正的计数器使能信号 **CNT\_EN** 是在 **CEN** 的一个时钟周期后被设置。

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 **TIM2X\*\_PSC** 寄

存器中的)16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器，它能够在工作时被改变。新的预分频器参数在下一次更新事件到来时被采用。

图 14-2 和图 14-3 给出了在预分频器运行时，更改计数器参数的例子。

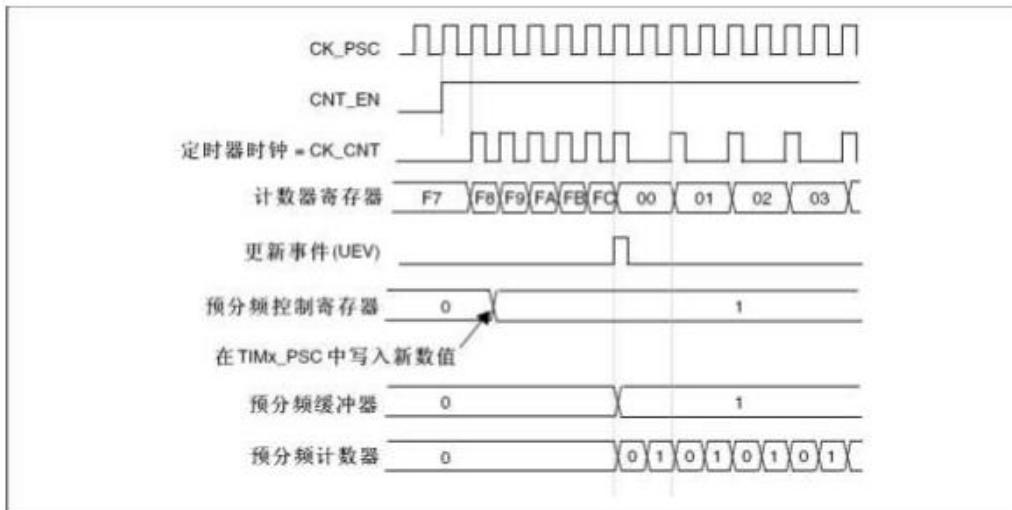


Figure 14-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

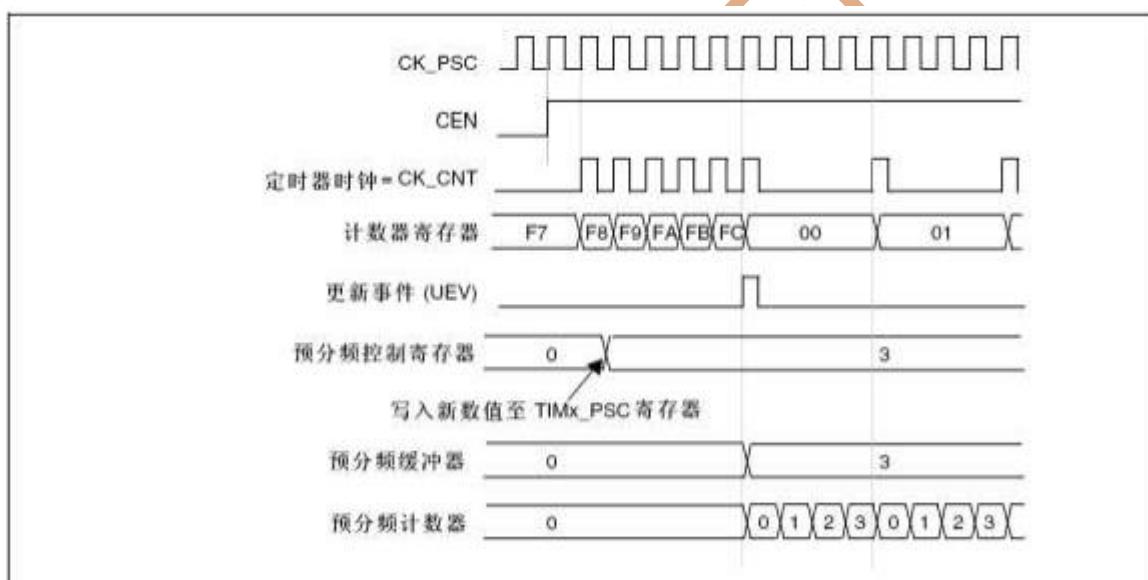


Figure 14-3 当预分频器的参数从 1 变到 4 时，计数器的时序图



### 14.3-2 计数器模式

#### 14.3-2.1 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( $\text{TIM2X*\_ARR}$  寄存器的内容)，然后重新从 0 开始计数并且产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件，在  $\text{TIM2X*\_EGR}$  寄存器中(通过软件方式或者使用从模式控制器)设置  $\text{UG}$  位也同样可以产生一个更新事件。

设置  $\text{TIM2X*\_CR1}$  寄存器中的  $\text{UDIS}$  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  $\text{UDIS}$  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(但预分频系数不变)。此外，如果设置了  $\text{TIM2X*\_CR1}$

寄存器中的  $\text{URS}$  位(选择更新请求)，设置  $\text{UG}$  位将产生一个更新事件  $\text{UEV}$ ，但硬件不设置  $\text{UIF}$  标志(即不产生中断或者  $\text{DMA}$  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  $\text{URS}$  位)设置更新标志位( $\text{TIM2X*\_SR}$  寄存器中的  $\text{UIF}$  位)。

- 预分频器的缓冲区被置入预装载寄存器的值( $\text{TIM2X*\_PSC}$  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( $\text{TIM2X*\_ARR}$ )。

下图给出一些例子，当  $\text{TIM2X*\_ARR}=0x36$  时计数器在不同时钟频率下的动作。

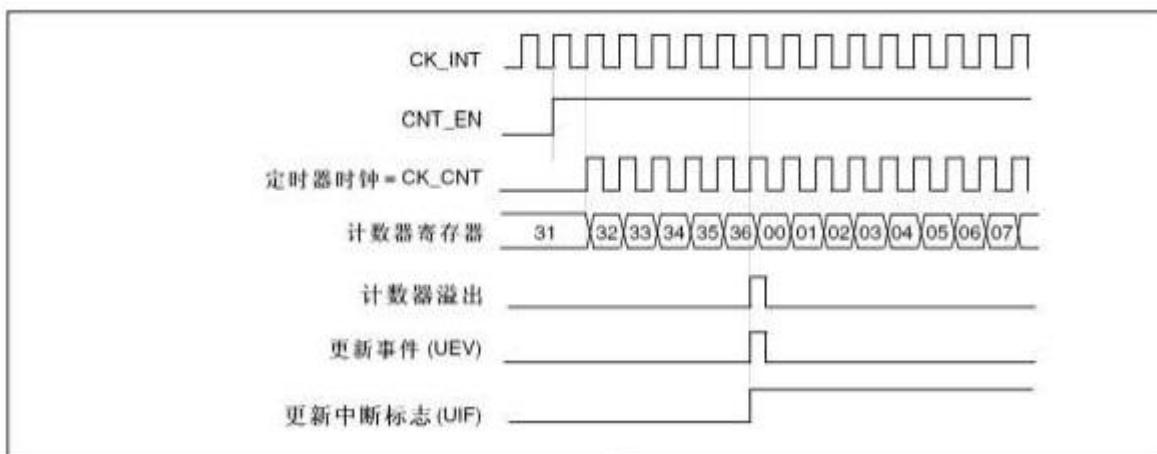


Figure 14-4 计数器时序图：内部时钟分频因子为 1

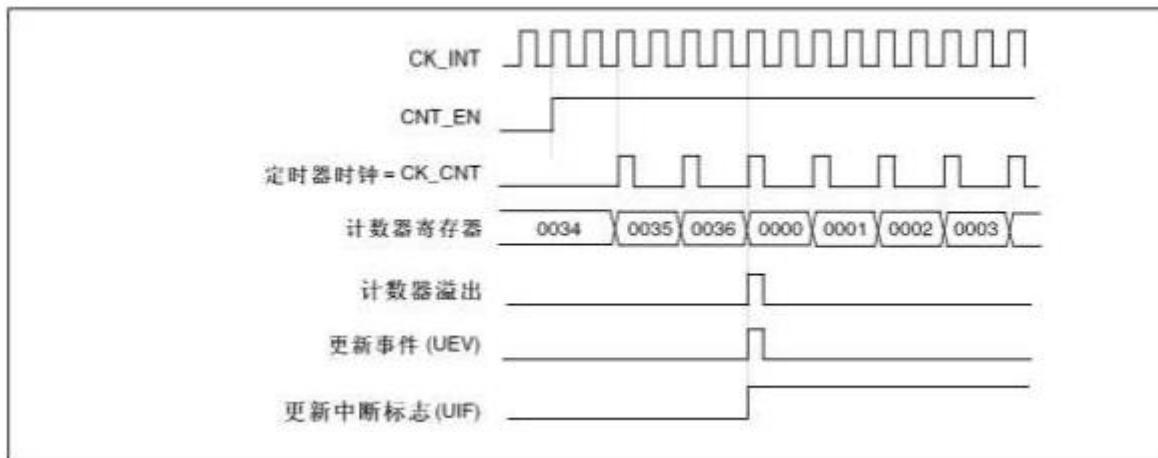


Figure 14-5 计数器时序图：内部时钟分频因子为 2

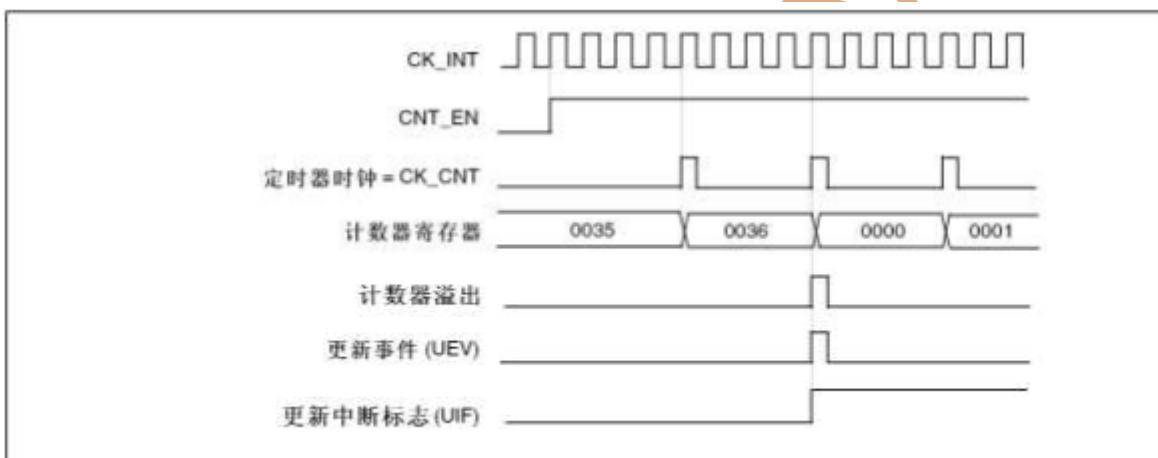


Figure 14-6 计数器时序图：内部时钟分频因子为 4

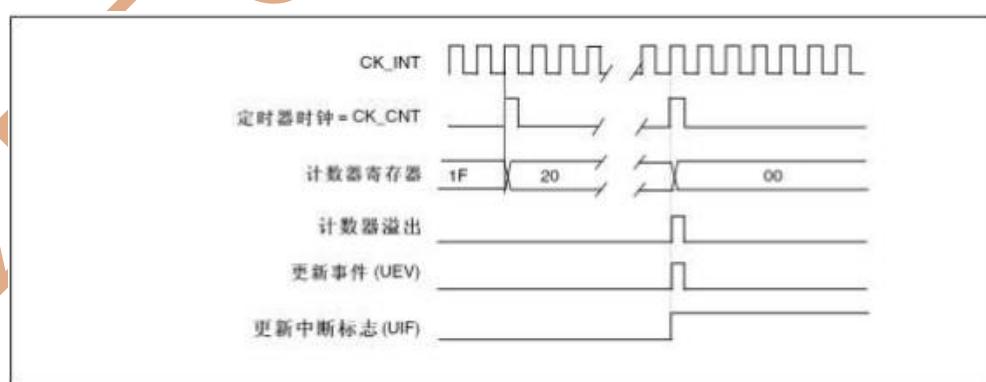


Figure 14-7 计数器时序图：内部时钟分频因子为 N

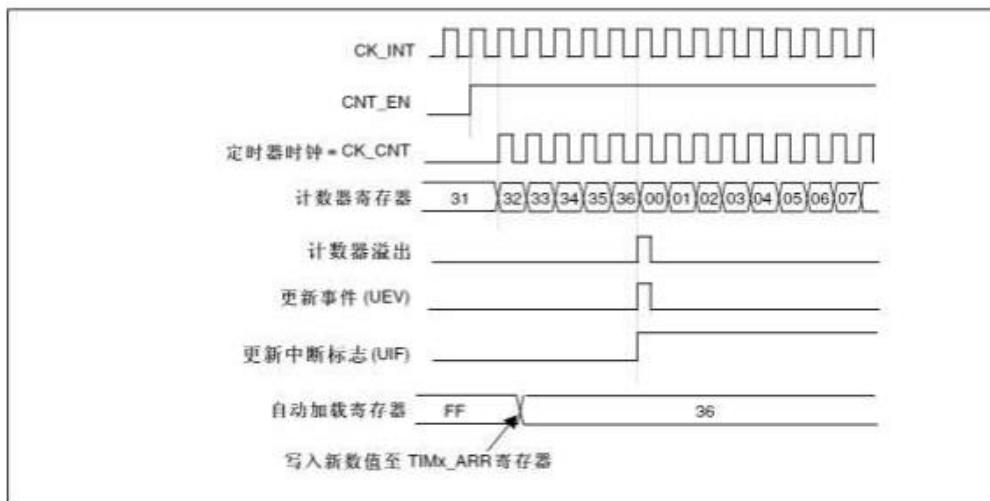


Figure 14-8 计数器时序图：当 ARPE=0 时的更新事件(TIM2X\*\_ARR 没有预装入)

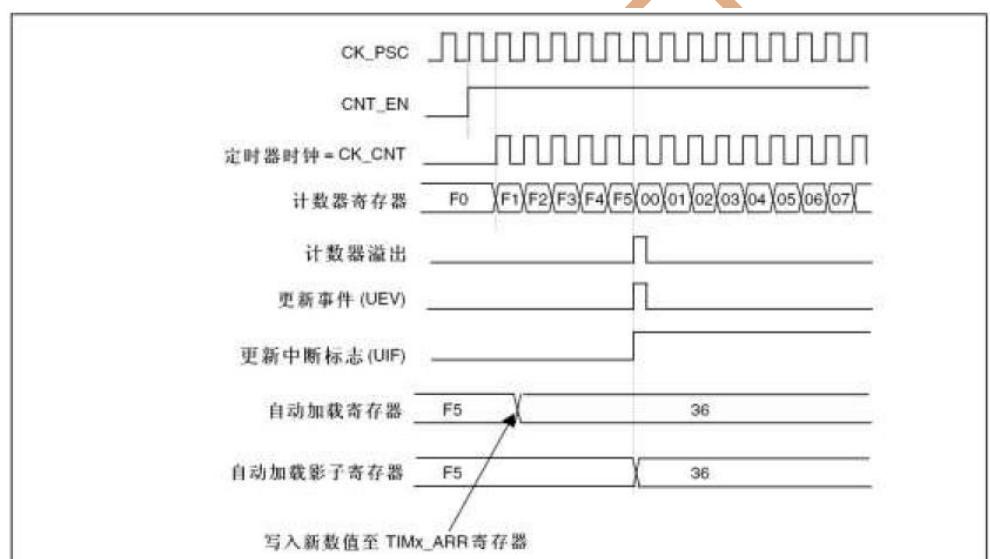


Figure 14-9 计数器时序图：当 ARPE=1 时的更新事件(预装入了 TIM2X\*\_ARR)



### 14.3-2.2 向下计数模式

在向下模式中，计数器从自动装入的值(TIM2X\*\_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。

每次计数器溢出时可以产生更新事件，在 TIM2X\*\_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 TIM2X\*\_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，同时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 TIM2X\*\_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(因此不产生中断或者 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(TIM2X\*\_SR 寄存器中的 UIF 位)也被设置。

- 预分频器的缓存器被置入预装载寄存器的值(TIM2X\*\_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值(TIM2X\*\_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 TIM2X\*\_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

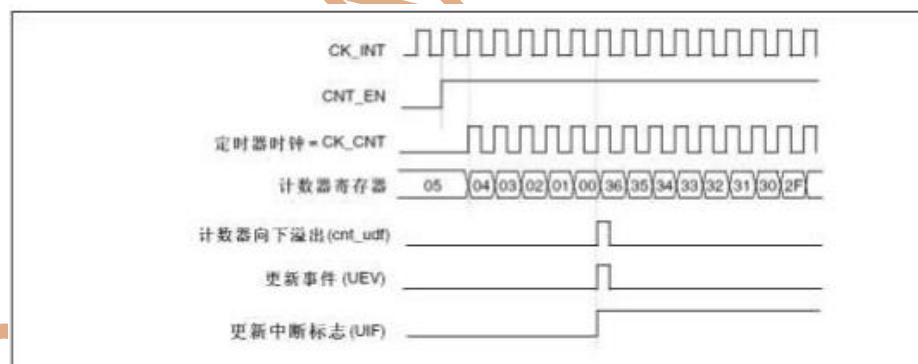


Figure 14-10 计数器时序图：内部时钟分频因子为 1

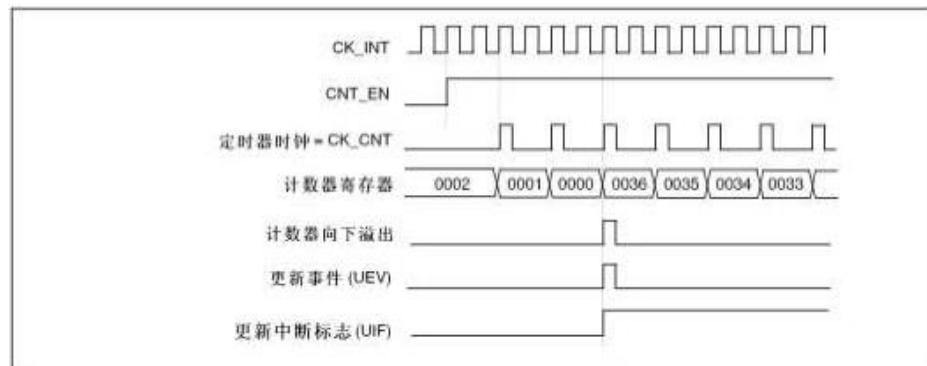


Figure 14-11 计数器时序图：内部时钟分频因子为 2

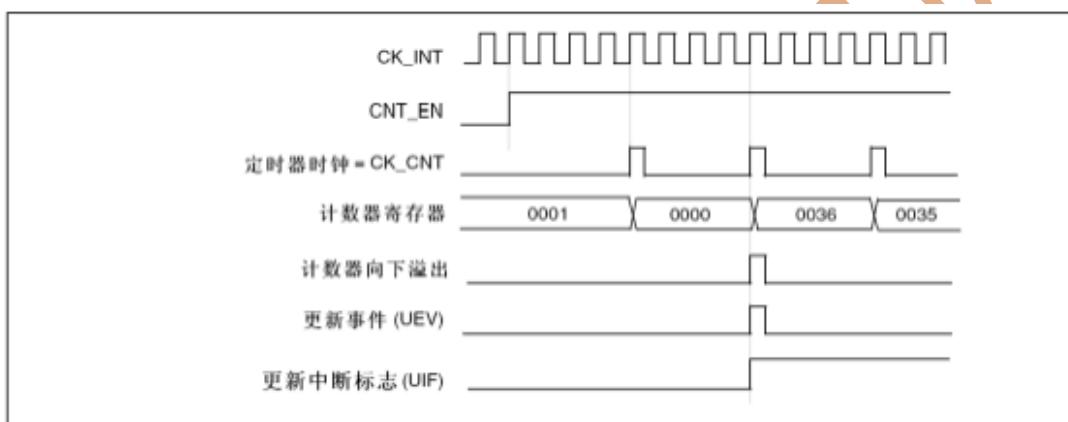


Figure 14-12 计数器时序图：内部时钟分频因子为 4

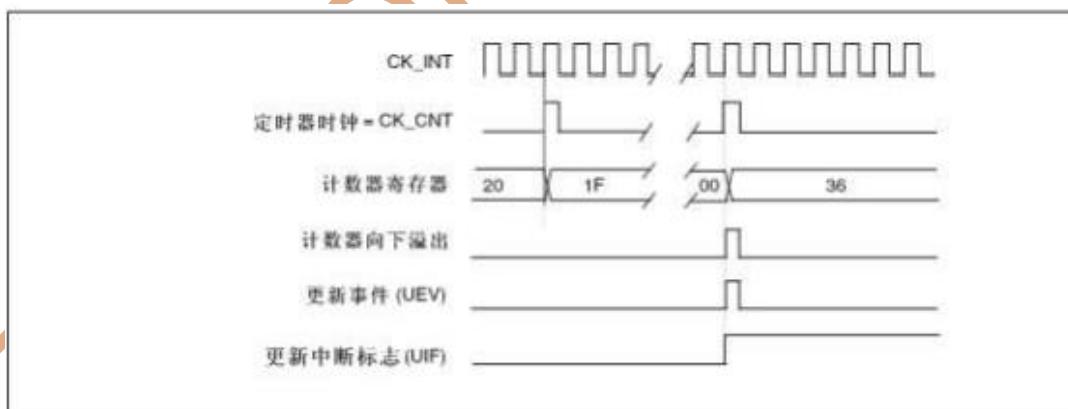


Figure 14-13 计数器时序图：内部时钟分频因子为 N

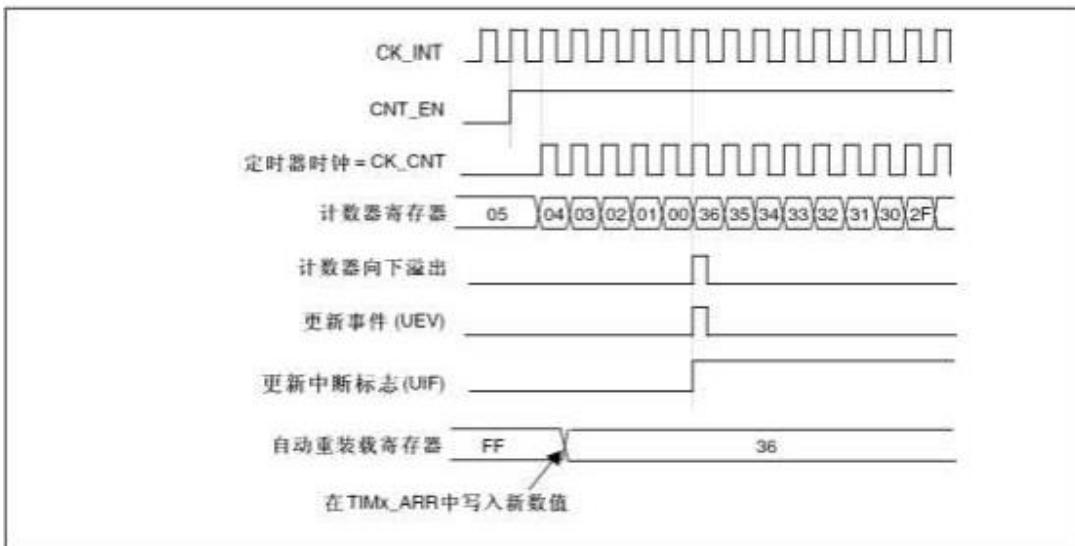


Figure 14-14 计数器时序图：当没有使用重复计数器时的更新事件

### 14.3-2.3 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(TIM2X\*\_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。

在这个模式，不能写入 TIM2X\*\_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 TIM2X\*\_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 TIM2X\*\_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 TIM2X\*\_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(因此不产生中断或者 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(TIM2X\*\_SR 寄存器中的 UIF 位)也被设置。

- 预分频器的缓存器被加载为预装载(TIM2X\*\_PSC 寄存器)的值。
- 当前的自动加载寄存器被更新为预装载值(TIM2X\*\_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:

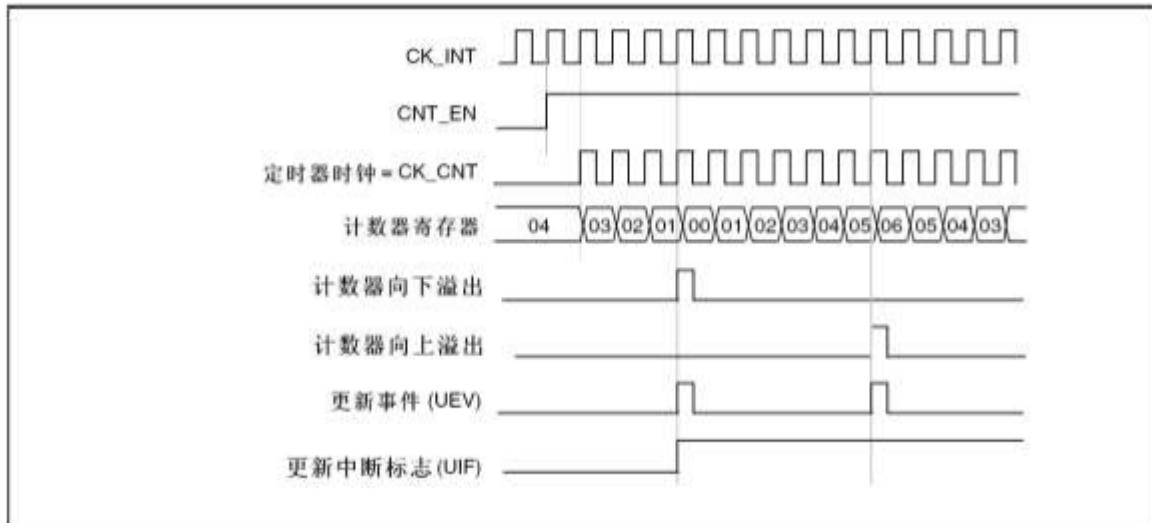


Figure 14-15 计数器时序图: 内部时钟分频因子为 1, TIM2X\*\_ARR=0x6

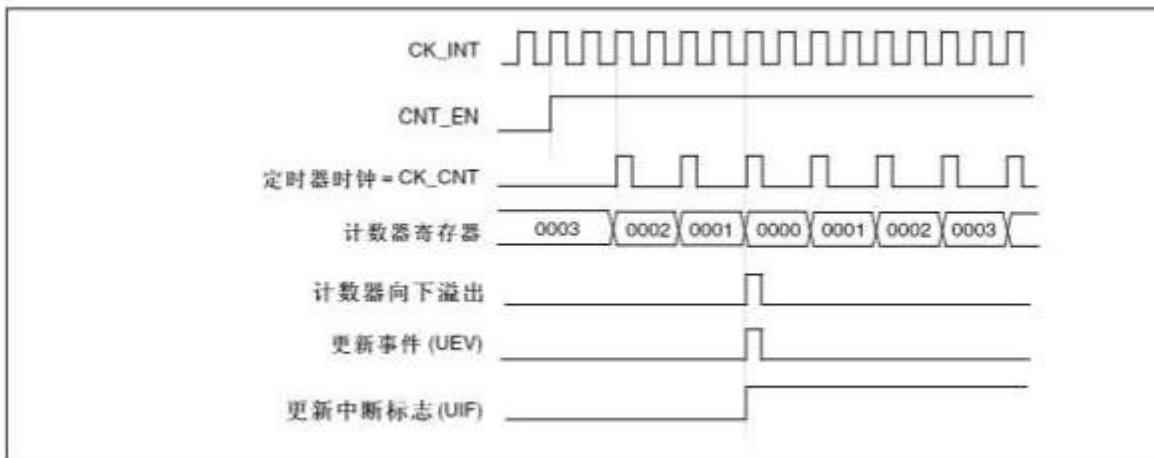


Figure 14-16 计数器时序图：内部时钟分频因子为 2

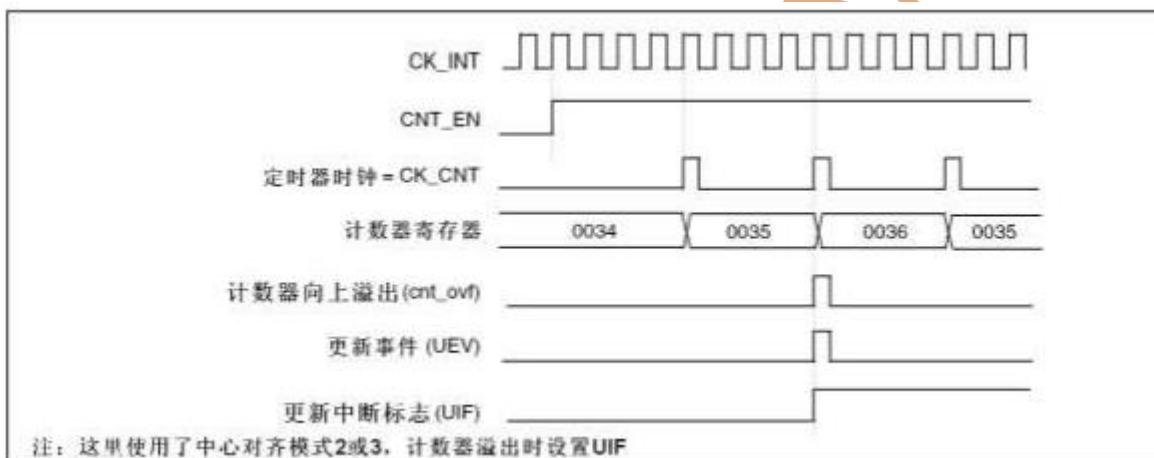


Figure 14-17 计数器时序图：内部时钟分频因子为 4, TIM2X\*\_ARR=0x36

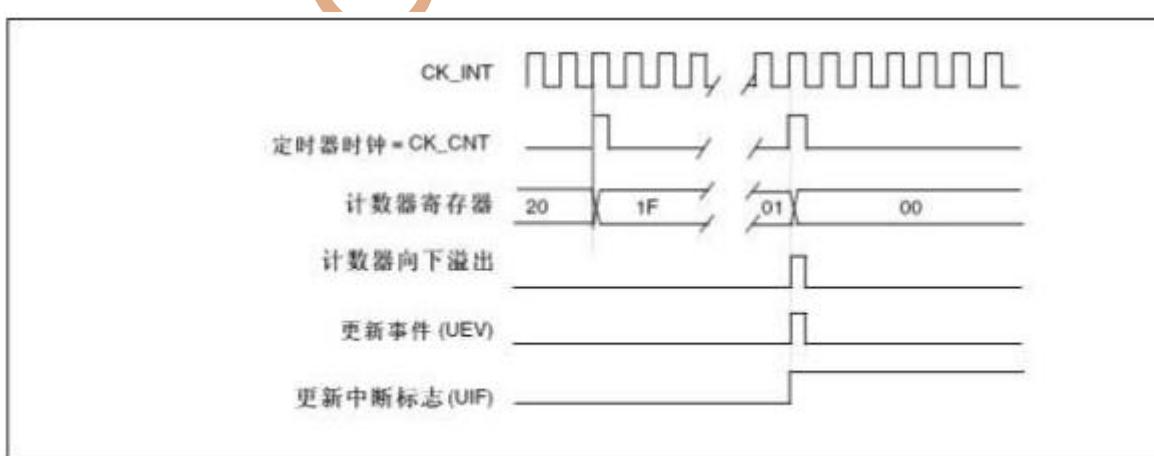


Figure 14-18 计数器时序图：内部时钟分频因子为 N

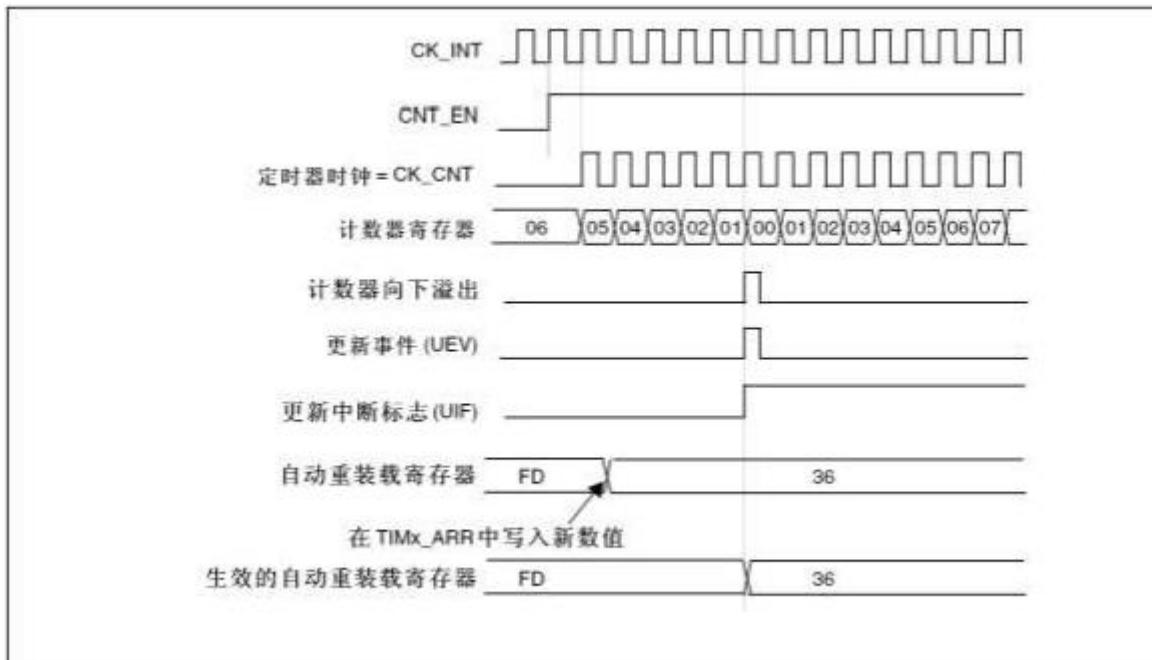


Figure 14-19 计数器时序图：ARPE=1 时的更新事件(计数器下溢)

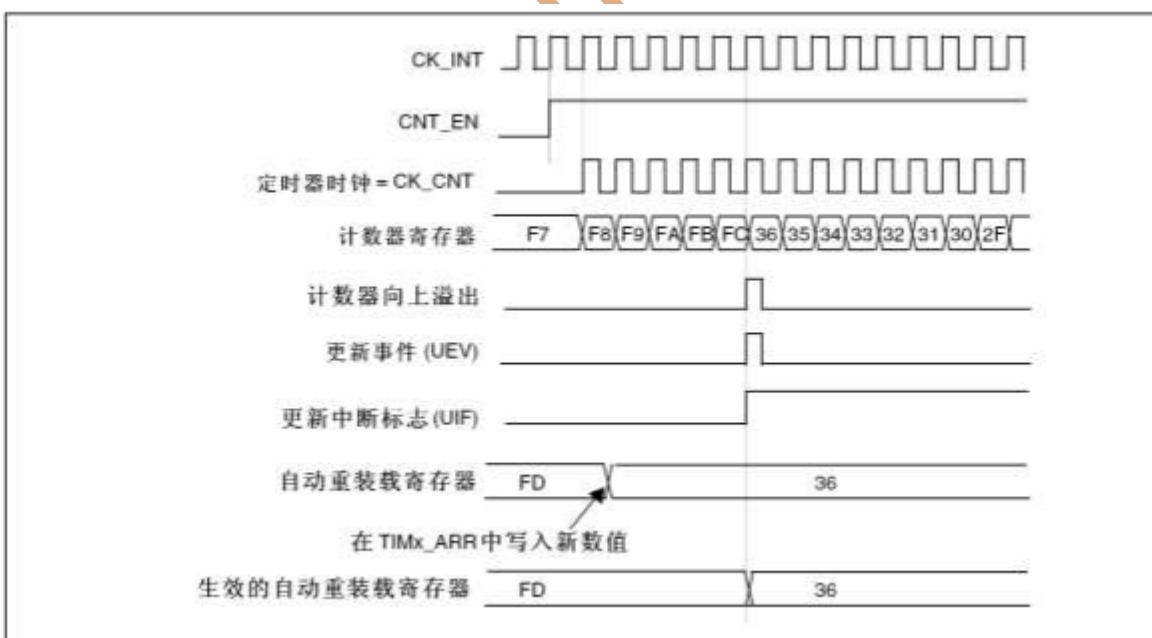


Figure 14-20 计数器时序图：ARPE=1 时的更新事件(计数器溢出)



### 14.3-3 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：外部输入脚(TIx)
- 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。

#### 14.3-3.1 内部时钟源(CK\_INT)

如果禁止了从模式控制器(TIM2X\*\_SMCR 寄存器的 SMS=000)，则 CEN、DIR(TIM2X\*\_CR1 寄存器)和

UG 位(TIM2X\*\_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。

下图显示了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。

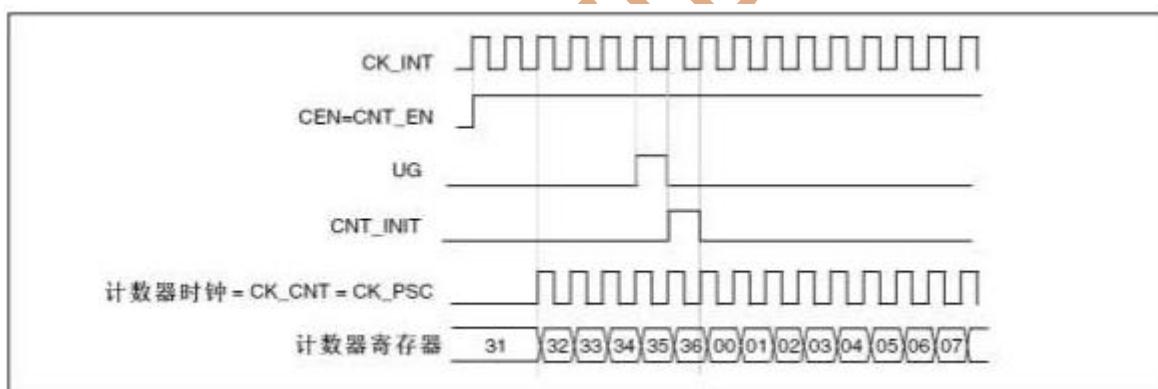


Figure 14-21 一般模式下的控制电路，内部时钟分频因子为 1



### 14.3-3.2 外部时钟源模式 1

当  $\text{TIM}2X^*\_\text{SMCR}$  寄存器的  $\text{SMS}=111$  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沿计数。

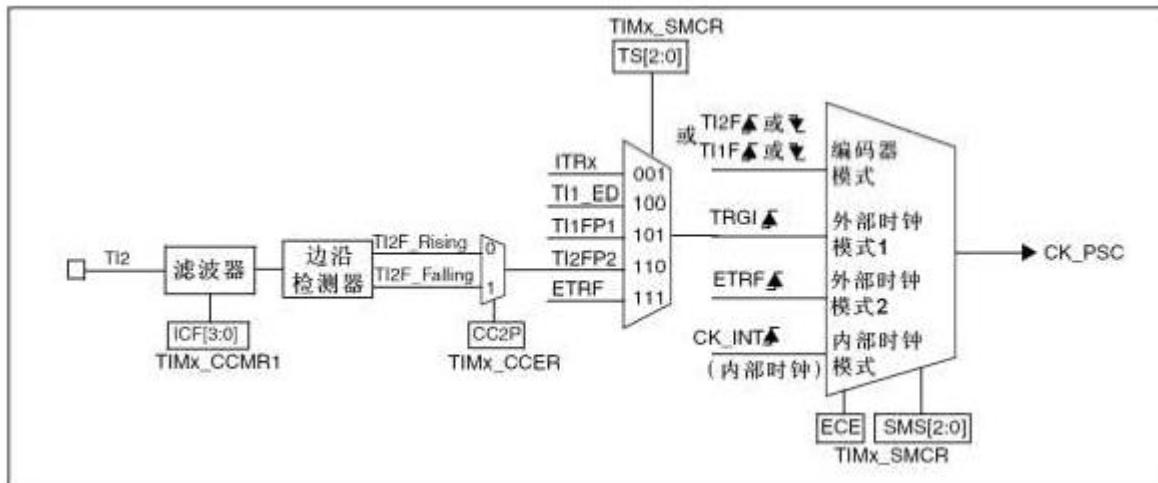


Figure 14-22 TI2 外部时钟连接例子

例如，要配置向上计数器在  $\text{T12}$  输入端的上升沿计数，使用下列步骤：

1. 配置  $\text{TIM}2X^*\_\text{CCMR1}$  寄存器  $\text{CC2S}= '01'$ ，配置信道 2 检测  $\text{TI2}$  输入的上升沿
2. 配置  $\text{TIM}2X^*\_\text{CCMR1}$  寄存器的  $\text{IC2F}[3:0]$ ，选择输入滤波器带宽(如果不需滤波器，保持  $\text{IC2F}=0000$ )

注：捕获预分频器不用作触发，所以不需要对它进行配置

3. 配置  $\text{TIM}2X^*\_\text{CCER}$  寄存器的  $\text{CC2P}= '0'$ ，选定上升沿极性
4. 配置  $\text{TIM}2X^*\_\text{SMCR}$  寄存器的  $\text{SMS}= '111'$ ，选择定时器外部时钟模式 1
5. 配置  $\text{TIM}2X^*\_\text{SMCR}$  寄存器中的  $\text{TS}= '110'$ ，选定  $\text{TI2}$  作为触发输入源
6. 设置  $\text{TIM}2X^*\_\text{CR1}$  寄存器的  $\text{CEN}= '1'$ ，启动计数器

当上升沿出现在  $\text{TI2}$ ，计数器计数一次，且  $\text{TIF}$  标志被设置。

在  $\text{TI2}$  的上升沿和计数器实际时钟之间的延时，取决于在  $\text{TI2}$  输入端的重新同步电路。

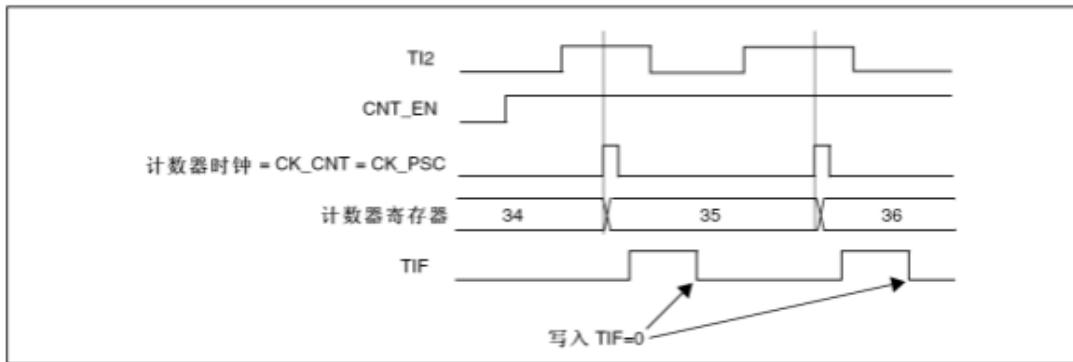


Figure 14-23 外部时钟模式 1 下的控制电路

#### 14.3-3.3 外部时钟源模式 2

选定此模式的方法为：令  $\text{TIM}2X^*\text{_SMCR}$  寄存器中的  $\text{ECE}=1$

计数器能够在外部触发  $\text{ETR}$  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：



Figure 14-24 外部触发输入框图

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

1. 本例中不需要滤波器，置  $\text{TIM}2X^*\text{_SMCR}$  寄存器中的  $\text{ETF}[3:0]=0000$
2. 设置预分频器，置  $\text{TIM}2X^*\text{_SMCR}$  寄存器中的  $\text{ETPS}[1:0]=01$
3. 设置在  $\text{ETR}$  的上升沿检测，置  $\text{TIM}2X^*\text{_SMCR}$  寄存器中的  $\text{ETP}=0$
4. 开启外部时钟模式 2，置  $\text{TIM}2X^*\text{_SMCR}$  寄存器中的  $\text{ECE}=1$
5. 启动计数器，置  $\text{TIM}2X^*\text{_CR1}$  寄存器中的  $\text{CEN}=1$

计数器在每 2 个  $\text{ETR}$  上升沿计数一次。

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

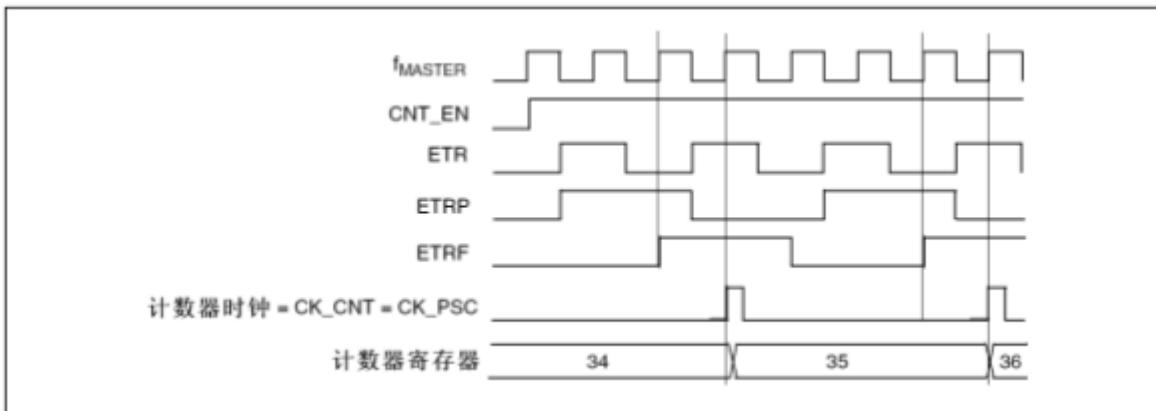


Figure 14-25 外部时钟模式 2 下的控制时序

#### 14.3-4 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)。

下面几张图是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘检测器产生一个信号(TIxFPx)，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(ICxPS)。

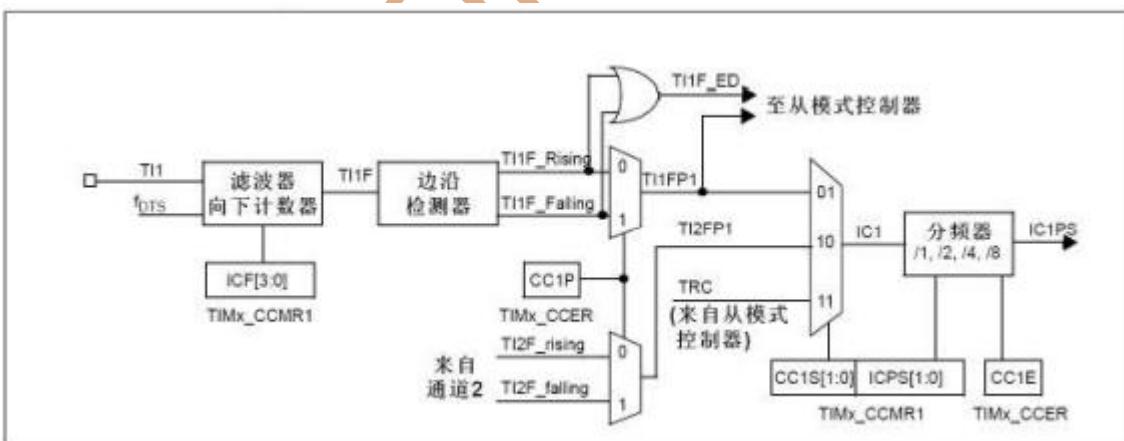


Figure 14-26 捕获/比较通道(如：通道 1 输入部分)

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

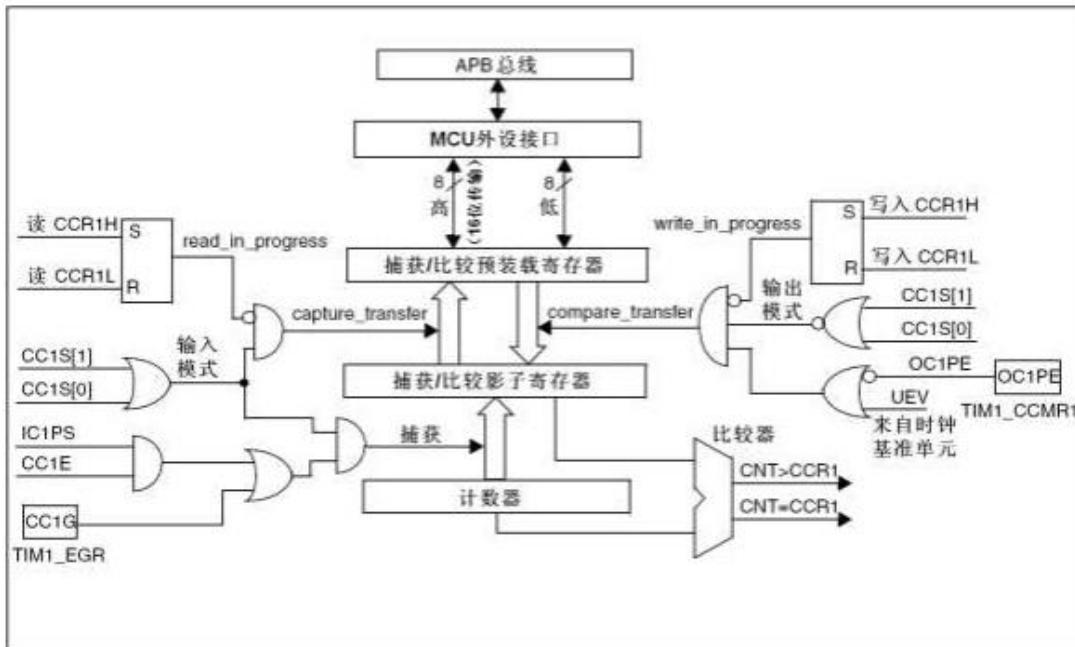


Figure 14-27 捕获/比较通道 1 的主电路

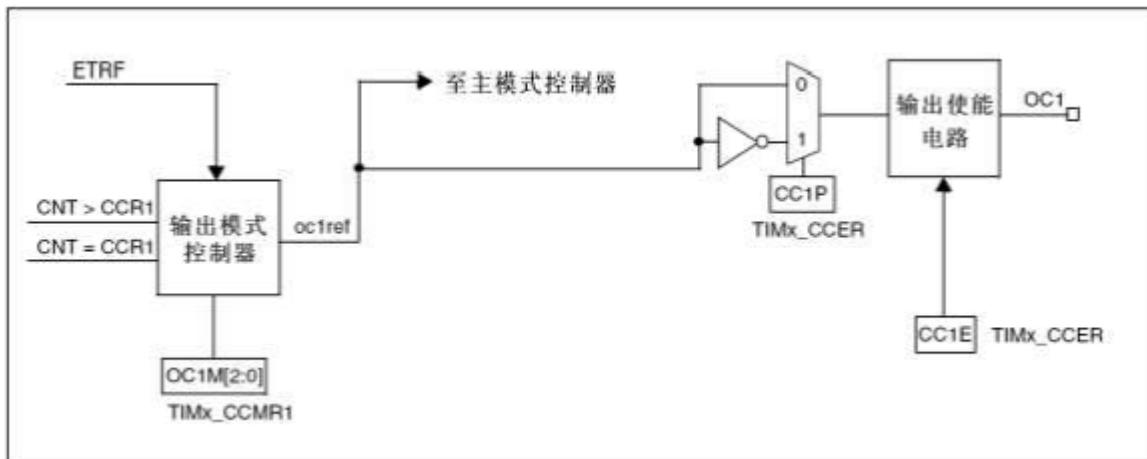


Figure 14-28 捕获/比较通道的输出部分(通道 1)

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。



### 14.3-5 输入捕获模式

在输入捕获模式下，当检测到 IC<sub>x</sub> 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 (TIM2X\*\_CCR<sub>x</sub>) 中。当捕获事件发生时，相应的 CC<sub>x</sub>IF 标志 (TIM2X\*\_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 操作。如果捕获事件发生时 CC<sub>x</sub>IF 标志已经为高，那么重复捕获标志 CC<sub>x</sub>OF (TIM2X\*\_SR 寄存器) 被置‘1’。写 CC<sub>x</sub>IF=0 可清除 CC<sub>x</sub>IF，或读取存储在 TIM2X\*\_CCR<sub>x</sub> 寄存器中的捕获数据也可清除 CC<sub>x</sub>IF。写 CC<sub>x</sub>OF=0 可清除 CC<sub>x</sub>OF。

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 TIM2X\*\_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIM2X\*\_CCR1 必须连接到 TI1 输入，所以写入 TIM2X\*\_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为‘00’，信道被配置为输入，并且 TM1X\*\_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<sub>x</sub> 时，输入滤波器控制位是 TIM2X\*\_CCMR<sub>x</sub> 寄存器中的 IC<sub>x</sub>F 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 TIM2X\*\_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIM2X\*\_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 TIM2X\*\_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIM2X\*\_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 TIM2X\*\_DIER 寄存器中的 CC1IE 位允许相关中断请求。  
通过设置 TIM<sub>x</sub>\_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 TIM2X\*\_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 TIM2X\*\_EGR 寄存器中相应的 CC<sub>x</sub>G 位，可以通过软件产生输入捕获中断和/或 DMA 请求。



### 14.3-6 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 IC<sub>x</sub> 信号被映射至同一个 TI<sub>x</sub> 输入。
  - 这 2 个 IC<sub>x</sub> 信号为边沿有效，但是极性相反。
  - 其中一个 TI<sub>x</sub>FP 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(TIM2X\*\_CCR1 寄存器)和占空比(TIM2X\*\_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 TIM2X\*\_CCR1 的有效输入：置 TIM2X\*\_CCMR1 寄存器的 CC1S=01(选择 TI1)。
  - 选择 TI1FP1 的有效极性(用来捕获数据到 TIM2X\*\_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。
  - 选择 TIM2X\*\_CCR2 的有效输入：置 TIM2X\*\_CCMR1 寄存器的 CC2S=10(选择 TI1)。
  - 选择 TI1FP2 的有效极性(捕获数据到 TIM2X\*\_CCR2)：置 CC2P=1(下降沿有效)。
  - 选择有效的触发输入信号：置 TIM2X\*\_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 TIM2X\*\_SMCR 中的 SMS=100。
  - 使能捕获：置 TIM2X\*\_CCER 寄存器中 CC1E=1 且 CC2E=1。

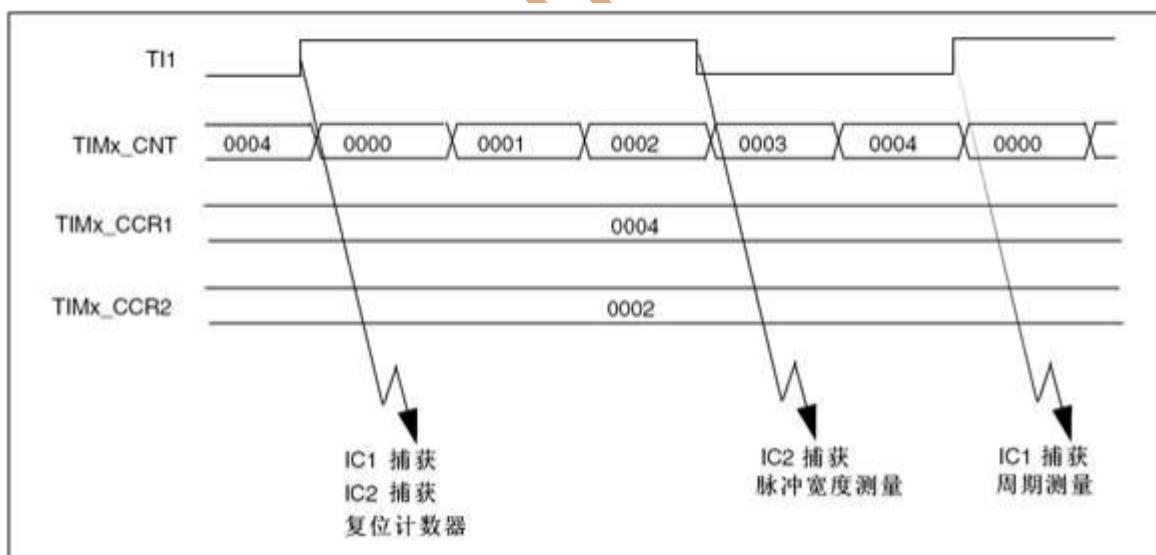


Figure 14-29 PWM 输入模式时序

由于只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIM2X\*\_CH1/TIM2X\*\_CH2 信号。



#### 14.3-7 强置输出模式

在输出模式(TIM2X\*\_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx)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 TIM2X\*\_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 极性位相反的值。

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

置 TIM2X\*\_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。该模式下，在 TIM2X\*\_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。

#### 14.3-8 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(TIM2X\*\_CCMRx 寄存器中的 OCxM 位)和输出极性(TIM2X\*\_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(TIM2X\*\_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIM2X\*\_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。

TIM2X\*\_CCMRx 中的 OCxPE 位选择 TIM2X\*\_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：

6. 选择计数器时钟(内部，外部，预分频器)
7. 将相应的数据写入 TIM2X\*\_ARR 和 TIM2X\*\_CCRx 寄存器中
8. 如果要产生一个中断请求和/或一个 DMA 请求，设置 CCxIE 位和/或 CCxDE 位。
9. 选择输出模式，例如当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚，CCRx 预装载未



用，开启 OCx 输出且高电平有效，则必须设置 OCxM= ‘011’、OCxPE= ‘0’、CCxP= ‘0’ 和 CCxE= ‘1’。

#### 10. 设置 TIM2X\*\_CR1 寄存器的 CEN 位启动计数器

TIM2X\*\_CCR<sub>x</sub> 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 TIM2X\*\_CCR<sub>x</sub> 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。

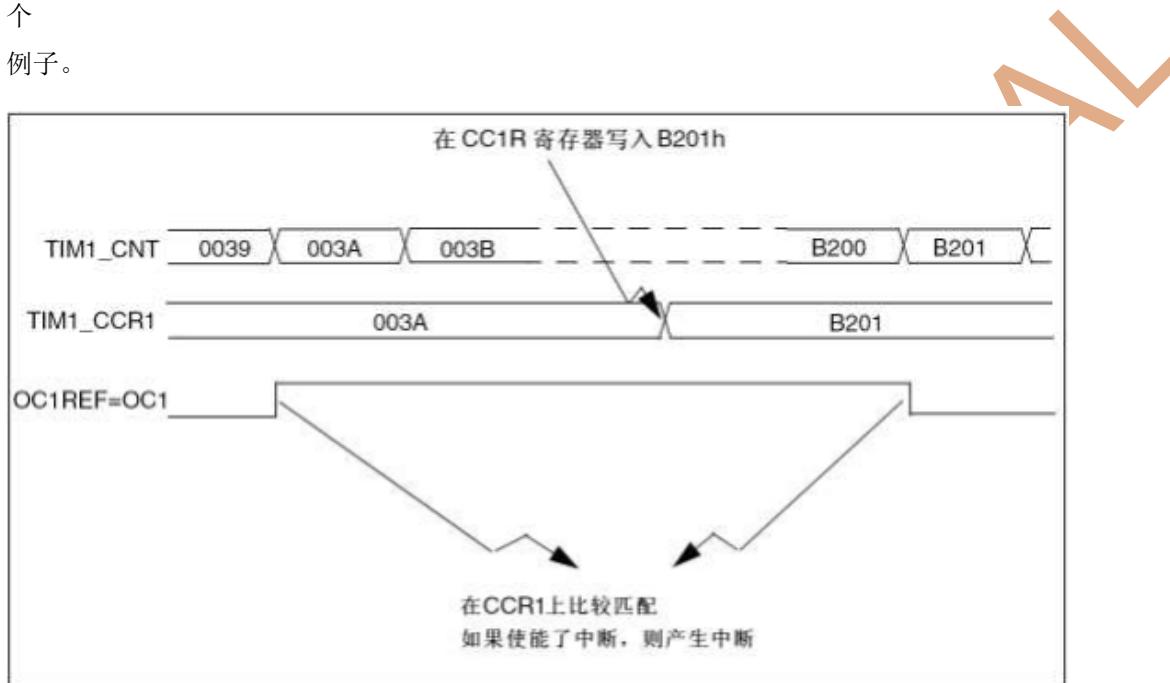


Figure 14-30 输出比较模式，翻转 OC1



### 14.3-9 PWM 模式

脉冲宽度调制模式可以产生一个由 `TIM2X*_ARR` 寄存器确定频率、由 `TIM2X*_CCRx` 寄存器确定占空比的信号。

在 `TIM2X*_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。必须设置 `TIM2X*_CCMRx` 寄存器 `OCxPE` 位以使能相应的预装载寄存器，最后还要设置 `TIM2X*_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 `TIM2X*_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 `TIM2X*_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。`TIM2X*_CCER` 寄存器中的 `CCxE` 位控制 `OCx` 输出使能。详见 `IM2_CCERx` 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下，`TIM2X*_CNT` 和 `TIM2X*_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 `TIM2X*_CCRx < TIM2X*_CNT` 或者 `TIM2X*_CNT < TIM2X*_CCRx`。然而为了与 `OCREF_CLR` 的功能(在下一个 PWM 周期之前，`ETR` 信号上的一个外部事件能够清除 `OCxREF`)一致，`OCxREF` 信号只能在下述条件下产生：

- 当比较的结果改变 或 当输出比较模式(`TIM2X*_CCMRx` 寄存器中的 `OCxM` 位)从“冻结”(无比较，`OCxM= '000'`)切换到某个 PWM 模式(`OCxM= '110'` 或' 111' )

这样在运行中可以通过软件强置 PWM 输出。

根据 `TIM2X*_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。

#### 14.3-9.1 PWM 边沿对齐模式

##### 14.3-9.1.1 向上计数配置

当 `TIM2X*_CR1` 寄存器中的 `DIR` 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 `TIM2X*_CNT < TIM2X*_CCRx` 时 PWM 信号参考 `OCxREF` 为高，否

则为低。如果 `TIM2X*_CCRx` 中的比较值大于自动重装载值(`TIM2X*_ARR`)，则 `OCxREF` 保持为' 1' 。如果

比较值为 0，则 `OCxREF` 保持为' 0' 。下图为 `TIM2X*_ARR=8` 时边沿对齐的 PWM 波形实例。

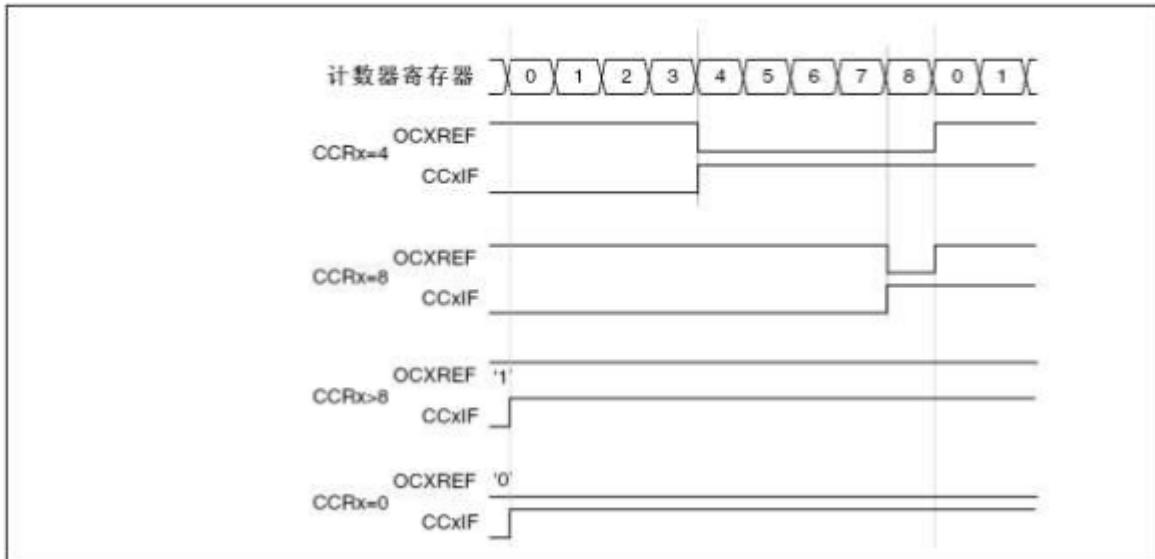


Figure 14-31 边沿对齐的 PWM 波形(ARR=8)

#### 14.3-9.1.2 向下计数的配置

当 TIM2X\*\_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 TIM2X\*\_CNT > TIM2X\*\_CCR<sub>x</sub> 时参考信号 OCxREF 为低，否则为高。如果 TIM2X\*\_CCR<sub>x</sub>

中的比较值大于 TIM2X\*\_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。

#### 14.3-9.2 PWM 中央对齐模式

当 TIM2X\*\_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'。TIM2X\*\_CR1 寄存器中的计数方向位 (DIR)

由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子

- TIM2X\*\_ARR=8
- PWM 模式 1
- TIM2X\*\_CR1 寄存器中的 CMS=01，在中央对齐模式 1 时，当计数器向下计数时设置比较标志。

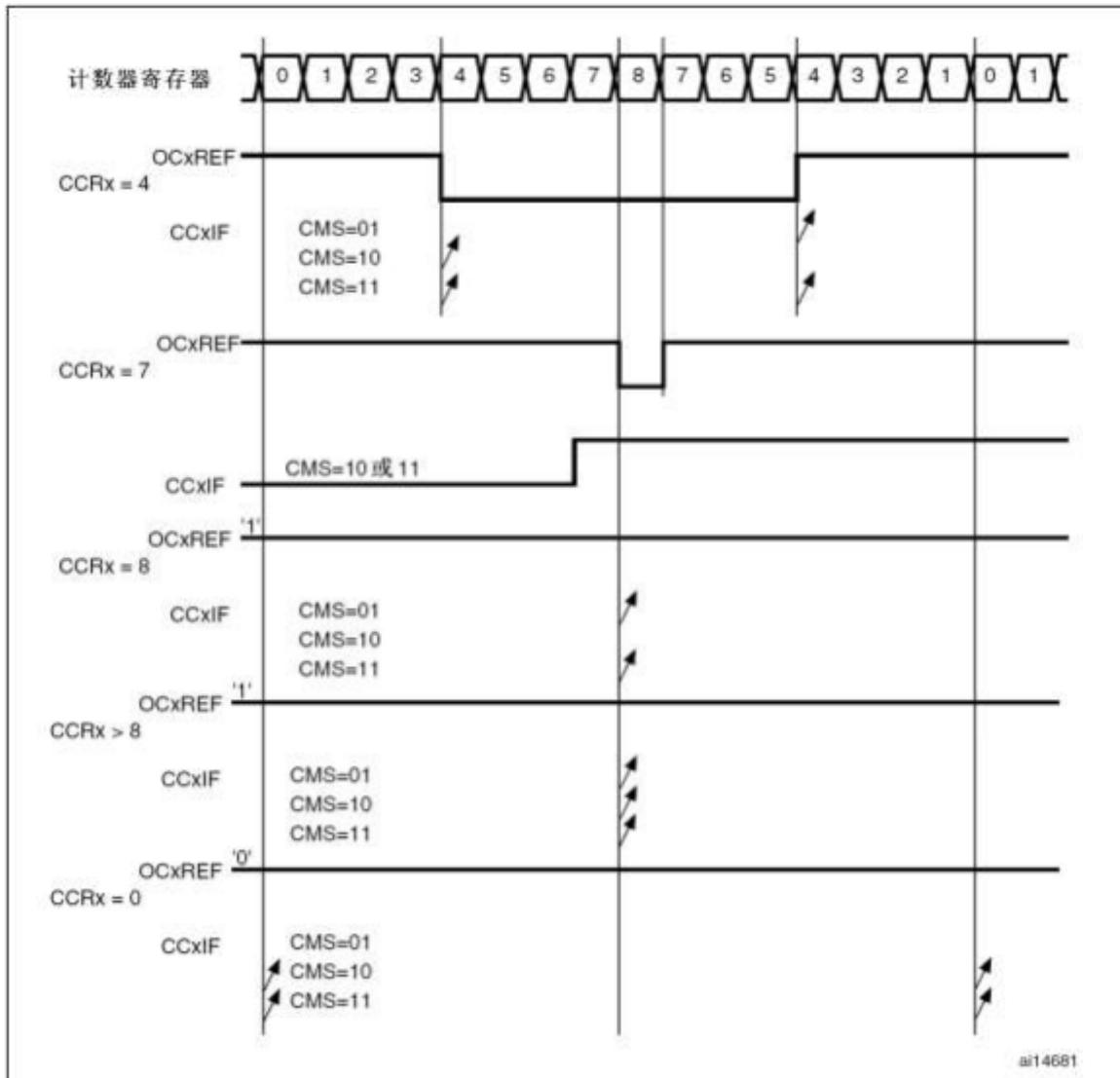


Figure 14-32 中央对齐的 PWM 波形(APR=8)

#### 14.3-9.2.1 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；这就意味着计数器向上还是向下计数取决于 `TIM2X*_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(`TIM2X*_CNT>TIM2X*_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 `TIM2X*_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 `TIM2X*_EGR` 位中的 UG 位)，不要在计数进行过程中修改计数器的值。



### 14.3-10 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，产生一个脉宽可编程控制的脉冲。

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 `TIM2X*_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： $CNT < CCRx \leq ARR$ (特别地， $0 < CCRx$ )，

向下计数方式： $CNT > CCRx$ 。

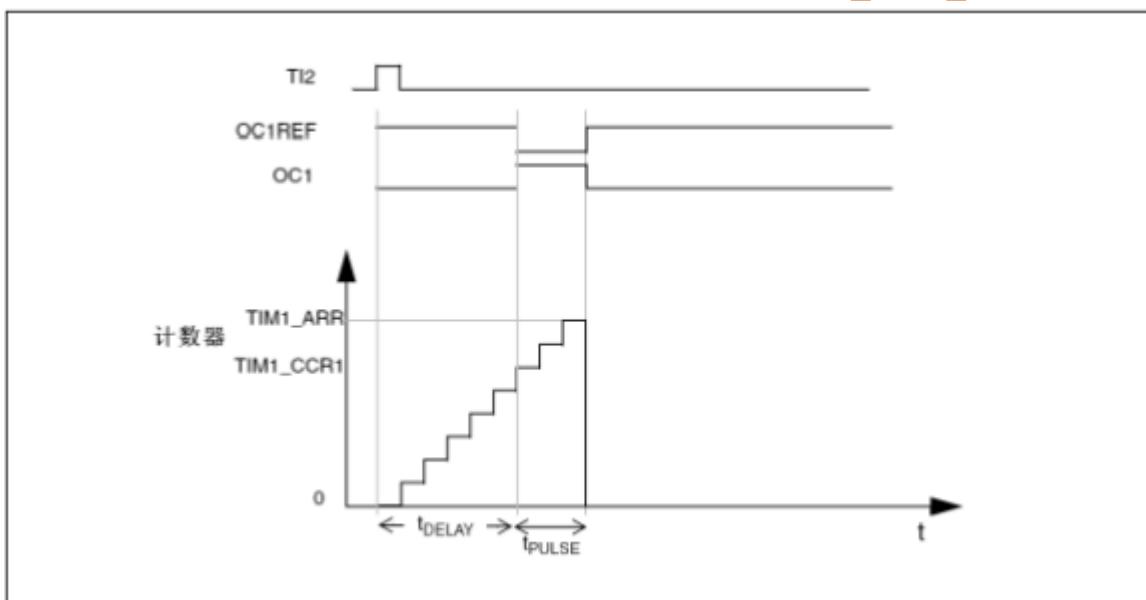


Figure 14-33 单脉冲模式的例子

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 `TIM2X*_CCMR1` 寄存器中的 `CC2S= '01'`，把 `TI2FP2` 映像到 `TI2`。
- 置 `TIM2X*_CCER` 寄存器中的 `CC2P= '0'`，使 `TI2FP2` 能够检测上升沿。
- 置 `TIM2X*_SMCR` 寄存器中的 `TS= '110'`，`TI2FP2` 作为从模式控制器的触发(`TRGI`)。
- 置 `TIM2X*_SMCR` 寄存器中的 `SMS= '110'` (触发模式)，`TI2FP2` 被用来启动计数器。

OPM 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)

- $t_{DELAY}$  由写入 `TIM2X*_CCR1` 寄存器中的值定义。



- tPULSE 由自动装载值和比较值之间的差值定义( $\text{TIM2X}^*\_\text{ARR} - \text{TIM2X}^*\_\text{CCR1}$ )。
- 假定当发生比较匹配时要产生从' 0' 到' 1' 的波形，当计数器到达预装载值时要产生一个从' 1' 到' 0' 的波形；首先要置  $\text{TIM2X}^*\_\text{CCMR1}$  寄存器的  $\text{OC1M} = '111'$ ，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置  $\text{TIM2X}^*\_\text{CCMR1}$  中的  $\text{OC1PE} = '1'$  和  $\text{TIM2X}^*\_\text{CR1}$  寄存器中的  $\text{ARPE}$ ；然后在  $\text{TIM2X}^*\_\text{CCR1}$  寄存器中填写比较值，在  $\text{TIM2X}^*\_\text{ARR}$  寄存器中填写自动装载值，修改 UG 位来产生一个更新事件，然后等待在  $\text{TI2}$  上的一个外部触发事件。本例中， $\text{CC1P} = '0'$ 。在这个例子中， $\text{TIM2X}^*\_\text{CR1}$  寄存器中的  $\text{DIR}$  和  $\text{CMS}$  位应该置低。因为只需一个脉冲，所以必须设置  $\text{TIM2X}^*\_\text{CR1}$  寄存器中的  $\text{OPM} = '1'$ ，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。  
**特殊情况：OCx 快速使能**  
在单脉冲模式下，在  $\text{TIx}$  输入脚的边沿检测逻辑设置  $\text{CEN}$  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  $t\text{DELAY}$ 。  
如果要以最小延时输出波形，可以设置  $\text{TIM2X}^*\_\text{CCMRx}$  寄存器中的  $\text{OCxFE}$  位；此时  $\text{OCxREF}$ (和  $\text{OCx}$ )被强制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。 $\text{OCxFE}$  只在通道配置为 PWM1 和 PWM2 模式时起作用。



### 14.3-11 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 `TIM2X*_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：

外部触发预分频器必须处于关闭：`TIM2X*_SMCR` 寄存器中的 `ETPS[1:0]= '00'`。

必须禁止外部时钟模式 2：`TIM2X*_SMCR` 寄存器中的 `ECE= '0'`。

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 `TIM2X*` 被置于 `PWM` 模式。

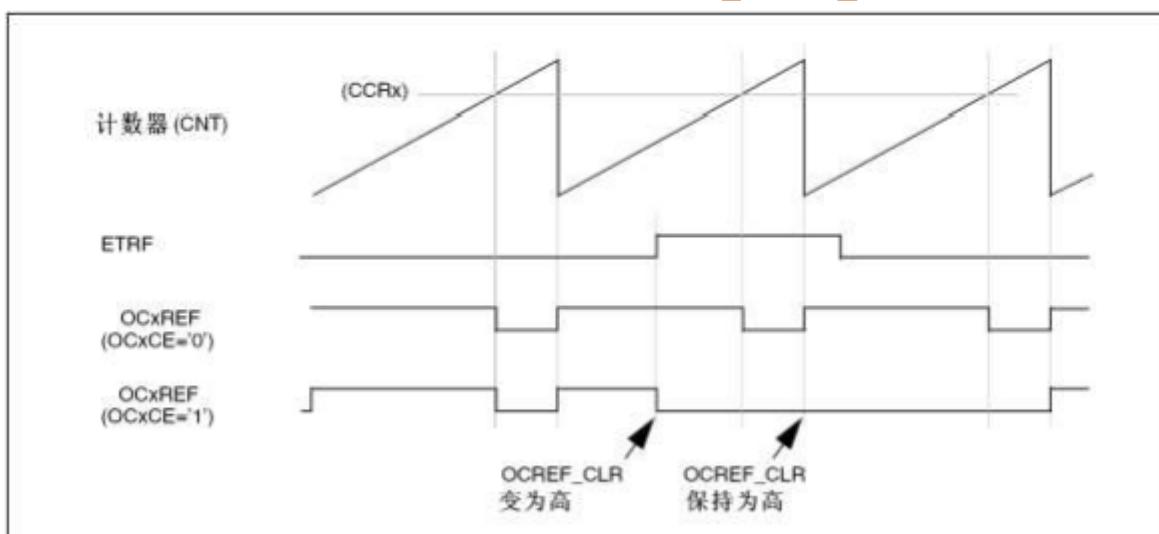


Figure 14-34 清除 TIM2X\* 的 OCxREF



### 14.3.12 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 TIM2X\*\_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 TIM2X\*\_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 14-1，假定计数器已经启动(TIM2X\*\_CR1 寄存器中的 CEN= '1' )，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 TIM2X\*\_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 TIM2X\*\_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 TIM2X\*\_ARR；同样，捕获器、比较器、预分频器、触发输出特性等仍工作如常。

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。

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

Table 14-1 计数方向与编码器信号的关系

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' (TIM2X\*\_CCMR1 寄存器, IC1FP1 映射到 TI1)
- CC2S= '01' (TIM2X\*\_CCMR2 寄存器, IC2FP2 映射到 TI2)
- CC1P= '0' (TIM2X\*\_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- CC2P= '0' (TIM2X\*\_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)
- SMS= '011' (TIM2X\*\_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效)
- CEN= '1' (TIM2X\*\_CR1 寄存器, 计数器使能)

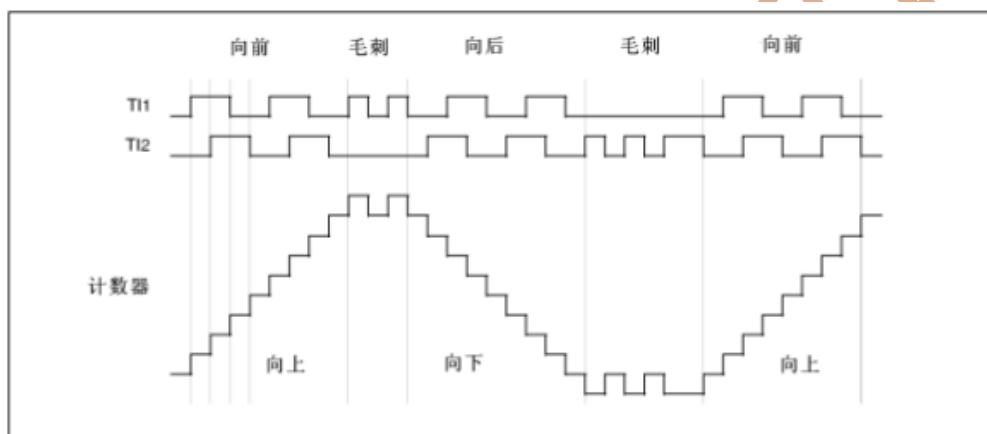


Figure 14-35 编码器模式下的计数器操作实例

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)

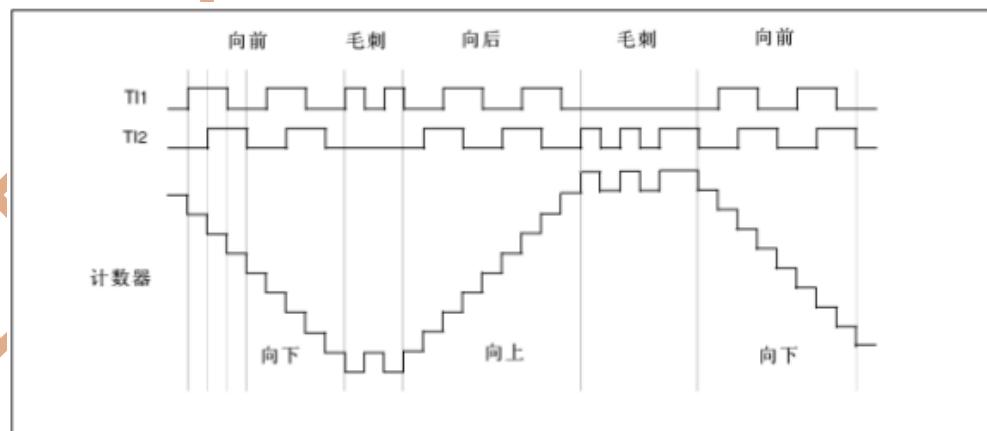


Figure 14-36 IC1FP1 反相的编码器接口模式实例

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)。它也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

#### 14.3-13 定时器输入异或功能

TIM2X\*\_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 TIM2X\*\_CH1、TIM2X\*\_CH2 和 TIM2X\*\_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。

#### 14.3-14 定时器和外部触发的同步

TIM2X\* 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。

##### 14.3-14.1 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 TIM2X\*\_CR1 寄存器的 URS 位为低，还会产生一个更新事件 UEV；然后所有的预装载寄存器(TIM2X\*\_ARR, TIM2X\*\_CCRx)都会被更新。

在下面的例子中，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，所以不需要配置它。CC1S 位只选择输入捕获源，即 TIM2X\*\_CCMR1 寄存器中 CC1S=01。置 TIM2X\*\_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- 置 TIM2X\*\_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM2X\*\_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM2X\*\_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(TIM2X\*\_SR 寄存器中的 TIF 位)被设置，根据 TIM2X\*\_DIER 寄存器中 TIE(中断使能)位，产生一个中断请求和 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 TIM2X\*\_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。

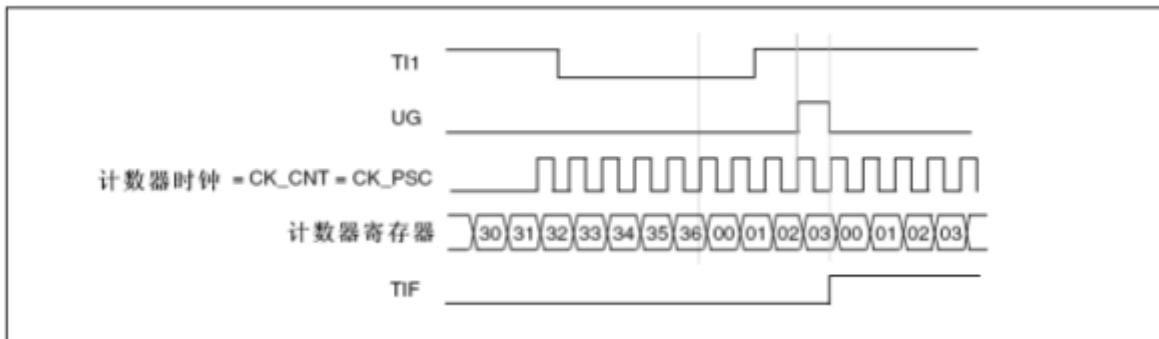


Figure 14-37 复位模式下的控制电路

#### 14.3-14.2 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 TIM2X\*\_CCMR1 寄存器中 CC1S=01。置 TIM2X\*\_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIM2X\*\_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM2X\*\_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM2X\*\_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，在 TI1 变高时停止计数。当计数器开始或停止时都设置 TIM2X\*\_SR 中的 TIF 标置。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。

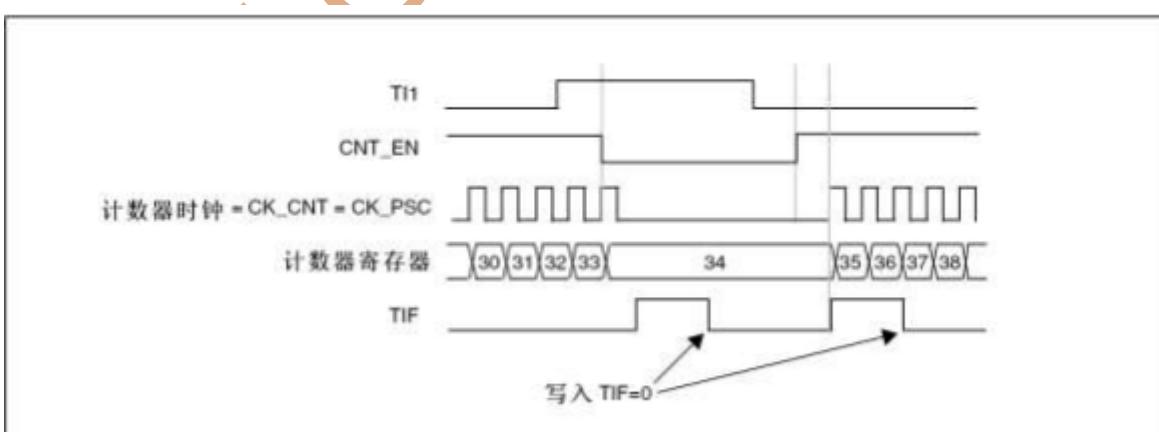


Figure 14-38 门控模式下的控制电路



#### 14.3-14.3 从模式：触发模式

输入端上选中的事件使能计数器。

● 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 TIM2X\*\_CCMR1 寄存器中 CC2S=01。置 TIM2X\*\_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。

● 置 TIM2X\*\_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM2X\*\_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。

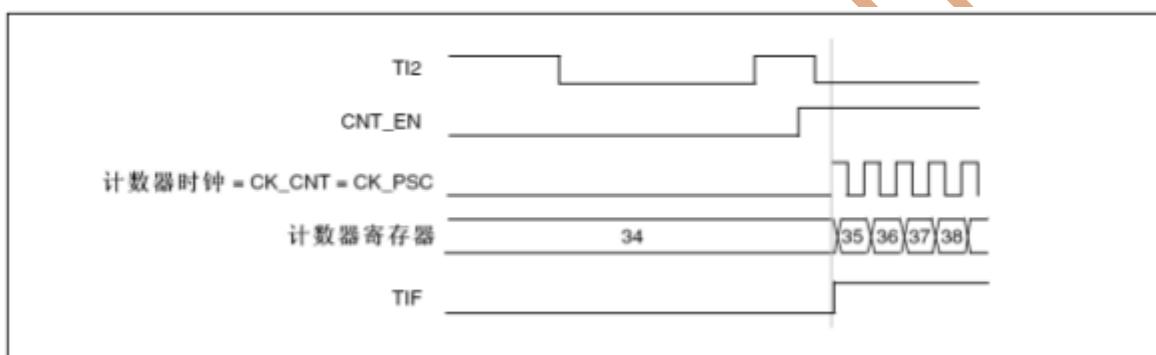


Figure 14-39 触发器模式下的控制电路

#### 14.3-14.4 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时可以选择另一个输入作为触发输入。不建议使用 TIM2X\*\_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

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

通过 TIM2X\*\_SMCR 寄存器配置外部触发输入电路：

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

按如下配置信道 1，检测 TI 的上升沿：

- IC1F=0000: 没有滤波
- 触发操作中不使用捕获预分频器，不需要配置
- 置 TIM2X\*\_CCMR1 寄存器中 CC1S=01，选择输入捕获源
- 置 TIM2X\*\_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)



置 `TIM2X*_SMCR` 寄存器中 `SMS=110`，配置定时器为触发模式。置 `TIM2X*_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。

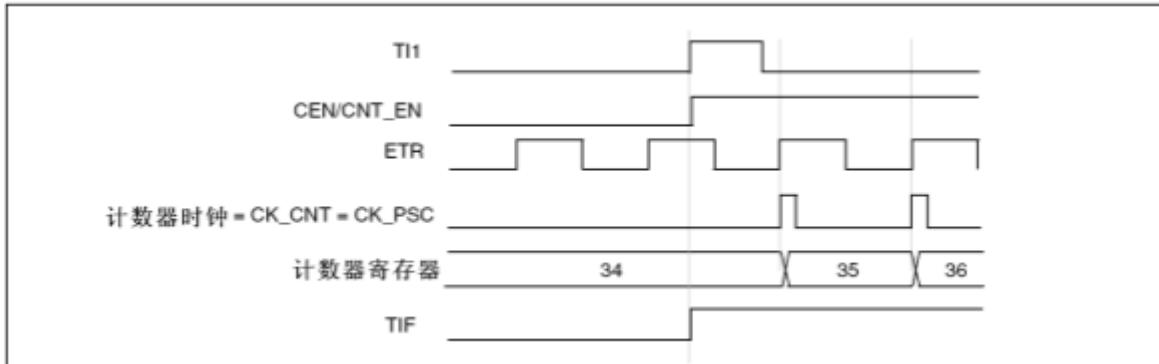


Figure 14-40 外部时钟模式 2 + 触发模式下的控制电路

### 14.3-15 定时器同步

所有 `TIM2X*` 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况。

#### 14.3-15.1 使用一个定时器作为另一个定时器的预分频器

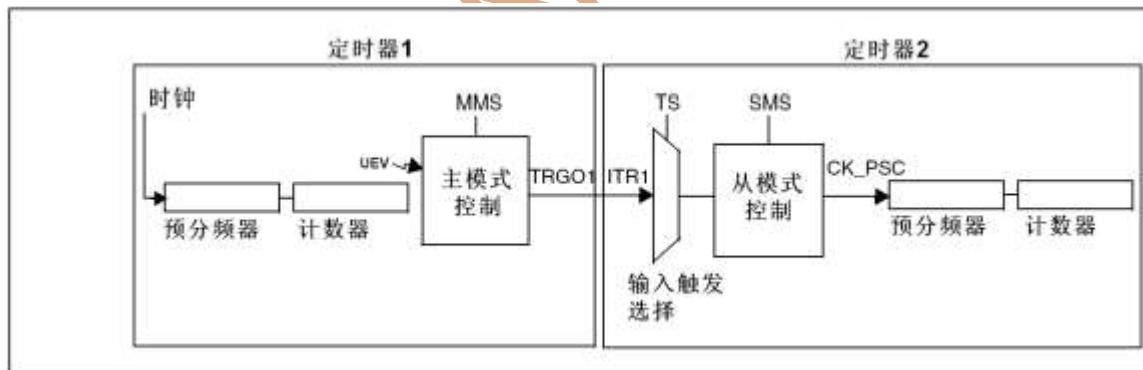


Figure 14-41 主/从定时器的例子

如：可以配置定时器 1 作为定时器 2 的预分频器。参考图 14-41，进行下述操作：

- 配置定时器 1 为主模式，它可以在每一个更新事件 `UEV` 时输出一个周期性的触发信号。在 `TIM1X*_CR2` 寄存器的 `MMS= '010'` 时，每当产生一个更新事件时在 `TRGO1` 上输出一个上升沿信号。
- 连接定时器 1 的 `TRGO1` 输出至定时器 2，设置 `TIM2X*_SMCR` 寄存器的 `TS= '000'`，配置定时器 2 为使用 `ITR1` 作为内部触发的从模式。



● 然后把从模式控制器置于外部时钟模式 1(TIM2X\*\_SMCR 寄存器的 SMS=111); 这样定时器 2 即可由定时器 1 周期性的上升沿(即定时器 1 的计数器溢出)信号驱动。

● 最后，必须设置相应(TIM2X\*\_CR1 寄存器)的 CEN 位分别启动两个定时器。

注：如果 OCx 已被选中为定时器 1 的触发输出(MMS=1xx)，它的上升沿用于驱动定时器 2 的计数器。

#### 14.3-15.2 使用一个定时器使能另一个定时器

在这个例子中，定时器 2 的使能由定时器 1 的输出比较控制。参考图 13-41 的连接。只当定时器 1 的 OC1REF 为高时，定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK\_INT 除以 3(fCK\_CNT=fCK\_INT/3)得到。

● 配置定时器 1 为主模式，送出它的输出比较参考信号(OC1REF)为触发输出(TIM1X\*\_CR2 寄存器的 MMS=100)

● 配置定时器 1 的 OC1REF 波形(TIM1X\*\_CCMR1 寄存器)

● 配置定时器 2 从定时器 1 获得输入触发(TIM2X\*\_SMCR 寄存器的 TS=000)

● 配置定时器 2 为门控模式(TIM2X\*\_SMCR 寄存器的 SMS=101)

● 置 TIM2X\*\_CR1 寄存器的 CEN=1 以使能定时器 2 (3/4/5)

● 置 TIM1X\*\_CR1 寄存器的 CEN=1 以启动定时器 1 (8/12)

注：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

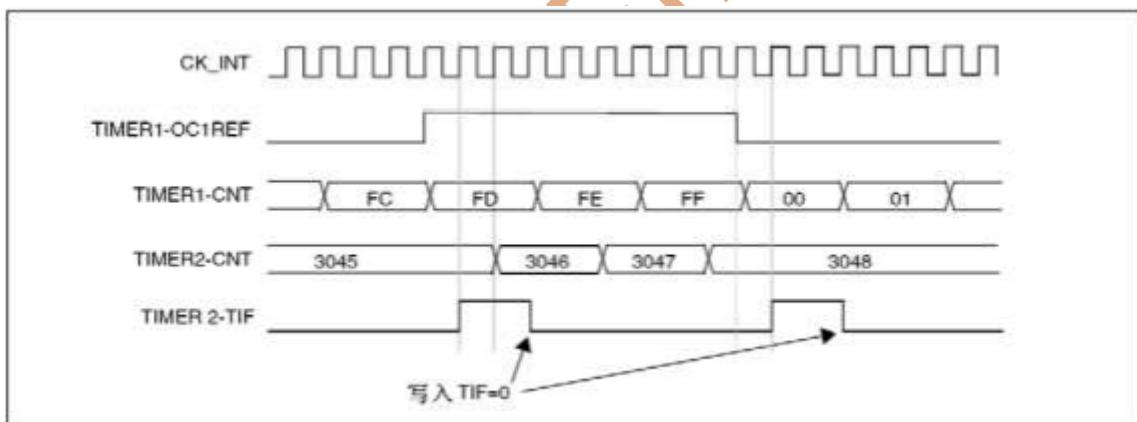


Figure 14-42 的 OC1REF 控制定时器 2,3,4,5

在图 14-42 的例子中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 TIM2X\*\_EGR 寄存器的 UG 位即可复位定时器。

在下一个例子中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模式并从 0xE7 开始；2 个定时器的预分频器系数相同。写‘0’到 TIM1X\*\_CR1 的 CEN 位将禁止定时器 1，定时器 2 随即停止。

● 配置定时器 1 为主模式，送出输出比较 1 参考信号(OC1REF)做为触发输出(TIM1X\*\_CR2 寄存器的 MMS=100)。



- 配置定时器 1 的 OC1REF 波形(TIM1X\*\_CCMR1 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发(TIM2X\*\_SMCR 寄存器的 TS=000)
- 配置定时器 2 为门控模式(TIM2X\*\_SMCR 寄存器的 SMS=101)
- 置 TIM1X\*\_EGR 寄存器的 UG= '1' , 复位定时器 1。
- 置 TIM2X\*\_EGR 寄存器的 UG= '1' , 复位定时器 2。
- 写' 0xE7' 至定时器 2 的计数器(TIM2X\*\_CNTL), 初始化它为 0xE7。
- 置 TIM2X\*\_CR1 寄存器的 CEN= '1' 以便能定时器 2。
- 置 TIM1X\*\_CR1 寄存器的 CEN= '1' 以启动定时器 1。
- 置 TIM1X\*\_CR1 寄存器的 CEN= '0' 以停止定时器 1。

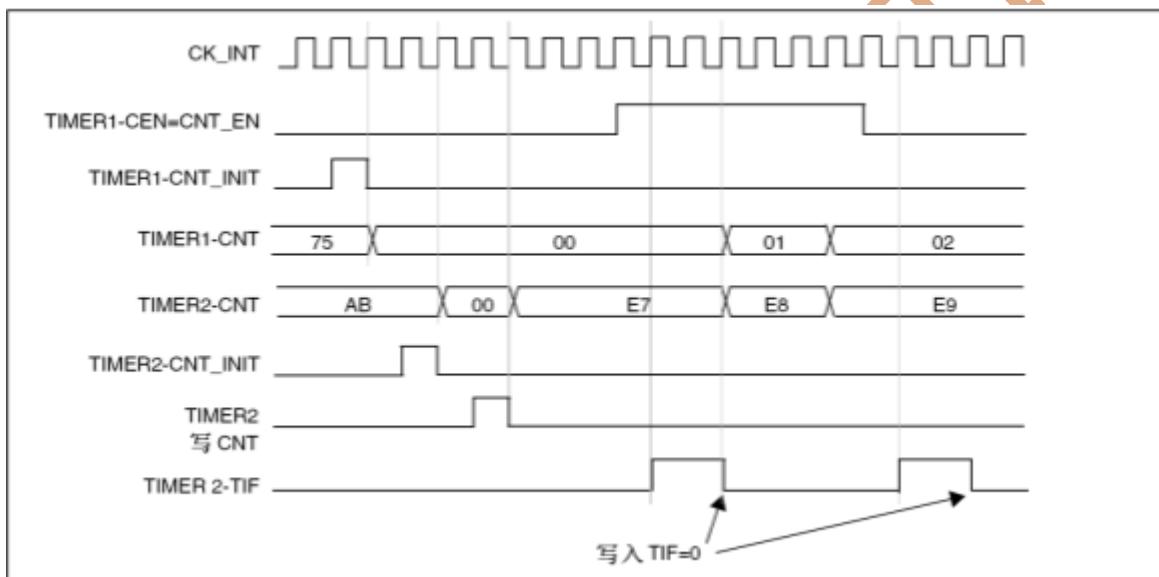


Figure 14-43 通过使能定时器 1 可以控制定时器 2

#### 14.3-15.3 使用一个定时器去启动另一个定时器

在这个例子中, 使用定时器 1 的更新事件使能定时器 2。参考图 13-41 的连接。一旦定时器 1 产生更新事件, 定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时, 定时器 2 的 CEN 位被自动地置' 1' , 同时计数器开始计数直到写' 0' 到 TIM2\_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK\_INT 除以 3( $f_{CK\_CNT}=f_{CK\_INT}/3$ )。

- 配置定时器 1 为主模式, 送出它的更新事件(UEV)做为触发输出(TIM1X\*\_CR2 寄存器的 MMS=010)。
- 配置定时器 1 的周期(TIM1X\*\_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发(TIM2X\*\_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式(TIM2X\*\_SMCR 寄存器的 SMS=110)



- 置 TIM1X\*\_CR1 寄存器的 CEN=1 以启动定时器 1。

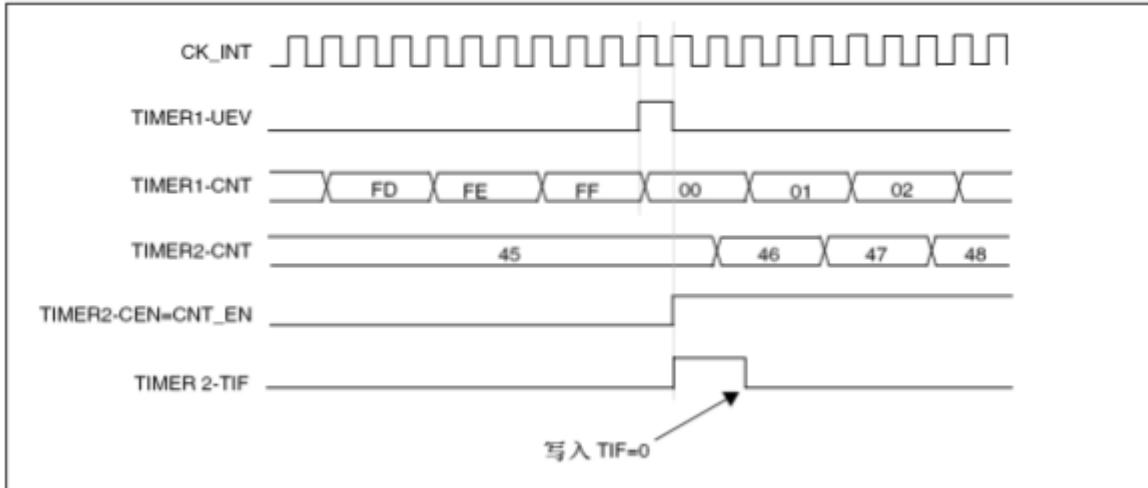


Figure 14-44 使用定时器 1 的更新触发定时器 2

在上一个例子中，可以在启动计数之前初始化两个计数器。图 13-45 显示在相同配置情况下，使用触发模式而不是门控模式(TIM2X\*\_SMCR 寄存器的 SMS=110)的动作。

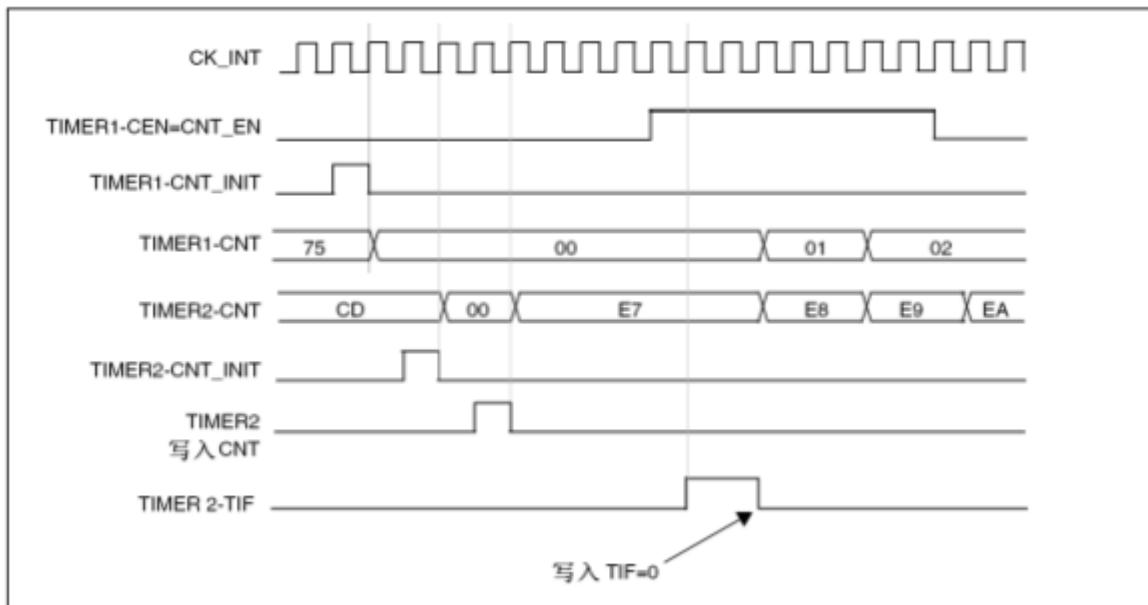


Figure 14-45 利用定时器 1 的使能触发定时器 2



#### 14.3-15.4 使用一个定时器作为另一个的预分频器

这个例子使用定时器 1 作为定时器 2 的预分频器。参考图 13-41 的连接，配置如下：

- 配置定时器 1 为主模式，送出它的更新事件 UEV 做为触发输出(TIM1X\*\_CR2 寄存器的 MMS= '010' )。然后每次计数器溢出时输出一个周期信号。
- 配置定时器 1 的周期(TIM1X\*\_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发(TIM2X\*\_SMCR 寄存器的 TS=000)
- 配置定时器 2 使用外部时钟模式(TIM2X\*\_SMCR 寄存器的 SMS=111)
- 置 TIM1X\*\_CR2 寄存器的 CEN=1 以启动定时器 2。
- 置 TIM1X\*\_CR1 寄存器的 CEN=1 以启动定时器 1。

#### 14.3-15.5 使用一个外部触发同步地启动 2 个定时器

这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见图 13-41。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 TI1 为从，对应定时器 2 为主)：

- 配置定时器 1 为主模式，送出它的使能做为触发输出(TIM1X\*\_CR2 寄存器的 MMS= '001' )
- 配置定时器 1 为从模式，从 TI1 获得输入触发(TIM1X\*\_SMCR 寄存器的 TS= '100' )
- 配置定时器 1 为触发模式(TIM1X\*\_SMCR 寄存器的 SMS= '110' )
- 配置定时器 1 为主/从模式，TIM1X\*\_SMCR 寄存器的 MSM= '1'
- 配置定时器 2 从定时器 1 获得输入触发(TIM2X\*\_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式(TIM2X\*\_SMCR 寄存器的 SMS= '110' )

当定时器 1 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TIF 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化(设置相应的 UG 位)，两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器(TIM2X\*\_CNT)在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNT\_EN 和 CK\_PSC 之间有个延迟。

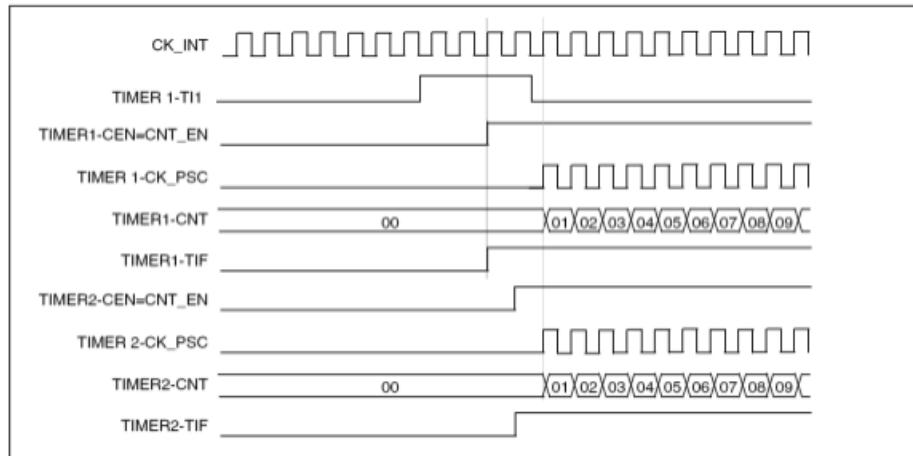


Figure 14-46 利用定时器 1 的使能触发定时器 2

#### 14.3-16 调试模式

当微控制器进入调试模式(Cortex-M0+核心停止), 根据 DBG 模块中 DBG\_TIM2X\*\_STOP 的设置, TIM2X\*

计数器或者继续正常操作, 或者停止。



#### 14.4 TIM2X\* 寄存器列表

可以用字(32位)的方式操作这些外设寄存器。

**TIM2 基地址 : 0x4000 0000**

**TIM2A 基地址 : 0x4000 0400**

**TIM2B 基地址 : 0x4000 0800**

**TIM2C 基地址 : 0x4000 0C00**

偏移地址	名称	描述	复位值
0x00	TIM2X*_CR1	TIM2X* 控制寄存器 1	0x0000 0000
0x04	TIM2X*_CR2	TIM2X* 控制寄存器 2	0x0000 0000
0x08	TIM2X*_SMCR	TIM2X* 从模式控制寄存器	0x0000 0000
0x0C	TIM2X*_DIER	TIM2X* 中断使能寄存器	0x0000 0000
0x10	TIM2X*_SR	TIM2X* 状态寄存器	0x0000 0000
0x14	TIM2X*_EGR	TIM2X* 事件产生寄存器	0x0000 0000
0x18	TIM2X*_CCMR1	TIM2X* 捕获/比较模式寄存器 1	0x0000 0000
0x1C	TIM2X*_CCMR2	TIM2X* 捕获/比较模式寄存器 2	0x0000 0000
0x20	TIM2X*_CCER	TIM2X* 捕获/比较使能寄存器	0x0000 0000
0x24	TIM2X*_CNT	TIM2X* 计数器	0x0000 0000
0x28	TIM2X*_PSC	TIM2X* 预分频器	0x0000 0000
0x2C	TIM2X*_ARR	TIM2X* 自动重装载寄存器	0x0000 0000
0x30	保留	保留	保留
0x34	TIM2X*_CCR1	TIM2X* 捕获/比较寄存器 1	0x0000 0000
0x38	TIM2X*_CCR2	TIM2X* 捕获/比较寄存器 2	0x0000 0000
0x3C	TIM2X*_CCR3	TIM2X* 捕获/比较寄存器 3	0x0000 0000
0x40	TIM2X*_CCR4	TIM2X* 捕获/比较寄存器 4	0x0000 0000
0x44	保留	保留	保留
0x48	TIM2X*_DCR	TIM2X* DMA控制寄存器	0x0000 0000
0x4C	TIM2X*_DMAR	TIM2X* 连续模式的DMA地址	0x0000 0000

Table 14-2 TIM2X\* 寄存器列表和复位值

**TIM2X\* : TIM2,TIM2A,TIM2B,TIM2C**



## 14.5 TIM2X\* 寄存器说明

### 14.5-1 TIM2X\* 控制寄存器 1(TIM2X\*\_CR1)

偏移地址: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						CKD[1:0]	ARP E	CMS[1:0]	DIR	OPM	URS	UDIS	CEN		
						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

位	标记	功能描述	复位值	读写
31:10	-	保留, 始终读为 0。	0	-
9:8	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: tDTS = tCK_INT 01: tDTS = 2 x tCK_INT 10: tDTS = 4 x tCK_INT 11: 保留, 不要使用这个配置	0	R/W
7	ARPE	自动重装载预装载允许位(Auto-reload preload enable)0: TIM2_ARR 寄存器没有缓冲; 1: TIM2_ARR 寄存器被装入缓冲器。	0	R/W
6:5	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(TIM2_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(TIM2_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(TIM2_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。	0	R/W
4	DIR	方向(Direction) 0: 计数器向上计数; 1: 计数器向下计数。	0	R/W



位	标记	功能描述	复位值	读写
		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。		
3	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。	0	R/W
2	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，则下述任一事件产生更新中断： – 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，则只有计数器溢出/下溢才产生更新中断。	0	R/W
1	UDIS	禁止更新(Update disable) 软件通过该位允许/禁止 UEV 事件的产生	0	R/W
0	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。	0	R/W

Note : 主从模式 TIM1 做主，TIM2 做从，需要設定 CCMR1[0]=1



### 14.5.2 TIM2X\* 控制寄存器 2(TIM2X\*\_CR2)

偏移地址: 0x04 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OIS4 R/W	OIS3 N R/W	OIS3 N R/W	OIS2 N R/W	OIS2 N R/W	OIS1 N R/W	OIS1 N R/W	TI1S R/W	MMS[1:0] R/W	保留	CCU S R/W	保留	CCP C R/W		

位	标记	功能描述	复位值	读写
31:8	-	保留, 始终读为 0。	0x0	-
7	TI1S	TI1 选择(TIM2X*_CH1 引脚连到 TI1 输入; 0: TIM2X*_CH1 引脚连到 TI1 输入; 1: TIM2X*_CH1、TIM2X*_CH2 和 TIM2X*_CH3 引脚经异或后连到 TI1 输入。)	0	R/W
6:4	MMS[2:0]	主模式选择(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 – TIM2X*_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 TIM2_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)。	0x0	R/W
3:0	-	保留, 始终读为 0。	0	-



### 14.5.3 TIM2X\* 从模式控制寄存器(TIM2X\*\_SMCR)

偏移地址: 0x08 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15	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。	0	R/W
14	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。	0	R/W
13:12	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 TIM1X*CLK 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。	0	R/W
11:8	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fDTS 采样 1000: 采样频率fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 ;1001: 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 ;1010: 采样频率	0x0	R/W



位	标记	功能描述	复位值	读写
		<p>fSAMPLING=fDTS/16, N=5</p> <p>0011: 采样频率 fSAMPLING=fCK_INT, N=8 ;1011: 采样频率 fSAMPLING=fDTS/16, N=6</p> <p>0100: 采样频率 fSAMPLING=fDTS/2, N=6;1100: 采样频率 fSAMPLING= fDTS /16, N=8</p> <p>0101: 采样频率 fSAMPLING=fDTS/2, N=8;1101: 采样频率 fSAMPLING= fDTS /32, N=5</p> <p>0110: 采样频率 fSAMPLING=fDTS/4, N=6 ;1110: 采样频率 fSAMPLING= fDTS /32, N=6</p> <p>0111: 采样频率 fSAMPLING=fDTS/4, N=8 ;1111: 采样频率 fSAMPLING= fDTS /32, N=8</p>		
7	MSM	<p>主/从模式(Master/slave mode)0: 无作用;</p> <p>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。</p>	0	R/W
6:4	TS[2:0]	<p>触发选择(Trigger selection)</p> <p>这3位选择用于同步计数器的触发输入。</p> <p>000: 内部触发 0(ITR0) ;100: TI1 的边沿检测器(TI1F_ED)</p> <p>001: 内部触发 1(ITR1) ;101: 滤波后的定时器输入 1(TI1FP1)</p> <p>010: 内部触发 2(ITR2) ;110: 滤波后的定时器输入 2(TI2FP2)</p> <p>011: 内部触发 3(ITR3) ; 111: 外部触发输入(ETRF)</p> <p>更多有关ITRx 的细节, 参见Table 14-3。</p> <p>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。</p>	0x0	R/W
3	-	保留, 始终读为 0。	0	-



位	标记	功能描述	复位值	读写
2:0	SMS[2:0]	<p>从模式选择(Slave mode selection)</p> <p>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)</p> <p>000: 关闭从模式 – 如果CEN=1，则预分频器直接由内部时钟驱动。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p>	0	R/W

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	tim1_trgo	irq_timer10	irq_timer11	irq_pca
TIM2A	tim1A_trgo	irq_timer10	irq_timer11	irq_pca
TIM2B	TIM1B_trgo	irq_timer10	irq_timer11	irq_pca
TIM2C	TIM1B_trgo	irq_timer10	irq_timer11	irq_pca

Table 14-3 TIM2X\* 内部触发连接

注：如果某个产品中没有相应的定时器，则对应的触发信号 ITRx 也不存在。



#### 14.5.4 TIM2X\* 中断与 DMA 请求使能寄存器(TIM2X\*\_DIER)

偏移地址: 0x0C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	TDE	--	CC4 DE	CC3 DE	CC2 DE	CC1 DE	UD E	--	TIE	--	CC4 IE	CC3 IE	CC2 IE	CC1 IE	UIE
R/ W	R/W	R/W	R/W	R/W	R/ W	R/ W	R/ W	R/ W	R/ W						

位	标记	功能描述	复位值	读写
31:15	-	保留, 始终读为 0。	0	-
14	TDE	始能DMA请求(TDREQ) 0: 无DMA请求 1:始能触发DMA请求(TDREQ)	0	R/W
13	Revered	Revered	0	-
12	CC4DE	始能捕获/比较4DMA请求 (CC4DREQ) 0: 无DMA请求 1:始能捕获/比较4DMA请求 (CC4DREQ)	0	R/W
11	CC3DE	始能DMA请求(UDREQ) 0: 无DMA请求 1:始能捕获/比较3DMA请求 (CC3DREQ)	0	R/W
10	CC2DE	始能DMA请求(UDREQ) 0: 无DMA请求 1:始能捕获/比较2DMA请求 (CC2DREQ)	0	R/W



位	标记	功能描述	复位值	读写
9	CC1DE	始能DMA请求(UDREQ) 0: 无DMA请求 1: 始能捕获/比较1DMA请求 (CC1DREQ)	0	R/W
8	UDE	始能DMA请求(UDREQ) 0: 无DMA请求 1: 始能更新DMA请求(UDREQ)	0	R/W
7	Revered	Revered	0	-
6	TIE	触发中断使能(Trigger interrupt enable)0: 禁止触发中断; 1: 使能触发中断。	0	R/W
5	-	保留, 始终读为 0。	0	-
4	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。	0	R/W
3	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。	0	R/W
2	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。	0	R/W
1	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。	0	R/W
0	UIE	允许更新中断(Update interrupt enable)0: 禁止更新中断; 1: 允许更新中断。	0	R/W



#### 14.5.5 TIM2X\* 状态寄存器(TIM2X\*\_SR)

偏移地址: 0x10 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CC4OF	CC3OF	CC2OF	CC1OF	保留	TIF	保留	CC4IF	CC3IF	CC2IF	CC1IF	UIF	RCW0	RCW0	RCW0
	RCW0	RCW0	RCW0	RCW0		RCW0		RCW0	RCW0	RCW0	RCW0				

位	标记	功能描述	复位值	读写
31:13	-	保留, 始终读为 0。	0	-
12	CC4OF	捕获/比较4 重复捕获标记(Capture/Compare 4 overcapture flag) 参见CC1OF 描述。	0	RCW0
11	CC3OF	捕获/比较3 重复捕获标记(Capture/Compare 3 overcapture flag) 参见CC1OF 描述。	0	RCW0
10	CC2OF	捕获/比较2 重复捕获标记(Capture/Compare 2 overcapture flag) 参见CC1OF 描述。	0	RCW0
9	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TIM2X*_CCR1 寄存器时, CC1IF 的状态已经为'1'。	0	RCW0
8:7	-	保留, 始终读为 0。	0	-
6	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。	0	RCW0
5	-	保留, 始终读为 0。	0	-
4	CC4IF	捕获/比较4 中断标记(Capture/Compare 4 interrupt flag) 参考CC1IF 描述。	0	RC



位	标记	功能描述	复位值	读写
				W 0
3	CC3IF	捕获/比较3中断标记(Capture/Compare 3 interrupt flag) 参考CC1IF描述。	0	RC W 0
2	CC2IF	捕获/比较2中断标记(Capture/Compare 2 interrupt flag) 参考CC1IF描述。	0	RC W 0
1	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TIM2X*_CR1 寄存器的CMS 位)。它由软件清‘0’。 0：无匹配发生； 1：TIM2X*_CNT 的值与TIM2X*_CCR1 的值匹配。 当 TIM2X*_CCR1 的内容大于 TIM2X*_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 TIM2X*_CCR1 清‘0’。 0：无输入捕获产生； 1：计数器值已被捕获(拷贝)至TIM2X*_CCR1(在 IC1 上检测到与所选极性相同的边沿)。	0	RC W 0
0	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 TIM2X*_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 TIM2X*_CR1 寄存器的 URS=0、UDIS=0，当设置 TIM2X*_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 TIM2X*_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 13.5.3：从模式控制寄存器 (TIM2X*_SMCR))。	0	RC W 0



#### 14.5.6 TIM2X\* 事件产生寄存器(TIM2X\*\_EGR)

偏移地址:0x14 复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TG	保留	CC4 G	CC3 G	CC2 G	CC1 G	UG	
WO								WO	保留	WO	WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:7	-	保留，始终读为 0。	0	-
6	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: TIM2X*_SR 寄存器的TIF=1，若开启对应的中断，则产生相应的中断。	0	WO
5	-	保留，始终读为 0。	0	-
4	CC4G	产生捕获/比较 4 事件(Capture/Compare 4 generation) 参考 CC1G 描述。	0	WO
3	CC3G	产生捕获/比较 3 事件(Capture/Compare 3 generation) 参考 CC1G 描述。	0	WO
2	CC2G	产生捕获/比较 2 事件(Capture/Compare 2 generation) 参考 CC1G 描述。	0	WO
1	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，则产生相应的中断。若信道 CC1 配置为输入： 当前的计数器值被捕获至 TIM2X*_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。	0	WO



0	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 TIM2X* ARR 的值。	0	WO
---	----	--	---	----

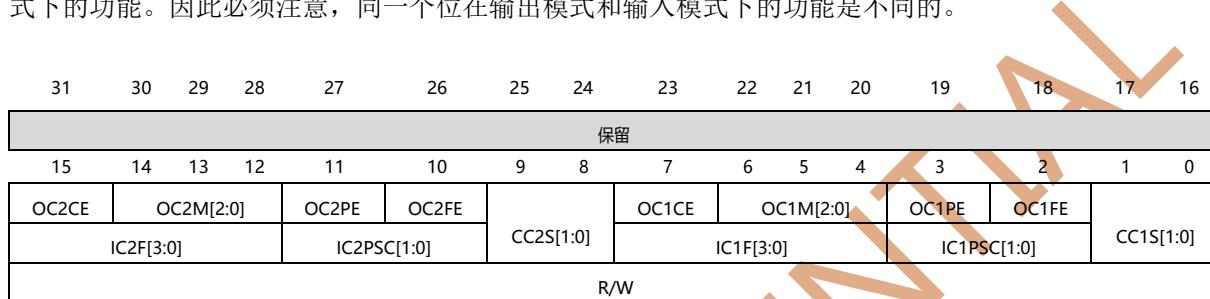
ZBT CONFIDENTIAL



#### 14.5.7 TIM2X\* 捕获/比较模式寄存器 1(TIM2X\*\_CCMR1)

偏移地址: 0x18 复位值: 0x0000 0000

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 CC<sub>x</sub>S 位定义。该寄存器其它位的作用在输入和输出模式下不同。OC<sub>xx</sub> 描述了信道在输出模式下的功能, IC<sub>xx</sub> 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:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15	OC2CE	输出比较 2 清 0 使能(Output Compare 2 clear enable)	0	R/W
14:12	OC2M[2:0]	输出比较 2 模式(Output Compare 2 mode)	0	R/W
11	OC2PE	输出比较 2 预装载使能(Output Compare 2 preload enable)	0	R/W
10	OC2FE	输出比较 2 快速使能(Output Compare 2 fast enable)	0	R/W
9:8	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 TIM2X*_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC2E=0) 才是可写的。	0	R/W



位	标记	功能描述	复位值	读写
7	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。	0	R/W
6:4	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 TIM2X*_CCR1 与计数器 TIM2X*_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 TIM2X*_CNT 的值与捕获/比较寄存器 1(TIM2X*_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 TIM2X*_CNT 的值与捕获/比较寄存器 1(TIM2X*_CCR1)相同时, 强制 OC1REF 为低。011: 翻转。当 TIM2X*_CCR1=TIM2X*_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 TIM2X*_CNT<TIM2X*_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 TIM2X*_CNT>TIM2X*_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 TIM2X*_CNT<TIM2X*_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 TIM2X*_CNT>TIM2X*_CCR1 时通道 1 为有效电平, 否则为无效电平。 注: 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。	0	R/W



位	标记	功能描述	复位值	读写
3	OC1PE	<p>输出比较 1 预装载使能(Output Compare 1 preload enable)</p> <p>0: 禁止TIM2X*_CCR1 寄存器的预装载功能, 可随时写入TIM2X*_CCR1寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启TIM2X*_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2X*_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注: 仅在单脉冲模式下(TIM2X*_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。</p>	0	R/W
2	OC1FE	<p>输出比较 1 快速使能(Output Compare 1 fast enable)</p> <p>该位用于加快 CC 输出对触发输入事件的响应。</p> <p>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。</p> <p>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。</p> <p>OCFE 只在信道被配置成 PWM1 或 PWM2 模式时起作用。</p>	0	R/W
1:0	CC1S[1:0]	<p>捕获/比较 1 选择。(Capture/Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 信道被配置为输出;</p> <p>01: CC1 信道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 信道被配置为输入, IC1 映射在 TI2 上;</p> <p>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 TIM2X*_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC1S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC1E=0)才是可写的。</p>	0	R/W

Note: 主从模式 TIM1 做主, TIM2 做从, 需要設定 CCMR1[0]=1

#### 输入捕获模式:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:12	IC2F[3:0]	输入捕获 2 滤波器(Input capture 2 filter)	0x0	R/W



位	标记	功能描述	复位值	读写
11:10	IC2PSC[1:0]	输入/捕获2预分频器(Input capture 2 prescaler)	0x0	R/W
9:8	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 TIM2X*_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2X*_CCER 寄存器的CC2E=0)才是可写的。	0x0	R/W
7:4	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: 0000: 无滤波器, 以 fDTS 采样 1000: 采样频率 $f_{SAMPLING} = fDTS/8, N=6$ 0001: 采样频率 $f_{SAMPLING} = fCK\_INT, N=2$ 1001: 采样频率 $f_{SAMPLING} = fDTS/8, N=8$ 0010: 采样频率 $f_{SAMPLING} = fCK\_INT, N=4$ 1010: 采样频率 $f_{SAMPLING} = fDTS/16, N=5$ 0011: 采样频率 $f_{SAMPLING} = fCK\_INT, N=8$ 1011: 采样频率 $f_{SAMPLING} = fDTS/16, N=6$ 0100: 采样频率 $f_{SAMPLING} = fDTS/2, N=6$ 1100: 采样频率 $f_{SAMPLING} = fDTS/16, N=8$ 0101: 采样频率 $f_{SAMPLING} = fDTS/2, N=8$ 1101: 采样频率 $f_{SAMPLING} = fDTS/32, N=5$ 0110: 采样频率 $f_{SAMPLING} = fDTS/4, N=6$ 1110: 采样频率 $f_{SAMPLING} = fDTS/32, N=6$ 0111: 采样频率 $f_{SAMPLING} = fDTS/4, N=8$ 1111: 采样频率	0x0	R/W



位	标记	功能描述	复位值	读写
		$f_{\text{SAMPLING}} = f_{\text{DTS}}/32, N=8$		
3:2	IC1PSC[1:0]	<p>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<math>\text{CC1E}=0</math>(TIM2X*_CCER寄存器中), 则预分频器复位。</p> <p>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。</p>	0x0	R/W
1:0	CC1S[1:0]	<p>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:</p> <p>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TIM2X*_SMCR寄存器的TS位选择)。</p> <p>注: CC1S仅在通道关闭时(TIM2X*_CCER寄存器的CC1E=0)才是可写的。</p>	0x0	R/W



#### 14.5-8 TIM2X\* 捕获/比较模式寄存器 2(TIM2X\*\_CCMR2)

偏移地址: 0x1C 复位值: 0x0000 0000

参看以上 CCMR1 寄存器的描述。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]	OC4PE	OC4FE	CC4S[1:0]	OC3CE	OC3M[2:0]	OC3PE	OC3FE	CC3S[1:0]						
IC4F[3:0]	IC4PSC[1:0]				IC3F[3:0]	IC3PSC[1:0]									
R/W															

输出比较模式:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15	OC4CE	输出比较 4 清0 使能(Output Compare 4 clear enable)	0	R/W
14:12	OC4M[2:0] ]	输出比较 4 模式(Output Compare 4 mode)	0x0	R/W
11	OC4PE	输出比较 4 预装载使能(Output Compare 4 preload enable)	0	R/W
10	OC4FE	输出比较 4 快速使能(Output Compare 4 fast enable)	0	R/W
9:8	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 TIM2X*_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC4E=0)才是可写的。	0x0	R/W
7	OC3CE	输出比较 3 清'0' 使能(Output Compare 3 clear enable)	0	R/W
6:4	OC3M[2:0] ]	输出比较 3 模式(Output Compare 3 mode)	0x0	R/W
3	OC3PE	输出比较 3 预装载使能(Output Compare 3 preload enable)	0	R/W



2	OC3FE	输出比较 3 快速使能(Output Compare 3 fast enable)	0	R/W
1:0	CC3S[1:0]	<p>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 TIM2X*_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC3S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC3E=0) 才是可写的。</p>	0x0	R/W

输入捕获模式:

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:12	IC4F[3:0]	输入捕获 4 滤波器(Input capture 4 filter)	0	R/W
11:10	IC4PSC[1:0]	输入/捕获 4 预分频器(Input capture 4 prescaler)	0	R/W
9:8	CC4S[1:0]	<p>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 TIM2X*_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC4S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC4E=0) 才是可写的。</p>	0	R/W
7:4	IC3F[3:0]	输入捕获 3 滤波器(Input capture 3 filter)	0	R/W
3:2	IC3PSC[1:0]	输入/捕获 3 预分频器(Input capture 3 prescaler)	0	R/W
1:0	CC3S[1:0]	<p>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 TIM2X*_SMCR 寄存器的 TS</p>	0	R/W



	位选择)。 注：CC3S 仅在通道关闭时(TIM2X*_CCER 寄存器的CC3E=0) 才是可写 的。		
--	---	--	--

ZBIT CONFIDENTIAL



#### 14.5.9 TIM2X\* 捕获/比较使能寄存器(TIM2X\*\_CCER)

偏移地址: 0x20 复位值: 0x0000 0000

位	标记	功能描述	复位值	读写
31:14	-	保留, 始终读为 0。	0x0	-
13	CC4P	输入/捕获4输出极性(Capture/Compare 4 output polarity) 参考CC1P的描述。	0	R/W
12	CC4E	输入/捕获4输出使能(Capture/Compare 4 output enable) 参考CC1E的描述。	0	R/W
11:10	-	保留, 始终读为 0。	0x0	-
9	CC3P	输入/捕获3输出极性(Capture/Compare 3 output polarity) 参考CC1P的描述。	0	R/W
8	CC3E	输入/捕获3输出使能(Capture/Compare 3 output enable) 参考CC1E的描述。	0	R/W
7:6	-	保留, 始终读为 0。	0x0	-
5	CC2P	输入/捕获2输出极性(Capture/Compare 2 output polarity) 参考CC1P的描述。	0	R/W
4	CC2E	输入/捕获2输出使能(Capture/Compare 2 output enable) 参考CC1E的描述。	0	R/W
3:2	-	保留, 始终读为 0。	0x0	-
1	CC1P	CC1P: 输入/捕获1输出极性 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。	0	R/W



0	CC1E	CC1E: 输入/捕获1输出使能 CC1信道配置为输出: 0: 关闭—OC1禁止输出。 1: 开启—OC1信号输出到对应的输出引脚。 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; 0: 捕获使能。	0	R/W
---	------	--	---	-----

CCxE 位	OCx 输出状态
0	禁止输出(OCx=0, OCx_EN=0)
1	OCx=OCxREF + 极性, OCx_EN=1

注: 连接到标准 OCx 通道的外部 I/O 引脚状态, 取决于 OCx 通道状态和 GPIO 以及 AFIO 寄存器

#### 14.5-10 TIM2X\* 计数器(TIM2X\*\_CNT)

偏移地址: 0x24 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0x0	-
15:0	CNT[15:0]	计数器的值(Counter value)	0x0	R/W

#### 14.5-11 TIM2X\* 预分频器(TIM2X\*\_PSC)

偏移地址: 0x28 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写



31:16	-	保留, 始终读为 0。	0x0	-
15:0	PSC[15:0]	预分频器的值(Prescaler value) 计数器的时钟频率(CK_CNT)等于 fCK_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' 。	0x0	R/W

ZBT CONFIDENTIAL



#### 14.5.12 TIM2X\* 自动重装载寄存器(TIM2X\*\_ARR)

偏移地址:0x2C 复位值:0x0000 0000

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:0	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。	0	R/W

#### 14.5.13 TIM2X\* 捕获/比较寄存器 1(TIM2X\*\_CCR1)

偏移地址: 0x34 复位值: 0x0000 0000

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-

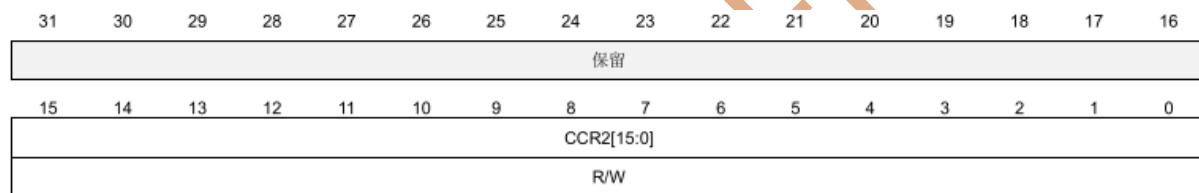
位	标记	功能描述	复位值	读写
31:16	-	捕获/比较通道 1 的值(Capture/Compare 1 value) 若 CC1 信道配置为输出： CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。如果在 TIM2X*_CCMR1 寄存器(OC1PE 位)中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2X*_CNT 的比	0	R/W



		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。		
--	--	--	--	--

#### 14.5-14 TIM2X\* 捕获/比较寄存器 2(TIM2X\*\_CCR2)

偏移地址: 0x38 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:16	-	保留，始终读为 0。	0	-
15:0	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 TIM2X*_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TIM2X*_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。	0	R/W



#### 14.5-15 TIM2X\* 捕获/比较寄存器 3(TIM2X\*\_CCR3)

偏移地址: 0x3C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:0	CCR3[15:0]	<p>捕获/比较通道 3 的值(Capture/Compare 3 value)            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 TIM2X*_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TIM2X*_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。</p>	0	R/W

#### 14.5-16 TIM2X\* 捕获/比较寄存器 4(TIM2X\*\_CCR4)

偏移地址: 0x40 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
R/W															

位	标记	功能描述	复位值	读写



31:16	-	保留, 始终读为 0。	0	-
15:0	CCR4[15:0]	<p>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 TIM2X*_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TIM2X*_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。</p>	0	R/W



#### 14.5.17 TM2X\*DMA 控制寄存器(TIM2X\*\_DCR)

地址偏移: 0x48 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DBL[4:0]					保留				DBA[4:0]				
保留		rw rw rw rw rw					保留				rw rw rw rw rw				

位	标记	功能描述	复位值	读写
31:13	-	保留, 始终读为 0。	0x0	-
12:18	DBL[4:0]	<p>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TIM2x_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:  <math>(TIM2x_CR1的地址) + DBA + (DMA索引)</math>, 其中 DMA索引 = DBL            其中(TIM2x_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。</p>	0x0	R/W
7:5	保留	-	0x0	-
		<p>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TIM2x_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: TIM2x_CR1,</p>	0x0	R/W



位	标记	功能描述	复位值	读写
4:0	DBA[4:0]	00001: TIM2x_CR2, 00010: TIM2x_SMCR, .....		

#### 14.5-18 TM2X\*连续模式的 DMA 地址(TIM2X\*\_DMAR)

地址偏移: 0x4C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
DMAB[31:16]														
rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw															

位	标记	功能描述	复位值	读写
31:0	DMAB[31:0]	DMAB[31:0]: DMA连续传送寄存器(DMA register for burst accesses)  对TIM2X*_DMAR寄存器的读或写会导致对以下地址的寄存器的存取操作：  “TIM2X*_CR1 Address + (DBA + DMA index)x4”，其中： “TIM2X*_CR1地址”是控制寄存器1的地址； “DBA”是TIM2X*_DCR寄存器中定义的基地址； “DMA指针”是由DMA自动控制的偏移量，它取决于TIM2X*_DCR寄存器中定义的DBL。	0x0000	R/W



## 【15】可编程计数阵列(PCA)

### 15.1 PCA 简介

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

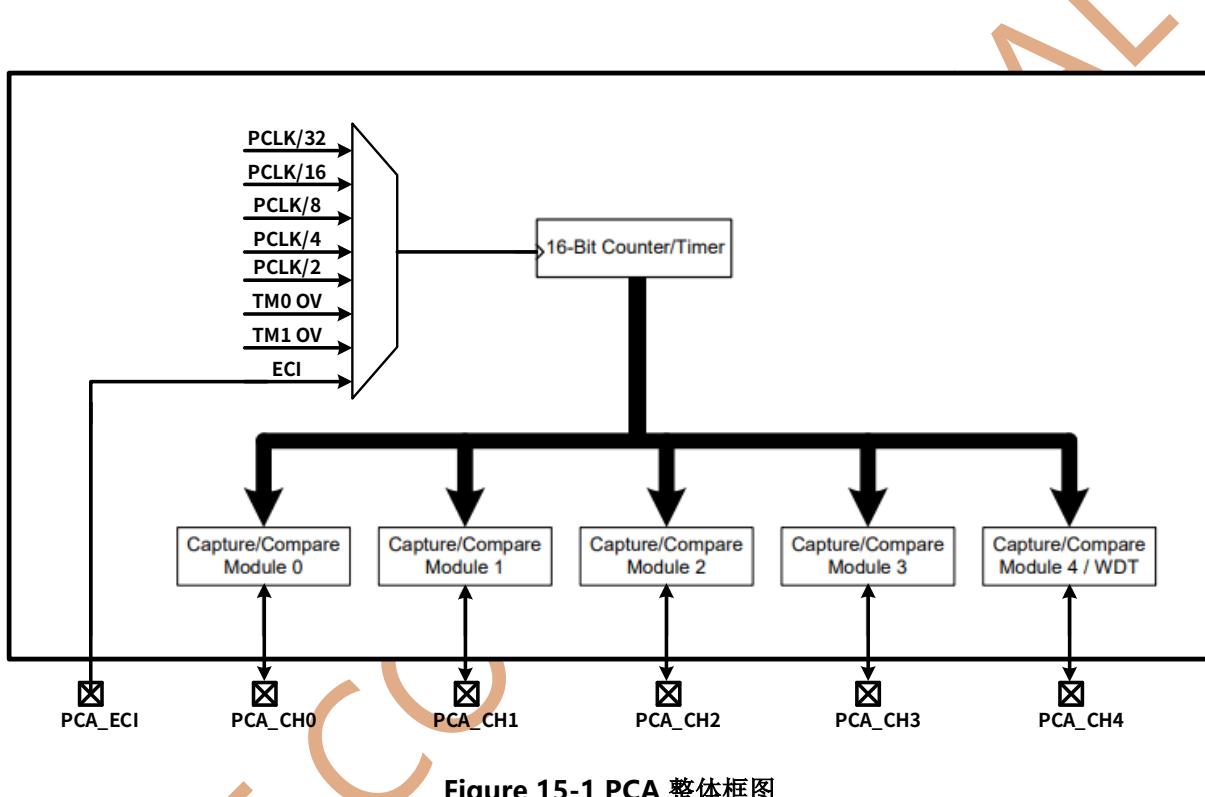


Figure 15-1 PCA 整体框图



## 15.2 PCA 功能描述

每个模块都可被配置为独立工作，有三种工作方式：边沿触发捕捉、输出比较、8位脉宽调制。每个模块在系统控制器中都有属于自己的功能寄存器，这些寄存器用于配置模块的工作方式和与模块交换数据。每组比较/捕获模块是由一个比较/捕获寄存器组(CCAPx)、1个16位比较器和各种逻辑门控制组成。寄存器组用来存储时间或次数，针对外部触发捕获条件，或内部触发比较条件。在 PWM 模式下，寄存器(CCAPxL)用来控制输出波形的占空比。每个模块都可以独立编程的操作在任何以下模式：

1. 16位捕获模式的上升沿，下降沿或任意沿触发。
2. 比较模式：16位软件定时器，16位高速输出或8位脉冲宽度调制。
3. 未启动。

比较/捕获模块模式寄存器(CCAPMx)确定相应的工作模式。对于比较/捕获模块进行编程时，他们是基于共同的时间计数。定时器/计数器打开和关闭通过 CR.CR 位即可控制 PCA 定时/计数器的运行。如果相应的使能位(CCAPMx.CCIE)被设置，当出现匹配或捕获时，比较/捕获标志(CR.CCFx)被设置并产生 PCA 中断请求。CPU 可以在任何时候读写 CCAPx 寄存器。

### 15.2-1 PCA 定时/计数器

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

1. 系统时钟 PCLK 的 32 分频
2. 系统时钟 PCLK 的 16 分频
3. 系统时钟 PCLK 的 8 分频
4. 系统时钟 PCLK 的 4 分频
5. 系统时钟 PCLK 的 2 分频
6. 定时器 TIM10 的溢出(overflow): 每次定时器 TIM10 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
7. 定时器 TIM11 的溢出(overflow): 每次定时器 TIM11 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
8. ECI: CPU 每过 4 个 PCLK 时钟周期就对 PCA ECI 进行采样，当每次采样结果从高变低时，CNT\_L(CNT low 8 bit)自动加 1，因此最高的 ECI 输入频率不能高于系统时钟 PCLK 的 1/8，以满足采样需求。



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

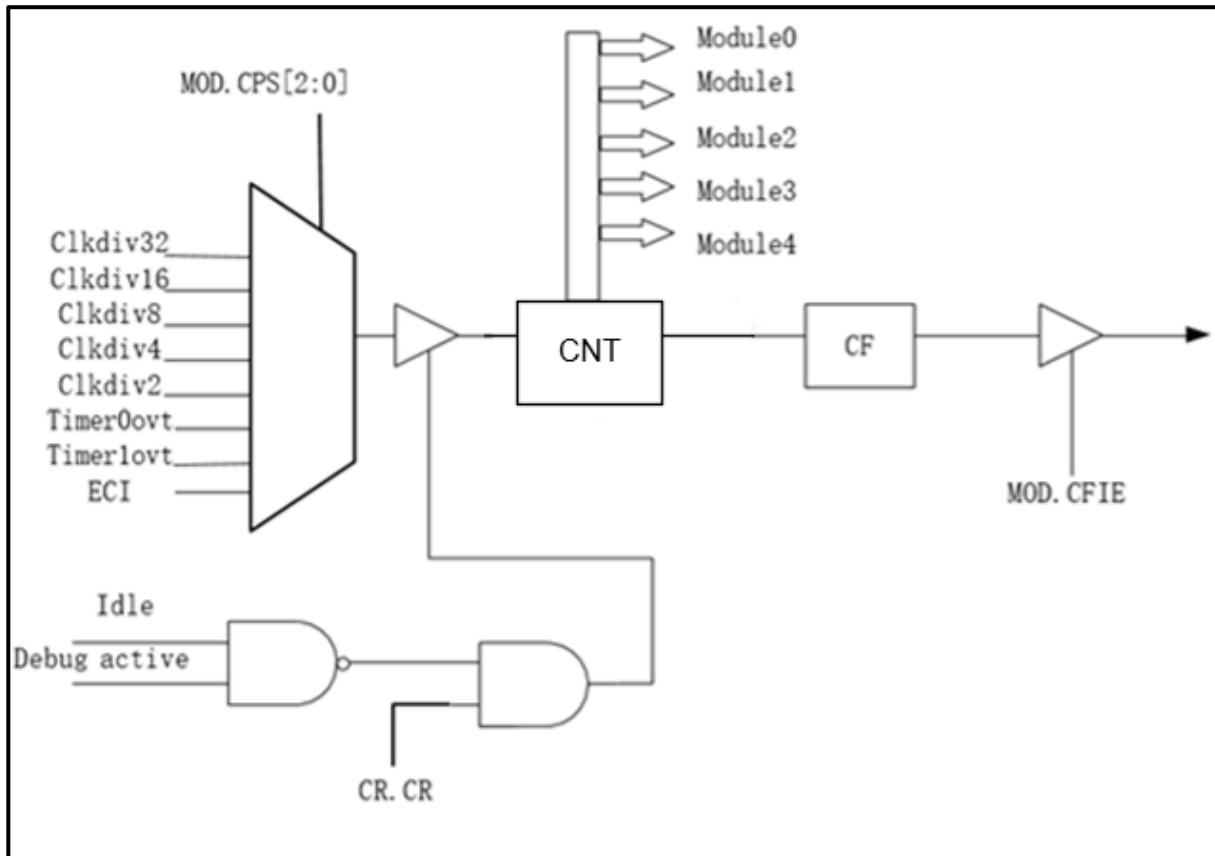


Figure 15-2 PCA 计数器框图

### 15.2-2 捕获功能

PCA 捕获模式提供了 5 路 PCA 测量脉冲周期、脉冲宽、占空比和相位差的功能。引脚上出现的电平跳变导致 PCA 捕获 PCA 计数器/定时器的值并将其装入到对应模块的 16 位捕捉/比较寄存器(CCAPPx)。CCAPMx.CAPP 以及 CCAPMx.CAPN 位用于选择触发捕捉的电平变化类型：低电平到高电平(正沿)、高电平到低电平(负沿)或任何变化(正沿或负沿)。当捕捉发生时，CR 中的捕捉/比较标志(CCFx)被置为逻辑 ‘1’ 并产生一个中断请求(如果 CCF 中断被允许)。当 CPU 转向中断服务程序时，CR.CCFx 位不能被硬件自动清除，用户软件写 INTCLR.CCFx 寄存器清除此标志位。如果 CCPMx.CAPP 以及 CCAPMx.CAPN 位都被设置为逻辑 ‘1’，可以通过直接读对应端口引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。分辨率等于定时器/计数器的时钟。输入信号必须在高电平或低电平期间至少保持 2 个时钟周期，以保证输入信号能够被硬件识别。



CPU 可以在任何时候读取或写入 CCAPx 的寄存器。捕获设置：

1. 当需要在外部上升沿进行捕获, CCPMx.CAPP = "1" 以及 CCAPMx.CAPN = "0"
2. 当需要在外部下降沿进行捕获, CCPMx.CAPP = "0" 以及 CCAPMx.CAPN = "1"
3. 当需要在外部上升、下降沿进行捕获, CCPMx.CAPP = "1" 以及 CCAPMx.CAPN = "1"

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

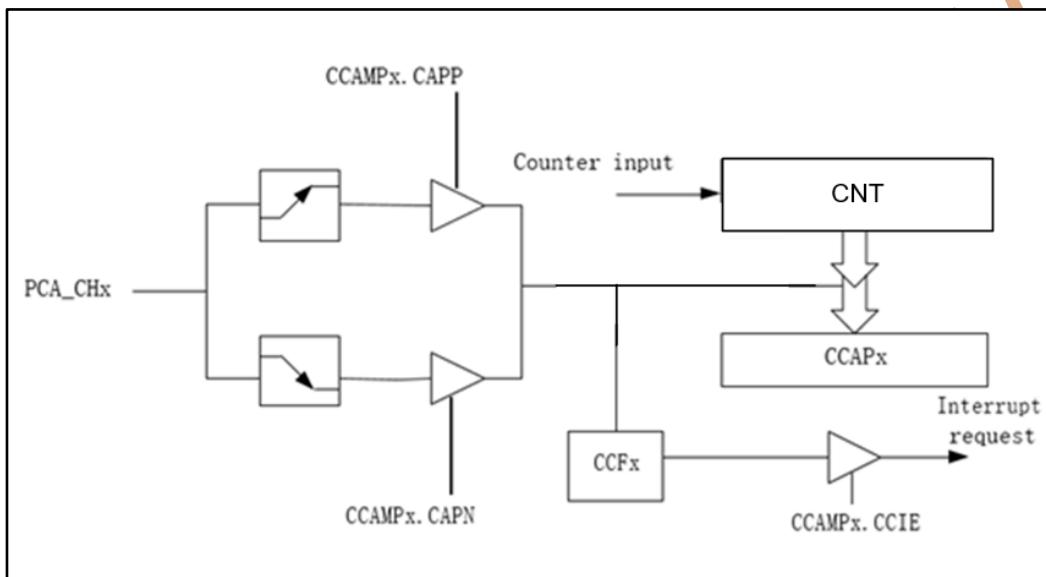


Figure 15-3 PCA 捕获功能框图

### 15.2.3 PCA 比较功能

PCA 比较功能提供如下功能：定时器、事件计数器、脉冲宽度调制。PCA 比较功能可以提供三种模式：16 位软件定时器模式，高速输出模式，PWM 模式。在前两个模式下，比较/捕获模块比较 16 位 PCA 定时器/计数器的值与预先加载到该模块的 CCAPx 寄存器中的 16 位值。在 PWM 模式下，PCA 模块不断地将 PCA 定时器/计数器低字节寄存器(CNT)与一个在 CCAPxL 寄存器的 8 位的值进行比较。每 4 个时钟周期比较一次，即与最快的 PCA 定时器/计数器的时钟速率相匹配。设置 CCAPMx.ECOM 位选择该模块的比较功能。若要正确使用在比较模式下的模块，请遵守以下的一般程序：

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



### 15.2-3.1 16 位软件计数器模式

要设定一个比较/捕获模块工作在 16 位软件定时器模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.MAT 位。一旦在 PCA 定时器/计数器和比较/捕获的寄存器(CCAPx)之间发生了匹配，这将设置模块的比较/捕获标志 (CR.CCFx)。这将产生一个中断请求，如果相应的中断使能位 (CCAPMx.CCIE) 设置。由于硬件并不清除比较/捕获标志(CR.CCFx)，用户必须清除软件标志。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器(CCAPx)。注意：在更新这些寄存器时，为了防止无效的匹配发生，用户软件应该先写 CCAPxL，后写 CCAPxH。一旦写入 CCAPxL 就会清除禁用比较功能 ECOMx 位，而写入 CCAPxH 会同时设置的 ECOMx 位，重新启用比较功能。即当向 PCA0 的捕捉/比较寄存器写入一个 16 位数值时，应先写低字节。

### 15.2-3.2 高速输出模式

在高速输出模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器(CCAPx)发生匹配时，模块 PCA 的 PCA\_CCAPOx 寄存器（对应于 PCA\_CHx 引脚）的值将发生翻转。这可以提供比切换 IO 输出有更高精度，因为这个高速输出不会响应中断而影响输出频率，靠 CPU 来切换 IO 输出的话，功耗、精度都有所欠缺。

要设定一个比较/捕获模块的高速输出模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位。PCA 定时器/计数器和比较/捕获的寄存器(CCAPx)之间的匹配翻转 PCA 的 PCA\_CCAPOx 寄存器的值，并设置模块的比较/捕获标志(CR.CCFx)。

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

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

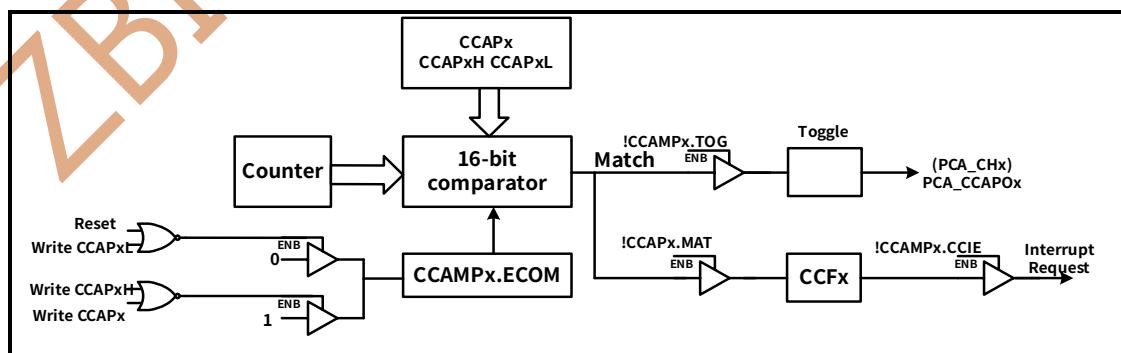


Figure 15-4 PCA 比较功能框图



### 15.2-3.3 8 位脉宽调制(PWM)功能

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

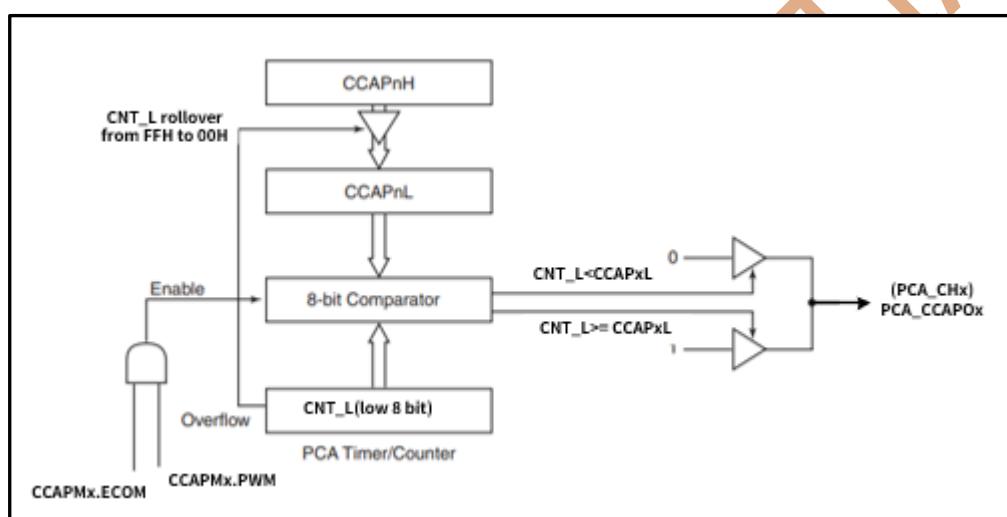


Figure 15-5 PCA PWM 功能框图

在这种模式下，PCA定时器/计数器CNT\_L的值是不断在低字节比较/捕获寄存器(CCAPxL)的值相比。当  $CNT\_L < CCAPxL$ ，输出波形为低。当两者匹配时( $CNT\_L = CCAPxL$ )，输出波形去到高，直到  $CNT\_L$  溢出从FFH到00H，结束期间仍然很高。在溢出时，在CCAPxH的值自动装载到CCAPxL内，一个新的周期的开始。

在CCAPxL的值决定当前波形的占空比。在CCAPxH的值确定下一个波形的占空比。改变CCAPxL中的值即可更改的脉冲宽度调制。正如图所示，8位值在CCAPxL可以从0(100%占空比)，到255(0.4%占空比)。要改变CCAPxL值而不产生毛刺，需要在高字节寄存器(CCAPxH)写入一个新值。当CL超过0xFF滚动到0x00，这个值是由硬件自动加载到CCAPxL。

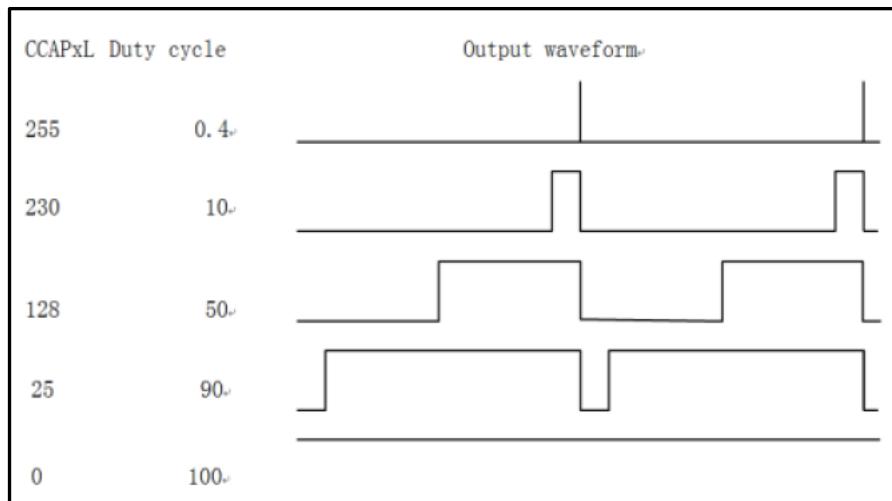


Figure 15-6 PCA PWM 输出波形

要设定一个比较/捕获模块工作在 PWM 模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.PWM 位。

另外 PCA 定时器/计数器由编程 MOD.CSP[2:0]可以选择输入计数信号频率。在 CCAPxL 输入一个 8 位的值指定第一个 PWM 波形的占空比。在 CCAPxH 输入一个 8 位的值会指定第二个 PWM 波形的占空比。设置定时器/计数器运行控制位(CR.CR)启动 PCA 定时器/计数器。

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	工作方式
X	1	0	0	0	0	X	用CCPn 的正沿触发捕捉
X	0	1	0	0	0	X	用CCPn 的负沿触发捕捉
X	1	1	0	0	0	X	用CCPn 的跳变触发捕捉
1	0	0	1	0	0	X	软件定时器
1	0	0	1	1	0	X	高速输出
1	0	0	0	0	1	0	8 位脉冲宽度调制器

Table 15-1 PCA 比较/捕获功能模块设置



### 15.3 PCA 模块与其他模块互连及控制

#### 15.3-1 ECI 互连

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

#### 15.3-2 PCACAP0

通道 0 的捕获输入可以是：

- (1) 外部的 IO MUX 的输入端口，外部 UART 的 RX 的 MUX 输入
- (2) 内部的 VC 的比较滤波后的输出 UART 选择控制在 PCA 捕获通道控制寄存器 SYSCON\_PCACR 中，VC 输出控制寄存器在 VC 控制模块。

#### 15.3-3 PCACAP1/2/3/4

通道 1/2/3/4 的捕获输入可以是：

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

UART 选择控制 UART 选择控制在 PCA 捕获通道控制寄存器 SYSCON\_PCACR 中



#### 15.4 PCA 寄存器列表

基址地址 0x4000 1800

地址偏移	名称	描述	复位值
0x000	PCA_CR	PCA 控制寄存器	0x0000 0000
0x004	PCA_MOD	PCA 模式寄存器	0x0000 0000
0x008	PCA_CNT	PCA 计数寄存器	0x0000 0000
0x00C	PCA_INTCLR	PCA 中断清除寄存器	0x0000 009F
0x010	PCA_CCAPM0	PCA 比较/捕获模块 0 模式寄存器	0x0000 0000
0x014	PCA_CCAPM1	PCA 比较/捕获模块 1 模式寄存器	0x0000 0000
0x018	PCA_CCAPM2	PCA 比较/捕获模块 2 模式寄存器	0x0000 0000
0x01C	PCA_CCAPM3	PCA 比较/捕获模块 3 模式寄存器	0x0000 0000
0x020	PCA_CCAPM3	PCA 比较/捕获模块 4 模式寄存器	0x0000 0000
0x030	PCA_CCAP0L	PCA 比较/捕获模块 0 低 8 位寄存器	0x0000 0000
0x034	PCA_CCAP0H	PCA 比较/捕获模块 0 高 8 位寄存器	0x0000 0000
0x038	PCA_CCAP1L	PCA 比较/捕获模块 1 低 8 位寄存器	0x0000 0000
0x03C	PCA_CCAP1H	PCA 比较/捕获模块 1 高 8 位寄存器	0x0000 0000
0x040	PCA_CCAP2L	PCA 比较/捕获模块 2 低 8 位寄存器	0x0000 0000
0x044	PCA_CCAP2H	PCA 比较/捕获模块 2 高 8 位寄存器	0x0000 0000
0x048	PCA_CCAP3L	PCA 比较/捕获模块 3 低 8 位寄存器	0x0000 0000
0x04C	PCA_CCAP3H	PCA 比较/捕获模块 3 高 8 位寄存器	0x0000 0000
0x050	PCA_CCAP4L	PCA 比较/捕获模块 4 低 8 位寄存器	0x0000 0000
0x054	PCA_CCAP4H	PCA 比较/捕获模块 4 高 8 位寄存器	0x0000 0000
0x058	PCA_CCAPO	PCA PWM 与高速输出标志寄存器	0x0000 0000
0x05C	PCA_POCR	PCA 端子输出控制寄存器	0x0000 0000
0x060	PCA_CCAP0	PCA 比较/捕获模块 0 的 16 位寄存器	0x0000 0000
0x064	PCA_CCAP1	PCA 比较/捕获模块 1 的 16 位寄存器	0x0000 0000
0x068	PCA_CCAP2	PCA 比较/捕获模块 2 的 16 位寄存器	0x0000 0000
0x06C	PCA_CCAP3	PCA 比较/捕获模块 3 的 16 位寄存器	0x0000 0000
0x070	PCA_CCAP4	PCA 比较/捕获模块 4 的 16 位寄存器	0x0000 0000

Table 15-2 PCA 寄存器列表和复位值



## 15.5 寄存器说明

### 15.5.1 控制寄存器(PCA\_CR)

地址偏移: 0x000 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留位	0x00	-
7	CF	PCA 计数器溢出标志: 0: 无溢出 1: 发生计数器溢出 当PCA 计数溢出时, CF 由硬件置位 当MOD 寄存器的CFIE 位为 1, CF 标志可以产生中断	0	RO
6	CR	PCA 计数器运行控制位 0: 关闭PCA 计数器计数 1: 启动PCA 计数器计数	0	R/W
5	-	保留位	0	-
4	CCF4	PCA 计数器模块 4 比较/捕获标志 位: 当出现匹配或捕获时, 该位由硬件置位 当CCAPM3.CCIE 置位时, 这个标志位会产生一个PCA 中断 0: 无匹配或捕获 1: 匹配或捕获发生	0	RO
3	CCF3	PCA 计数器模块 3 比较/捕获标志 位: 当出现匹配或捕获时, 该位由硬件置位 当CCAPM3.CCIE 置位时, 这个标志位会产生一个PCA 中断 0: 无匹配或捕获 1: 匹配或捕获发生	0	RO
2	CCF2	PCA 计数器模块 2 比较/捕获标志 位: 当出现匹配或捕获时, 该位由硬件置位 当CCAPM2.CCIE 置位时, 这个标志位会产生一个PCA 中断 0: 无匹配或捕获 1: 匹配或捕获发生	0	RO



1	CCF1	PCA 计数器模块 1 比较/捕获标志 位：当出现匹配或捕获时，该位由硬件置位 当CCAPM1.CCIE 置位时，这个标志位会产生一个PCA 中断 0: 无匹配或捕获 1: 匹配或捕获发生	0	RO
0	CCF0	PCA 计数器模块 0 比较/捕获标志 位：当出现匹配或捕获时，该位由硬件置位 当CCAPM0.CCIE 置位时，这个标志位会产生一个PCA 中断 0: 无匹配或捕获 1: 匹配或捕获发生	0	RO

ZBT CONFIDENTIAL



### 15.5-2 模式寄存器(PCA\_MOD)

地址偏移: 0x004 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:8	-	保留位	0x0	-
7	CIDL	空闲模式IDLE 下, PCA 是否停止工作 0: 休眠模式(Sleep)下, PCA 继续工作 1: 休眠模式(Sleep)下, PCA 停止工作	0	R/W
6:4	-	保留位	0x0	-
3:1	CPS	时钟分频选择及时钟源选择 000: PCLK/32 001: PCLK/16 010: PCLK/8 011: PCLK/4 100: PCLK/2 101: Timer0 overflow 110: Timer1 overflow 111: ECI 外部时钟, 时钟PCLK 四分频采样	0	R/W
0	CFIE	PCA 计数器中断使能控制信号 0: 关闭中断 1: 使能中断	0	R/W



### 15.5-3 计数寄存器(PCA\_CNT)

地址偏移: 0x008 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:16	-	保留位	0x0	-
15:0	CNT	定时器计数器的值: 只有在PCA 停止状态, CNT 才可以写入, 否则写入无效	0x0	R/W



### 15.5-4 中断清除寄存器(PCA\_INTCLR)

地址偏移: 0x00C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
保留																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留								CF	保留		CCF4	CCF3	CCF2	CCF1	CCF0		
								WO	WO		WO	WO	WO	WO	WO		
位	标记	功能描述												复位值	读写		
31:8	-	保留位												0x0	-		
7	CF	PCA 计数器溢出标志清除: 0: 写 0 无效 1: 软件写 1 清零对应的标志												0	WO		
6:5	-	保留位												0x0	-		
4	CCF4	PCA 计数器模块 4 比较/捕获标志位清除: 0: 写 0 无效 1: 软件写 1 清零对应的标志												0	WO		
3	CCF3	PCA 计数器模块 3 比较/捕获标志位清除: 0: 写 0 无效 1: 软件写 1 清零对应的标志												0	WO		
2	CCF2	PCA 计数器模块 2 比较/捕获标志位清除: 0: 写 0 无效 1: 软件写 1 清零对应的标志												0	WO		
1	CCF1	PCA 计数器模块 1 比较/捕获标志位清除: 0: 写 0 无效 1: 软件写 1 清零对应的标志												0	WO		
0	CCF0	PCA 计数器模块 0 比较/捕获标志位清除: 0: 写 0 无效 1: 软件写 1 清零对应的标志												0	WO		



### 15.5.5 比较捕获模式寄存器(PCA\_CCAPM0~4)

地址偏移:

1. CCAPM0: 0x010
2. CCAPM1: 0x014
3. CCAPM2: 0x018
4. CCAPM3: 0x01C
5. CCAPM3: 0x020

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								E COM	C APP	C APN	M AT	T OG	P WM	C CIE	
R/W								R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:7	-	保留位	0x00	-
6	E COM	允许比较器功能控制位: 0: 禁止比较强功能 1: 允许比较器功能 当PCA 用于软件计数器, 高速输出和PWM 模式时, 要置位 E COM; 当写CCAMPHx 或CCAMPx 寄存器会自动置位E COM; 当写CCAMPLx 寄存器会自动清除E COM 位;	0	R/W
5	C APP	正沿捕获控制位: 0: 禁止上升沿捕获 1: 允许上升沿捕获	0	R/W
4	C APN	负沿捕获控制位 0: 禁止下降沿捕获 1: 允许下降沿捕获	0	R/W
3	M AT	允许匹配控制位: 0: 禁止匹配功能; 1: PCA 计数值与模块的比较/捕获寄存器的值一旦匹配, 将置位 CR 寄存器的中断标志CCFx	0	R/W
2	T OG	翻转控制位: 0: 禁止翻转功能 1: 工作在PCA 高速输出模式时, PCA 计数器的值与模块的比较/捕获寄存器的值一旦匹配, PCA_CHx 引脚翻转	0	R/W



1	PWM	脉宽调制控制位: 0: 禁止PWM 脉宽调制功能 1: 允许PCA_CHx 引脚作为PWM 输出 只有CCAPMx[6:0]=100 0010' b 时, PWM 功能才有效	0	R/W
0	CCIE	PCA 使能中断: 0: PCA 比较/捕获功能中断禁止 1: 使能比较/捕获中断	0	R/W

### 15.5-6 比较捕获数据寄存器低 8 位(PCA\_CCAP0~4L)

地址偏移:

1. CCAP0L: 0x030
2. CCAP1L: 0x038
3. CCAP2L: 0x040
4. CCAP3L: 0x048
5. CCAP4L: 0x050

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留位	0x0	-
7:0	CCAPx[7:0]	比较/捕获模式低 8 位寄存器: 当 PCA 模式用于比较/捕获模式时, 用于保存 16 位捕获计数值的低8 位; 写CCAPxH 寄存器会自动清除寄存器CCAPMx 的ECOM 位。当 PCA 模式用于 PWM 模式时, 用于控制输出占空比比较寄存器, 在 PWM 模式, 计数器的低 8 位的值小于CCAPx[7:0]的值 PWM 输出低电平, 否则PWM 输出高电平。	0x0	R/W



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBIT CONFIDENTIAL



### 15.5.7 比较捕获数据寄存器高 8 位(PCA\_CCAP0~4H)

地址偏移:

1. CCAP0H: 0x034
2. CCAP1H: 0x03C
3. CCAP2H: 0x044
4. CCAP3H: 0x04C
5. CCAP4H: 0x054

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留位	0x0	-
7:0	CCAPx[15:8] ]	比较/捕获模式高 8 位寄存器: 当 PCA 模式用于比较/捕获模式时，用于保存 16 位捕获计数值的高 8 位，写 CCAPxH 寄存器会自动置位寄存器 CCAPMx 的 ECOM 位。 当PCA 模式用于 PWM 模式时，用于控制输出占空比装载寄存器， 在计数器低 8 位溢出时，装载寄存器会自动更新到 PWM 比较寄存器	0x0	R/W



### 15.5.8 比较高速输出标志寄存器(PCA\_CCAPO)

地址偏移: 0x058 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
								CCA PO4	CCA PO3	CCA PO2	CCA PO1	CCA PO0			
								R/W	R/W	R/W	R/W	R/W			

位	标记	功能描述	复位值	读写
31:5	-	保留位	0x0	-
4	CCAPO4	高速模式比较模块 4 的输出值 0: 输出 0 1: 输出 1	0	R/W
3	CCAPO3	高速模式比较模块 3 的输出值 0: 输出 0 1: 输出 1	0	R/W
2	CCAPO2	高速模式比较模块 2 的输出值 0: 输出 0 1: 输出 1	0	R/W
1	CCAPO1	高速模式比较模块 1 的输出值 0: 输出 0 1: 输出 1	0	R/W
0	CCAPO0	高速模式比较模块 0 的输出值 0: 输出 0 1: 输出 1	0	R/W



### 15.5.9 端子输出控制寄存器(PCA\_POCR)

地址偏移: 0x05C 复位值: 0x0000 0000

位	标记	功能描述	复位值	读写
31:13	-	保留位	0x00	-
12	POINV4	比较通道 4 的输出极性反转 0 = 禁止 PWM0 输出极性反转 1 = 使能 PWM0 输出极性反转	0	R/W
11	POINV3	比较通道 3 的输出极性反转 0 = 禁止 PWM0 输出极性反转 1 = 使能 PWM0 输出极性反转	0	R/W
10	POINV2	比较通道 2 的输出极性反转 0 = 禁止 PWM0 输出极性反转 1 = 使能 PWM0 输出极性反转	0	R/W
9	POINV1	比较通道 1 的输出极性反转 0 = 禁止 PWM0 输出极性反转 1 = 使能 PWM0 输出极性反转	0	R/W
8	POINV0	比较通道 0 的输出极性反转 0 = 禁止 PWM0 输出极性反转 1 = 使能 PWM0 输出极性反转	0	R/W
7:5	-	保留位	0x0	-
4	POE4	比较通道 4 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
3	POE3	比较通道 3 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
2	POE2	比较通道 2 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
1	POE1	比较通道 1 的输出使能 0: 输出禁止 1: 输出使能	0	R/W



0	POE0	比较通道 0 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
---	------	------------------------------------	---	-----

ZBT CONFIDENTIAL

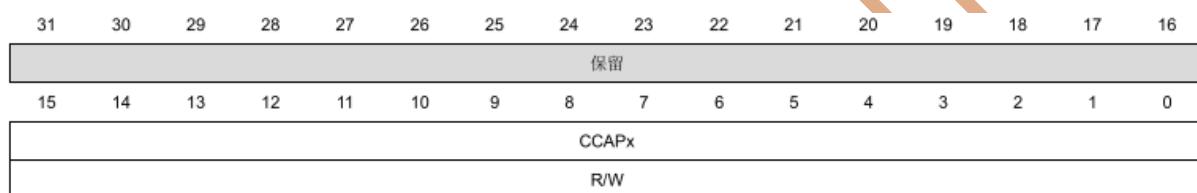


### 15.5.10 比较捕获 16 位寄存器(PCA\_CCAP0~4)

地址偏移:

1. CCAP0: 0x060
2. CCAP1: 0x064
3. CCAP2: 0x068
4. CCAP3: 0x06C
5. CCAP4: 0x070

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:16	-	保留位	0x0	-
15:0	CCAPx	比较/捕获模式 16 位寄存器: 当 PCA 用于比较/捕获模式时，用于保存 16 位捕获计数值；写 CCAPx 寄存器会置位寄存器 CCAPMx 的 ECOM 位。写 CCAPx 寄存器相当于写 CCAPxL 及 CCAPxH 这两个 8 位寄存器。在比较/捕获模式下，可以直接读写这个寄存器；在 PWM 模式下，使用 CCAPxL 及 CCAPxH 寄存器。	0x0	R/W



## 【16】基础定时器 Base Timer(TIM10/TIM11)

### 16.1 Base Timer 简介

基础定时器 Base Timer 包含两个定时器 TIM10/11。TIM10/11 功能完全相同。TIM10/11 是同步定时 / 计数器，可以作为 16/32 位自动重装载功能的定时/计数器，也可以作为 16/32 位无重载功能的定时 / 计数器。TIM10/11 可以对外部脉冲进行计数或者实现系统定时。

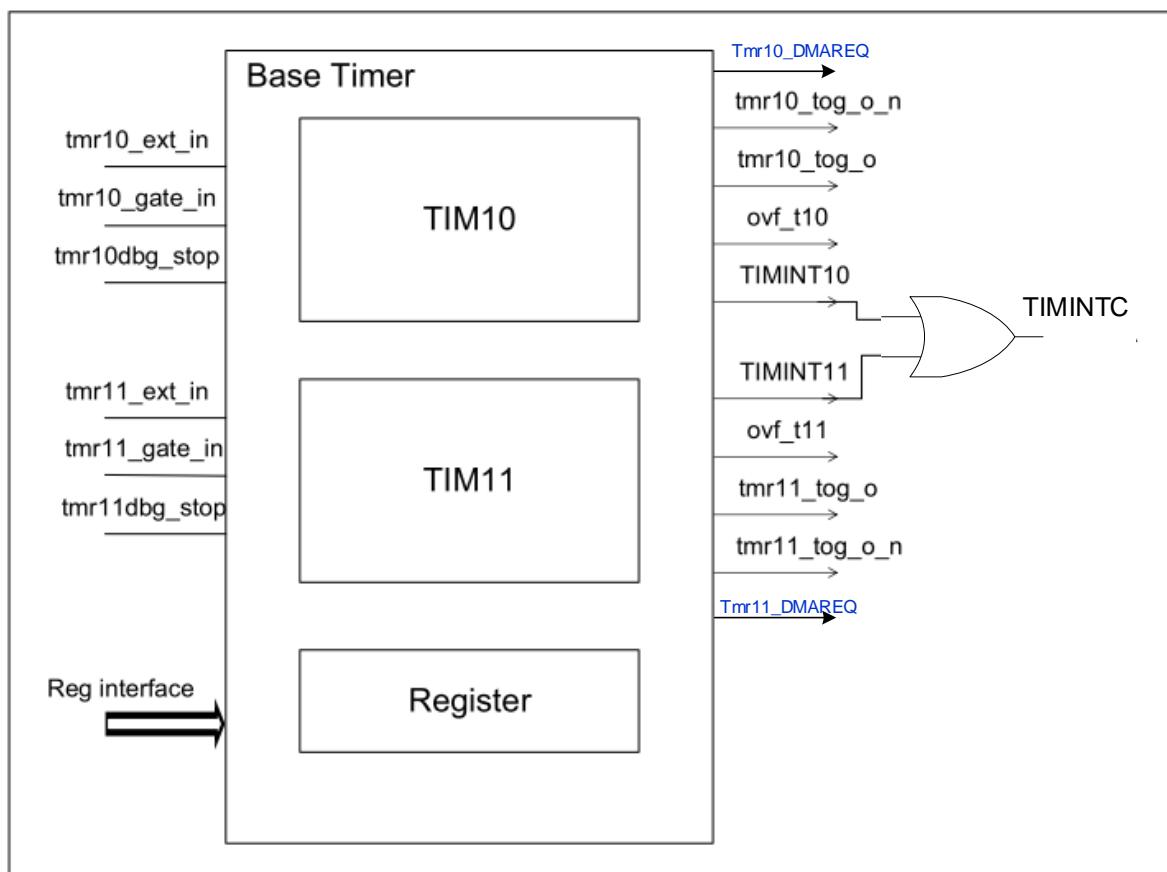


Figure 16-1 Base Timer 框图



## 16.2 Base Timer 功能描述

TIM10/11 每个定时/计数器都有独立的控制启动信号、外部输入时钟和门控信号。

当 TIM10/11 使用 EXT、GATE 来进行计数功能时，EXT 用于计数器的外部输入时钟信号，GATE 用于有效电平计数使能信号。当门控功能使能后，当且仅当外部输入 GATE 电平有效时，计数器才会计数，否则计数器处于保持状态。门控使能使用 TIMx\_CR.GATE\_EN 控制。默认门控功能关闭。门控电平选择使用 TIMx\_CR.GATE\_P 控制。默认高电平为门控有效电平；设置 TIMx\_CR.GATE\_P 为 1 后，门控低电平为有效电平。

当 TIM10/11 使用 PCLK、GATE 来进行定时功能时，PCLK 用于定时器的内部输入时钟信号，GATE 可用于有效电平定时使能信号。当门控功使能后，当且仅当外部输入 GATE 电平有效时，定时器才会计数，否则定时器处于定时计数器停止状态。门控使能使用 TIMx\_CR.GATE\_EN 控制。默认门控功能关闭。门控电平选择使用 TIMx\_CR.GATE\_P 控制。默认高电平为门控有效电平；设置为 1 后门控有效电平是低电平。定时功能可以配置预除频。TIMx\_CR.TMR\_PRE 控制分频比。

TMR_PRE [2:0]	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	128

TIM10/11 支持定时/计数器两种功能，可通过设置定时器控制寄存器(TIMx\_CR)中 CT\_SEL 进行配置。

每种功能支持 2 种模式，模式 1 为 16/32 位自由计数模式，模式 2 是 16/32 位重载模式。

### ● 在模式 1 自由计数模式

计数到最大值(16 位 Max=0xFFFF，32 位最大值为 0xFFFFFFFF)溢出后产生中断，定时/计数器清零，然后继续计数。

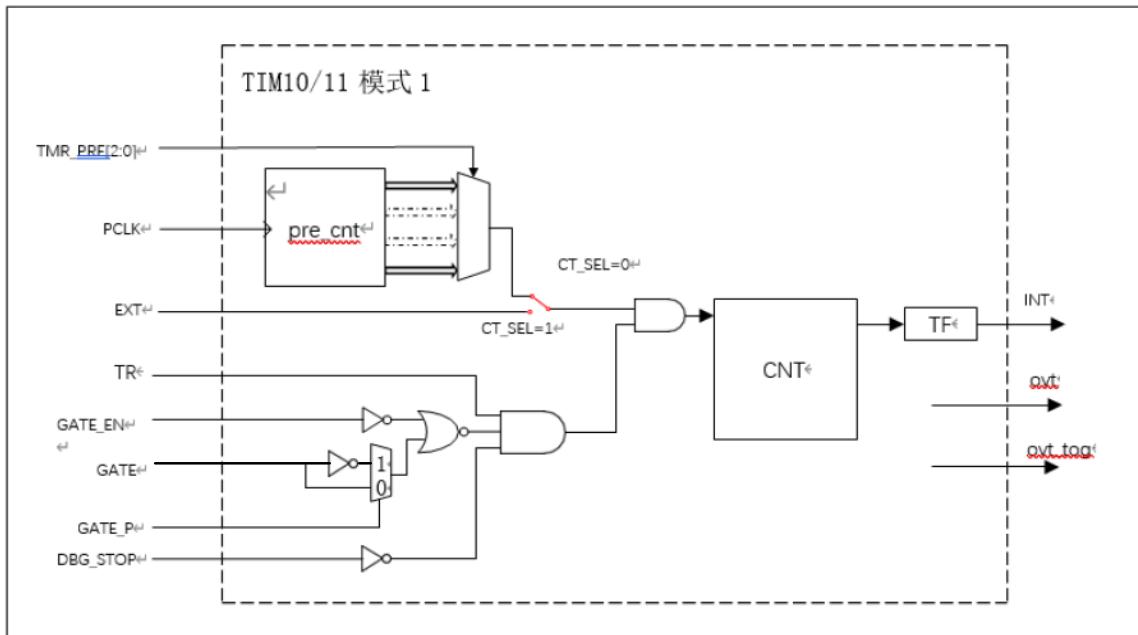


Figure 16-2 Timer 模式 1 框图

- 模式 2 重载模式

重载模式，计数到最大值后溢出，产生中断，定时/计数器的值被装载为 BGLOAD 的值，然后继续向上计数。在重载模式下，定时时间设的小的情况下需要考虑软件处理速度，否则中断会来不及处理而造成中断丢失。

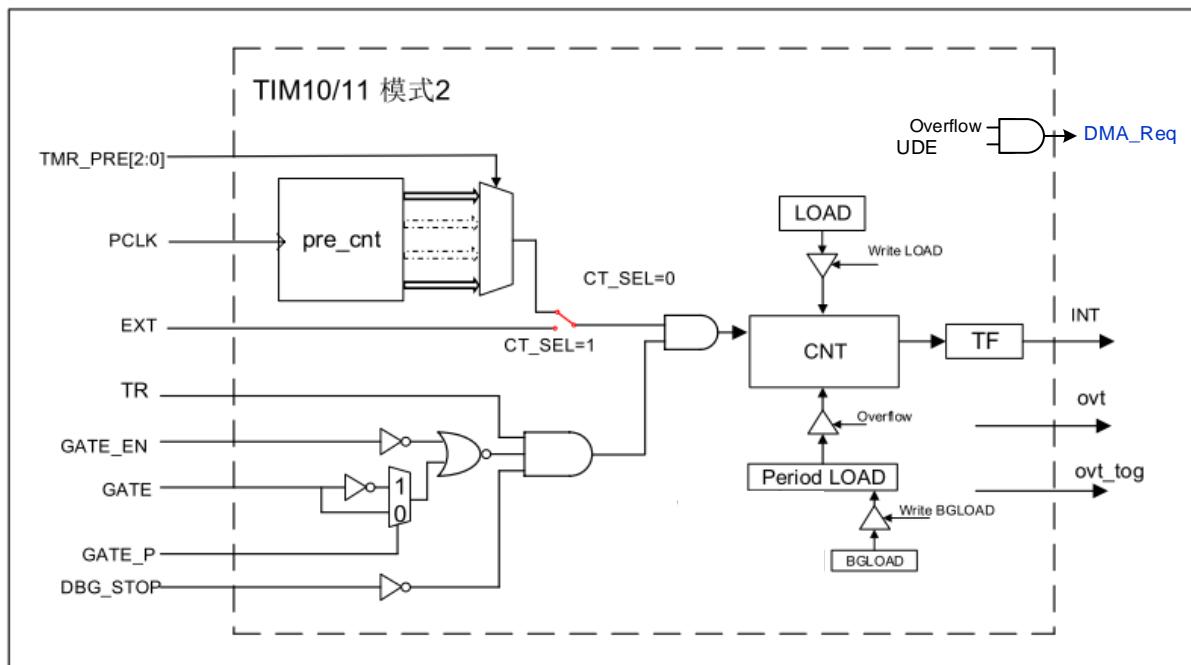


Figure 16-3 Timer 模式 2 框图

当设置对应定时器 `TIMx_CR.TR` 为 1 后定时器开始运行。模式 1，启动后从寄存器设定的初值开始计数，计数到最大后产生溢出中断，然后从 0 继续计数。模式 2，启动后从寄存器初值 `CNT` 开始向上计数，计数到最大值后产生中断后重载寄存器 `BGLOAD` 的值到计数器 `CNT` 中，继续向上计数。不论是自由计数模式，还是重载模式，只要是写 `LOAD` 值，都会立即更新定时/计数器的值，然后继续向上计数。

### 16.2-1 计数功能

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

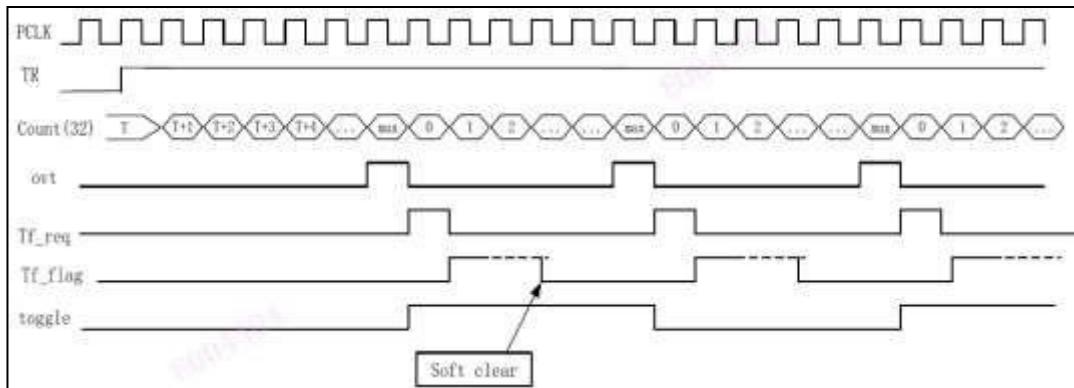


Figure 16-4 32 位模式 1 时序图(max=0xFFFF FFFF)

### 16.2-2 定时功能

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

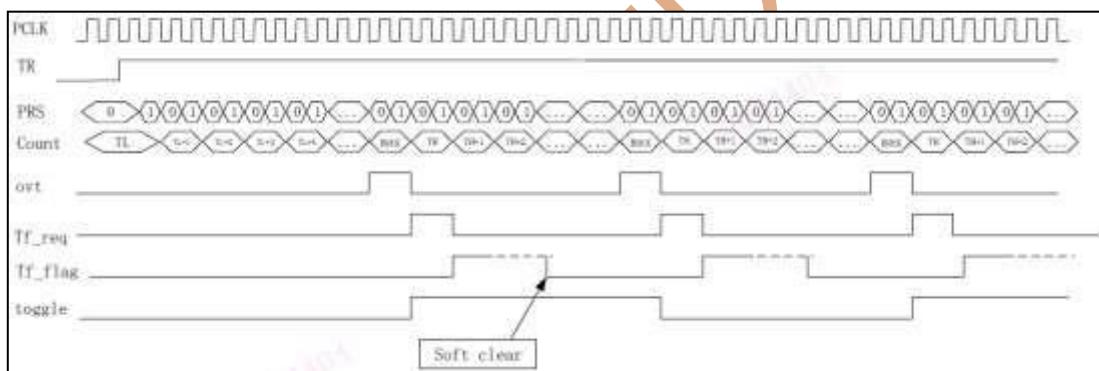


Figure 16-5 32 位模式 2 时序图(PCLK 二分频, max=0xFFFF FFFF)

### 16.2-3 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。TIMx\_CR.TOG\_EN 为 1 时，TOG、TOGN 输出反向。设置 TIMx\_CR.TOG\_EN 为 0 可以同时设置端口 TOG、TOGN 输出为 0。在计数时钟为 4M 情况下 Buzzer 输出不同频率的 Timer 重载模式配置如下（16 位 Max=0xFFFF）：

Buzzer 频率	计数周期	计数值	重载值	CNT 初始值	LOAD 重载值
1KHz	0.5ms	2000	63536	0xF830	0xF830
2KHz	0.25ms	1000	64536	0xFC18	0xFC18
4KHz	0.125ms	500	65036	0xFE0C	0xFE0C



### 16.3 Base Timer 互连

#### 16.3-1 GATE 互连

GATE 输入可以从端口直接输入，也可以输入 UART/LPUART 的 RX 信号；还可以配置为 VC 的输出作为 GATE 信号。TIM10/11 的 GATE 都可以配置。

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

配置选择 RX 输入在 SYSCON\_PORTCR 寄存器控制，VC 控制在 VC\_OUTCFG 寄存器控制。端口选择时，UART/LPUART 输入选择与 VC 输出选择作为门控输入只能选择一个有效。VC 的输出选择优先级最高。

#### 16.3-2 Toggle 输出互连

TIM10 的翻转输出 tmr10\_tog\_o 到内部模块 UART0，控制 UART0 的波特率；TIM11 的翻转输出 tmr11\_tog\_o 到内部模块 UART1，控制 UART1 的波特率；TIM10/11 的翻转输出还输出到端口上，可以驱动 Buzzer 实现蜂鸣器的控制。



#### 16.4 Base Timer(TIM10/TIM11) 寄存器列表

x=10 或 11;

	基地址	描述
TIM10	0x4000 1000	TIM10 基地址
TIM11	0x4000 1400	TIM11 基地址

地址偏移	名称	描述	复位值
0x00	TIMx_CR	控制寄存器	0x0000 0000
0x04	TIMx_LOAD	32 位立即重载寄存器	0x0000 0000
0x08	TIMx_CNT	读计数器寄存器, 只读	0x0000 0000
0x0C	TIMx_RAWINTS_R	读原始中断寄存器,	0x0000 0000
0x10	TIMx_MSKINTS_R	读中断寄存器	0x0000 0000
0x14	TIMx_INTCLR	中断清除寄存器	0x0000 0000
0x18	TIMx_BGLOAD	32 位周期重载寄存器	0x0000 0000

Table 16-1 PCA 寄存器列表和复位



## 16.5 Base Timer 寄存器说明

### 16.5-1 控制寄存器(TIMx\_CR)

地址偏移: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留								保留								
1 5	1 4	1 3	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留				UDE	TE_P	GA	GAT_EN	RN_EN	TOG_SEL	CT	TR	MODE	INTEN	TMR_SIZE	SHOT	ONE
r/w				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位	符号	描述	复位值	读写
31:13	-	保留位, 读为 0	0x0	-
12	UDE	DAM请求始能控制, 写 1 后使能DAM请求  DMA更新请求  0 : 禁用DMA更新请求 1: 使能DMA更新请求	0	R/W
11	GATE_P	端口GATE 极性控制, 默认高电平 GATE 有效, 设置为 1 后低电平有效 0: 高电平有效 1: 低电平有效	0	R/W
10	GATE_EN	定时器门控 0: 无门控, TR=1 时定时器工作; 1: 只有端口GATE 有效并且TR=1 时才工作;	0	R/W
9	TOG_EN	TOG 输出使能 0: TOG、TOGN 同时输出 0 1: TOG、TOGN 输出相位相反的信号。可供buzzer 使用。	0	R/W
8	CT_SEL	计数器/定时器功能选择 0: 定时器功能, 定时器由PCLK 来进行计数。 1: 计数器功能, 计数器由外部输入的下降沿进行计数。外	0	R/W



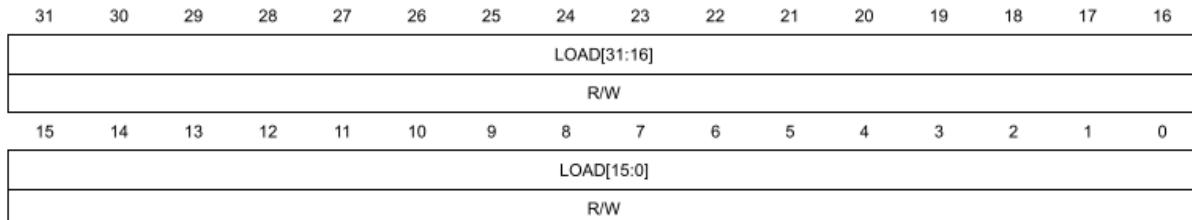
位	符号	描述	复位值	读写
		外部输入由PCLK 采样，外部输入时钟频率要低于 1/2 采样时钟。		
7	TR	定时器运行控制 0: 定时器停止 1: 定时器运行	0	R/W
6	MODE	定时器工作模式 0: 模式 1 计数器/定时器 1: 模式 2 自动重装载计数器/定时器	0	R/W
5	INTEN	中断使能控制，写 1 后使能中断	0	R/W
4	TMR_SIZE	TimerSize=0: max count value=0xFFFF; TimerSize=1: max count value=0xFFFFFFFF;	0	R/W
3	ONESHOT	计数器运行一次使能 0: 重复模式 1: oneshot 模式	0	R/W
2:0	TMR_PRSC[2:0]	TIM 预除频选择。 000: 分频数 1; 001: 分频数 2; 010: 分频数 4; 011: 分频数 8; 100: 分频数 16; 101: 分频数 32; 110: 分频数 64; 111: 分频数 128	0x0	R/W



### 16.5-2 立即重载寄存器(TIMx\_LOAD)

地址偏移: 0x04

复位值: 0x0000 0000

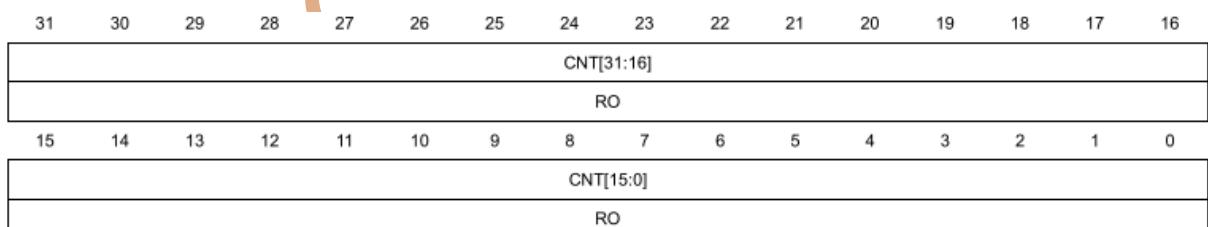


位	符号	描述	复位值	读写
31:0	LOAD[31:0]	立即重载寄存器 写此寄存器，会立即更新计数器寄存器 CNT 的值 注：读 TIMx_LOAD 和 TIMx_BGLOAD，可以读到最近更新的 LOAD 或者 BGLOAD 寄存器的值	0x0	R/ W

### 16.5-3 计数器寄存器(TIMx\_CNT)

地址偏移: 0x08

复位值: 0x0000 0000



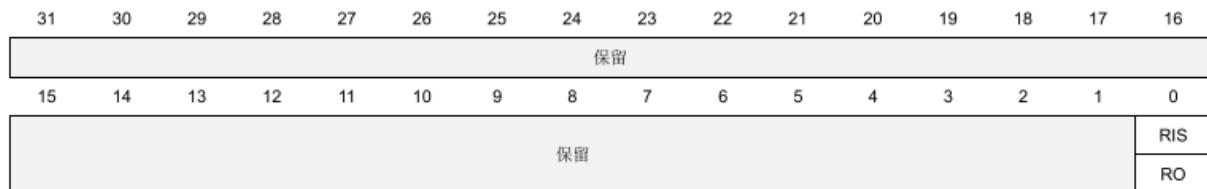
位	符号	描述	复位值	读写
31:0	CNT[31:0]	计数器寄存器	0x0	RO



#### 16.5-4 原始中断状态寄存器(TIMx\_RAWINTSR)

地址偏移: 0x0c

复位值: 0x0000 0000



位	符号	描述	复位值	读写
31:1	-	保留位, 读为 0	0x0	-
0	RIS	中断标志, 硬件置位;不论IntEnable=0 or 1, 都可以读中断	0	RO

#### 16.5-5 中断标志寄存器(TIMx\_MSKINTSR)

地址偏移: 0x10

复位值: 0x0000 0000



位	符号	描述	复位值	读写
31:1	保留	保留位, 读为 0	0x0	-
0	TF	INTEN=1,才可以读中断寄存器	0	RO



### 16.5-6 中断清除寄存器(TIMx\_INTCLR)

地址偏移: 0x14

复位值: 0x0000 0000

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

位	符号	描述	复位值	读写
31:1	保留	保留位, 读为 0	0x0	-
0	INTCLR	中断标志清除, 写 1 清除, 写 0 无效	0	WO

### 16.5-7 周期重载寄存器(TIMx\_BGLOAD)

地址偏移: 0x18

复位值: 0x0000 0000

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

位	符号	描述	复位值	读写
31:0	BGLOAD[31:0] ]	BackGround 周期重载寄存器, 写此寄存器不会立即更新计数器寄存器 CNT 的值。只有当 CNT 值溢出时, 才会重新装载 BGLOAD 的值到 CNT 寄存器里面。 注: 读 TIMx_LOAD 和 TIMx_BGLOAD, 可以读到最近更新的 LOAD 或者BGLOAD 寄存器的值	0x0	R/W



## 【17】低功耗定时器(LPTIM)

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

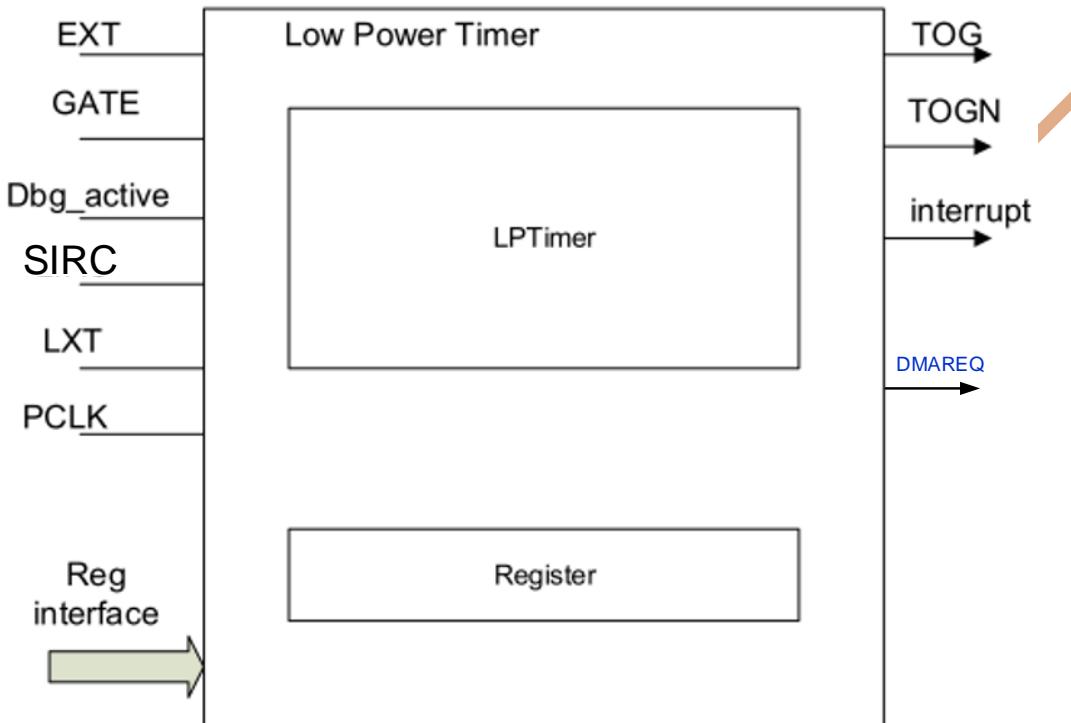


Figure 17-1 LPTIMER 结构框图

### 17.1 LPTIM 功能描述

LPTIM 具有独立的控制启动信号、外部输入时钟和门控信号。

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

LPTIM 的定时器支持两种工作模式，通过设置定时器控制寄存器(LPTIM\_CR)中 MODE 选择工作模式。

- 模式 1 为 16 位自由计数模式，计数器从 LPTIM\_LOAD 设定的值开始计数，溢出后计数值从 0x0000 开始重新计数。

- 模式 2 为 16 位重载模式，LPTIM 启动时会自动装载重载寄存器 LPTIM\_LOAD 的值到计数器中，当溢出后会自动装载 Period LOAD 的值到计数器中。用户写 LPTIM\_LOAD 或者 LPTIM\_BGLOAD 都会更新 Period LOAD 的值，Period LOAD 的值为用户最近一次跟新 LPTIM\_LOAD 或 LPTIM\_BGLOAD 的值。

LPTIM 可选三种时钟作为定时器时钟，通过控制寄存器 LPTIM\_CR.TCK\_SEL 来选择。默认选择



PCLK。时钟选择如表：

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

当选择相应的时钟源，然后设定 TCK\_EN 为 1 后可以打开计数器的计数时钟源。

当时钟源选择并打开后，设置对应定时器的 TIM\_RUN 为 1 后定时器开始运行。

对于模式 1 和模式 2，如果设置 LOAD 值，任何时刻，计数值会被立即更新为 LOAD 值，计数器从 LOAD 值开始重新计数。设置 LOAD 值的优先级高于其他计数器被更新的优先级。

在模式 2，设定 BGLOAD 的值，只有在计数器发生溢出后，设定值才会被更新到计数器。

模式 1：如果没有设置 LOAD 值，计数器从 0 开始计数，计数到最大 0xFFFF 后产生溢出中断。计数器计数到最大 0xFFFF 后，计数器再次从 0 开始计数。

模式 2：如果没有设置 LOAD 值，计数器从 0 开始计数，计数到最大 0xFFFF 后产生溢出中断。计数器计数到最大 0xFFFF 后，计数值会被更新为 Period LOAD 的值，然后向上计数。

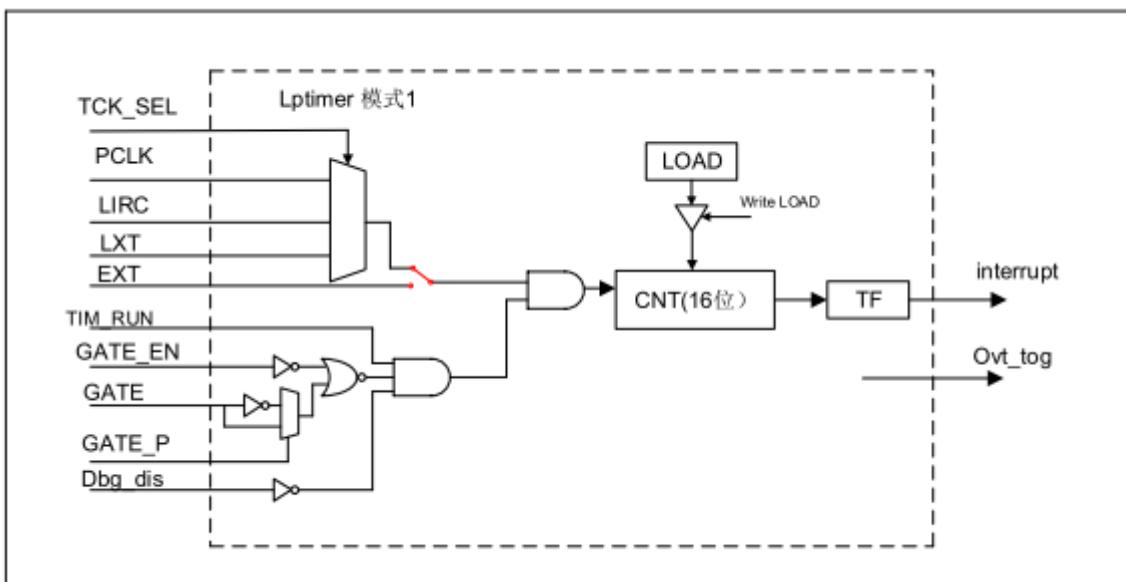


Figure 17-2 LPTIMER 模式 1

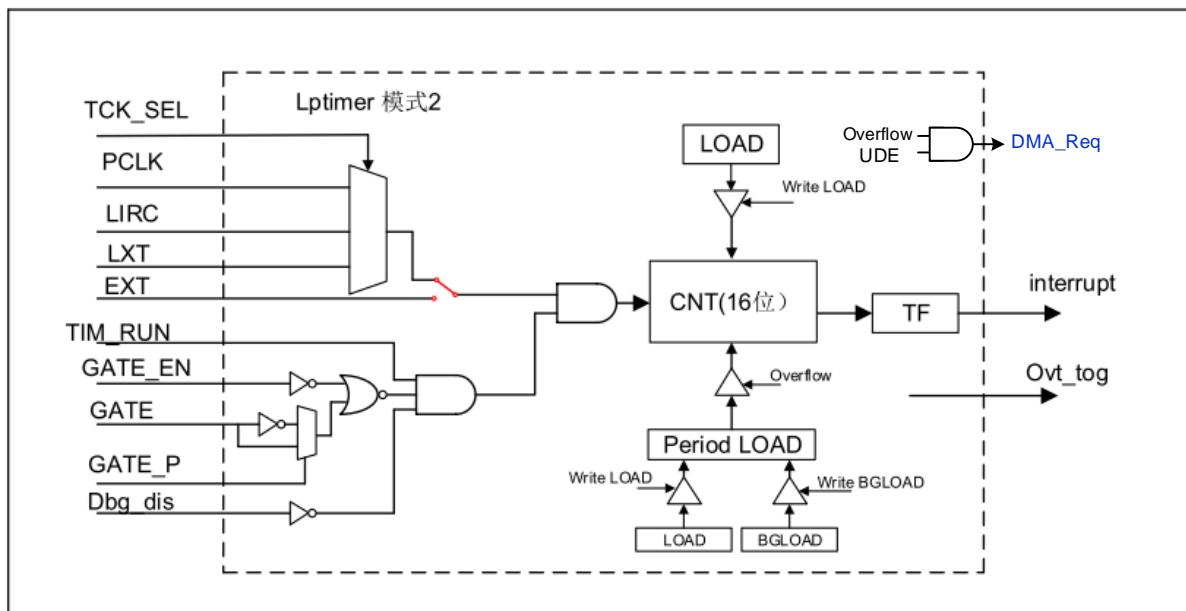


Figure 17-3 LPTIMER 模式 2

### 17.1-1 计数功能

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

### 17.1-2 定时功能

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



## 17.2 LPTIM 互连

### 17.2-1 GATE 互联

GATE 输入可以从端口直接输入，也可以输入 UART 的 RX 信号；

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

配置选择 RX 输入在 SYSCON\_PORTCR 寄存器控制，VC 控制在 VC 控制寄存器控制。

### 17.2-2 EXT 互联

EXT 输入可以从端口直接输入，也配置为 VC 的输入作为 EXT 信号。

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

### 17.2-3 TOGGLE 输出互联

LPTIM 的翻转输出到端口上，可以驱动 BUZZER 实现蜂鸣器的控制。



### 17.3 LPTIM 寄存器列表

基址: 0x4000 F800

偏移地址	名称	描述	复位值
0x00	LPTIM_CNTVAL	LPTIM 计数值只读寄存器	0x0000 0000
0x04	LPTIM_CR	LPTIM 控制寄存器	0x0000 0000
0x08	LPTIM_LOAD	LPTIM 立即重载寄存器	0x0000 0000
0x0C	LPTIM_INTSR	LPTIM 中断寄存器	0x0000 0000
0x10	LPTIM_INTCLR	LPTIM 中断清除寄存器	0x0000 0000
0x14	LPTIM_BGLOAD	LPTIM 周期重载寄存器	0x0000 0000

Table 17-1 LPTIM 寄存器列表和复位值



#### 17.4 LPTIM 寄存器说明

##### 17.4-1 LPTIM 计数值只读寄存器(LPTIM\_CNTVAL)

地址偏移: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPT_CNT[15:0]															
RO															

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15:0	LPT_CNT[15:0]	计数值只读寄存器	0x0	RO



### 17.4-2 LPTIM 控制寄存器(LPTIM\_CR)

地址偏移: 0x04 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															WT_FLAG
保留															RO
1	1	1	1	1	10	9	8	7	6	5	4	3	2	1	0
5	4	3	2	1	UDE	TCK_EN	INT_EN	GATE_P	GATE	TCK_SEL[1:0]	TOG_EN	CT_SEL	MODE	TIER_RUN	
					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:17	-	保留	0x0	-
16	WT_FLAG	WT 写同步标志 0: 同步完成, 此时可更改LOAD/BGLOAD 1: 正在同步, 此时写LOAD/BGLOAD 无效	0x0	RO
15:11		保留	0x0	-
10	UDE	DAM请求始能控制.写 1 后使能DAM请求  DMA更新请求  0 : 禁用DMA更新请求 1: 使能DMA更新请求	0	R/W
9	TCK_EN	LPTIM 计数时钟使能 0: LPTIM 计数时钟关闭 1: LPTIM 计数时钟使能  只有计数时钟使能后才能进行LOAD/BGLOAD 的配置	0	R/W
8	INT_EN	中断使能控制, 写 1 后使能中断	0	R/W



位	标记	功能描述	复位值	读写
7	GATE_P	输入 GATE 的有效极性 默认高电平 GATE 有效，设置为 1 后低电平有效	0	R/W
6	GATE_EN	定时器门控 0: 无门控 1: 有门控	0	R/W
5:4	TCK_SEL[1:0]	LPTIM 时钟选择 00: PCLK; 10: LXT; 11: SIRC	0x0	R/W
3	TOG_EN	TOG 输出使能 0: TOG, TOGN 同时输出 0 1: TOG, TOGN 输出相位相反的信号。可供 BUZZER 使用。	0	R/W
2	CT_SEL	计数器/定时器功能选择 0: 定时器功能，定时器使用 TCK_SEL 选择的时钟进行计数。 1: 计数器功能，计数器使用外部输入的下降沿进行计数。采样时钟使用 TCK_SEL 选择的时钟，外部输入时钟要低于 1/2 采样时钟。	0	R/W
1	MODE	定时器工作模式 0: 模式 1 无重载模式 16 位计数器/定时器 1: 模式 2 自动重装载 16 位计数器/定时器	0	R/W
0	TIM_RUN	定时器运行控制位 0: 定时器停止 1: 定时器运行	0	R/W



### 17.4.3 LPTIM 立即重载寄存器(LPTIM\_LOAD)

地址偏移: 0x08 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15:0	LOAD[15:0]	即刻重载寄存器 与 Timer 是否运行无关，与 MODE 无关。 写 LOAD 前需要读取 LPTIM_CR.WT_FLAG，当且仅当 WT_FLAG 为 0 时，才能写入数据。写 LOAD 寄存器完成后 WT_FLAG 会变低。写该寄存器时会立即更新计数器的值。 读该寄存器时返回最新更新到 LOAD 或者 BGLOAD 的值	0x0	R/W



#### 17.4-4 LPTIM 中断寄存器(LPTIM\_INTSR)

地址偏移: 0x0C

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	INTF	中断标志： 0: 未发生中断 1: 发生溢出中断	0x0	RO



#### 17.4.5 LPTIM 中断寄存器(LPTIM\_INTCLR)

地址偏移: 0x10

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	ICLR	写1清除中断标志，写0无效	0x0	WO



#### 17.4.6 LPTIM 周期重载寄存器(LPTIM\_BGLOAD)

地址偏移: 0x14

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15:0	BGLOAD[15:0]	BackGround 周期重载寄存器 写 BGLOAD 前需要读取 LPTIM_CR.WT_FLAG， 当且仅当WT_FLAG 为 0 时，才能写入数据。写 BGLOAD 寄存器完成后WT_FLAG 会变低。 写该寄存器时，当计数器溢出后，设定值会更新到计数器。 读该寄存器时返回最新更新到LOAD 或者 BGLOAD 的值。	0x0	R/W



## 【18】自唤醒定时器(AWK)

ZB32F103 有一个专用的自唤醒定时器(AWK)，为芯片在低功耗模式下提供一个唤醒事件基准。AWK 在 Sleep 或 Deep Sleep 模式保持计数。当 AWK 用作唤醒定时器时，AWK 要在进入省电模式之前开启。AWK 可以配置内部低速时钟源 SIRC、外部低速时钟源 LXT、外部高速时钟 HXT 分频后的时钟。注意系统时钟频率必须大于 AWK 时钟两倍以上。如果 AWK 开始计数，在设备进入 Sleep 或 Deep Sleep 模式时，选择的时钟源会也要保持工作。注意选择的 AWK 时钟源不会连同 AWK 的配置自动使能，用户应该手动使能选择的时钟源并等待它稳定来确保操作的成功。

AWK 配备了一个简单的 8 位自动重载向上计数定时器。它的预分频可选择从 1/2 到 1/65536，通过 AWK\_CR.DIVSEL[3:0] 来设置。用户填写重装载值到 AWK\_RLOAD 寄存器来决定它的溢出速率。

AWK\_CR.AWKEN 置位后，当 CPU 进入 Sleep/Deep Sleep 模式时装载 AWK\_RLOAD 寄存器的值到内部 8 位计数器并开始计数。当计数器溢出，AWK\_SR.AWUF 置为 1，唤醒 CPU。

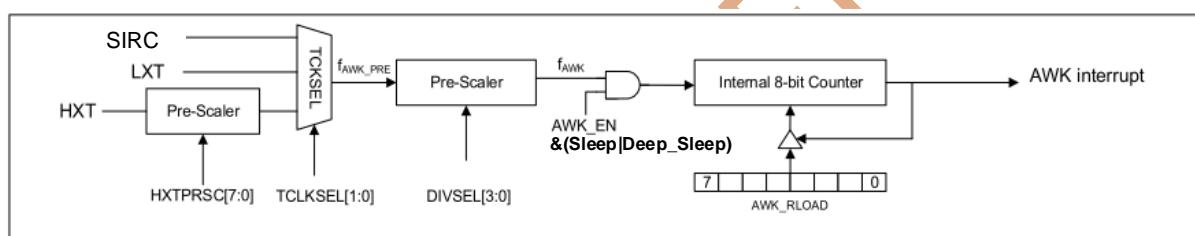


Figure 18-1 自唤醒定时器结构图



### 18.1 寄存器列表

基址: 0x4000 2000

地址偏移	名称	描述	复位值
0x000	AWK_CR	自唤醒定时器控制寄存器	0x0000 BC00
0x004	AWK_RLOAD	自唤醒定时器重装载数据寄存器	0x0000 0000
0x008	AWK_SR	自唤醒定时器状态寄存器	0x0000 0000
0x00C	AWK_INTCLR	自唤醒中断清除寄存器	0x0000 0000

Table 18-1 AWK 寄存器列表和复位值



## 18.2 寄存器说明

### 18.2-1 自唤醒定时器控制寄存器(AWK\_CR)

地址偏移: 0x00 复位值: 0x0000 BC00



位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15:8	HXTPRSC[7:0]	HXT 时钟分频系数: $f_{HXT}/(N+1)$	0xBC	R/W
7	-	保留	0x0	-
6:5	TCLKSEL[1:0]	AWK 计数时钟源选择 00: 停止 01: SIRC 时钟 10: HXT 分频后的时钟 11: LXT 时钟	0x0	R/W
4	AWKEN	AWK 使能 1: 使能 0: 不使能	0	R/W
3:0	DIVSEL[3:0]	计数器用时钟源选择位 0000: FAWK_PRE/2^1 0001: FAWK_PRE/2^2 0010: FAWK+PRE/2^3 ... 1111: FAWK PRE/2^16	0x0	R/W



### 18.2-2 自唤醒定时器重装载数据寄存器(AWK\_RLOAD)

地址偏移: 0x04

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	RLDVAL[7:0]	计数器	0x0	R/W



### 18.2.3 自唤醒定时器状态寄存器(AWK\_SR)

地址偏移: 0x08

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
保留															

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	AWUF	自动唤醒发生, 硬件置 1, 软件清零 0: 未发生自动唤醒 1: 计数器溢出, 自动唤醒发生	0	RO



#### 18.2-4 自唤醒中断清除寄存器(AWK\_INTCLR)

地址偏移: 0x0C

复位值: 0x0000 0000





## 【19】独立看门狗(IWDG)

### 19.1 IWDG 概述

IWDG 的作用是为了防止软件系统在异常情况下，程序执行错误，导致系统异常工作或崩溃。而 IWDG 复位可以帮助系统自动恢复。工作原理是软件系统出错时，在固定的时间(这个时间可以配置)产生一个复位或者中断，让程序重新执行或者按照中断服务程序执行，而不至于系统崩溃。从而增加了软件系统的安全性能。

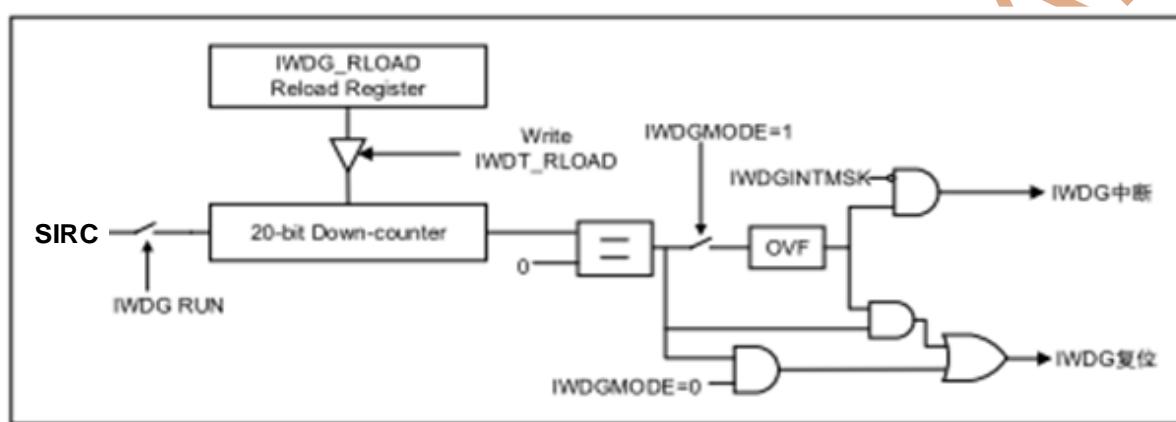


Figure 19-1 IWDG 整体框图

### 19.2 IWDG 的功能

- (1) IWDG 模块是一个 20 位的计数器，每一个 SIRC 时钟(假定为 38.4KHz)计数累加一次，计数时间可配置为 26us-27s;
- (2) 内部低速时钟(SIRC)计数;
- (3) 当计数溢出时，支持中断和复位 2 种方式;
- (4) 可配置的计数溢出时间;
- (5) 启动并清零 IWDG 具有操作序列要求，增加安全性能;
- (6) 部分寄存器写保护功能，防止程序意外操作。



### 19.2-1 超时周期

看门狗超时周期由计数器数值决定，下表列出了它们的数值

看门狗超时周期(假定 SIRC 为 38.4KHz)

IWDG_RLOAD[19:0]寄存器配置值	超时周期
0x000000	26us
...	...
0x003FF	26.6ms
...	...
0xFFFFF	27s

### 19.2-2 IWDG 溢出后产生中断

在本模式下，IWDG 将按所设定的时间周期性的产生中断。在中断服务程序中需要清除 IWDG 溢出标志。配置方法如下所示：

1. 写 0x55AA6699 到 IWDG\_UNLOCK 解除 IWDG 的寄存器写保护，如果 IWDG\_UNLOCK.IWDGREN 为 1 可省略本步骤。
2. 配置 IWDG\_CFGR.IWDGMODE 为 1，选择中断方式。
3. 选择配置 IWDG\_CFGR.IWDGINTMSK 为 0，则 CPU 响应 IWDG 的中断信号；配置为 1，则中断被屏蔽，CPU 只能通过读取 IWDG\_SR.IWDGOVF 来判断是否计数溢出。
4. 配置 IWDG\_RLOAD 寄存器。选择 IWDG 计数溢出时间。
5. 写任意非 0x55AA6699 的值到 IWDG\_UNLOCK 开启 IWDG 的寄存器写保护。
6. 写 0x55 到 IWDG\_CMDCR，启动 IWDG。
7. 如果中断产生，在中断服务程序中先解除 IWDG 的寄存器保护，然后对 IWDG\_INTCLR 写 1 清除中断标记。



### 19.2-3 IWDG 溢出后产生复位

在本模式下，IWDG 计数器溢出后会产生 Reset 信号，该信号会复位 MCU。用户需要在 IWDG 溢出前重装载 IWDG 计数器，从而避免产生 IWDG 复位。

配置方法如下：

1. 写 0x55AA6699 到 IWDG\_UNLOCK 解除 IWDG 的寄存器写保护，如果 IWDG\_UNLOCK.IWDGREN 为 1 可省略本步骤。
2. 配置 IWDG\_CFGR.IWDGMODE 为 0，选择复位方式。
3. 配置 IWDG\_RLOAD 寄存器。选择 IWDG 计数溢出时间。
4. 写任意非 0x55AA6699 的值到 IWDG\_UNLOCK 开启 IWDG 的寄存器写保护。
5. 写 0x55 到 IWDG\_CMDCR，启动 IWDG。
6. 在计数溢出前写 0xAA 到 IWDG\_CMDCR，刷新 IWDG 计数器。

### 19.3 寄存器列表

基址地址：0x4000 3000

地址偏移	名称	描述	默认值
0x00	IWDG_CMDCR	IWDG 控制命令寄存器	0x0000 0000
0x04	IWDG_CFGR	IWDG 配置寄存器	0x0000 0000
0x08	IWDG_RLOAD	IWDG 计数器重装载寄存器	0x000F FFFF
0x0c	IWDG_CNTVAL	IWDG 计数器值	0x000F FFFF
0x10	IWDG_SR	IWDG 中断状态寄存器	0x0000 0000
0x14	IWDG_INTCLR	IWDG 中断清除寄存器	0x0000 0000
0x18	IWDG_UNLOCK	IWDG 寄存器访问保护	0x0000 0000

Table 19-1 IWDG 寄存器列表和复位值



## 19.4 寄存器说明

### 19.4-1 IWDG 控制命令寄存器(IWDG\_CMDCR)

地址偏移: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留	CMD[7:0]														
保 留	WO														

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	CMD[7:0]	0x55: IWDG 启动命令 0xAA: IWDG 重装载刷新命令	0x0	WO

注意: 在 IWDG 运行时才能写重装载命令



#### 19.4.2 IWDG 配置寄存器(IWDG\_CFGR)

地址偏移: 0x04 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IWDGRUNF	IWDGINTMSK	IWDGMODE												
保留	RO	R/W	R/W												

	标记	功能描述	复位值	读写
31:3	-	保留	0x0	-
2	IWDGRUNF	IWDG 运行标志 0: IWDG 停止 1: IWDG 运行(即在计数)	0	RO
1	IWDGINTMSK	IWDG 中断屏蔽 0: 中断不屏蔽(通知给CPU) 1: 中断被屏蔽	0	R/W
0	IWDGMODE	IWDG 计数溢出模式选择位 0: 复位方式 1: 中断方式, 生成一个中断信号, 然后重启计数器, 如果中断没有在第二次超时发生之前被清除的话, 则生成一个系统复位信号。注意: IWDG 产生复位后会复位整个系统	0	R/W

注意: 受 IWDG\_UNLOCK 保护



#### 19.4.3 IWDG 计数器重装载寄存器(IWDG\_RLOAD)

地址偏移: 0x08 复位值: 0x000F FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														IWDGRLOAD[19:16]	
保留														R/W	

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

位	标记	功能描述	复位值	读写
31:20	-	保留位		
19:0	IWDGRLOAD[19:0]	IWDG 重装载寄存器	0xFFFF	R/W

注意: 受 IWDG\_UNLOCK 保护



#### 19.4.4 IWDG 计数器值寄存器(IWDG\_CNTVAL)

地址偏移: 0x0C 复位值: 0x000F FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												IWDGCNT [19:16]			
保留												RO			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDGCNT [15:0]															
RO															

位	标记	功能描述	复位值	读写
31:20	-	保留位	0x0	-
19:0	IWDGCNT[19:0]	IWDG 计数值寄存器,	0xFFFF	RO



#### 19.4.5 IWDG 中断状态寄存器(IWDG\_SR)

地址偏移: 0x10 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IWDGOVF														
保留	RO														

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	IWDGOVF	<p>IWDG 溢出中断标志位 0: IWDG 无溢出中断发生 1: IWDG 有溢出中断发生 注意: 1. 当 IWDG 配置为复位方式, 不管 IWDG 计数器是否溢出, 这位都不会置高。 2. 当 IWDG 配置为中断方式, 不管 IWDGINTMSK 位是否置高, 只要 IWDG 计数器溢出, 这位就置高。</p>	0	RO



#### 19.4.6 IWDG 中断清除寄存器(IWDG\_INTCLR)

地址偏移: 0x14 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IWDGINTCLR														
保留	WO														

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	IWDGINTCLR	IWDG 中断清除 0: 写 0, 无任何动作 1: 写 1, 清除 IWDG 中断标志	0x0	WO

注意: 受 IWDG\_UNLOCK 保护



#### 19.4.7 IWDG 保护寄存器(IWDG\_UNLOCK)

地址偏移: 0x18 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	IWDGREN	IWDG 中断清除 0: 不能更改IWDG 的相关寄存器 1: 可以更改IWDG 的相关寄存器	0	R/W

注: 写 0x55AA6699 到 IWDG\_UNLOCK 可以解除 IWDG 的寄存器写保护, 写任意非 0x55AA6699 的值到 IWDG\_UNLOCK 可以开启 IWDG 的寄存器写保护(IWDG\_CFG、IWDG\_RLOAD、IWDG\_INTCLR)。

#### 19.5 注意

- (1) 喂狗指令到看门狗定时器被更新需要两个看门狗计数时钟源时钟的延迟。
- (2) 系统操作看门狗时, 两次喂狗间隔需要允许有至少三个看门狗时钟。



## 【20】 系统窗口看门狗(WWDG)

### 20.1 概述

窗口看门狗定时器(WWDG)的目的是在一个指定的窗口周期中执行系统复位，防止软件在任何不可预知的条件下进入不可控制的状态。

### 20.2 特征

- (1) 一个 8 位向下数计数器(WWDG\_CNT)和一个 8 位比较值(WINCMP)使 WWDG 超时窗口周期可调
- (2) 支持 20 位值(PRSC)选择看门狗预分频值
- (3) 支持窗口计数值比较中断和计数溢出、加载计数值出错复位

### 20.3 结构框图

窗口看门狗定时器框图如下：

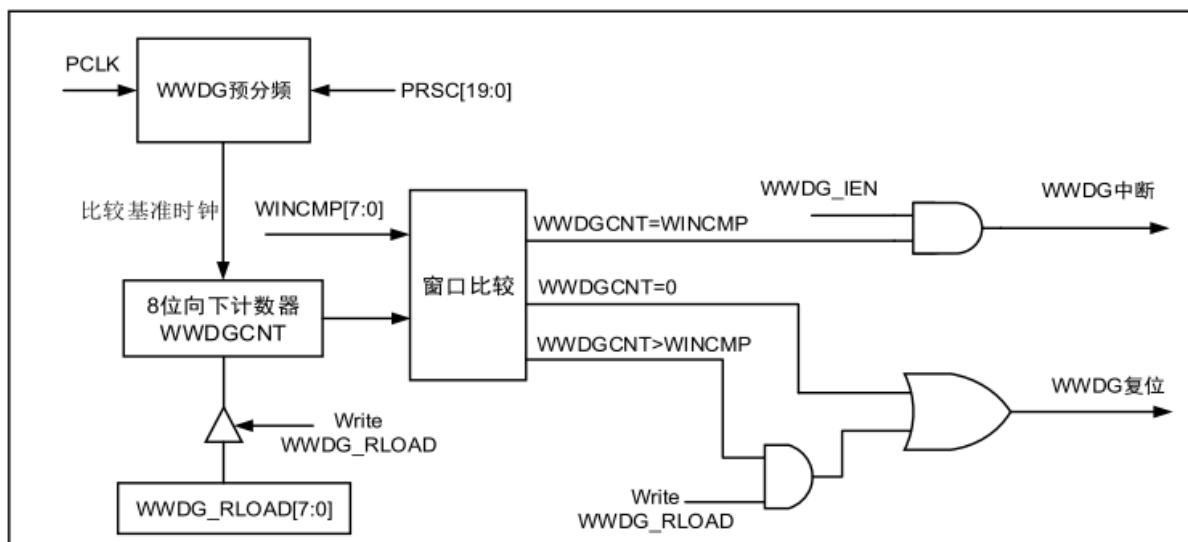


Figure 20-1 WWDG 结构框图

### 20.4 基本配置

WWDG 外设时钟源通过 RCC\_PCLKEN.WWDGCKEN 来使能。

1. 通过 WWDG\_RLOAD[19:0]配置窗看门狗计数初始值
2. 通过 WWDG\_CR.PRSC[19:0]配置计数时钟预分频
3. 通过 WWDG\_CR.WINCMP[7:0]配置窗比较值
4. 根据是否需要使能中断，来配置 WWDG\_INTEN.WWDGIEN
5. 通过 WWDG\_CR.WWDGEN 写 1 来开启窗看门狗



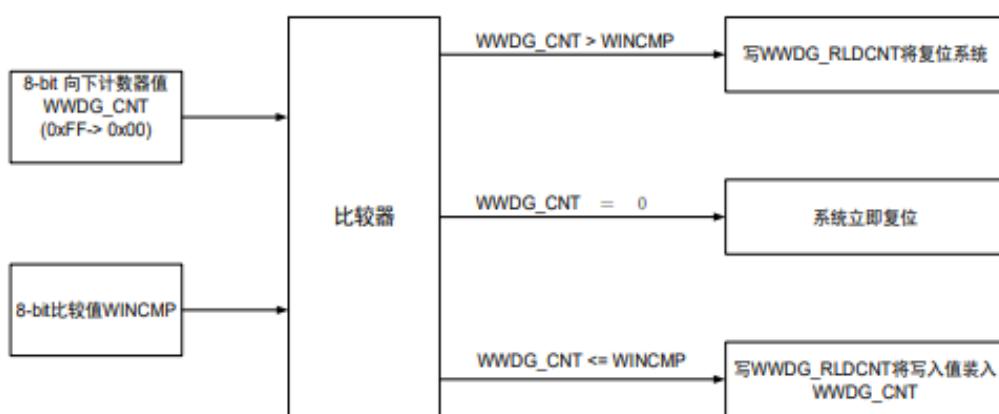
## 20.5 功能描述

窗口看门狗定时器(WWDG)是一个 8 位向下计数器，该计数器带一个可选择预分频值，不同的预分频值对应不同的看门狗定时溢出时间。8 位窗口看门狗定时器的时钟源是 PCLK 时钟经分频后的时钟，看门狗的时钟源带一个可选择的 20 位预分频值，该值可通过 WWDG\_CR.PRSC[19:0]位来设置选择，对应预分频值如下表。

PRSC[19:0]	预分频值	定时溢出周期	定时溢出间隔 PCLK=24MHz
0x00000	1	1PCLK * 1	41.7ns
0x00001	2	1PCLK * 2	83.4ns
0x00002	3	1PCLK * 3	125.1ns
0x00003	4	1PCLK * 4	166.8ns
0x00004	5	1PCLK * 5	208.5ns
...	...	...	...
0x80000	524289	1PCLK * 524289	21.9ms
...	...	...	...
0xFFFFF	1048576	1PCLK * 1048576	43.8ms

### 20.5-1 窗口看门狗定时器的计数

当 WWDG\_CR.WWDGEN 位被使能，窗口看门狗向下计数器将会从 WWDG\_CNT[7:0]向下递减计数到 0，并且不能够被软件关闭。为了防止程序在非用户指定位置关闭窗口看门狗定时器，窗口看门狗定时器控制寄存器的 WWDGEN 在芯片上电或复位后仅可写一次。当 WWDG\_CR.WWDGEN 位被软件使能以后，用户不能禁止窗口看门狗定时器 WWDG\_CR.WWDGEN，修改计数器预分频周期 WWDG\_CR.PRSC[19:0]，或修改窗口比较值 WWDG\_CR.WINCM[7:0]，除非芯片复位。窗口看门狗定时器在 CPU 进入 Sleep 模式或是 Deep Sleep 模式时会停止计数，CPU 被唤醒后恢复正常工作。





### 20.5-2 窗口看门狗定时器比较中断

窗口看门狗定时器向下计数过程中，当窗口看门狗定时器计数值 `WWDG_CNT[7:0]` 等于窗口比较值 `WWDG_CR.WINCM[7:0]` 时，`WWDG_SR.WWDGIF` 会被置 1 并且 `WWDG_SR.WWDGIF` 可以被软件清零。如果 `WWDG_INTEN.WWDGIEN` 位被使能，当 `WWDG_SR.WWDGIF` 位被硬件置 1，就会产生窗口看门狗比较匹配中断。

### 20.5-3 窗口看门狗定时器复位系统

当 `WWDG` 计数器的值计到 0 时，`RCC_RSTSR.WWDGRST` 会被置 1。在 `WWDG` 计数器下数到 0 前，用户必须通过对 `WWDG_RLOAD` 写值来进行重载，从而阻止 `WWDG` 复位的发生。重载的动作只能在计数器的值小于或等于 `WINCM` 值时进行。如果 `WWDG` 计数器当前值大于 `WINCM` 的值，用户对 `WWDG_RLOAD` 寄存器写，窗口看门狗定时器复位系统信号将立刻产生，并导致芯片复位。

### 20.5-4 窗口看门狗定时器的窗口设置限制

当用户对 `WWDG_RLOAD` 寄存器写重载 `WWDG` 的值的时候，

$$TPCLK = THCLK * (2 * RCC_PCLKDIV.APBCKDIV[7:0])$$

设定时间间隔：

$$T = TPCLK * (WWDG_CR.WWDG_PRSC[19:0] + 1) * (WWDG_RLOAD.WWDG_RLOAD[7:0] + 1)$$

用户可以根据需要来配置分频寄存器 `WWDG_PRSC[19:0]` 和 `WWDG_RLOAD[7:0]` 的值来达到想要的时间间隔。

为了保证正常工作，`WWDG_RLOAD[7:0]` 的值要大于等于 1。

## 20.6 与独立看门狗定时器(IWDG)比较

### 20.6-1 复位条件和复位延时

IWDG 和 `WWDG` 通常是应用在系统跑到不可控制的状态后复位系统。IWDG 只有一个条件可以触发复位信号，`WWDG` 有两种条件可以触发 `WWDG` 产生复位信号：

`WWDG_CNT = 0`; `WWDG_CNT` 大于 `WINCM` 时往 `WWDG_RLOAD` 写入。

一旦 `WWDGRST` 被置 1，`WWDG` 将立即复位系统。

### 20.6-2 唤醒功能

IWDG 支持唤醒功能并且在 Deep Sleep 模式下继续工作。相比之下，`WWDG` 不支持唤醒功能并且 `WWDG` 的计数器在 Deep Sleep 模式下会停止计数。



## 20.7 寄存器列表

WWDG: 基址: 0x4000 2000

地址偏移	名称	描述	默认值
0x00	WWDG_RLOAD	窗口看门狗定时器重载计数寄存器	0x0000 00FF
0x04	WWDG_CR	窗口看门狗定时器控制寄存器	0x0800 00FF
0x08	WWDG_INTEN	窗口看门狗定时器中断使能寄存器	0x0000 0000
0x0C	WWDG_SR	窗口看门狗定时器状态寄存器	0x0000 0000
0x10	WWDG_INTCLR	窗口看门狗定时器中断清除寄存器	0x0000 0000
0x14	WWDG_CNTVAL	窗口看门狗定时器计数器值寄存器	0x0000 00FF

Table 20-1 WWDG 寄存器列表和复位值

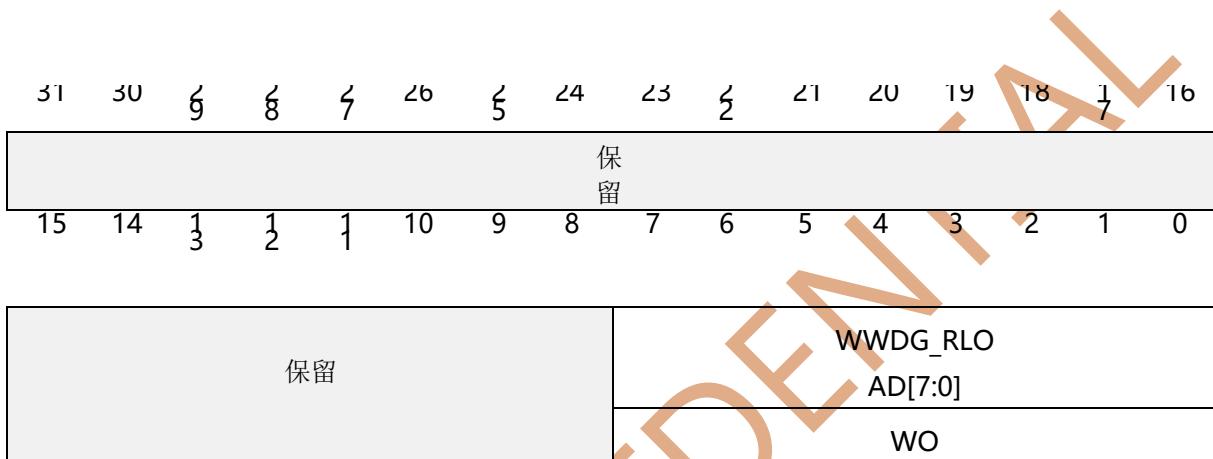


## 20.8 寄存器说明

### 20.8-1 窗口看门狗定时器重载计数寄存器(WWDG\_RLOAD)

地址偏移: 0x00 复位值: 0x0000 00FF

复位值: 0x0000 00FF



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	WWDG_RLOAD[7:0]	窗口看门狗定时器重载计数寄存器。写入值大于 0	0xFF	WO



## 20.8-2 窗口看门狗定时器控制寄存器(WWDG\_CR)

地址偏移: 0x04 复位值: 0x0800 00FF

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

位	标记	功能描述	复位值	读写
31:29	-	保留	0x0	-
28	WWDGEN	窗口看门狗使能位 设置该位使能窗口看门狗定时器 0: 禁止窗口看门狗定时器功能 1: 使能窗口看门狗定时器功能	0	R/W
27:8	PRSC[19:0]	WWDG 预分频 $FPCLK/(PRSC+1)$	0x8000 0	R/W
7:0	WINCMP[7:0]	WWDG 窗口比较寄存器 设置该寄存器调整有效的重载窗口。 注: 仅当WWDG 计数器值在 0 和 WINCMP 之间时, 软件才能写WWDG_RLOAD。当WWDG 计数器值大于WINCMP 时, 如果软件写WWDG_RLOAD, WWDG 会产生复位信号	0xFF	R/W

注: 当 WWDGEN 设定为 1 后, 该寄存器的软件配置将被禁止。



### 20.8-3 窗口看门狗定时器中断使能寄存器(WWDG\_INTEN)

地址偏移: 0x08 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	WWDGIE N	WWDG 中断使能位 设置该位使能窗口看门狗定时器中断功能。 0: 禁止窗口看门狗定时器中断功能 1: 使能窗口看门狗定时器中断功能	0	R/W



#### 20.8-4 窗口看门狗定时器状态寄存器(WWDG\_SR)

地址偏移: 0x0C 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	WWDGIF	WWDG 比较匹配中断标志 0: 无窗口看门狗定时器中断 1: 有窗口看门狗定时器中断 当 WINCMP 和 WWDG 计数器匹配，该位置 1，软件对 WWDG INTCLR.INTCLR 写 1 清 0 该位。	0	RO



### 20.8-5 窗口看门狗定时器中断清除寄存器(WWDG\_INTCLR)

地址偏移: 0x10 复位值: 0x0000 0000

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	INTCLR	WWDG 比较匹配中断标志清除 软件写 1 清 0 对应的中断标志位 WWDG_SR.WWDGIF。	0	WO



### 20.8-6 窗口看门狗定时器计数器值寄存器(WWDG\_CNTVAL)

地址偏移: 0x14 复位值: 0x0000 00FF



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	WWDGCNT[7:0]	WWDG 计数器值 该寄存器表示窗口看门狗计数器当前的值，该寄存器只读	0xFF	RO



## 【21】通用异步/同步收发器(USART0/1/3,UART4/5)

### 21.1 概述

- USART0/1(No DMA request)

本产品带有 2 个通用 UART 模块(UART4/5)，支持半双工和全双工传输；支持 8bit、9bit 数据格式；支持 Mode0/1/2/3 四种不同传输模式；USART0 的波特率可以由 TIM10 产生或者自动波特率发生器产生，USART4 的波特率可以由 TIM11 产生或者自动波特率发生器产生；支持多机通讯模式；支持自动地址识别；支持给定地址和广播地址。

通用 UART(UART4/5)只有一个时钟输入 PCLK，寄存器配置逻辑和数据收发逻辑都工作在该时钟。

- USART(support DMA request)

通用同步异步收发器（USART）提供了一种灵活的方法来与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。USART 利用分数波特率发生器提供宽范围的波特率选择。

它支持同步单向通信和半双工单线通信。它也支持 LIN(局部互连网)，智能卡协议和 IrDA(红外数据组织)SIR ENDEC 规范，以及调制解调器(CTS/RTS)操作。它还允许多处理器通信。

使用多缓冲器配置的 DMA 方式，可以实现高速数据通信。

主要特性：

- 全双工的，异步通信
- NRZ 标准格式
- 分数波特率发生器系统
  - 发送和接收共享的可编程波特率，最高到 4.5Mbits/s
- 可编程数据字长度（8 位或 9 位）
- 可配置的停止位-支持 1 或 2 个停止位
- LIN 主发送同步断开符的能力以及 LIN 从检测断开符的能力
  - 当 USART 硬件配置成 LIN 时，生成 13 位断开符；检测 10/11 位断开符
- 发送方为同步传输提供时钟
- IRDA SIR 编码器解码器
  - 在正常模式下支持 3/16 位的持续时间
- 智能卡模拟功能
  - 智能卡接口支持 ISO7816-3 标准里定义的异步协议智能卡
  - 智能卡用到的 0.5 和 1.5 个停止位
- 单线半双工通信
- 可配置的使用 DMA 的多缓冲器通信
  - 在 SRAM 里利用集中式 DMA 缓冲接收/发送字节
- 单独的发送器和接收器使能位



- 检测标志
  - 接收缓冲器满
  - 发送缓冲器空
  - 传输结束标志
- 校验控制
  - 发送校验位
  - 对接收数据进行校验
- 四个错误检测标志
  - 溢出错误
  - 噪音错误
  - 帧错误
  - 校验错误
- 10 个带标志的中断源
  - CTS 改变
  - LIN 断开符检测
  - 发送数据寄存器空
  - 发送完成
  - 接收数据寄存器满
  - 检测到总线为空闲
  - 溢出错误
  - 帧错误
  - 噪音错误
  - 校验错误
- 多处理器通信 -- 如果地址不匹配，则进入静默模式
- 从静默模式中唤醒（通过空闲总线检测或地址标志检测）
- 两种唤醒接收器的方式：地址位(MSB, 第 9 位), 总线空闲



## 21.2 结构框图

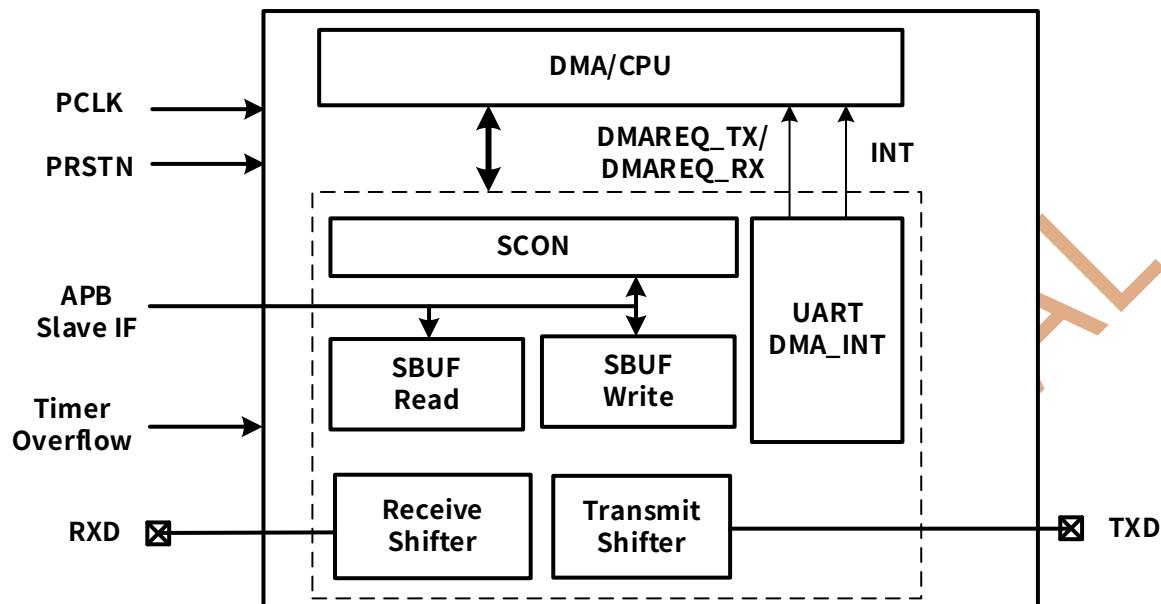


Figure 21-1 UART 结构图

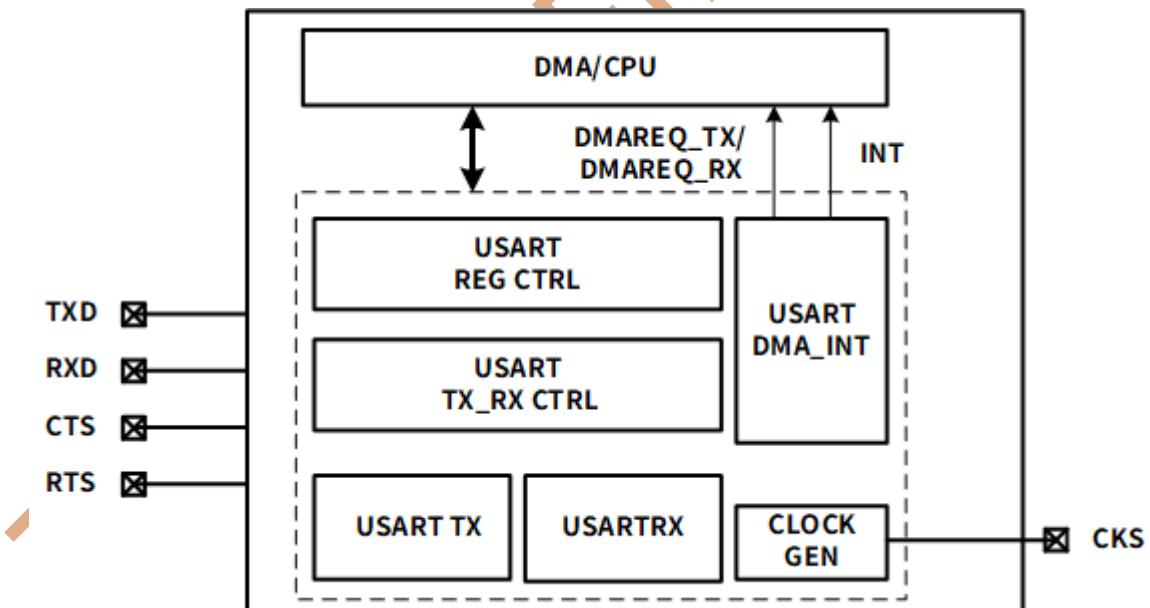


Figure 21-2 USART 结构图

RX: 接收数据串行输入。通过过采样技术来区别数据和噪音，从而恢复数据。

TX: 发送数据输出。当发送器被禁止时，输出引脚恢复到它的 I/O 端口配置。当发送器被激活，并且没东西发送时，TX 引脚处于高电平。在单线和智能卡模式里，此 I/O 口被用来发送和接收数据。

在同步模式中需要下列引脚：



**CKS:** 发送器时钟输出。此引脚输出用于同步传输的时钟，(在 Start 位和 Stop 位上没有时钟脉冲，软件可选地，可以在最后一个数据位送出一个时钟脉冲)。数据可以在 RX 上同步被接收。这可以用来控制带有移位寄存器的外部设备(例如 LCD 驱动器)。时钟相位和极性都是软件可编程的。在智能卡模式里，SCLK 可以为智能卡提供时钟。

在 IrDA 模式里需要下列引脚：

**IrDA\_RDI:** IrDA 模式下的数据输入。

**IrDA\_TDO:** IrDA 模式下的数据输出。

下列引脚在硬件流控模式中需要：

**CTS:** 清除发送，若是高电平，在当前数据传输结束时阻断下一次的数据发送。

**RTS:** 发送请求，若是低电平，表明 USART 准备好接收数据



### 21.3 UART 工作模式

#### 21.3-1 Mode 0(同步模式, 半双工)

当工作在 Mode0 时, UART 工作在同步模式, 其波特率为固定的 PCLK 时钟的 1/12。UART 接收数据由 RXD 输入、UART 发送数据由 TXD 输出, RXD 此时为输入输出端口。UART 同步移位时钟由 TXD 输出, TXD 此时为输出端口。注意, 本模式只能作为主机发送同步移位时钟, 不可以作为从机从外部接收移位时钟。该模式下, 传输的数据位宽只能是 8 位的, 没有起始位和结束位。

将 UARTx\_SCON.SM0 和 UARTx\_SCON.SM1 清零, 可进入 Mode0 工作模式。

##### 21.3-1.1 发送数据

发送数据时, 清除 UARTx\_SCON.REN 位, 并将数据写入 UARTx\_SBUF 寄存器。此时, 发送数据将从 TXD 输出(低位在先, 高位在后), 同步移位时钟从 TXD 输出。

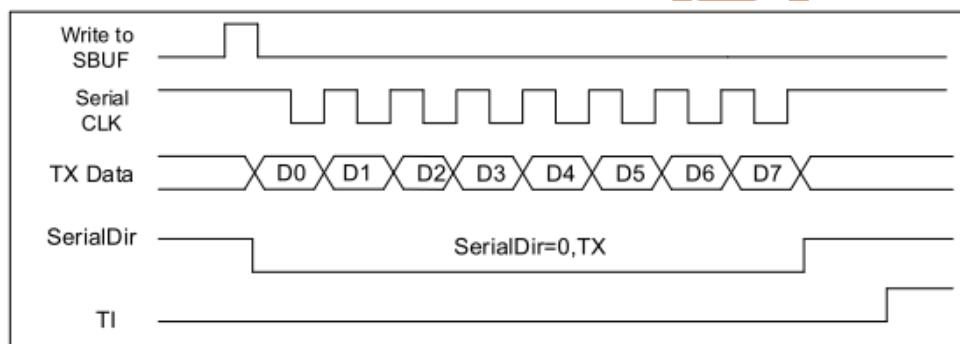


Figure 21-3 Mode0 发送数据

##### 21.3-1.2 接收数据

接收数据时, 将 UARTx\_SCON.REN 位置 1, 并将 UARTx\_INTSR.RI 位清零。当接收结束, 数据可从 UARTx\_SBUF 寄存器读出。此时, 接收数据从 RXD 输入(低位在先, 高位在后), 同步移位时钟从 TXD 输出。

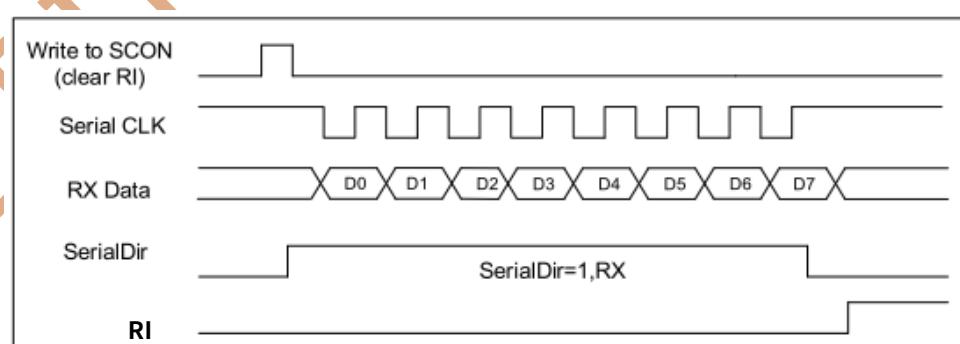


Figure 21-4 Mode0 接收数据



### 21.3-2 Mode 1(异步模式，全双工)

当工作在 Mode1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 10 位组成：起始位 “0” 开始，紧接着 8 位数据位(低位在先，高位在后)，最后是结束位 “1” 。该模式下，波特率可以由

可编程定时器模块产生，也可以由模块内部的自动波特率发生器产生。

当 `UARTx_BAUDCR.SELF_BRG` 为 0 时，选择由定时器产生波特率时，`UART0, UART2` 的波特率由 `TIM10` 产生，`UART1, UART3` 的波特率由 `TIM11` 产生；当 `UARTx_BAUDCR.SELF_BRG` 置 1 时，`UART0,1,2,3` 波特率都由各自内部的自动波特率发生器产生。波特率产生公式请参考 21.3.5.2 节 Mode1/3。

将 `UARTx_SCON.SM0` 清 0，`UARTx_SCON.SM1` 置 1，可进入 Mode1 工作模式。

#### 21.3-2.1 发送数据

发送数据时，与 `UARTx_SCON.REN` 的值无关，将所发送数据写入 `UARTx_SBUF` 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

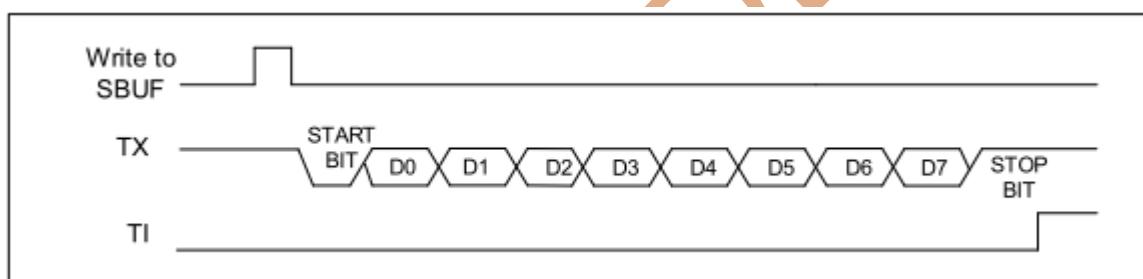


Figure 21-5 Mode1 发送数据



### 21.3-2.2 接收数据

接收数据时，需将 `UARTx_SCON.REN` 位置 1，并将 `UARTx_INTSR.RI` 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 `UARTx_SBUF` 寄存器读出。

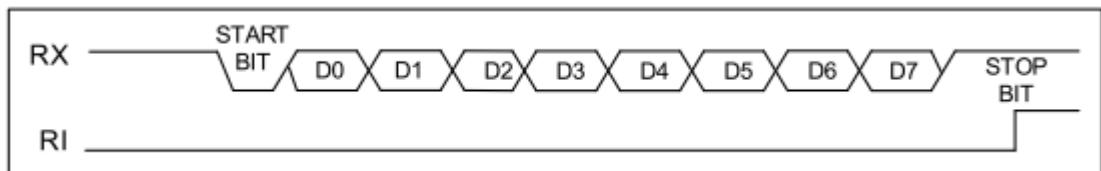


Figure 21-6 Mode1 接收数据

### 21.3-3 Mode 2(异步模式，全双工)

当工作在 Mode2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 11 位组成：起始位 “0” 开始，接着是 8 个数据位，1 个 TB8 位和结束位。额外的 TB8 位是用来在多机通讯环境下使

用，当 `TB8=1`，表明所接收的是地址帧；当 `TB8=0`，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位来使用。该模式下，波特率可以独立产生，不需要外部定时器模块产生。

将 `UARTx_SCON.SM0` 置 1，`UARTx_SCON.SM1` 清 0，可进入 Mode2 工作模式。

### 21.3-3.1 发送数据

发送数据时，与 `UARTx_SCON.REN` 的值无关，并将所发送数据写入 `UARTx_SBUF` 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

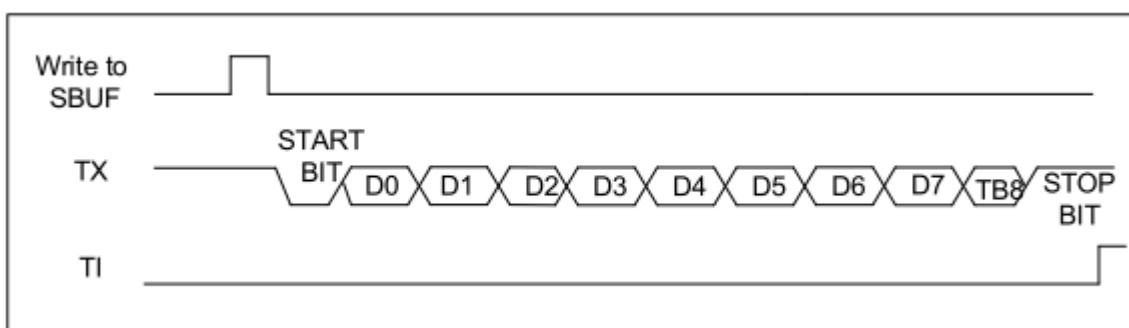


Figure 21-7 Mode2 发送数据



### 21.3.3.2 发送数据

接收数据时，需将 `UARTx_SCON.REN` 位置 1，并将 `UARTx_INTSR.RI` 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 `UARTx_SBUF` 寄存器读出。

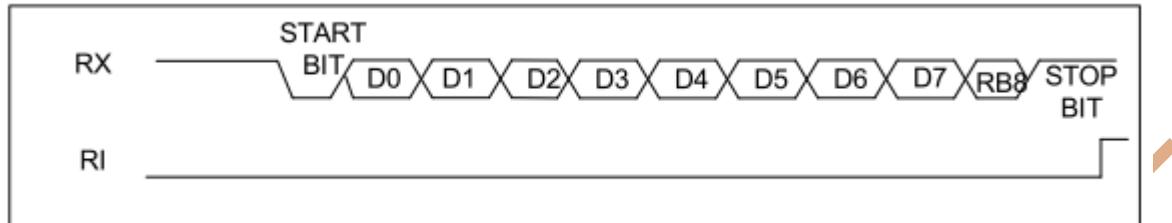


Figure 21-8 Mode2 接收数据



### 21.3-4 Mode 3(异步模式，全双工)

Mode3 的数据格式，传输时序以及操作方式都与 Mode2 相同，唯一的区别是 Mode3 的波特率选择由可编程定时器产生或者内部自动波特率发生器产生，而不是像 Mode2 只由设备自己独立产生。

Mode3 的波特率是可编程的，波特率生成方式与 Mode1 相同。

将 UARTx\_SCON.SM0 置 1，UARTx\_SCON.SM1 置 1，可进入 Mode3 工作模式。

### 21.3-5 波特率编程

#### 21.3-5.1 Mode 0

当工作在 Mode0 时，波特率被固定在 PCLK 的 1/12，不需要可编程定时器(Timer)的支持。

#### 21.3-5.2 Mode 1/3

当工作在 Mode1 或者 Mode3 时，波特率产生公式如下图所示：

UARTx\_BAUDCR

.SELF\_BRG=0，使用可编程定时器(Timer)波特率模式：

$$\text{BaudRate} = \frac{(\text{UARTx_SCON.DBAUD} + 1) * F_{\text{PCLK}}}{32 * (2^{16} - \text{TIMx_BGLOAD}[15:0])}$$

UARTx\_BAUDCR.SELF\_BRG=1，使用自身波特率生成模式：

$$\text{BaudRate} = \frac{(\text{UARTx_SCON.DBAUD} + 1) * F_{\text{PCLK}}}{32 * (\text{UARTx_BAUDCR.BRG}[15:0] + 1)}$$

其中，UARTX\_SCON.DBAUD 表示双倍波特率，FPCLK 为 PCLK 时钟频率，TIMx\_BGLOAD 为 Timer 的周期装载计数值。注意，Timer 必须配置为 16 位自动重载模式，立即重载寄存器(TIMx\_LOAD)和周期重载寄存器(TIMx\_BGLOAD)要写入相同的初始值。



### 21.3-5.3 Mode 2

当工作在 Mode 2 时，波特率被固定在如下公式所得值：

$$\text{BaudRate} = \frac{(\text{UARTX\_SCON.DBAUD} + 1) * \text{F}_{\text{PCLK}}}{64}$$

其中，UARTX\_SCON.DBAUD 表示双倍波特率，FPCLK 为 PCLK 时钟频率。

### 21.3-6 帧错误检测

Mode 1/2/3 具有帧错误检测功能，硬件会自动检测接收到的帧数据是否带有效的 Stop 位。如果没有收到有效 Stop 位，则 UARTx\_INTSR.FE 置 1。UARTx\_INTSR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有效 Stop 位，也不会把 UARTx\_INTSR.FE 标志清 0。

### 21.3-7 多机通讯

Mode 2/3 具有多机通讯功能，为此在其帧格式中增加了 1 位 TB8/RB8。将 UARTx\_SCON.SM2 置“1”，可开启多机通讯位。当开启多机通讯位后，发送数据时，主机可以通过 UARTx\_SCON.TB8 来区分当前帧是地址帧(UARTx\_SCON.TB8=1)还是数据帧(UARTx\_SCON.TB8=0)。接收数据时，从机会忽略 RB8 位(第 9 位)为“0”的当前接收帧。当收到帧的 RB8 位(第 9 位)为“1”表明其是地址帧，从机会继续判断接收到的地址与其自身地址是否相等。如果匹配，则从机会对 UARTx\_SCON.RB8 置“1”，并对 UARTx\_INTSR.RI 置“1”，以表明该帧为地址帧并且地址已经匹配。从机软件看到 UARTx\_SCON.RB8=1 并且 UARTx\_INTSR.RI=1 后，先把 UARTx\_SCON.SM2 位清“0”，然后准备接受给它的数据帧。如果地址不等，表明主机并不是寻址该从机，从机硬件保持 UARTx\_SCON.RB8 和 UARTx\_INTSR.RI 为“0”，软件保持 UARTx\_SCON.SM2 位为“1”，从机继续处于地址监听状态。

### 21.3-8 自动地址识别

当开启多机通讯位后(UARTx\_SCON.SM2 置“1”），自动地址识别功能也将开启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收端会给出 UARTx\_INTSR.RI 接收标志。如果地址不匹配，则接收端不会给出任何接收标志。

如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 Stop 位代替。当从机接收到匹配的地址帧和有效的 Stop 位时，UARTx\_INTSR.RI 会被置“1”。为了支持自动地址识别，定义了广播地址和给定地址的概念。



### 21.3-9 给定地址

UART 设备的 `UARTx_SADDR` 寄存器用来表示自己的设备给定地址，`UARTx_SADEN` 寄存器是地址掩码，可以用来定义地址中的无关位。当 `UARTx_SADEN` 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，`UARTx_SADEN` 寄存器必须设为 0xFF。

$$\text{GivenAddr} = \text{UARTx\_SADDR} \& \text{UARTx\_SADEN}$$

### 21.3-10 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 0xFF。

$$\text{BoardCastAddr} = \text{UARTx\_SADDR} | \text{UARTx\_SADEN}$$

给定地址和广播地址举例

假设某从机的 `UARTx_SADDR` 和 `UARTx_SADEN` 配置如下：

`UARTx_SADDR`: 0b01101001

`UARTx_SADEN`: 0b11111011

那么其给定地址和广播地址如下：

Given: 0b01101x01

Broadcast: 0b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

0b01101001 和 0b01101101 (given address)

0b11111011 和 0b11111111 (broadcast address)

### 21.3-11 收发端缓存

#### 21.3-11.1 接收缓存

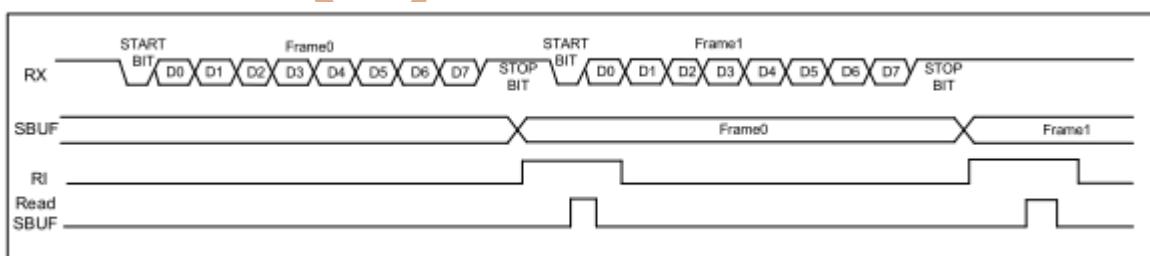


Figure 21-9 接收缓存

#### 21.3-11.2 发送缓存

通用 UART(UART4/5)发送端不支持发送缓存。如果在发送数据过程中，填写

`UARTx_SBUF`

寄存器，将会破坏当前正在发送数据。软件应该避免这种操作



### 21.3-12 IrDA 红外功能

IrDA SIR 物理层规定使用反相归零调制方案(RZI)，该方案用一个红外光脉冲代表逻辑' 0'，见图 21-9

IrDA 结构框图。SIR 发送编码器对从 UART 输出的 NRZ(非归零)比特流进行调制。输出脉冲流被传送到一个外部输出驱动器和红外 LED。对于 SIRENDEC 应用，UART 最高只支持到 115.2Kbps 速率。在正常模式里，脉冲宽度规定为一个位周期的 3/16。SIR 接收解码器对来自红外接收器的归零位比特流进行解调，并将接收到的 NRZ 串行比特流输出到 UART。在空闲状态里，解码器输入通常是高(标记状态)。发送编码器输出的极性和解码器的输入相反。当解码器输入低时，检测到一个起始位。

● IrDA 是一个半双工通信协议。如果发送器忙(也就是 UART 正在送数据给 IrDA 编码器)，IrDA 接收线上的任何数据都将被 IrDA 解码器所忽略。如果接收器忙(也就是 UART 正在接收从 IrDA 解码器来的解码数据)，从 UART 的 TX 上到 IrDA 的数据将不会被 IrDA 编码。当接收数据时，应该避免发送，因为将被发送的数据可能被破坏。

● SIR 发送逻辑把' 0' 作为高脉冲发送，把' 1' 作为低电平发送；或者取反发送。脉冲的宽度规定为

正常模式时位周期的 3/16，见图 21-10 IrDA 收发脉冲。

● SIR 解码器把接收到的 IrDA 信号转变成比特流后发送给 UART。  
● SIR 接收逻辑把高电平状态解释为' 0'，把低脉冲解释为' 1'；或者取反接收。  
● 发送编码器输出与解码器输入有着相反的极性。当空闲时，SIR 输出处于低状态。  
● IrDA 规范要求脉冲要宽于 1.41us。脉冲宽度是可编程的。接收器端的尖峰脉冲检测电路会过滤宽度小于 2 个 PSC 周期的脉冲进行过滤操作(PSC 是在 UARTx\_IRDACR 中编程的预分频值)。宽度小于 1 个 PSC 周期的脉冲一定会被过滤掉，但是那些宽度大于 1 个而小于 2 个 PSC 周期的脉冲可能被接收或滤除，那些宽度大于 2 个周期的将被视为一个有效的脉冲。  
● IrDA 接收器可以与另一 IrDA 低功耗发送器通信。



### 21.3-12.1 IrDA 低功耗模式

IrDA 可以工作在正常模式，也可以工作在低功耗模式。选择低功耗模式需要把 UART\_IRDACR.IRLPMODE 寄存器置 1。

发送器

在低功耗模式，脉冲宽度不再持续 3/16 个位周期。取而代之，脉冲的宽度是低功耗波特率时钟周期的 3 倍，该波特率的频率最小可以是 1.42MHz。通常这个值是 1.8432MHz ( $1.42\text{MHz} < \text{PSC} < 2.12\text{MHz}$ )

一个低功耗模式可编程分频器把系统时钟进行分频以达到这个值。

接收器

低功耗模式的接收类似于正常模式的接收。为了滤除尖峰干扰脉冲，UART 应该滤除宽度短于 1 个周期的脉冲。只有持续时间大于 2 个周期的 IrDA 低功耗波特率时钟(UART\_IRDACR 中的 PSC)的低电平信号才被接受为有效的信号。

注意：

宽度小于 2 个大于 1 个 PSC 周期的脉冲可能会也可能不会被滤除。

接收器的建立时间应该由软件管理。IrDA 物理层技术规范规定了在发送和接收之间最小要有 10ms 的延时(IrDA 是一个半双工协议)。

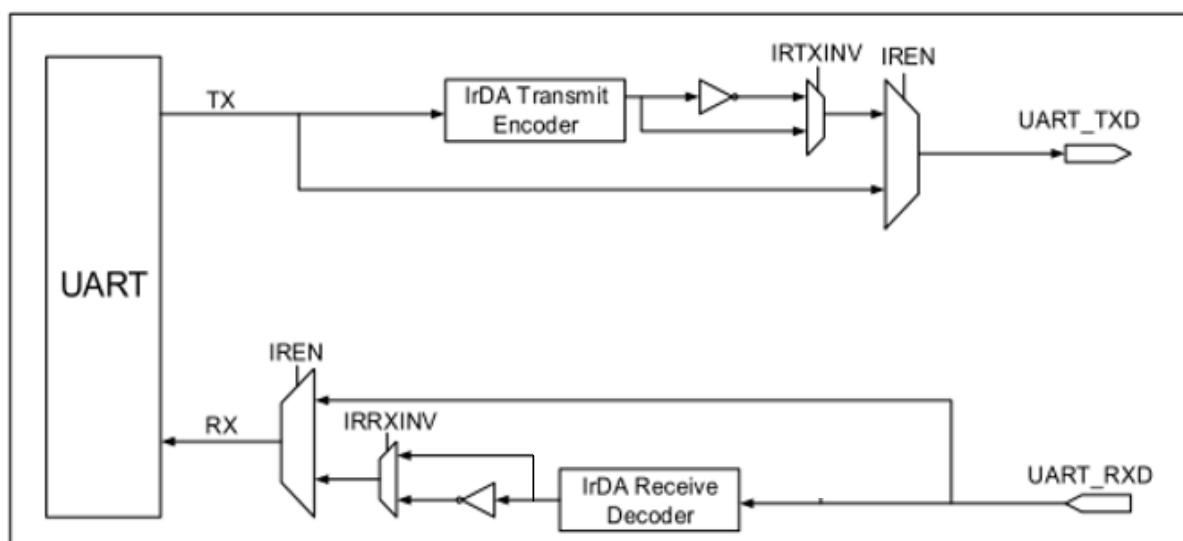


Figure 21-10 IrDA 结构框图

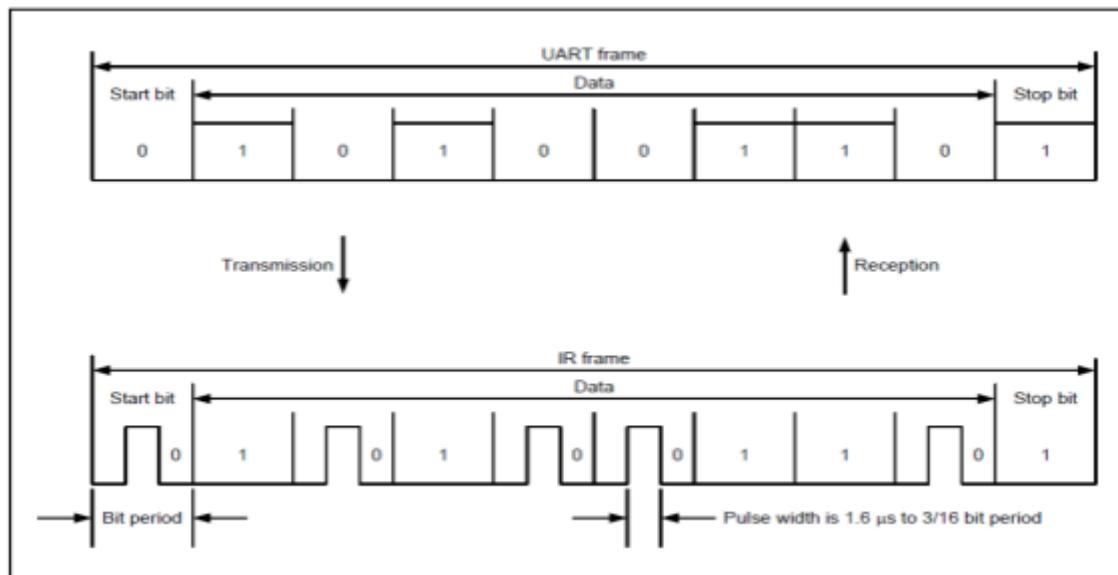


Figure 21-11 IrDA 收发脉冲



## 21.4 USART 工作模式

USART 工作模式 选择 如下表:

Table 21-1 USART 工作模式 选择表

LINEN	CLKEN	HDSEL	SCEN	IREN	STOP[1:0]	MODE
1	0	0	0	0	00	局域互联网(LIN)
0	1	0	0	0	X	同步(Synchronous)
0	0	1	0	0	X	单线半双工(Half)
0	1	0	1	0	X	智能卡(Smartcard)
0	0	0	0	1	00	红外线数据传输(IrDA)

Table 21-2 USART 工作模式控制表

MODE	Data length	STOP length	Baud rate	Break sequence detection	CK output
Asynchronous	USART_CR1.M (M=0:8 位; M=1:9 位)	0.5/1/1.5/2	USART_BRR 配置	10 or 11 "0"	none
LIN	8 位	1	USART_BRR 配置	13 "0"	none
Synchronous	USART_CR1.M (M=0:8 位; M=1:9 位)	0.5/1/1.5/2	USART_BRR 配置	10 or 11 "0"	Have output and sync with TX
Half	USART_CR1.M (M=0:8 位; M=1:9 位)	0.5/1/1.5/2	USART_BRR 配置	10 or 11 "0"	None
Smartcard	8 位	1.5 (0.5/1.5 when receiving)	USART_BRR 配置	none	Output, not related to TX
IrDA	USART_CR1.M (M=0:8 位; M=1:9 位)	1	PSC 配置 (low-power)	10 or 11 "0"	none



#### 21.4-1 USART 特征描述

字长可以通过编程 USART\_CR1 寄存器中的 M 位，选择成 8 或 9 位(见 图 166)。在起始位期间，TX 脚处于低电平，在停止位期间处于高电平。

空闲符号被视为完全由 “1” 组成的一个完整的数据帧，后面跟着包含了数据的下一帧的开始位。

断开符号 被视为在一个帧周期内全部收到 “0” (包括停止位期间，也是 “0” )。

在断开帧结束时，发送器再插入 1 或 2 个停止位来应答起始位。

发送和接收由一共享的波特率发生器驱动，当发送器和接收器的使能位分别置位时，分别为其产生时钟。

每个功能块的详细资料如下给出。

ZBT CONFIDENTIAL

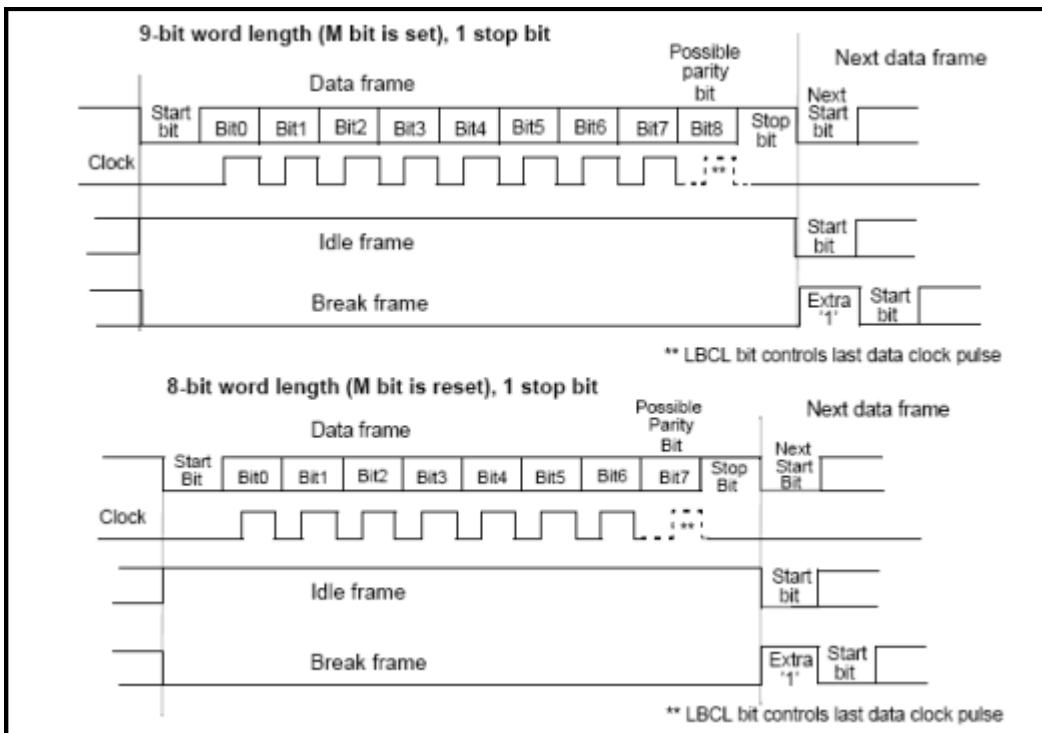


Figure 21-12 字长设置

## 21.4-2 发送器

发送器根据 M 位的状态发送 8 位或 9 位的数据字。当发送使能位 (TE) 被设置时，发送移位寄存器中的数据在 TX 脚上输出，相应的时钟脉冲在 SCLK 脚上输出。

### ● 可配置的停止位

随每个字符发送的停止位的位数可以通过控制寄存器 2 的位 13、12 进行编程。

- (1) 1 个停止位：停止位位数的默认值。
- (2) 2 个停止位：可用于常规 USART 模式、单线模式以及调制解调器模式。
- (3) 0.5 个停止位：在智能卡模式下接收数据时使用。
- (4) 1.5 个停止位：在智能卡模式下发送数据时使用。

空闲帧包括了停止位。断开帧(Frame)是 10 位低电平，后跟停止位 (当 m=0 时)；或者 11 位低电平，后跟停止位 (m=1 时)。不可能传输更长的断开帧 (长度大于 10 或者 11 位)

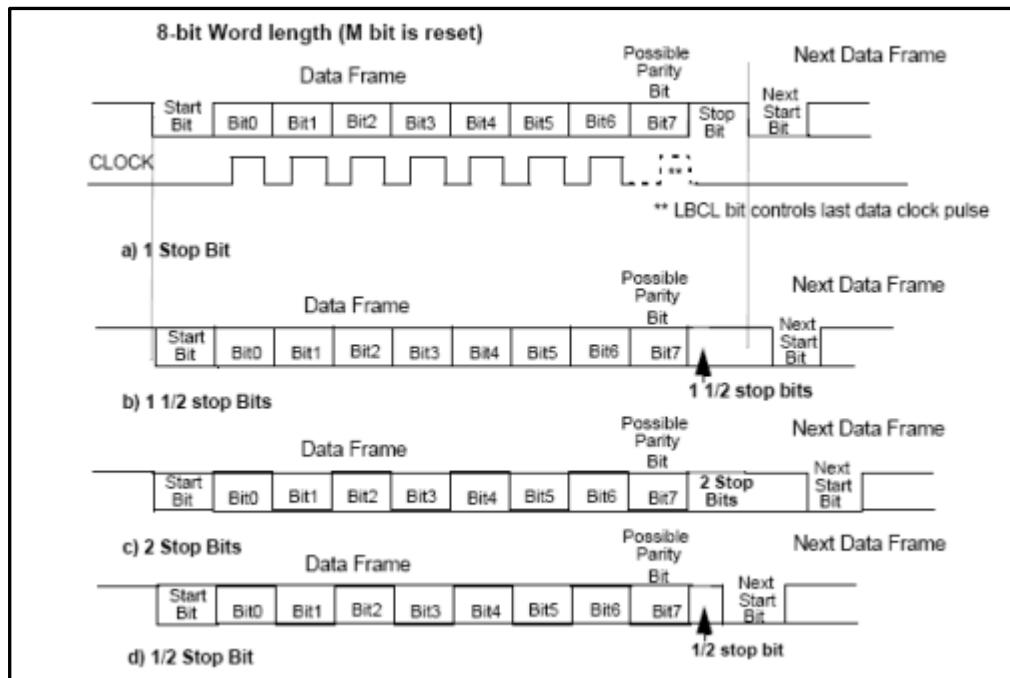


Figure 21-13 配置停止位

配置步骤：

- (1) 通过在 USART\_CR1 寄存器上置位 UE 位来激活 USART
- (2) 编程 USART\_CR1 的 M 位来定义字长。
- (3) 在 USART\_CR2 中编程停止位的位数。
- (4) 如果采用多缓冲器通信，配置 USART\_CR3 中的 DMA 使能位(DMAT)。按多缓冲器通信中的描述配置 DMA 寄存器。
- (5) 设置 USART\_CR1 中的 TE 位，发送一个空闲帧作为第一次数据发送。
- (6) 利用 USART\_BRR 寄存器选择要求的波特率。
- (7) 把要发送的数据写进 USART\_DR 寄存器(此动作清除 TXE 位)。在只有一个缓冲器的情况下，对每个待发送的数据重复步骤 7。



### ● 单字节通信

清零 TXE 位总是通过对数据寄存器的写操作来完成的。TXE 位由硬件来设置，它

表明：

- (1) 数据已经从 TDR 移送到移位寄存器，数据发送已经开始
- (2) TDR 寄存器被清空
- (3) 下一个数据可以被写进 USART\_DR 寄存器而不会覆盖先前的数据

如果 TXEIE 位被设置，此标志将产生一个中断。

如果此时 USART 正在发送数据，对 USART\_DR 寄存器的写操作把数据存进 TDR 寄存器，并在当前传输结束时把该数据复制进移位寄存器。

如果此时 USART 没有在发送数据，处于空闲状态，对 USART\_DR 寄存器的写操作直接把数据放进移位寄存器，数据传输开始，TXE 位立即被置起。当一帧发送完成时（停止位发送后），TC 位被置起，并且如果 USART\_CR1 寄存器中的 TCIE 位被置起时，中断产生。

在 USART\_DR 寄存器中写入了最后一个数据字后，在关闭 USART 模块之前或设置微控制器进入低功耗模式(详见下图)之前，必须先等待 TC=1。

使用下列软件过程清除 TC 位：

- (1) 先读一下 USART\_SR 寄存器。
- (2) 再写一下 USART\_DR 寄存器。

注意： TC 位也可以通过对它软件写 0 来清除。此清零方式只在多缓冲器通信模式下推荐使用。

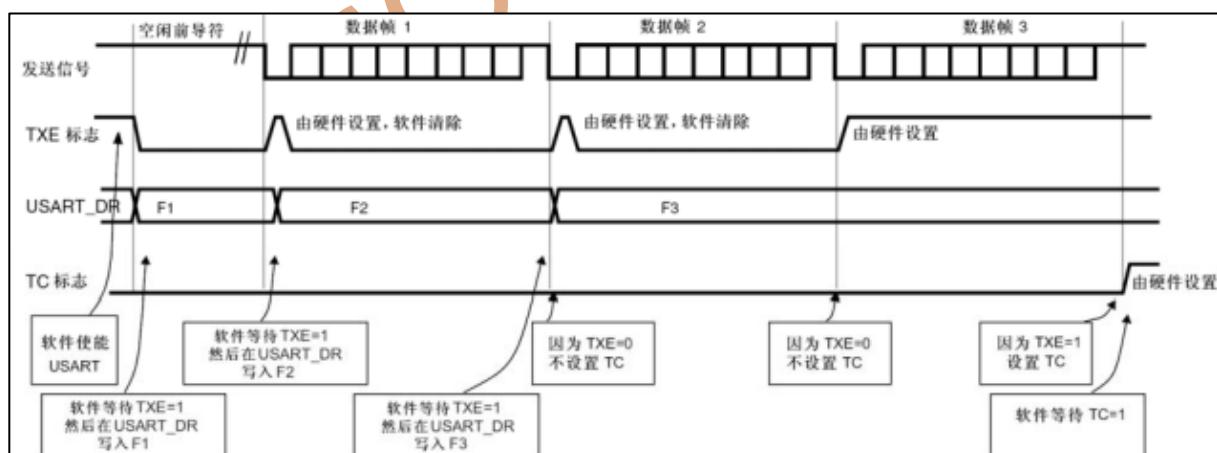


Figure 21-14 配置停止位



- 断开符号

置位 SBK 位可发送一个断开符号。断开帧长度取决 M 位(见图 Fig. 22-14)。如果 SBK 位被置 1，在完成当前数据发送后，将在 TX 线上发送一个断开符号。断开字符发送完成时（在断开符号的停止位时）SBK 被硬件复位。USART 在最后一个断开帧的结束处插入一逻辑 1 位，以保证能识别下一帧的起始位。

注意：如果在开始发送断开帧之前，软件又复位了 SBK 位，断开符号将不被发送。如果要发送两个连续的断开帧，SBK 位应该在前一个断开符号的停止位之后置。

- 空闲符号

置位 TE 将使得 USART 在第一个数据帧前发送一空闲帧。

#### 21.4.3 接收器

USART 可以根据 USART\_CR1 的 M 位接收 8 位或 9 位的数据字

- 字符接收

在 USART 接收期间，数据的最低有效位首先从 RX 脚移进。在此模式里，USART\_DR 寄存器包含的缓冲器位于内部总线和接收移位寄存器之间。

配置步骤：

1. 将 USART\_CR1 寄存器的 UE 置 1 来激活 USART。
2. 编程 USART\_CR1 的 M 位定义字长
3. 在 USART\_CR2 中编写停止位的个数
4. 如果需多缓冲器通信，选择 USART\_CR3 中的 DMA 使能位 (DMAT)。按多缓冲器通信所要求的配置 DMA 寄存器。
5. 利用波特率寄存器 USART\_BRR 选择希望的波特率。
6. 设置 USART\_CR1 的 RE 位。激活接收器，使它开始寻找起始位。

当一字符被接收到时，

- 1) RXNE 位被置位。它表明移位寄存器的内容被转移到 RDR。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志）。
- 2) 如果 RXNEIE 位被设置，产生中断。
- 3) 在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起，
- 4) 在多缓冲器通信时，RXNE 在每个字节接收后被置起，并由 DMA 对数据寄存器的读操作而清零。
- 5) 在单缓冲器模式里，由软件读 USART\_DR 寄存器完成对 RXNE 位清除。RXNE 标志也可以通过对它写 0 来清除。RXNE 位必须在下一字符接收结束前被清零，以避免溢出错误。

Note： 在接收数据时，RE 位不应该被复位。如果 RE 位在接收时被清零，当前字节的接收被丢失。



- 断开符号

当接收到一个断开帧时，USART 像处理帧错误一样处理它。

- 空闲符号

当一空闲帧被检测到时，其处理步骤和接收到普通数据帧一样，但如果 IDLEIE 位被设置将产生一个中断。

- 溢出错误

如果 RXNE 还没有被复位，又接收到一个字符，则发生溢出错误。数据只有当 RXNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。RXNE 标记是接收到每个字节后被置位的。如果下一个数据已被收到或先前 DMA 请求还没被服务时，RXNE 标志仍是置起的，溢出错误产生。

当溢出错误产生时：

- 1) ORE 位被置位。
- 2) RDR 内容将不会丢失。读 USART\_DR 寄存器仍能得到先前的数据。
- 3) 移位寄存器中以前的内容将被覆盖。随后接收到的数据都将丢失。
- 4) 如果 RXNEIE 位被设置或 EIE 和 DMAR 位都被设置，中断产生。
- 5) 顺序执行对 USART\_SR 和 USART\_DR 寄存器的读操作，可复位 ORE 位

注意：当 ORE 位置位时，表明至少有 1 个数据已经丢失。有两种可能：

- 1) 如果 RXNE=1，上一个有效数据还在接收寄存器 RDR 上，可以被读出。
- 2) 如果 RXNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。当上一个有效数据在 RDR 中被读取的同时又接收到新的（也就是丢失的）数据时，此种情况可能发生。在读序列期间（在 USART\_SR 寄存器读访问和 USART\_DR 读访问之间）接收到新的数据，此种情况也可能发生。

- 噪音错误

当在接收帧中检测到噪音时：

- 1) NE 在 RXNE 位的上升沿被置起。
- 2) 无效数据从移位寄存器移到 USART\_DR 寄存器。
- 3) 在单个字节通信情况下，没有中断产生。然而，NE 这个位和 RXNE 位同时置起，后者自己产生中断。在多缓冲器通信情况下，如果 USART\_CR3 寄存器中 EIE 位被置位的话，将产生一中断顺序执行对 USART\_SR 和 USART\_DR 寄存器的读操作，可复位 NE 位

- 帧(Frame)错误

当以下情况发生时检测到帧错误：

由于没有同步上或大量噪音的原因，停止位没有在预期的时间上接和收识别出来。

当帧错误被检测到时：

- 1) FE 位被硬件置起



- 2) 无效数据从移位寄存器传送到 USART\_DR 寄存器。
- 3) 在单个字节通信情况下，没有中断产生。然而，这个位和 RXNE 位同时置起，后者自己产生中断。在多缓冲器通信情况下，如果 USART\_CR3 寄存器中 EIE 位被置位的话，将产生一中断。  
顺序执行对 USART\_SR 和 USART\_DR 寄存器的读操作，可复位 FE 位。

#### 21.4-4 多处理器通信

通过 USART 可以实现多处理器通信（将几个 USART 连在一个网络里）。例如某个 USART 设备可以是主，它的 TX 输出和其他 USART 从设备的 RX 输入相连接；USART 从设备各自的 TX 输出逻辑地与在一起，并且和主设备的 RX 输入相连接。

在多处理器配置中，我们通常希望只有被寻址的接收者才被激活，来接收随后的数据，这样就可以减少由未被寻址的接收器的参与带来的多余的 USART 服务开销。

未被寻址的设备可启用其静默功能置于静默模式。在静默模式里：

- 1) 任何接收状态位都不会被设置。
- 2) 所有接收中断被禁止。
- 3) USART\_CR1 寄存器中的 RWU 位被置 1。RWU 可以被硬件自动控制或在某个条件下由软件写。

根据 USART\_CR1 寄存器中的 WAKE 位状态，USART 可以用二种方法进入或退出静默模式。

- 如果 WAKE 位被复位：进行空闲总线检测(WAKE=0)。

当 RWU 位被写 1 时，USART 进入静默模式。当检测到一空闲帧时，它被唤醒。

然后 RWU 被硬件清零，但是 USART\_SR 寄存器中的 IDLE 位并不置起。RWU 还可以被软件写 0。

Fig.22-17 给出利用空闲总线检测来唤醒和进入静默模式的一个例子

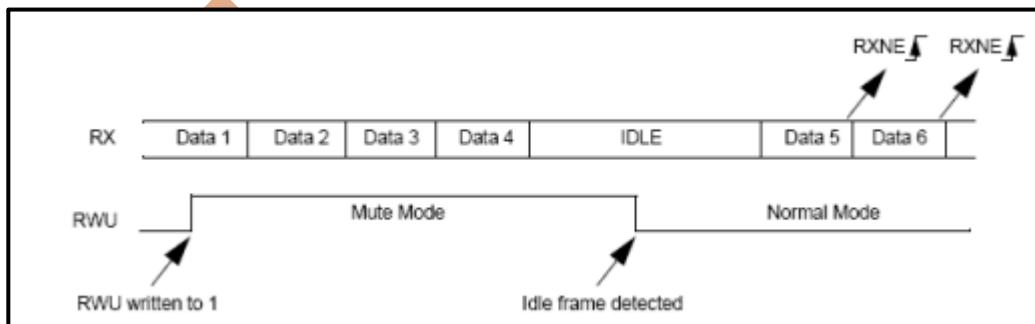


Figure 21-15 利用空闲总线检测的静默模式

- 如果 WAKE 位被设置：进行地址标记检测(WAKE=1)。

在这个模式里，如果 MSB 是 1，该字节被认为是地址，否则被认为是数据。在一个地址字节中，目标接收器的地址被放在 4 个 LSB 中。这个 4 位地址被接收器同它自己地址做比较，接收器的地址被编程



在 USART\_CR2 寄存器的 ADD。如果接收到的字节与它的编程地址不匹配时，USART 进入静默模式。该字节的接收既不会置起 RXNE 标志也不会产生中断或发出 DMA 请求，因为 USART 已经在静默模式。当接收到的字节与接收器内编程地址匹配时，USART 退出静默模式。然后 RWU 位被清零，随后的字节被正常接收。匹配的地址字节将置位 RXNE 位，因为 RWU 位已被清零。

当接收缓冲器不包含数据时（USART\_SR 的 RXNE=0），RWU 位可以被写 0 或 1。否则，该次写操作被忽略。Fig.22-18 给出利用地址标记检测来唤醒和进入静默模式的例子。

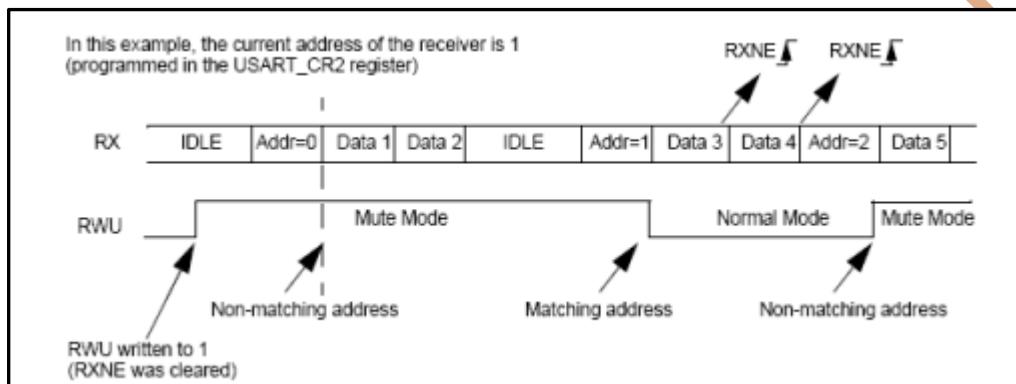


Figure 21-16 利用地址标记检测的静默模式

#### 21.4-5 校验控制

奇偶控制（发送时生成一个奇偶位，接收时进行奇偶校验）可以通过设置 USART\_CR1 寄存器上的 PCE 位而激活。根据 M 位定义的帧长度，可能的 USART 帧格式列在表 68 中。

M位	PCE位	USART帧
0	0	起始位 8位数据 停止位
0	1	起始位 7位数据 奇偶检验位 停止位
1	0	起始位 9位数据 停止位
1	1	起始位 8位数据 奇偶检验位 停止位

Table 21-3 USART 帧格式

注意：在用地址标记唤醒设备时，地址的匹配只考虑到数据的 MSB 位，而不用关心校验位。（MSB 是数据位中最后发出的，后面紧跟校验位或者停止位）**偶校验：**校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中“1”的个数为偶数。

例如：data=00110101，有 4 个“1”，如果选择偶校验（在 USART\_CR1 中的 PS=0），校验位将是 0。

**奇校验：**此校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中“1”的个数为奇数。

例如：data=00110101，有 4 个“1”，如果选择奇校验（在 USART\_CR1 中的 PS=1），校验位将是 1。



传输模式：如果 USART\_CR1 的 PCE 位被置位，写进数据寄存器的数据的 MSB 位被校验位替换后发送出去（如果选择偶校验偶数个 1，如果选择奇校验奇数个 1）。如果奇偶校验失败，USART\_SR 寄存器中的 PE 标志被置 1，并且如果 USART\_CR1 寄存器的 PEIE 在被预先设置的话，中断产生。

#### 21.4-6 LIN (局域互联网) 模式

LIN 模式是通过设置 USART\_CR2 寄存器的 LINEN 位选择的。在 LIN 模式里，下列位必须保持清零：

- 1) USART\_CR2 寄存器的 CLKEN 位
- 2) USART\_CR3 寄存器的 STOP[1:0], SCEN, HDSEL 和 IREN

##### 21.4-6.1 LIN 发送

21.4-2 节里所描述的同样步骤适用于 LIN 主发送，但和正常 USART 发送有以下区别：

- 1) 清零 M 位以配置 8-位字长
- 2) 置位 LINEN 位以进入 LIN 模式。这时，置位 SBK 将发送 13 位 “0” 作为断开符号。然后发一位 “1”，以允许对下一个开始位的检测。

##### 21.4-6.2 LIN 接收

当 LIN 模式被使能时，断开符号检测电路被激活。该检测完全独立于 USART 接收器。断开符号只要一出现就能检测到，不管是在总线空闲时还是在发送某数据帧期间，数据帧还未完成，又插入了断开符号发送。

当接收器被激活时(USART\_CR1 的 RE=1)，电路监测 RX 上的起始信号。监测起始位的方法同检测断开符号或数据是一样的。当起始位被检测到后，电路对每个接下来的位，在每个位的第 8, 9, 10 个过采样时钟点上进行采样。如果 10 个(当 USART\_CR2 的 LBDL = 0)或 11 个(当 USART\_CR2 的 LBDL = 1)连续位都是 “0”，并且又跟着一个定界符，USART\_SR 的 LBD 标志被设置。如果 LBDIE 位=1，中断产生。在确认断开符号前，要检查定界符，因为它意味 RX 线已经回到高电平。

如果在第 10 或 11 个采样点之前采样到了 “1”，检测电路取消当前检测并重新寻找起始位。如果 LIN 模式被禁止，接收器继续如正常 USART 那样工作，不需要考虑检测断开符号。

如果 LIN 模式被激活(LINEN=1)，只要一发生帧错误(也就是停止位检测到 “0”，这种情况出现在断开帧)，接收器就停止，直到断开符号检测电路接收到一个 “1” (这种情况发生于断开符号没有完整的发出来)，或一个定界符(这种情况发生于已经检测到一个完整的断开符号)。

如下圖 Fig 21-17 说明了断开符号检测器状态机的行为和断开符号标志的关系。

如下圖 Fig 21-18 给了一个断开帧的例子。

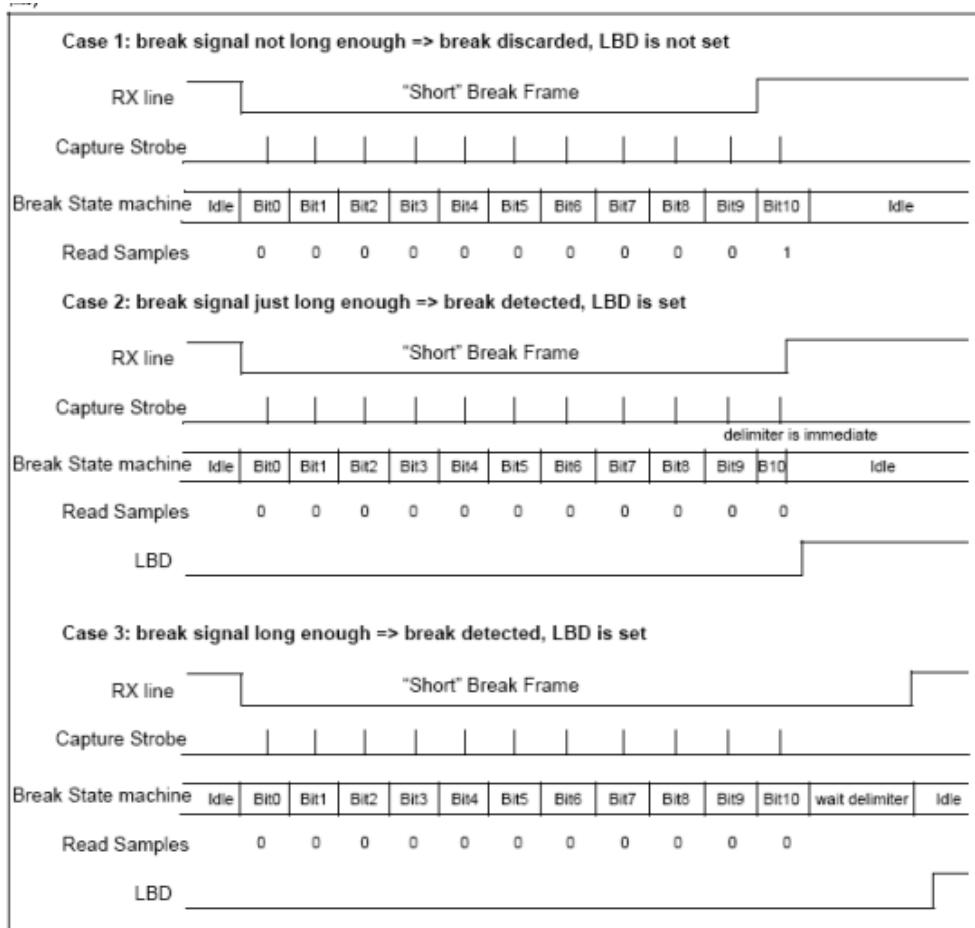


Figure 21-17 LIN 模式下的断开检测(11 位断开长度 – 设置了 LBDL 位)

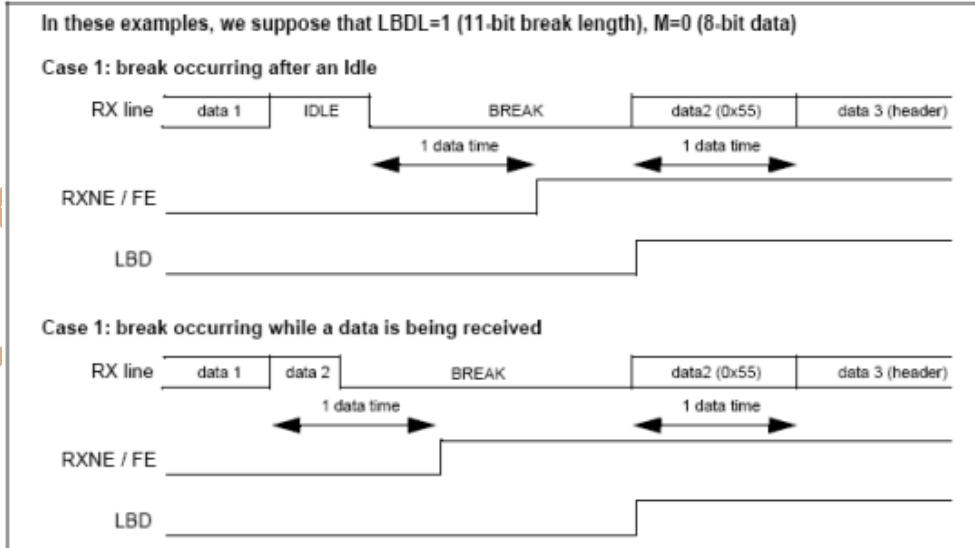


Figure 21-18 LIN 模式下的断开检测与帧错误的检测



## 21.4-7 USART 同步模式

通过在 USART\_CR2 寄存器上写 CLKEN 位选择同步模式

在同步模式里，下列位必须保持清零状态：

- 1) USART\_CR2 寄存器中的 LINEN 位
- 2) USART\_CR3 寄存器中的 SCEN,HDSEL 和 IREN 位

USART 允许用户以主模式方式控制双向同步串行通信。SCLK 脚是 USART 发送器时钟的输出。在起始位和停止位期间，SCLK 脚上没有时钟脉冲。根据 USART\_CR2 寄存器中 LBCL 位的状态，决定在最后一个有效数据位期间产生或不产生时钟脉冲。USART\_CR2 寄存器的 CPOL 位允许用户选择时钟极性，USART\_CR2 寄存器上的 CPHA 位允许用户选择外部时钟的相位(见 Fig22-24,25,26)。

在总线空闲期间，实际数据到来之前以及发送断开符号的时候，外部 SCLK 时钟不被激活。

同步模式时，USART 发送器和异步模式里工作一模一样。但是因为 SCLK 是与 TX 同步的(根据 CPOL 和 CPHA)，所以 TX 上的数据是随 SCLK 同步发出的。

同步模式的 USART 接收器工作方式与异步模式不同。如果 RE=1，数据在 SCLK 上采样(根据 CPOL 和 CPHA 决定在上升沿还是下降沿)，不需要任何的过采样。但必须考虑建立时间和持续时间(取决于波特率，1/16 位时间)。

注意：1. SCLK 脚同 TX 脚一起联合工作。因而，只有在发送器被激活(TE=1)，且数据被发送时(USART\_DR 寄存器被写入)才提供时钟。这意味着在没有发送数据时是不可能接收一个同步数据的。

2. LBCL,CPOL 和 CPHA 位的正确配置，应该在发送器和接收器都被禁止时；当发送器或接收器被激活时，这些位不能被改变
3. 建议在同一条指令中设置 TE 和 RE，以减少接收器的建立时间和保持时间。
4. USART 只支持主模式：它不能用来自其他设备的输入时钟接收或发送数据(SCLK 永远是输出)。

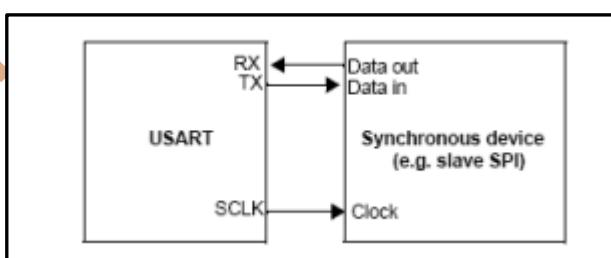


Figure 21-19 USART 同步传输的例子

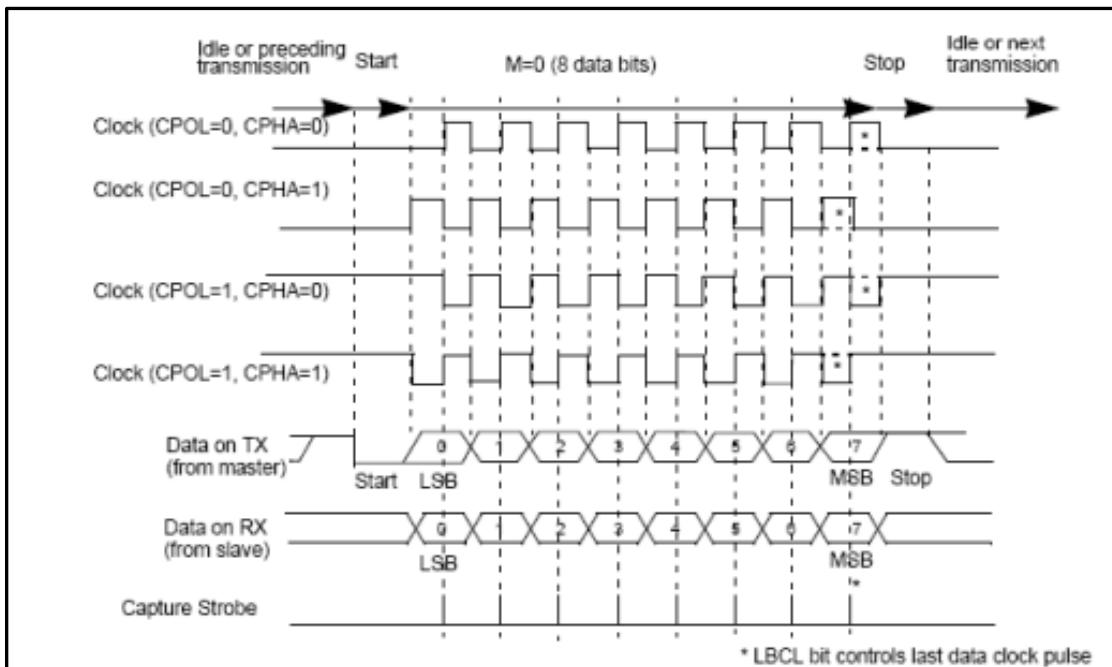


Figure 21-20 USART 数据时钟时序示例( $M=0$ )

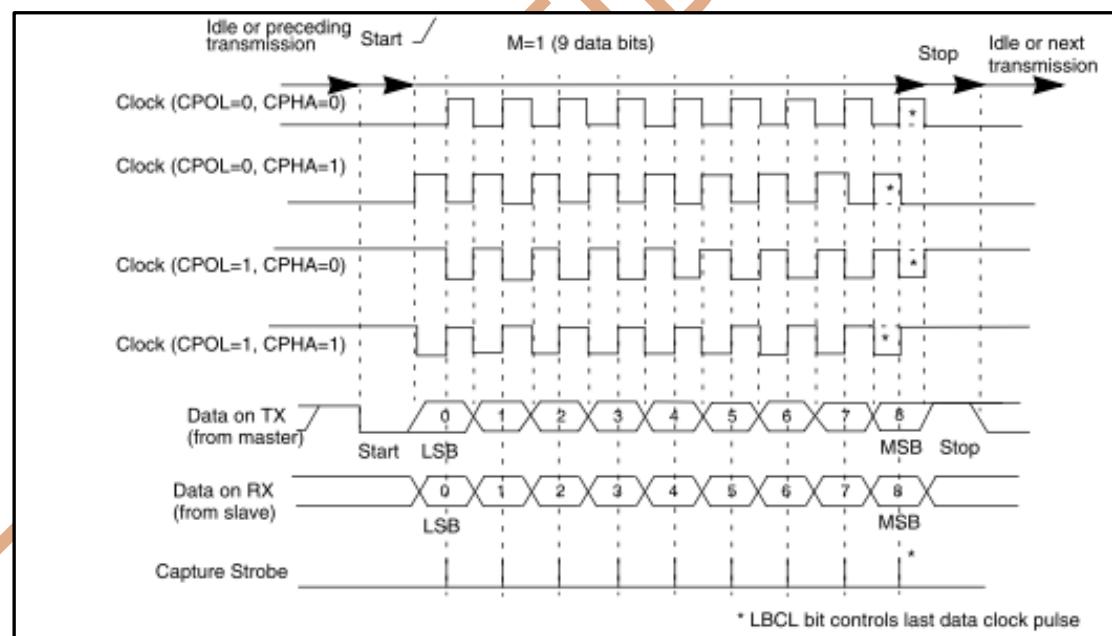


Figure 21-21 USART 数据时钟时序示例( $M=1$ )

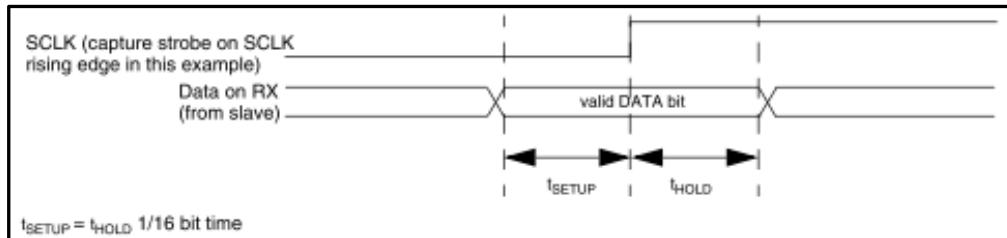


Figure 21-22 RX 数据采样/保持时间

#### 21.4-8 单线半双工通信模式

单线半双方模式通过设置 USART\_CR3 寄存器的 HDSEL 位选择。在这个模式里，下面的位必须保持清零状态：

- USART\_CR2 寄存器的 LINEN 和 CLKEN 位
- USART\_CR3 寄存器的 SCEN 和 IREN 位

USART 可以配置成遵循单线半双工协议。半双工和全双工通信是用控制位 "HALF DUPLEX SEL" 选择的。当 HDSEL 写 "1" 时

- RX 不再被使用
- 当没有数据传输时，TX 总是被释放。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口。这就意味该 I/O 在不被 USART 驱动时，必须配置成悬空输入（或开漏的输出高）。

除此以外，通信与正常 USART 模式类似。由软件来管理线上的冲突（例如通过使用一个中央仲裁器）。特别的是，发送从不会被硬件所阻碍。当 TE 位被设置时，只要数据一写到数据寄存器上，发送就继续。



### 21.4-9 智能卡模式

智能卡模式通过设置 USART\_CR3 寄存器的 SCEN 位选择。在智能卡模式里，下列位必须保持清零：

- USART\_CR2 寄存器的 LINEN 位
- USART\_CR3 寄存器的 HDSEL 位和 IREN 位

此外，CLKEN 位可以被设置，以提供时钟给智能卡。智能卡接口设计成 ISO7816-3 标准所定义的那样支持异步协议的智能卡。USART 应该被设置为：

- 8 位数据位加校验位：此时 USART\_CR1 寄存器 M=1,PCE=1，并且下列条件满足其一：
  - 接收时 0.5 个停止位：即 USART\_CR2 寄存器的 STOP=01
  - 发送时 1.5 个停止位：即 USART\_CR2 寄存器的 STOP=11

Fig. 22-25 给出的例子说明了数据线上，在有校验错误和没校验错误两种情况下的信号。

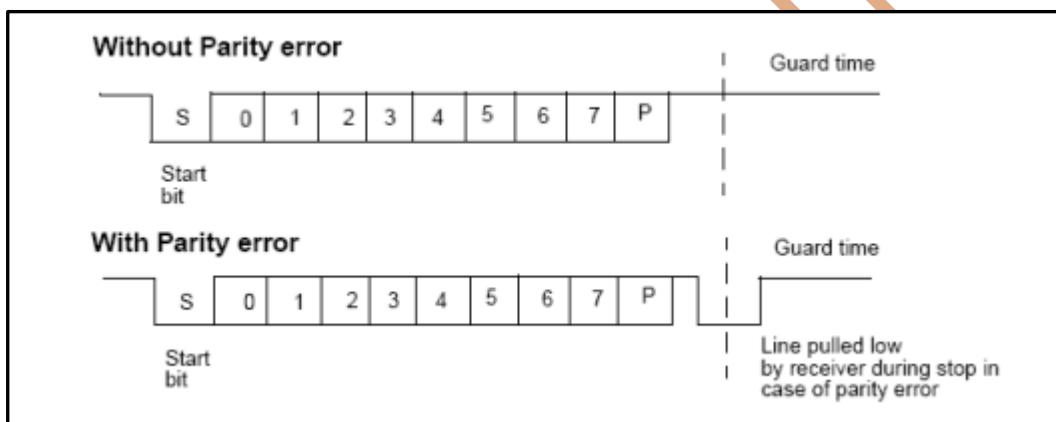


Figure 21-23 ISO7816-3 异步协议

当与智能卡相连接时，USART 的 TX 驱动一根智能卡也驱动的双向线。为了做到这点，SW\_RX 必须和 TX 连接到相同的 I/O 口。在发送开始位和数据字节期间，发送器的输出使能位 TX\_EN 被置起，在发送停止位期间被释放（弱上拉），因此在发现校验错误的情况下接收器可以将数据线拉低。如果 TX\_EN 不被使用，在停止位期间 TX 被拉到高电平：这样的话，只要 TX 配置成开漏，接收器也可以驱动这根线。



智能卡是一个单线半双工通信协议：

- (1) 从发送移位寄存器把数据发送出去，要被延时最小  $1/2$  波特时钟。在正常操作时，一个满的发送移位寄存器将在下一个波特时钟沿开始向外移出数据。在智能卡模式里，此发送被延迟  $1/2$  波特时钟。
- (2) 如果在接收数据帧期间，检测到一校验错误，该帧接收完成后(也就是在 0.5 停止位结束时)，发送线被拉低一个波特时钟周期。这是告诉智能卡发送到 USART 的数据没有被正确接收到。此 NACK 信号（拉低发送线一个波特时钟周期）在发送端将产生一个帧错误(发送端被配置成 1.5 个停止位)。应用程序可以根据协议处理重新发送的数据。如果 NACK 控制位被设置，发生校验错误时接收器会给出一个 NACK 信号；否则就不会发送 NACK。
- (3) TC 标志的置起可以通过编程保护时间寄存器得以延时。在正常操作时，当发送移位寄存器变空并且没有新的发送请求出现时，TC 被置起。在智能卡模式里，空的发送移位寄存器将触发保护时间计数器开始向上计数，直到保护时间寄存器中的值。TC 在这段时间被强制拉低。当保护时间计数器达到保护时间寄存器中的值时，TC 被置高。
- (4) TC 标志的撤销不受智能卡模式的影响。
- (5) 如果在发送器端检测到一个帧错误(收到接收器的 NACK 信号)，发送器的接收功能块不会把 NACK 当作起始位检测。根据 ISO 协议，接收到的 NACK 的持续时间可以是 1 或 2 波特时钟周期。
- (6) 在接收器这边，如果一个校验错误被检测到，并且 NACK 被发送，接收器不会把 NACK 检测成起始位。

#### 21.4-10 红外线数据传输(IrDA)

IrDA 模式是通过设置 USART\_CR3 寄存器的 IREN 位选择的。在 IRDA 模式里，下列位必须保持清零

- USART\_CR2 寄存器的 LINEN,STOP 和 CLKEN 位
- USART\_CR3 寄存器的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零调制方案 (RZI)，该方案用一个红外光脉冲代表逻辑 0 (如图 Figure 22.26 所示)。SIR 发送编码器对从 USART 输出的 NRZ (非归零) 比特流进行调制。输出脉冲流被传送到一个外部输出驱动器和红外 LED。USART 为 IR ENDEC 最高只支持到 115.2Kbps 速率。在正常模式里，脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器对来自红外接收器的归零位比特流进行解调，并将接收到的 NRZ 串行比特流输出到 USART。在空闲状态里，解码器输入通常是高(标记状态 marking state)。发送编码器输出的极性和解码器的输入相反。当解码器输入低时，检测到一个起始位。

- IrDA 是一个半双工通信协议。如果发送器忙(也就是 USART 正在送数据给 IrDA 编码器)，IrDA 接收线上的任何数据将被 IrDA 解码器忽视。如果接收器忙(也就是 USART 正在接收从 IrDA 解码器来的



解码数据), 从 USART 到 IrDA 的 TX 上的数据将不会被 IrDA 编码。当接收数据时, 应该避免发送, 因为将被发送的数据可能被破坏。

- SIR 发送逻辑把 “0” 作为高脉冲发送, 把 “1” 作为低电平发送。脉冲的宽度规定为正常模式时位周期的 3/16(如图 Figure 22.27 所示)。
- SIR 接收逻辑把高电平状态解释为 “1”, 把低脉冲解释为 “0”。
- 发送编码器输出与解码器输入有着相反的极性。当空闲时, SIR 输出处于低状态。
- SIR 解码器把 IrDA 兼容的接收信号转变成给 USART 的比特流。
- IrDA 规范要求脉冲要宽于 1.41us。脉冲宽度是可编程的。接收器端的尖峰脉冲检测逻辑滤除宽度小于 2 个 PSC 周期的脉冲 (PSC 是在 IrDA 低功耗波特率寄存器 USART\_GTPR 中编程的预分频值)。宽度小于 1PSC 周期的脉冲一定被滤除掉, 但是那些宽度大于 1 个而小于 2 个 PSC 周期的脉冲可能被接收或滤除, 那些宽度大于 2 个周期的将被视为一个有效的脉冲。当 PSC=0 时, IrDA 编码器/解码器不工作。
- 接收器可以与一低功耗发送器通信。
- 在 IrDA 模式里, USART\_CR2 寄存器上的 STOP 位必须配置成 1 个停止位。



#### 21.4-10.1 IrDA 低功耗模式

##### 发送器

在低功耗模式，脉冲宽度不再持续  $3/16$  个位周期。取而代之，脉冲的宽度是低功耗波特率的 3 倍，它最小可以是 1.42MHz。通常这个值是 1.8432MHz ( $1.42 \text{ MHz} < \text{PSC} < 2.12 \text{ MHz}$ )。一个低功耗模式可编程分频器把系统时钟进行分频以达到这个值。

##### 接收器

低功耗模式的接收类似于正常模式的接收。为了滤除尖峰干扰脉冲，USART 应该滤除宽度短于 1 个 PSC 的脉冲。只有持续时间大于 2 个周期的 IrDA 低功耗波特率时钟（USART\_GTPR 中的 PSC）的低电平信号才被接受为有效的信号。

Note : (1) 宽度小于 2 个大于 1 个 PSC 周期的脉冲可能会也可能不被滤除。

(3) 接收器的建立时间应该由软件管理。IrDA 物理层技术规范规定了在发送和接收之间最小要有 10ms 的延时（IrDA 是一个半双工协议）。

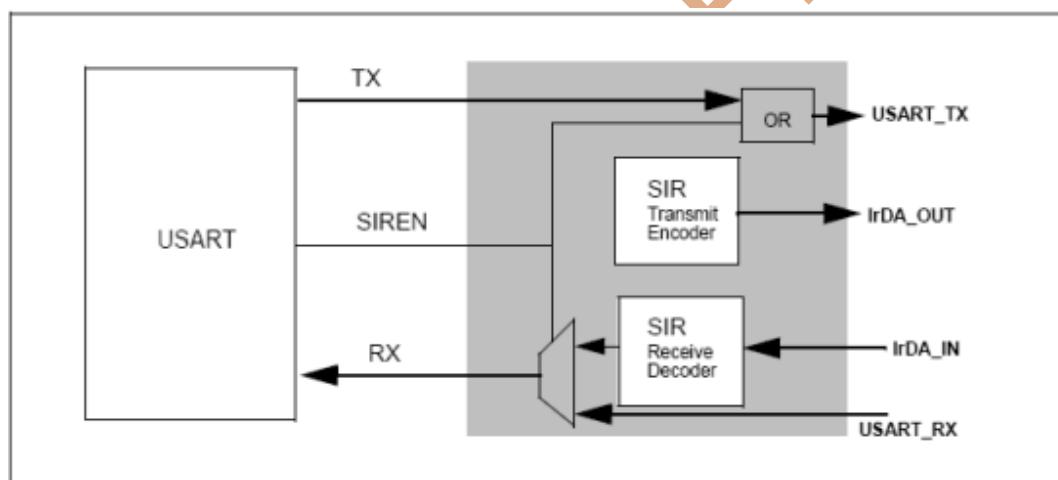


Figure 21-24 IrDA 框图

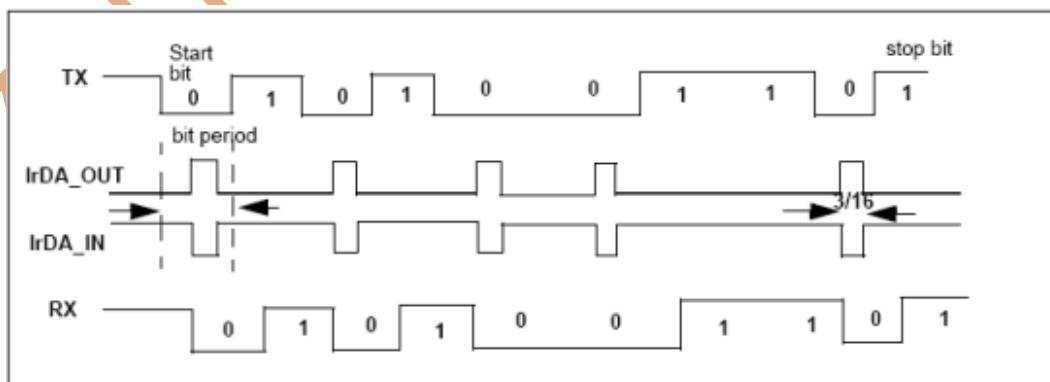
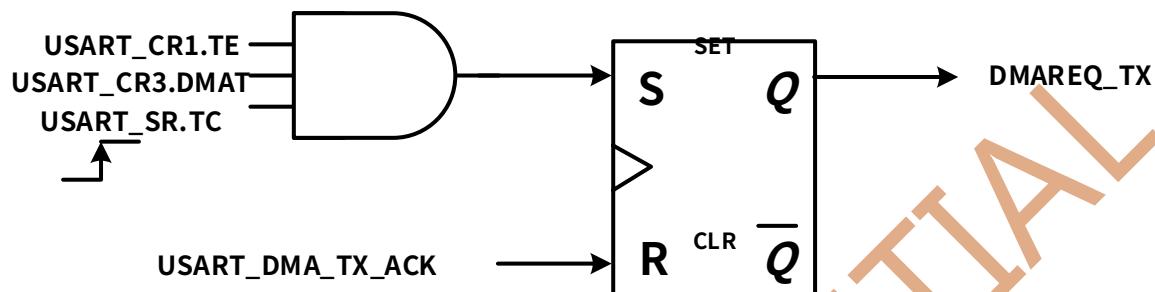


Figure 21-25 IrDA 数据调制

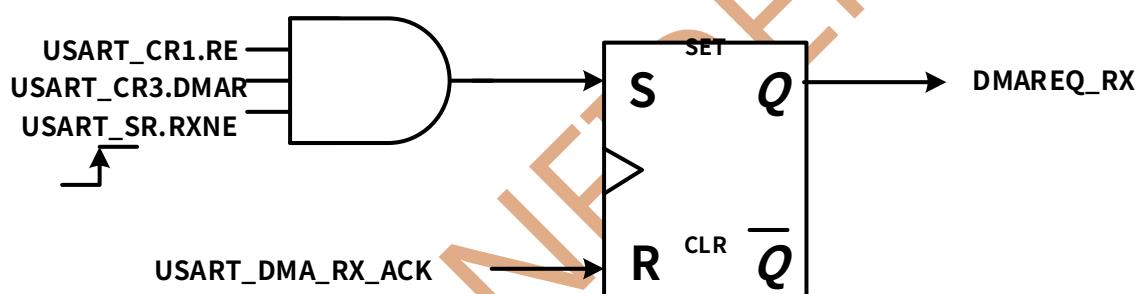


#### 21.4-11 使用 DMA 通信

- DMA 传送请求条件



- DMA 接收请求条件



#### 21.4-12 硬件流控制

利用 nCTS 输入和 nRTS 输出可以控制 2 个设备间的串行数据流。图 181 表明在这个模式里如何连接 2 个设备。

通过将 UASRT\_CR3 中的 RTSE 和 CTSE 置位，可以分别独立地使能 RTS 和 CTS 流控制。

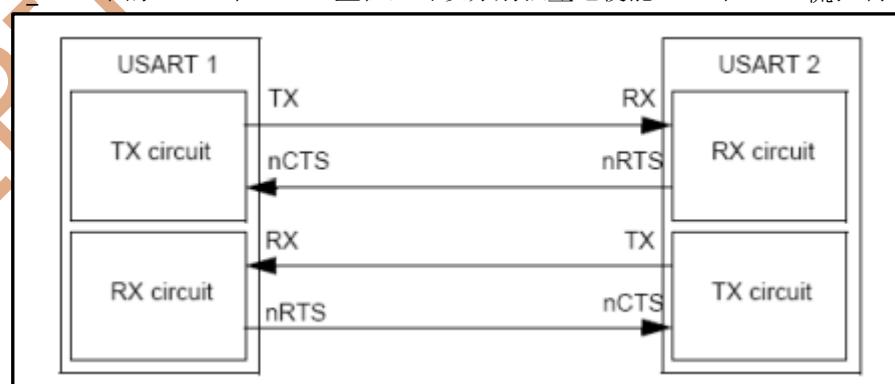


Figure 21-26 两个 USART 间的硬件流控制



#### 21.4-12.1 RTS 流控制

如果 RTS 流控制被使能(RTSE=1)，只要 USART 接收器准备好接收新的数据，nRTS 就变成有效(接低电平)。当接收寄存器内有数据到达时，nRTS 被释放，由此表明希望在当前帧结束时停止数据传输。

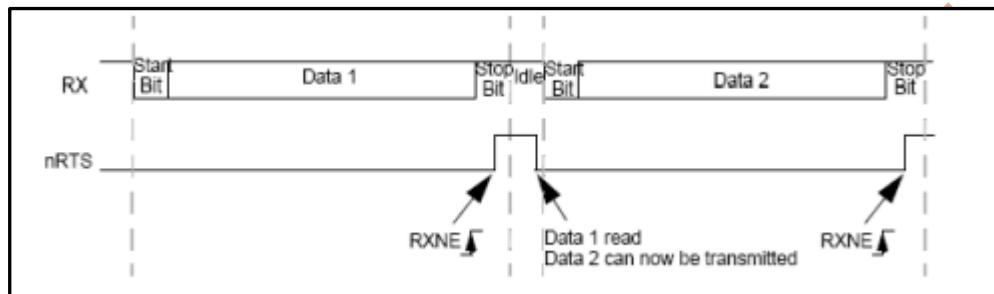


Figure 21-27 RTS 流控制

#### 21.4-12.2 CTS 流控制

如果 CTS 流控制被使能(CTSE=1)，发送器在发送下一帧前检查 nCTS 输入。如果 nCTS 有效(被拉成低电平)，则下一个数据被发送(假设那个数据是准备发送的，也就是 TXE=0)，否则下一帧数据不被发出去。若 nCTS 在传输期间被变成无效，当前的传输完成后停止发送。

当 CTSE=1 时，只要 nCTS 输入一变换状态，CTSIF 状态位就自动被硬件设置。它表明接收器是否准备进行通信。如果 USART\_CT3 寄存器的 CTSIE 位被设置，中断产生.

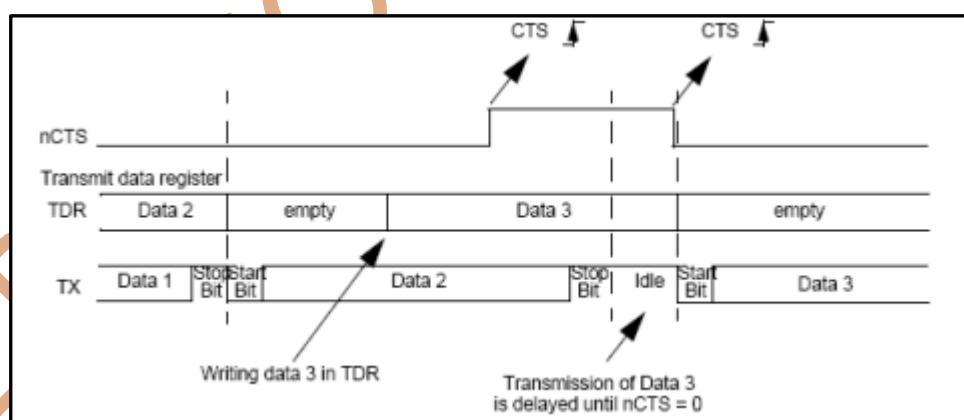


Figure 21-28 CTS 流控制



#### 21.4-13 中断请求

中断请求设定如下表：

Table 21-4 USART 中断请求

中断事件	事件标志	使能位
发送数据寄存器空	TXE	TXEIE
CTS标志	CTS	CTSIE
发送完成	TC	TCIE
接收数据就绪可读	TXNE	TXNEIE
检测到数据溢出	ORE	
检测到空闲线路	IDLE	IDLEIE
奇偶检验错	PE	PEIE
断开标志	LBD	LBDIE
噪声标志，多缓冲通信中的溢出错误和帧错误	NE或ORT或FE	EIE

USART 的各种中断事件被连接到同一个中断向量(见 图 184)，有以下各种中断事件：

- 发送期间：发送完成中断、清除发送中断、发送数据寄存器空中断。
- 接收期间：空闲总线检测中断、溢出错误中断、接收数据寄存器非空中断、校验错误中断、LIN 断开符号检测中断、噪音中断（仅在多缓冲器通信）和帧错误中断（仅在多缓冲器通信）。

如果对应的使能控制位被设置，这些事件就会产生各自的中断。

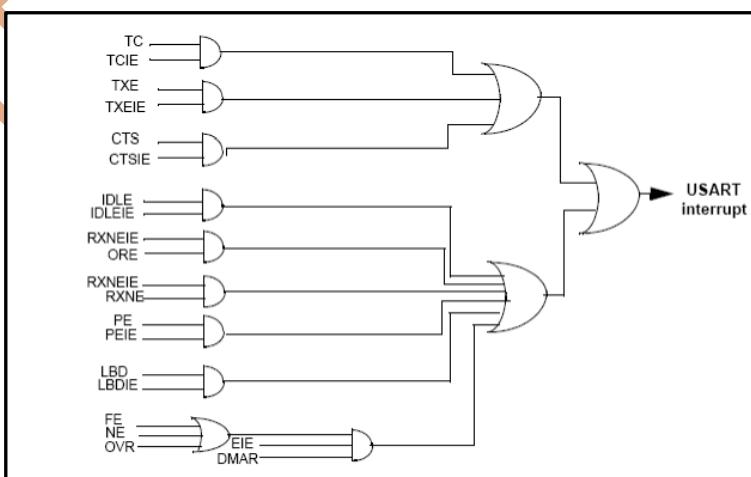


Figure 21-29 USART 中断映像图



## 21.5 不同波特率的分频设置

波特率	PCLK = 1 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	26	2403.85	0.16%	13	2403.85	0.16%
4800	13	4807.69	0.16%	7	4464.29	-6.99%
9600	7	8928.57	-6.99%	3	10416.67	8.51%
19200	3	20833.33	8.51%	2	15625.00	-18.62%
38400	2	31250.00	-18.62%	1	31250.00	-18.62%
57600	1	62500.00	8.51%	1	31250.00	-45.75%
76800	1	62500.00	-18.62%	0	-	-
115200	1	62500.00	-45.75%	0	-	-

波特率	PCLK = 4 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	104	2403.85	0.16%	52	2403.85	0.16%
4800	52	4807.69	0.16%	26	4807.69	0.16%
9600	26	9615.38	0.16%	13	9615.38	0.16%
19200	13	19230.77	0.16%	7	17857.14	-6.99%
38400	7	35714.29	-6.99%	3	41666.67	8.51%
57600	4	62500.00	8.51%	2	62500.00	8.51%
76800	3	83333.33	8.51%	2	62500.00	-18.62%
115200	2	125000.00	8.51%	1	125000.00	8.51%



波特率	PCLK = 10 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	260	2403.85	0.16%	130	2403.85	0.16%
4800	130	4807.69	0.16%	65	4807.69	0.16%
9600	65	9615.38	0.16%	33	9469.70	-1.36%
19200	33	18939.39	-1.36%	16	19531.25	1.73%
38400	16	39062.50	1.73%	8	39062.50	1.73%
57600	11	56818.18	-1.36%	5	62500.00	8.51%
76800	8	78125.00	1.73%	4	78125.00	1.73%
115200	5	125000.00	8.51%	3	104166.67	-9.58%

波特率	PCLK = 14 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	365	2397.26	-0.11%	182	2403.85	0.16%
4800	182	4807.69	0.16%	91	4807.69	0.16%
9600	91	9615.38	0.16%	46	9510.87	-0.93%
19200	46	19021.74	-0.93%	23	19021.74	-0.93%
38400	23	38043.48	-0.93%	11	39772.73	3.57%
57600	15	58333.33	1.27%	8	54687.50	-5.06%
76800	11	79545.45	3.57%	6	72916.67	-5.06%
115200	8	109375.00	-5.06%	4	109375.00	-5.06%



波特率	PCLK = 20 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	521	2399.23	-0.03%	260	2403.85	0.16%
4800	260	4807.69	0.16%	130	4807.69	0.16%
9600	130	9615.38	0.16%	65	9615.38	0.16%
19200	65	19230.77	0.16%	33	18939.39	-1.36%
38400	33	37878.79	-1.36%	16	39062.50	1.73%
57600	22	56818.18	-1.36%	11	56818.18	-1.36%
76800	16	78125.00	1.73%	8	78125.00	1.73%
115200	11	113638.36	-1.36%	5	125000.00	8.51%

波特率	PCLK = 24 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	625	2400.00	0.00%	313	2396.17	-0.16%
4800	313	4792.33	-0.16%	156	4807.69	0.16%
9600	156	9615.38	0.16%	78	9615.38	0.16%
19200	78	19230.77	0.16%	39	19230.77	0.16%
38400	39	38461.54	0.16%	20	37500.00	-2.34%
57600	26	57692.31	0.16%	13	57692.31	0.16%
76800	20	75000.00	-2.34%	10	75000.00	-2.34%
115200	13	115384.62	0.16%	7	107142.86	-6.99%



波特率	PCLK = 2 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	52	2403.85	0.16%	26	2403.85	0.16%
4800	26	4807.69	0.16%	13	4807.69	0.16%
9600	13	9615.38	0.16%	7	8928.57	-6.99%
19200	7	17857.14	-6.99%	3	20833.33	8.51%
38400	3	41666.67	8.51%	2	31250.00	-18.62%
57600	2	62500.00	8.51%	1	62500.00	8.51%
76800	2	62500.00	-18.62%	1	62500.00	-18.62%
115200	1	125000.00	8.51%	1	62500.00	-45.75%

波特率	PCLK = 8 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	208	2403.85	0.16%	104	2403.85	0.16%
4800	104	4807.69	0.16%	52	4807.69	0.16%
9600	52	9615.38	0.16%	26	9615.38	0.16%
19200	26	19230.77	0.16%	13	19230.77	0.16%
38400	13	38461.54	0.16%	7	35714.29	-6.99%
57600	9	55555.56	-3.55%	4	62500.00	8.51%
76800	7	71428.57	-6.99%	3	83333.33	8.51%
115200	4	125000.00	8.51%	2	125000.00	8.51%



波特率	<b>PCLK = 11.0592 MHz</b>					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	288	2400.00	0.00%	144	2400.00	0.00%
4800	144	4800.00	0.00%	72	4800.00	0.00%
9600	72	9600.00	0.00%	36	9600.00	0.00%
19200	36	19200.00	0.00%	18	19200.00	0.00%
38400	18	38400.00	0.00%	9	38400.00	0.00%
57600	12	57600.00	0.00%	6	57600.00	0.00%
76800	9	76800.00	0.00%	5	69120.00	-10.00%
115200	6	115200.00	0.00%	3	115200.00	0.00%

波特率	<b>PCLK = 16 MHz</b>					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%
57600	17	58823.53	2.12%	9	55555.56	-3.55%
76800	13	76923.08	0.16%	7	71428.57	-6.99%
115200	9	111111.11	-3.55%	4	125000.00	8.51%



波特率	PCLK = 22.12 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	576	2400.17	0.01%	288	2400.17	0.01%
4800	288	4800.35	0.01%	144	4800.35	0.01%
9600	144	9600.69	0.01%	72	9600.69	0.01%
19200	72	19201.39	0.01%	36	19201.39	0.01%
38400	36	38402.78	0.01%	18	38402.78	0.01%
57600	24	57604.17	0.01%	12	57604.17	0.01%
76800	18	76805.56	0.01%	9	76805.56	0.01%
115200	12	115208.33	0.01%	6	115208.33	0.01%



## 21.6 UART/USART 寄存器列表

UART4 基地址: 0x4000 4C00

UART5 基地址: 0x4000 5000

偏移地址	名称	描述	复位值
0x00	UARTx_SCON	控制寄存器	0x0000 0000
0x04	UARTx_SBUF	数据寄存器	0x0000 0000
0x08	UARTx_SADDR	地址寄存器	0x0000 0000
0x0C	UARTx_SADEN	地址掩码寄存器	0x0000 0000
0x10	UARTx_INTSR	中断标志位寄存器	0x0000 0000
0x14	UARTx_INTCLR	中断标志位清除寄存器	0x0000 0000
0x18	UARTx_BAUDCR	波特率控制寄存器	0x0000 0000
0x1C	UARTx_IRDACR	IrDA 控制寄存器	0x0000 0000

Table 21-5 USART 寄存器列表

USART1 基地址: 0x4001 3800

USART2 基地址: 0x4000 4400

USART3 基地址: 0x4000 4800

偏移地址	名称	描述	复位值
0x00	USARTx_SR	状态寄存器	0x0000 00C0
0x04	USARTx_DR	数据寄存器	0x0000 0000
0x08	USARTx_BRR	波特比率寄存器寄存器	0x0000 0000
0x0C	USARTx_CR1	控制寄存器1寄存器	0x0000 0000
0x10	USARTx_CR2	控制寄存器2寄存器	0x0000 0000
0x14	USARTx_CR3	控制寄存器3寄存器	0x0000 0000
0x18	USARTx_GTPR	保护时间和预分频寄存器寄存器	0x0000 0000
0x1C			
0x20			

Table 21-6 USART 寄存器列表



## 21.7 UART4,5 寄存器说明

### 21.7-1 UART 控制寄存器(UARTx\_SCON)

地址偏移: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
DBAUD	FEEN	SM01[1:0]	SM2	REN	TB8	RB8	TIEN	RIEN							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							

位	标记	功能描述	复位值	读写																									
31:10	-	保留	0x0	-																									
9	DBAUD	双倍波特率 0: 单倍波特率 1: 双倍波特率	0	R/W																									
8	FEEN	接收帧错误中断使能 0: Disable 1: Enable	0	R/W																									
7:6	SM0: SM1	工作模式 00: Mode 0; 01: Mode 1; 10: Mode 2; 11: Mode 3 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th>SM0</th><th>SM1</th><th>MODE</th><th>描述</th><th>波特率</th></tr> <tr> <td>0</td><td>0</td><td>0</td><td>位移寄存器</td><td>CLK/12</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>8 位串口传输</td><td>可变波特率</td></tr> <tr> <td>1</td><td>0</td><td>2</td><td>9 位串口传输</td><td>CLK/32, CLK/64</td></tr> <tr> <td>1</td><td>1</td><td>3</td><td>9 位串口传输</td><td>可变波特率</td></tr> </table>	SM0	SM1	MODE	描述	波特率	0	0	0	位移寄存器	CLK/12	0	1	1	8 位串口传输	可变波特率	1	0	2	9 位串口传输	CLK/32, CLK/64	1	1	3	9 位串口传输	可变波特率	0x0	R/W
SM0	SM1	MODE	描述	波特率																									
0	0	0	位移寄存器	CLK/12																									
0	1	1	8 位串口传输	可变波特率																									
1	0	2	9 位串口传输	CLK/32, CLK/64																									
1	1	3	9 位串口传输	可变波特率																									

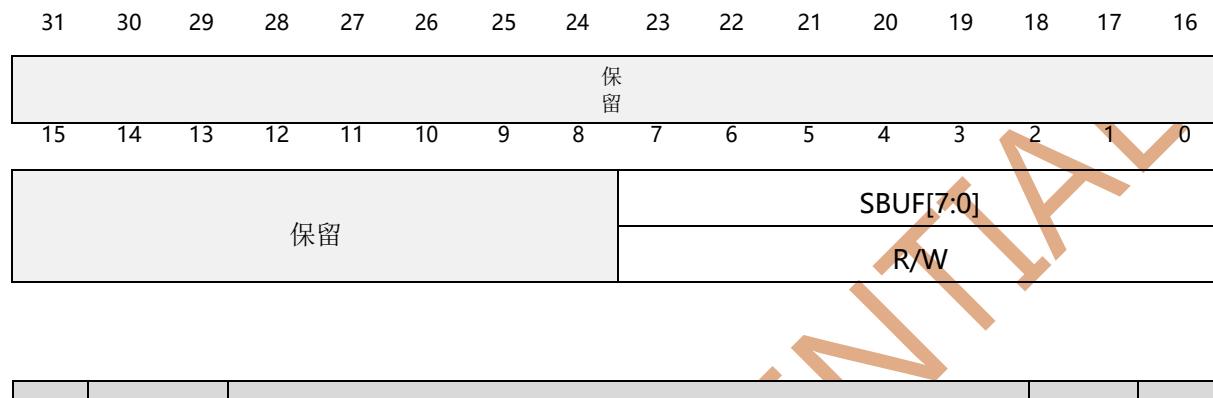


位	标记	功能描述	复位值	读写
5	SM2	<p>多主机通讯 0: Disable 1: Enable</p> <p>SM2: 软件配置多机通讯以及自动地址配模式1: 启动多从机通讯以及地址自动匹配</p> <p>0: 关闭多从机通讯以及地址自动匹配在模式 2 和模式 3 中:</p> <p>(1) 如果 SM2=1, 并且 REN=1, 则接收机处于地址帧监测模式, 可以使用接收到的第 9 位 RB8 来进行地址筛选。RB8=1 为地址帧, 通讯数据可以进入 SBUF, 置位 RI, 进入中断服务程序中进行地址比较; RB8=0 为数据帧, 接收机忽略这些数据帧并保持 RI=0。</p> <p>(2) 如果 SM2=0, 并且 REN=1, 则接收机不使用地址监测模式, 无论收到的 RB8 为 0 或 1, 都直接接收并且进入 SBUF, 置位 RI, RB8 在这种模式下为校验位。</p>	0	R/W
4	REN	<p>接收使能 Mode 0: 0: 发送, 1: 接收或其他: 0: 发送, 1: 接收/发送</p>	0	R/W
3	TB8	发送TB8 位	0	R/W
2	RB8	接收RB8 位	0	R/W
1	TIEN	<p>发送完成中断使能 0: Disable 1: Enable</p>	0	R/W
0	RIEN	<p>接收完成中断使能 0: Disable 1: Enable</p>	0	R/W



### 21.7-2 UART 数据寄存器(UARTx\_SBUF)

地址偏移: 0x04 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SBUF[7:0]	发送数据时，当发送数据写入该寄存器；接收数据时，数据接收完毕后，从该寄存器中读出。 注意：对该寄存器读的值实际是 RXBuffer 中的值，对该寄存器写的值实际是写到了 TXShifter 中。	0x0	R/W



### 21.7-3 UART 地址寄存器(UARTx\_SADDR)

地址偏移: 0x08 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SADDR[7:0]	从机设备地址寄存器	0x0	R/W



#### 21.7-4 UART 地址掩码寄存器(UARTx\_SADEN)

地址偏移: 0x0C 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SADEN[7:0]	从机设备地址掩码寄存器	0x0	R/W



### 21.7-5 UART 标志位寄存器(UARTx\_INTSR)

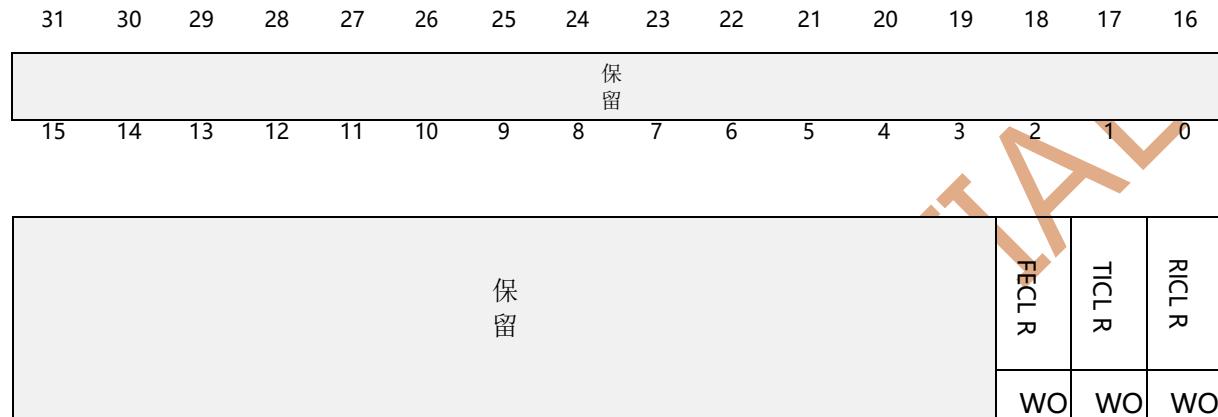
地址偏移: 0x10 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留															0	
保留															FE	
保留															TI	
保留															RI	
位	标记	功能描述													复位值	读写
31:3	-	保留													0x0	-
2	FE	接收帧错误标志位, 硬件置位, 软件清零 0: FE 中断无效 1: FE 中断有效													0	RO
1	TI	发送完成中断标志位, 硬件置位, 软件清零 0: TI 中断无效 1: TI 中断有效													0	RO
0	RI	接收完成中断标志位, 硬件置位, 软件清零 0: RI 中断无效 1: RI 中断有效													0	RO



### 21.7-6 UART 标志位清除寄存器(UARTx\_INTCLR)

地址偏移: 0x10 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:3	-	保留	0x0	-
2	FECLR	清除接收帧错误标志位; 写 1 清零, 写 0 无效	0x0	WO
1	TICLR	清除发送完成中断标志位; 写 1 清零, 写 0 无效	0x0	WO
0	RICLR	清除接收完成中断标志位; 写 1 清零, 写 0 无效	0x0	WO



### 21.7-7 UART 波特率控制寄存器(UARTx\_BAUDCR)

地址偏移: 0x18 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
保留														SELF_BRG
保留														R/W

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

位	标记	功能描述	复位值	读写
31:17	-	保留	0x0	-
16	SELF_BRG	UART 波特率选择位: 0: UART 的波特率由 timer 产生 1: UART 的波特率由 (DBAUD+1)*FPCLK/(32*(BRG[15:0]+1))生成	0	R/W
15:0	BRG[15:0]	UART 自动波特率生成配置位: 波特率=(DBAUD+1)*FPCLK/(32*(BRG[15:0]+1))	0x0	R/W



### 21.7-8 rDA 控制寄存器(UARTx\_IRDACR)

地址偏移: 0x1C 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
11	IRLP MODE	Ir 低功耗模式 0: Ir 普通模式 1: Ir 低功耗模式	0	R/W
10	IRRXINV	IrRXD 数据反转位 0: 不反转 1: 反转输入	0	R/W
9	IRTXINV	IrTXD 数据反转位 0: 不反转 1: 反转输出	0	R/W
8	IREN	IrDA 使能位 0: 无效 1: 使能	0	R/W
7:0	PSC	PSC[7:0] 红外模式发送, 接收模式滤波分频 对系统时钟分频已达到低功耗的频率	0x0	R/W



## 21.8 USART2,3 寄存器说明

### 21.8-1 状态寄存器(USARTx\_SR)

Address offset: 0x00

Reset value: 0x0000 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
r		CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE				
rc		rc	ro	rc	rc	rc	ro	ro	ro	ro	ro				

位	标记	功能描述	复位值	读写
31:10	Revered	Revered	--	--
9	CTS	<p>CTS: CTS 标志</p> <p>如果 CTSE 位置位, 当 nCTS 输入变化状态时, 该位被硬件置高。由软件将其清零。如果 USART_CR3 中的 CTSIE 为一, 产生中断</p> <p>0: nCTS 状态线上没有变化</p> <p>1: nCTS 状态线上发生变化</p>	0x0	rc
8	LBD	<p>LBD: LIN break 检测标志 (状态标志)</p> <p>0: 没有检测到 LIN break</p> <p>1: 检测到 LIN break</p> <p>注意: 若 LBDIE=1, 当 LBD 为 1 时要产生中断</p>	0x0	rc
7	TXE	<p>TXE:发送数据寄存器空</p> <p>当 TDR 寄存器中的数据被硬件转移到移位寄存器的时候, 该位被硬件置位。如果 USART_CR1 寄存器中的 TXEIE 为 1, 则产生中断。对 USART_DR 的写操作, 将该位清零。</p> <p>0: 数据还没有被转移到移位寄存器</p> <p>1: 数据已经被转移到移位寄存器</p> <p>注意: 单缓冲器传输中使用该位</p>	0x1	ro



位	标记	功能描述	复位值	读写
6	TC	TC: 发送完成  当包含有数据的一帧发送完成后，由硬件将该位置位。如果 USART_CR1 中的 TCIE 为 1，产生中断。由软件序列清除该位（先对 USART_SR 进行读操作，然后对 USART_DR 进行写操作）  0: 发送还未完成  1: 发送完成成	0x1	rc
5	RXNE	RXNE:读数据寄存器非空  当 RDR 移位寄存器中的数据被转移到 USART_DR 寄存器中，该位被硬件置位。如果 USART_CR1 寄存器中的 RXNEIE 为 1，中断产生。  对 USART_DR 的读操作可以将改位清零。  0: 数据没有收到  1: 收到数据，可以读出	0x0	rc
4	IDLE	IDLE:监测到 IDLE 总线 当检测到空闲总线时，该位被硬件置位。如果 USART_CR1 中的 IDLEIE 为 1，产生中断。由软件序列清除该位（先读 USART_SR，然后 USART_DR）  0: 没有检测到空闲总线  1: 检测到空闲总线  注意：IDLE 位不会再次被置高直到 RXNE 位被置起(即又检测到一次空闲总线)	0x0	r
3	ORE	ORE:过载错误  当 RXNE 还是 1 的时候，当前被接收在移位寄存器中的数据要往 RDR 寄存器中传送时，硬件将该位置位。如果 USART_CR1 中的 RXNEIE 为 1 的话，产生中断。由软件序列将其清零（先读 USART_SR，然后读 USART_CR）  0: 没有过载错误  1: 检测到过载错误  注意：该位被置位时，RDR 寄存器中的值不会丢失，但是移位寄存器中的数据会被覆盖。  如果 EIE 位被设置，在多缓冲器通信模式下，ORE 标志置位会产生中断的。	0x0	r
2	NE	NE: 噪声错误标志	0x0	r



位	标记	功能描述	复位值	读写
		<p>在接收到的帧检测到噪音时，由硬件对该位置位。由软件序列对其清零（先读 USART_SR，再读 USART_DR）</p> <p>0: 没有检测到噪音 1: 检测到噪音</p> <p>注意：该位不会产生中断，因为它和 RXNE 一起出现，后者自己会在 RXNE 标志置位时产生中断，如果 EIE 位被设置，并且工作在多缓冲区通信模式下</p>		
1	FE	<p>FE: 帧错误</p> <p>当检测到同步错位，过多的噪声或者检测到 break 符，该位被硬件置位。由软件序列将其清零（先读 USART_SR，再读 USART_DR）</p> <p>0: 没有检测到帧错误 1: 检测到帧错误或者 break 符</p> <p>注意：该位不会产生中断，因为它和 RXNE 一起出现，后者自己会在 RXNE 标志置位时产生中断。如果当前传输的数据既产生了帧错误，又产生了过载错误，还是会继续该数据的传输，并且只有 ORE 位会被置位。如果 EIE 位被置位，在多缓冲区通信模式下，随着 FE 标志被置位，中断产生。</p>	0x0	r
0	PE	<p>PE: 校验错误</p> <p>在接收模式下，如果出现校验错误，硬件对该位置位。由软件序列对其进行清零（依次读 USART_SR 和 USART_DR）。如果 USART_CR1 中的 PEIE 为 1，产生中断。</p> <p>0: 没有校验错误</p>	0x0	ro



### 21.8-2 数据寄存器(USARTx\_DR)

地址偏移: 0x04 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
8:0	DR	<p>DR[8:0]:数据值 包含了发送或接收的数据。由于它是由两个寄存器组成的，一个给发送用 (TDR)，一个给接收用 (RDR)，该寄存器兼具读和写的功能。 TDR 寄存器提供了内部总线和输出移位寄存器之间的并行接口（参见图 1）。RDR 寄存器提供了输入移位寄存器和内部总线之间的并行接口。 当使能校验位 (USART_CR1 中 PCE 位被置位) 进行发送时，写到 MSB 的值（根据数据的长度不同，MSB 是第 7 位或者第 8 位）会被后来的校验位取代。当使能校验位进行接收时，读到的 MSB 位是接收到的校验位。</p>	0x0	RW



### 21.8-3 波特比率寄存器(USARTx\_BRR)

地址偏移: 0x08 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
15:4	DIV_MANTISSA	DIV_Mantissa[11:0]:USARTDIV 的小数部分 这 12 位定义了 USART 分频器除法因子 (USARTDIV)的小数部分	0x0	RW
3:0	DIV_FRACTON	DIV_Fraction[3:0]:USARTDIV 的整数部分 这 4 位定义了 USART 分频器除法因子 (USARTDIV)的整数部分	0x0	RW

注意: 如果 TE 或 RE 被分别禁止, 波特计数器停止计数



#### 21.8-4 控制寄存器 1 (USARTx\_CR1)

地址偏移: 0x0C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
Revered															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	SYNSPI	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
-	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:14	Revered	Revered	--	--
14	SYNSPI	SPI 同步始能(用于 SPI slave Interface)  0: 正常操作。 1: 始能 SPI 同步模式。(仅用于同步传输)	0x0	rw
13	UE	UE: USART 使能  当该位被清零, USART 的分频器和输出在当前字节传输完成后停止工作以减少功耗。该位的置起和清零, 是由软件操作的。 0: USART 分频器和输出被禁止 1: USART 模块使能	0x0	rw
12	M	M: 字长  该位定义了数据字的长度, 由软件对其置位和清零操作 0: 一个起始位, 8 个数据位, n 个停止位 1: 一个起始位, 9 个数据位, 一个停止位  注意: 在数据传输过程中(发送或者接收时), 不能修改这个位	0x0	rw
11	WAKE	WAKE: 唤醒的方法  这位决定了把 USART 唤醒的方法, 由软件对该位置位或者清零。 0: 被空闲总线唤醒 1: 被地址标记唤醒	0x0	rw
10	PCE	PCE: 检验控制使能	0x0	rw



位	标记	功能描述	复位值	读写
		用该位来选择是否进行硬件校验控制（对于发送来说就是校验位的产生；对于接收来说就是校验位的检测）。当使能了该位，在发送数据的 MSB（如果 M=1，MSB 就是第 9 位；如果 M=0,MSB 就是第 8 位）插入校验位；对接收到的数据检查其校验位。软件对它置位或者清零。一旦该位被置位，当前字节传输完成后，校验控制才生效。 0: 校验控制被禁止 1: 校验控制被使能		
9	PS	PS: 校验选择 该位用来选择当校验控制使能后，是采用偶校验还是奇校验。软件对它置位或者清零。当前字节传输完成后，该选择生效 0: 偶校验 1: 奇校验	0x0	rw
8	PEIE	PEIE: PE 中断使能 软件对该位置位或者清零 0: 中断被禁止 1: 当 USART_SR 中的 PE 为 1 时，产生 USART 中断 register	0x0	rw
7	TXEIE	TXEIE: 发送缓冲区空中断使能 软件对该位置位或者清零 0: 中断被禁止 1: 当 USART_SR 中的 TXE 为 1 时，产生 USART 中断	0x1	rw
6	TCIE	TCIE: 发送完成中断使能 软件对该位置位或者清零 0: 中断被禁止 1: 当 USART_SR 中的 TC 为 1 时，产生 USART 中断	0x1	rw
5	RXNEIE	RXNEIE: 接收缓冲区非空中断使能 软件对该位置位或者清零 0: 中断被禁止 1: 当 USART_SR 中的 ORE 或者 RXNE 为 1 时，产生 USART 中断	0x0	rw
4	IDLEIE	IDLEIE: IDLE 中断使能 软件对该位置位或者清零 0: 中断被禁止 1: 当 USART_SR 中的 IDLE 为 1 时，产生 USART 中断	0x0	rw
3	TE	TE: 发送使能	0x0	rw



位	标记	功能描述	复位值	读写
		<p>该位使能发送器。软件对该位置位或者清零</p> <p>0: 发送被禁止</p> <p>1: 发送被使能</p> <p>注意:</p> <p>在数据传输过程中, 除了在智能卡模式下, 如果 TE 位上有个 0 脉冲 (即 “0” 之后来一个 “1”), 会在当前数据字传输完成后, 发送一个 “预备状态” (空闲总线) 当 TE 被设置后, 在真正发送开始之前, 有 一个比特时间的延迟。</p>		
2	RE	<p>RE: 接收使能</p> <p>软件对该位置位或者清零</p> <p>0: 接收被禁止</p> <p>1: 接收被使能, 开始搜寻 RX 引脚上的起始位。</p>	0x0	rw
1	RWU	<p>RWU: 接收唤醒</p> <p>该位用来决定是否把 USART 置于静默模式。软件对该位置位或者清零。当唤醒序列到来时, 硬件也会将其清零。</p> <p>0: 接收器处于正常工作模式</p> <p>1: 接收器处于静默模式</p> <p>注意:</p> <p>在把 USART 置于静默模式(设置 RWU 位)之前, USART 要已经先接收了一个数据字节。否则在静默模式下, 不能被空闲总线检测唤醒。当配置成地址标记检测唤醒(WAKE 位为 1), 在 RXNE 位被置位时, 不能用软件来修改 RWU 位。</p>	0x0	rW
0	SBK	<p>SBK: 发送断开帧</p> <p>使用该位来发送断开字符。软件可以对该位置位或者清零。应该由软件来置位它, 然后在断开帧的停止位时, 由硬件将该位复位。</p> <p>0: 没有发送断开字符</p> <p>1: 将要发送断开字符</p>	0x0	rw



### 21.8-5 控制寄存器 2(USART\_CR2)

地址偏移: 0x10 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LINEN	STOP	CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD				
r	RW	rw	rw	rw	rw	rw	r	rw	rw	r	r	w	w	w	r

位	标记	功能描述	复位值	读写
31:15	Revered	Revered	--	--
14	LINEN	<p>LINEN: LIN 模式使能 软件对该位置位或者清零。 0: LIN 模式被禁止 1: LIN 模式被使能 LIN 模式可以用 USART_CR1 寄存器中的 SBK 位发送 LIN 同步 breaks, 以及检测 LIN 同步 break</p>	0x0	rw
13:12	STOP	<p>STOP: 停止位 用来设置停止位的位数 00: 1 个停止位 01: 0.5 个停止位 10: 2 个停止位 11: 1.5 个停止位</p>	0x0	rw
11	CLKEN	<p>CLKEN: 时钟使能 该位用来使能 SCLK 引脚</p>	0x0	rw



位	标记	功能描述	复位值	读写
		0: SCLK 引脚被禁止 1: SCLK 引脚被使能		
10	CPOL	CPOL: 时钟极性  用户可以用该位来选择同步模式下 SCLK 引脚上时钟输出的极性。和 CPHA 位一起配合来产生用户希望的时钟/数据的采样关系 0: 总线空闲时 SCLK 引脚上保持低电平 1: 总线空闲时 SCLK 引脚上保持高电平	0x1	rw
9	CPHA	CPHA: 时钟相位  用户可以用该位来选择同步模式下 SCLK 引脚上时钟输出的相位。和 CPOL 位一起配合来产生用户希望的时钟/数据的采样关系(参见图 174 和 0) 0: 时钟第一个边沿进行数据捕获 1: 时钟第二个边沿进行数据捕获	0x1	rw
8	LBCL	LBCL: 最后一位时钟脉冲  使用该位来控制是否在同步模式下，在 SCLK 引脚上输出最后发送的那个数据字节(MSB)对应的时钟脉冲 0: 最后一位数据的时钟脉冲不从 SCLK 输出 1: 最后一位数据的时钟脉冲会从 SCLK 输出  注意：最后一个数据位就是第 8 或者第 9 个发送的位（根据 USART_CR1 寄存器中的 M 位所定义的 8 或者 9 位数据帧格式）.	0x0	rw
7	Reserved	Reserved	0x0	r
6	LBDIE	LBDIE: LIN break 检测中断使能  Break 中断掩码（使用 break 定界符来检测 break） 0: 中断被禁止 1: 只要 USART_SR 寄存器中的 LBD 为 1 就产生中断	0x0	rw



位	标记	功能描述	复位值	读写
5	LBDL	LBDL: LIN break 检测长度  该位用来选择是 11 位还是 10 位的 break 检测  0: 10 位的 break 检测  1: 11 位的 break 检测	0x0	rw
4	Reserved	Reserved	0x0	r
3:0	ADD	ADD[3:0]: 该 USART 节点的地址  该位域给出这个 USART 节点的地址  这是在多处理器通信下的静默模式中使用的，使用地址标记来唤醒某个 USART 设备	0x0	rw

Note : 在发送被使能后不能写这三个位 (CPOL、CPHA、LBCL)



### 21.8-6 控制寄存器 2(USART\_CR3)

地址偏移: 0x14 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:10	Revered	Revered	--	--
10	CTSIE	CTSIE: CTS 中断使能 0: 中断被禁止 1: 只要 USART_SR 寄存器中的 CTS 为 1 就产生中断	0x0	rw
9	CTSE	CTSE: CTS 使能 0: CTS 硬件流控制被禁止 1: CTS 模式使能, 只有 nCTS 输入信号有效 (拉成低电平) 时才能发送数据。如果在数据传输的过程中, nCTS 信号变成无效, 那么发完这个数据后, 传输就停止下来。如果当 nCTS 为无效的时候, 往数据寄存器里写了数据, 那么这个数据要等到 nCTS 有效的时候才会被发送出去。	0x0	rw
8	RTSE	RTSE: RTS 使能 0: RTS 硬件流控制被禁止 1: RTS 中断使能, 只有接收缓冲区内有空闲的空间时才请求下一个数据。当前数据发送完成后, 发送操作就需要暂停下来。如果可以接收数据了, 将 nRTS 输出置为有效 (拉至低电平)	0x0	rw
7	DMAT	DMAT: DMA 使能发送 由软件对该位清零或者置位	0x0	rw



位	标记	功能描述	复位值	读写
		1: 发送时的 DMA 模式使能 0: 发送时的 DMA 模式被禁止		
6	DMAR	DMAR: DMA 使能接收 由软件对该位清零或者置位 1: 接收时的 DMA 模式使能 0: 接收时的 DMA 模式被禁止	0x0	rw
5	SCEN	SCEN: 智能卡模式使能 该位用来使能智能卡模式 0: 智能卡模式使能 1: 智能卡模式被禁止	0x0	rw
4	NACK	NACK:智能卡 NACK 使能 0: 校验错误出现时, 不发送 NACK 1: 校验错误出现时, 发送 NACK 0: NACK transmission in case of parity error is disabled 1: NACK transmission during parity error is enabled This bit is not available for UART4 & UART5.	0x0	rw
3	HDSEL	HDSEL: 半双工选择 选择单线半双工模式 0: 不选择半双工模式 1: 选择半双工模式	0x0	rw
2	IRLP	IRLP: 红外低功耗 该位用来选择普通模式还是低功耗红外模式 0: 通常模式 1: 低功耗模式	0x0	rw
1	IREN	IREN: 红外模式使能 由软件对该位清零或者置位 0: 红外被禁止 1: 红外使能	0x0	rw
0	EIE	EIE: 错误中断使能 在多缓冲区通信模式下, 当有帧错误、过载或者噪声错误时(USART_SR 中 FFE=1, 或者 ORE=1, 或者 NE=1), 产生中断。	0x0	rw



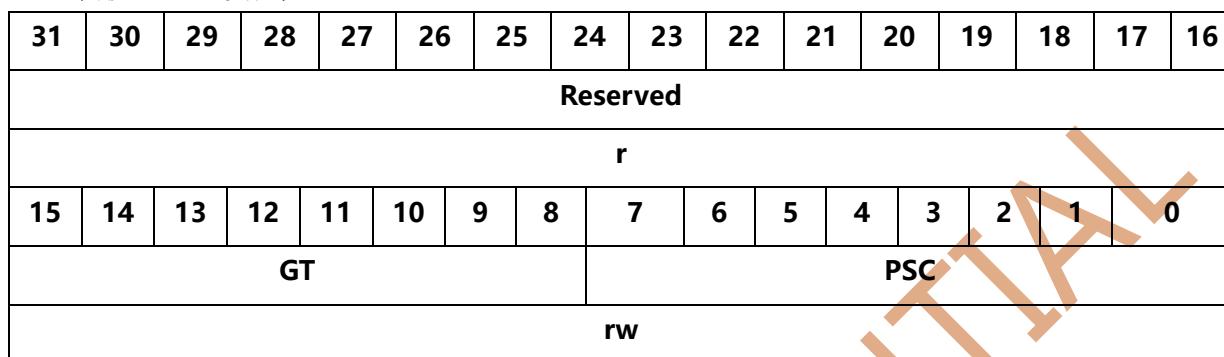
位	标记	功能描述	复位值	读写
		0: 中断被禁止 1: 只要 USART_CR3 中的 DMAR=1, 并且 USART_SR 中的 FE=1, 或者 ORE=1, 或者 NE=1, 产生中断		

ZBT CONFIDENTIAL



### 21.8-7 保护时间和预分频寄存器(USART\_GTPR)

地址偏移: 0x18 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:16	Revered	Revered	--	--
15:8	GT	<p>GT[7:0]: 保护时间值 该位域规定了以波特时钟为单位的保护时间的值。在智能卡模式下，需要这个功能。当保护时间过去后，发送完成标志才被置起。</p>	0x0	RW
7:0	PSC	<p>PSC[7:0]:预分频器值 - 在红外低功耗模式下： PSC[7:0] = 红外低功耗波特率 对系统时钟分频已到达低功耗的频率： 源时钟被寄存器中的值(仅有 8 位有效)分频 00000000: 保留 - 不要写入该值 00000001: 对源时钟 1 分频 00000010: 对源时钟 2 分频 ..... 00000010: divides the source clock by 2 - 在红外的通常模式下： PSC 只能设置为 0000001 - 在智能卡模式下： PSC[4:0]:预分频值 对系统时钟进行分频，给智能卡提供时钟。 寄存器中给出的值（5 个有效位）乘以 2 后，作为对源时钟的分频因子 00000: 保留 - 不要写入该值 00001: 对源时钟进行 2 分频</p>	0x0	RW



位	标记	功能描述	复位值	读写
		00010: 对源时钟进行 4 分频 00011: 对源时钟进行 6 分频 ..... 注意: 位[7:5]在智能卡模式下没有意义		

ZBT CONFIDENTIAL



## 【22】低功耗通用异步收发器(LPUART)

### 22.1 概述

本产品带有 1 个 LPUART 模块，支持半双工和全双工传输；支持 8BIT、9BIT 数据格式；支持 Mode 0/1/2/3 四种不同传输模式；LPUART 的波特率由 LPTIM 产生，也可以由内部自动波特率发生器产生；支持多机通讯模式；支持自动地址识别；支持给定地址和广播地址，支持低功耗模式。

LPUART 为支持低功耗应用，除了原本的 PCLK 时钟外，增加了一路 SCLK 时钟，并可以控制 LPUART 工作状态。LPUART 模块内部寄存器配置逻辑工作于 PCLK 时钟域，数据收发逻辑工作于 SCLK 时钟域。当系统进入低功耗模式且后且处在 LPUART 工作状态，关闭高频 PCLK 时钟，打开低频 SCLK 时钟，LPUART 仍旧可以进行正常的数据收发。关闭工作状态则停止波特率的生成。

SCLK 时钟来源可以选择：PCLK、外部低速时钟(LXT)，内部低速时钟(SIRC)。在 LPMODE=1 时，SCLK 时钟还支持 1/2/4/8/16/32/64/128 倍的预分频。

注意，在 LPMODE=0 时，LPUART 接收的是 LPTIM 时钟的 TOGGLE 输出信号而非 OVERFLOW 信号，因此必须使能 LPTIM 的 TOGGLE 输出。

### 22.2 结构框图

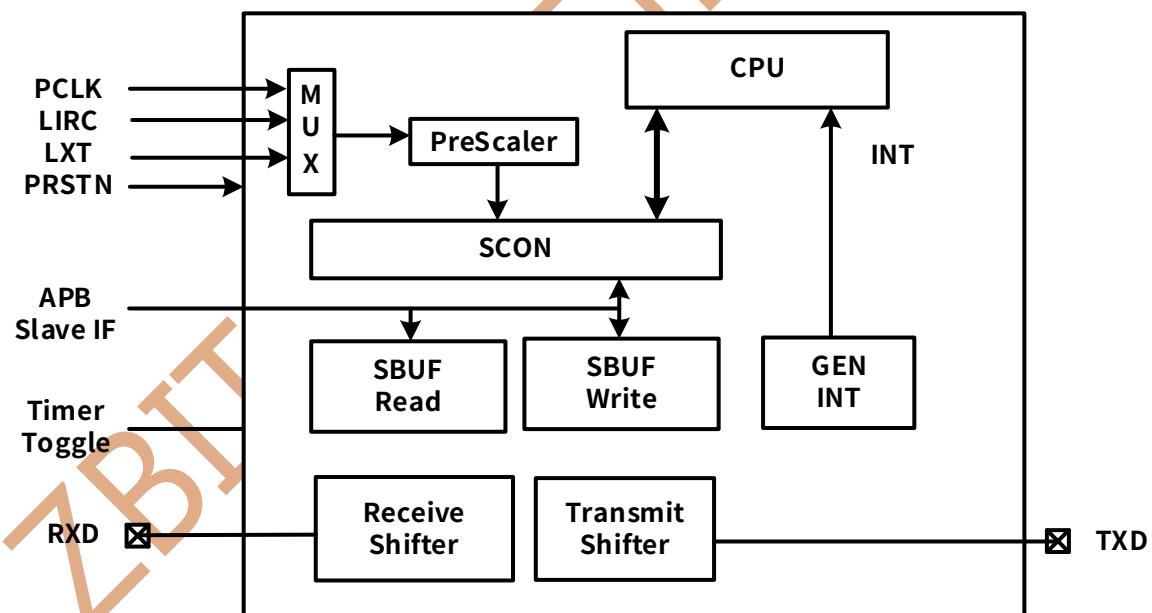


Figure 22-1 LPUART 结构框图



## 22.3 工作模式

与通用 UART(UART0/1)相比，LPUART 增加了一个 LPMODE 控制位。当该位置 “1” 时，只支持 Mode 1/3 工作模式，并且波特率生成方式也会发生改变。具体描述请参考以下章节。

### 22.3-1 Mode 0(同步模式，半双工)

当工作在 Mode 0 时，UART 工作在同步模式，其波特率为固定的 SCLK 时钟的 1/12。UART 接收数据由 RXD 输入、UART 发送数据有 RXD 输出，RXD 此时为输入输出端口。UART 同步移位时钟由 TXD 输出，TXD 此时为输出端口。注意，本模式只能作为主机发送同步移位时钟，不可以作为从机从外部接收该时钟。该模式下，传输的数据位宽只能是 8 位的，没有起始位和结束位。

将 LPUARAT\_SCON.SM0 和 LPUART\_SCON.SM1 清零，可进入 Mode 0 工作模式。

当 LPMODE=1 时，不支持 Mode 0 工作模式。

#### 22.3-1.1 发送数据

发送数据时，清除 LPUART\_SCON.REN 位，并将数据写入 LPUART\_SBUF 寄存器。此时，发送数据将从 RXD 输出(低位在先，高位在后)，同步移位时钟从 TXD 输出。

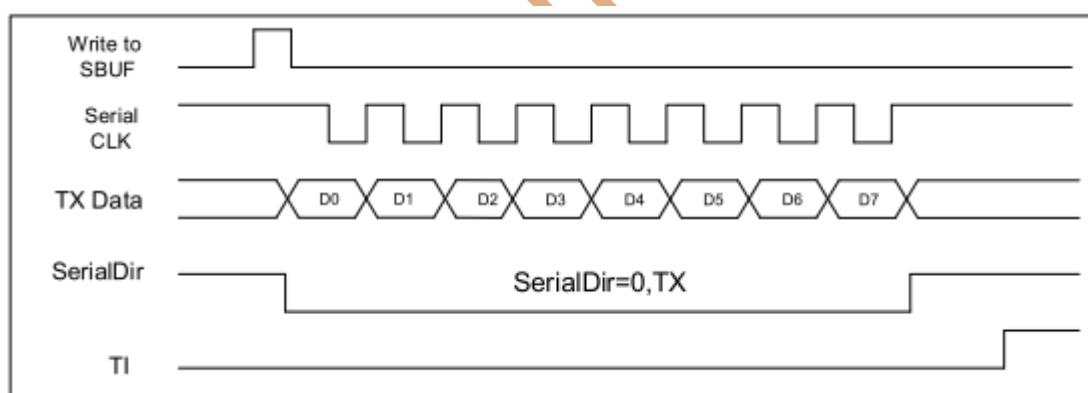


Figure 22-2 Mode 0 发送数据



### 22.3-1.2 接收数据

接收数据时，将 LPUART\_SCON.REN 位置 1，并将 LPUART\_INTSR.RI 位清零。当接收结束，数据可从 LPUART\_SBUF 寄存器读出。此时，接收数据从 RXD 输入(低位在先，高位在后)，同步移位时钟从 TXD 输出。

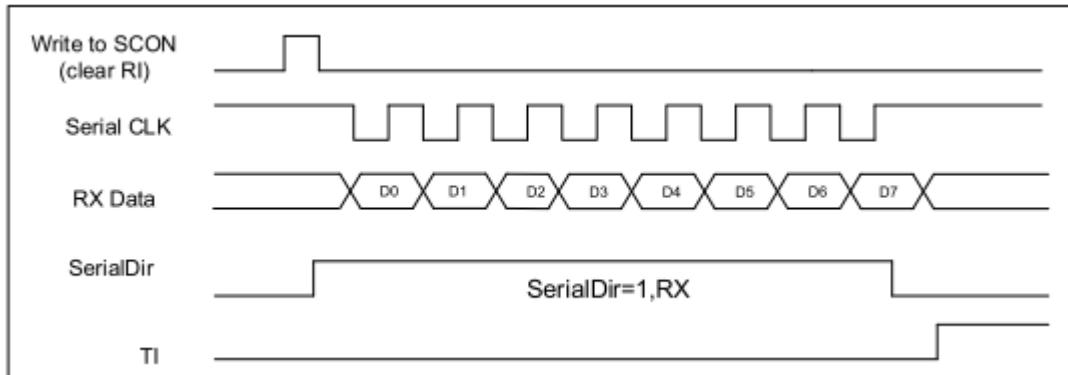


Figure 22-3 Mode 0 接收数据

### 22.3-2 Mode 1(异步模式，全双工)

当工作在 Mode 1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 10 位组成：起始位 “0” 开始，紧接着 8 位数据位(低位在先，高位在后)，最后是结束位 “1”。

将 LPUART\_SCON.SM0 清 0，LPUART\_SCON.SM1 置 1，可进入 Mode 1 工作模式。

该模式下，当 LPMODE=0 时，LPUART 的波特率可以选择由自动波特率发生器或者定时器 LPTIM 模块产生，并且是可编程的。

当 LPMODE=1 时，波特率计算方式发生改变，具体参考波特率编程章节。

### 22.3-2.1 发送数据

发送数据时，与 LPUART\_SCON.REN 的值无关，将所发送数据写入 LPUART\_SBUF 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

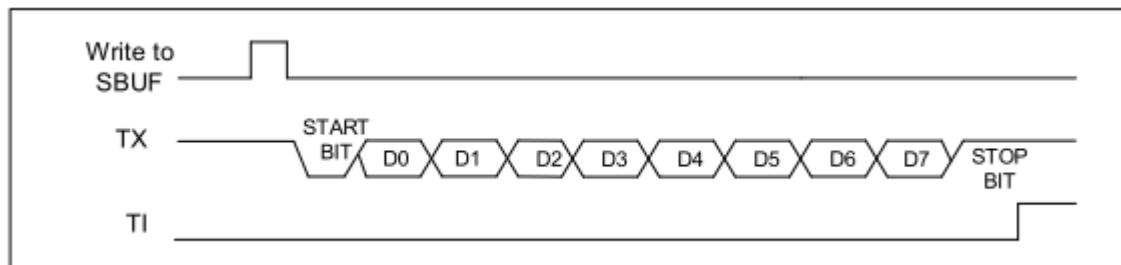


Figure 22-4 Mode 1 发送数据



### 22.3-2.2 接收数据

接收数据时，需将 LPUART\_SCON.REN 位置 1，并将 LPUART\_INTSR.RI 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 LPUART\_SBUF 寄存器读出。

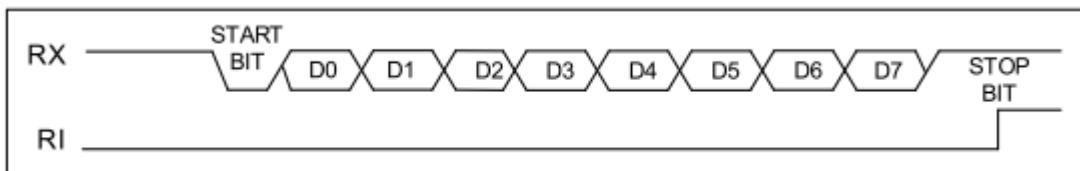


Figure 22-5 Mode 1 接收数据

### 22.3-3 Mode 2(异步模式，全双工)

当工作在 Mode 2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 11 位组成：起始位 “0” 开始，接着是 8 个数据位，1 个 TB8 位和结束位。额外的 TB8 位是用来在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位来使用。

将 LPUART\_SCON.SM0 置 1，LPUART\_SCON.SM1 清 0，可进入 Mode 2 工作模式。

该模式下，波特率可以独立产生，不需要外部 Timer 产生。

当 LPMODE=1 时，不支持 Mode 2 工作模式。

### 22.3-3.1 发送数据

发送数据时，与 LPUART\_SCON.REN 的值无关，并将所发送数据写入 LPUART\_SBUF 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

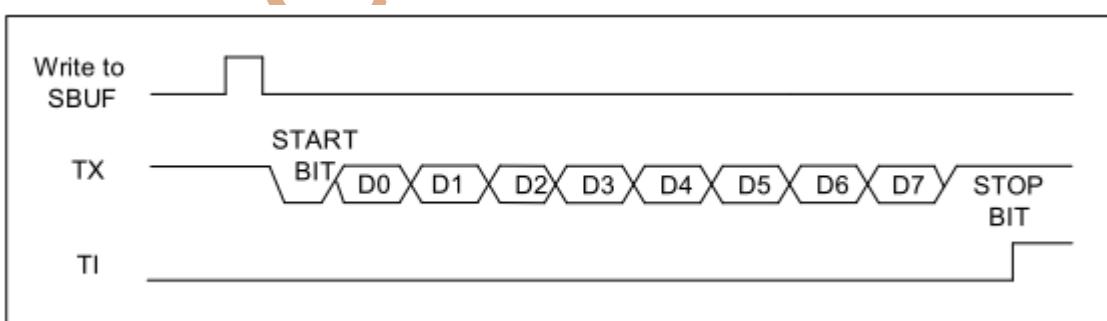


Figure 22-6 Mode 2 发送数据



### 22.3.3.2 接收数据

接收数据时，需将 LPUART\_SCON.REN 位置 1，并将 LPUART\_INTSR.RI 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 LPUART\_SBUF 寄存器读出。

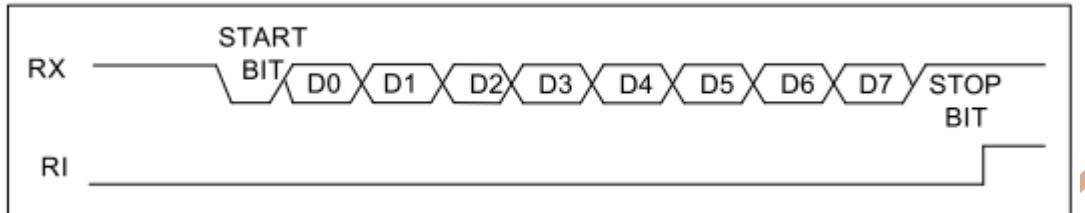


Figure 22-7 Mode 2 接收数据

### 22.3.4 Mode 3(异步模式，全双工)

Mode 3 的数据格式，传输时序以及操作方式都与 Mode 2 相同，唯一的区别是 Mode 3 的波特率由 LPTIM 产生或者内部自动波特率发生器产生，而不是像 Mode 2 由设备自己独立产生。Mode 3 的波特率是可编程的，波特率生成方式与 Mode 1 相同。

将 LPUART\_SCON.SM0 置 1，LPUART\_SCON.SM1 置 1，可进入 Mode 3 工作模式。

当 LPMODE=1 时，支持 Mode 3 工作模式。但是波特率计算方式发生改变，具体参考波特率编程章节。

## 22.4 波特率编程

### 22.4.1 Mode 0

LPMODE=0

当工作在 Mode 0 时，波特率被固定在 PCLK 的 1/12，不需要 LPTIM 的支持。

LPMODE=1

当 LPMODE=1 时，不支持该模式。



## 22.4-2 Mode 1/3

当工作在 Mode 1 或者 Mode 3 时，波特率可以由 LPTIM 的溢出时间决定。具体公式如下图所示：

$$\text{BaudRate} = ((\text{LPUART\_SCON.DBAUD} + 1) * \text{FSCLK}) / (32 * (216 - \text{LPTIM\_BGLOAD}[15:0]))$$

其中，LPUART\_SCON.DBAUD 表示双倍波特率，FSCLK 为 SCLK 时钟频率，LPTIM\_BGLOAD 为 LPTIM 的周期装载计数值。

注意，LPTIM 必须配置为 16 位自动重载模式，立即重载寄存器(LPTIM\_LOAD)和周期重载寄存器(LPTIM\_BGLOAD)要写入相同的初始值。

也可以使用自身波特率生成模式：

$$\text{BAUDRATE} = ((\text{LPUART\_SCON.DBAUD} + 1) * \text{FSCLK}) / (32 * (\text{LPUART_BAUDCR.BRG} + 1))$$

其中，UARTX\_SCON.DBAUD 表示双倍波特率，FSCLK 为 SCLK 时钟频率。

LPMODE=1

当 LPMODE 设为 “1” 时，波特计算公式与上述公式不同，简化为：

$$\text{BAUDRATE} = (\text{FSCLK}) / (4 * (\text{LPUART_SCON.PRSC}))$$

其中，FSCLK 为 SCLK 时钟频率，LPUART\_SCON.PRSC 为预分频系数。

## 22.4-3 Mode 2

LPMODE=0

当工作在 Mode 2 时，传输时钟只能选择 PCLK，波特率被固定在如下公式所得值：

$$\text{BAUDRATE} = ((\text{LPUART_SCON.DBAUD} + 1) * \text{FPCLK}) / 64$$

其中，LPUART\_SCON.DBAUD 表示双倍波特率，FPCLK 为 PCLK 时钟频率。

LPMODE=1

当 LPMODE=1 时，不支持该模式。



## 22.5 帧错误检测

Mode 1/2/3 具有帧错误检测功能，硬件会自动检测接收到的帧数据是否带有效的 STOP 位。如果没有收到有效 STOP 位，则 LPUART\_INTSR.FE 置 1。LPUART\_INTSR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有效 STOP 位，也不会把 LPUART\_INTSR.FE 标志清 0。

## 22.6 多机通讯

Mode 2/3 具有多机通讯功能，为此在其帧格式中增加了 1 位 TB8/RB8。将 LPUART\_SCON.SM2 置 “1”，可开启多机通讯位。当开启多机通讯位后，发送数据时，主机可以通过 LPUART\_SCON.TB8 来区分当前帧是地址帧(LPUART\_SCON.TB8=1)还是数据帧(LPUART\_SCON.TB8=0)。接收数据时，从机会忽略 RB8 位(第 9 位)为“0”的当前接收帧。当收到帧的 RB8 位(第 9 位)为“1”表明其是地址帧，从机会继续判断接收到的地址与其自身地址是否相等。如果匹配，则从机会对 LPUART\_SCON.RB8 置“1”，并对 LPUART\_INTSR.RI 置“1”，以表明该帧为地址帧并且地址已经匹配。从机软件看到 LPUART\_SCON.RB8=1 并且 LPUART\_INTSR.RI=1 后，先把 LPUART\_SCON.SM2 位清“0”，然后准备接受给它的数据帧。如果地址不等，表明主机并不是寻址该从机，从机硬件保持 LPUART\_SCON.RB8 和 LPUART\_INTSR.RI 为“0”，软件保持 LPUART\_SCON.SM2 位为“1”，从机继续处于地址监听状态。

## 22.7 自动地址识别

当开启多机通讯位后(LPUART\_SCON.SM2 置 “1”)，自动地址识别功能也将开启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收端会给出 LPUART\_INTSR.RI 接收标志。如果地址不匹配，则接收端不会给出任何接收标志。

如果有需要，也可以在 Mode 1 下开启多机通讯位，此时 TB8 位由 STOP 位代替。当从机接收到匹配的地址帧和有效的 STOP 位时，LPUART\_INTSR.RI 会被置“1”。为了支持自动地址识别，定义了广播地址和给定地址的概念。

## 22.8 给定地址

LPUART 设备的 LPUART\_SADDR 寄存器用来表示自己的设备给定地址，LPUART\_SADEN 寄存器是地址掩码，可以用来定义地址中的无关位。当 LPUART\_SADEN 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，LPUART\_SADEN 寄存器必须设为 0xFF。

$$\text{GIVENADDR} = \text{SADDR} \& \text{ SADEN}$$



## 22.9 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 0xHFF。

BoardCastAddr = SADDR | SADEN

给定地址和广播地址举例

假设某从机的 LPUART\_SADDR 和 LPUART\_SADEN 配置如下：

SADDR: 0b01101001

SADEN: 0b11111011

那么其给定地址和广播地址如下：

Given Address: 0b01101x01

BroadCast Address: 0b11111x11

可见，主机可以用四个地址寻址到本从机，分别是：

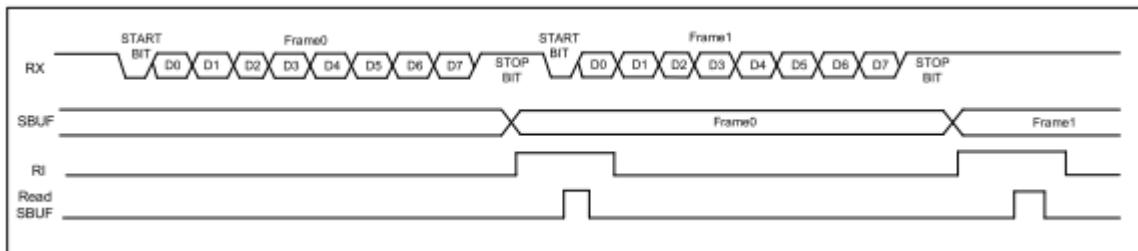
0b01101001 和 0b01101101(given address)

0b11111011 和 0b11111111(broadcast address)。

## 22.10 收发端缓存

### 22.10-1 接收缓存

LPUART 接收端有一个帧长度(8/9BITS)的接收缓存，也就是说当一帧数据接收完毕后，接收缓存中的数据会被一直保持，直到下一帧数据的 STOP 位接收完毕后，接收缓存才会更新为新一帧数据。



### 22.10-2 发送缓存

LPUART 发送端不支持发送缓存。如果在发送数据过程中，填写 LPUART\_SBUF 寄存器，将会屏蔽该写操作。软件应该避免这种操作。



## 22.11 寄存器列表

LPUART 基址: 0x4000 F400

地址偏移	名称	描述	复位值
0x00	LPUART_SBUF	数据寄存器	0x0000 0000
0x04	LPUART_SCON	控制寄存器	0x0000 E000
0x08	LPUART_SADDR	地址寄存器	0x0000 0000
0x0C	LPUART_SADEN	地址掩码寄存器	0x0000 0000
0x10	LPUART_INTSR	中断标志位寄存器	0x0000 0000
0x14	LPUART_INTCLR	中断标志位清除寄存器	0x0000 0000
0x18	LPUART_BAUDCR	波特率控制寄存器	0x0000 0000

Table 22-1 LPUART 寄存器列表

## 22.12 寄存器说明

### 22.12-1 LPUART 数据寄存器(LPUART\_SBUF)

地址偏移: 0x00 复位值: 0x0000 0000





## 22.12-2 LPUART 控制寄存器(LPUART\_SCON)

地址偏移: 0x04 复位值: 0x0000 E000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										保留			保留	DMATEN	DMAREN
保留										保留			R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PRSC[2:0]			SCLKSEL[1:0]		LPMODE	DBAUD	TEEN	SM0/SM1		SM2		REN	TB8	RB8	TIEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:17	-	保留	0x0	-
18	DMATEN	DMA TX enable 1 : Enable 0 : Disable	0	R/W
17	DMAREN	DMA RX enable 1 : Enable 0 : Disable	0	R/W
16	EN	Low-Power UART 工作使能 0: Low-Power UART 关闭, 不接收/发送数据 1: Low-Power UART 使能, 进行数据传输之前必须将该位置 1	0	R/W
15:13	PRSC[2:0]	传输时钟SCLK 预分频选择 000: DIV128; 001: DIV64; 010: DIV32; 011:16; 100:8; 101:4; 110: DIV2; 111: DIV1。 PRSC[2:0]只有当 LPMODE=1 时有效; 当 LPMODE=0 时, PRS[2:0]不会对SCLK 预分频。	0x7	R/W



位	标记	功能描述	复位值	读写																									
12:11	SCLKSEL[1:0]	传输时钟SCLK 选择 00 / 01: PCLK 10: LXT 11: SIRC	0x0	R/W																									
10	LPMODE	低功耗模式 0: 正常工作模式 1: 低功耗工作模式	0	R/W																									
9	DBAUD	双倍波特率 0: 单倍波特率 1: 双倍波特率	0	R/W																									
8	TEEN	发送缓存空中断使能 0: DISABLE 1: ENABLE	0	R/W																									
7:6	SM0:SM1	工作模式: 00: Mode 0; 01: Mode 1; 10: Mode 2; 11: Mode 3 <table border="1"><tr><th>SM0</th><th>SM1</th><th>MODE</th><th>描述</th><th>波特率</th></tr><tr><td>0</td><td>0</td><td>0</td><td>位移寄存器</td><td>PCLK/12</td></tr><tr><td>0</td><td>1</td><td>1</td><td>8 位串口传输</td><td>可变波特率</td></tr><tr><td>1</td><td>0</td><td>2</td><td>9 位串口传输</td><td>PCLK/32,PCLK/64</td></tr><tr><td>1</td><td>1</td><td>3</td><td>9 位串口传输</td><td>可变波特率</td></tr></table>	SM0	SM1	MODE	描述	波特率	0	0	0	位移寄存器	PCLK/12	0	1	1	8 位串口传输	可变波特率	1	0	2	9 位串口传输	PCLK/32,PCLK/64	1	1	3	9 位串口传输	可变波特率	0x0	R/W
SM0	SM1	MODE	描述	波特率																									
0	0	0	位移寄存器	PCLK/12																									
0	1	1	8 位串口传输	可变波特率																									
1	0	2	9 位串口传输	PCLK/32,PCLK/64																									
1	1	3	9 位串口传输	可变波特率																									
5	SM2	多主机通讯 : 0: DISABLE, 1: ENABLE SM2: 软件配置多机通讯以及自动地址匹配模式1: 启动多从机通讯以及地址自动匹配 0: 关闭多从机通讯以及地址自动匹配在模式 2 和模式 3 中: 如果 SM2=1, 并且 REN=1, 则接收机处于地址帧监测模式, 可以使用接收到的第 9 位 RB8 来进行地址筛选。 RB8=1 为地址帧, 通讯数据可以进入 SBUF, 置位 RI, 进入中断服务程序中进行地址比较; RB8=0 为数据帧, 接收机忽略这些数据帧并保持 RI=0 如果 SM2=0, 并且 REN=1, 则接收机不使用地址监测模式, 无论收到的 RB8 为 0 或 1, 都直接接收并且进入 SBUF, 置位 RI, RB8 在这种模式下为校验位。	0	R/W																									
4	REN	接收使能 Mode 0: 0: 发送, 1: 接	0	R/W																									



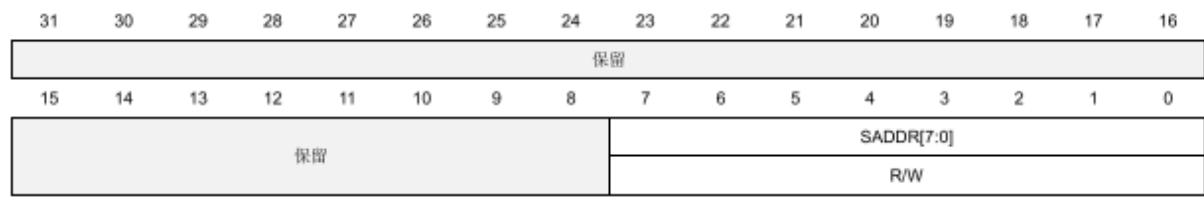
位	标记	功能描述	复位值	读写
		收其他: 0: 发送, 1: 接收 /发送		
3	TB8	发送TB8 位	0	R/W
2	RB8	接收RB8 位	0	R/W
1	TIEN	接收完成中断使能 0: DISABLE 1: ENABLE	0	R/W
0	RIEN	接收完成中断使能 0: DISABLE 1: ENABLE	0	R/W

ZBT CONFIDENTIAL



### 22.12-3 LPUART 地址寄存器(LPUART\_SADDR)

地址偏移: 0x08 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SADDR[7:0]	从机设备地址寄存器	0x0	R/W



#### 22.12-4 LPUART 地址掩码寄存器(LPUART\_SADEN)

地址偏移: 0x0C 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SADEN[7:0]	从机设备地址掩码寄存器	0x0	R/W

#### 22.12-5 LPUART 标志位寄存器(LPUART\_INTSR)

地址偏移: 0x10 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:3	-	保留	0x0	-
2	FE	接收帧错误标志位, 硬件置位, 软件清零 0: FE 中断无效 1: FE 中断有效	0	RO
1	TI	发送完成中断标志位, 硬件置位, 软件清零 0: TI 中断无效 1: TI 中断有效	0	RO
0	RI	接收完成中断标志位, 硬件置位, 软件清零 0: RI 中断无效 1: RI 中断有效	0	RO



## 22.12-6 LPUART 标志位清除寄存器(LPUART\_INTCLR)

地址偏移: 0x14 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:3	-	保留	0x0	-
2	FECLR	清除接收帧错误标志位; 写 1 清零, 写 0 无效	0x0	WO
1	TICLR	清除发送完成中断标志位; 写 1 清零, 写 0 无效	0x0	WO
0	RICLR	清除接收完成中断标志位; 写 1 清零, 写 0 无效	0x0	WO

## 22.12-7 LPUART 波特率控制寄存器(LPUART\_BAUDCR)

地址偏移: 0x18 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:17	-	保留	0x0	-
16	SELF_BRG	UART 波特率选择位: 0: UART 的波特率由 timer 产生 1: UART 的波特率由 (DBAUD+1)*FPCLK/(32*(BRG[15:0]+1))生成	0	R/W
15:0	BRG[15:0]	UART 自动波特率生成配置位: 波特率=(DBAUD+1)*FPCLK/(32*(BRG[15:0]+1))	0x0	R/W



## 【23】I2C 接口(I2C1/2-L032)

### 23.1 I2C 简介

I2C 是双线双向的串行总线，它为设备之间数据交换提供了一种简单高效的方法。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时请求控制总线时发生数据冲突。

I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。

I2C 总线使用连接设备的 "SCL" (串行时钟总线) 和 "SDA" (串行数据总线) 来传送信息。数据在主机与从机之间通过 SCL 时钟线控制在 SDA 数据线上实现一个字节一个字节的同步传输，每个字节为 8 位长度，一个 SCL 时钟脉冲传输一个数据位，数据由最高位 MSB 开始传输，每个传输字节后跟随一个应答位，每个位在 SCL 为高时采样；因此，SDA 线只有在 SCL 为低时才可以改变，在 SCL 为高时 SDA 保持稳定。当 SCL 为高时，SDA 线上的跳变视为命令中断(START 或 STOP)，I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器(I2C\_SR)能反映 I2C 总线控制器和 I2C 总线的状态。

### 23.2 I2C 主要特性

I2C 控制器支持以下特性：

- (1) 支持主机发送/接收，从机发送/接收四种工作模式
- (2) 支持标准(100Kbps)/快速(400Kbps)/高速(1Mbps)三种工作速率
- (3) 支持 7 位寻址功能
- (4) 支持噪声过滤功能
- (5) 支持广播地址
- (6) 支持中断状态查询功能
- (7) I2C1 支持 DMA 发送/接收

### 23.3 I2C 协议描述

通常标准 I2C 传输协议包含四个部分：

- (1) 起始信号或重复起始信号
- (2) 从机地址传输和 R/W 位传输
- (3) 数据传输
- (4) 停止信号

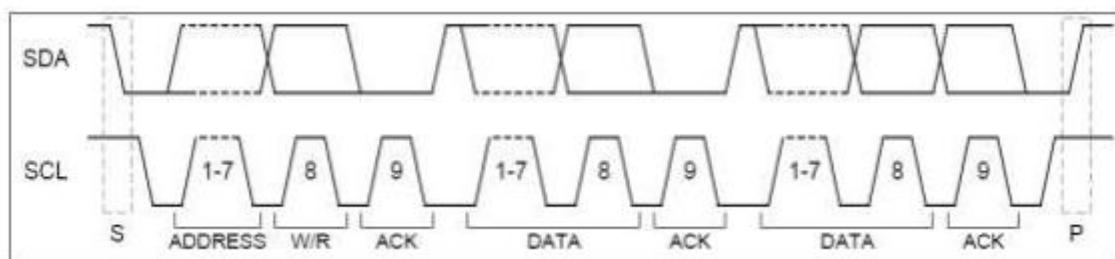


Figure 23-1 I2C 传输协议



### 23.3-1 I2C 总线上数据传输

主机发出从机接收 7 位地址(一个字节), 传输方向未改变。

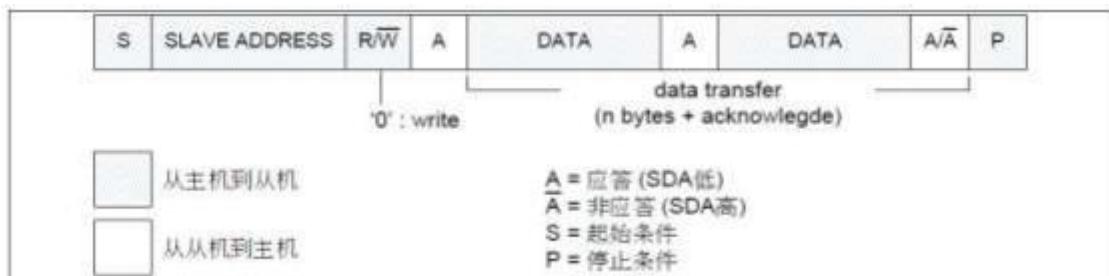


Figure 23-2 主机向从机传输数据

第一个字节后主机紧接着由从机读取数据(内容为从机地址), 传输方向改变。

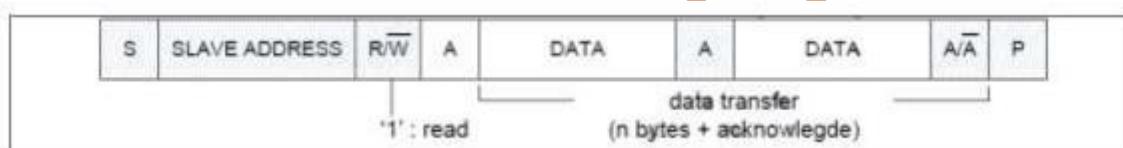


Figure 23-3 主机向从机传输数据

### 23.3-2 起始位或重复起始信号

当总线处于空闲状态下, 说明没有主机对总线发起传输请求(SCL 和 SDA 线同时为高), 主机可以通过发送一个 START 信号来发起传输请求。

起始信号: 通常表示为 S-bit, 当 SCL 线为高时, SDA 线上信号由高至低, 标示总线上产生起始信号, 新的传输开始。

重复起始信号(Sr): 即在两个 START 信号之间没有 STOP 信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信(例如: 从写入设备到从设备读出), 而不释放总线。

STOP 信号: 主机向总线发出停止信号结束数据传送。停止信号, 通常用 P-bit 表示, 当 SCL 线为高时, SDA 线上出现由低到高的信号, 被定义为停止信号。

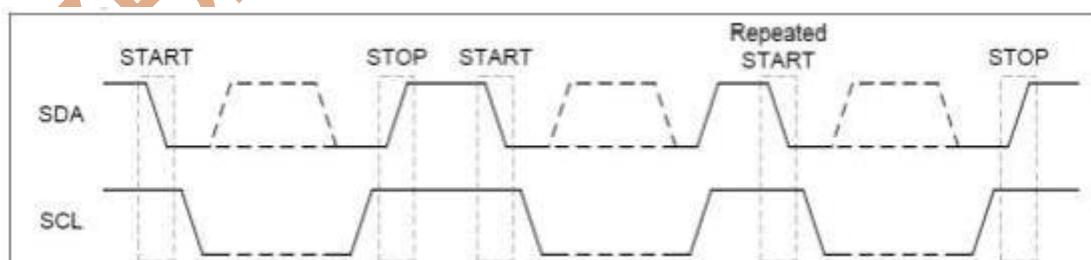


Figure 23-4 START 和 STOP 条件



### 23.3-3 从机地址传输

START 信号是从机地址时，主机立即传输数据的第一位。这是一个跟随有一个 RW 位的 7 位调用地址，RW 位控制从机的信号传输方向。系统中没有两个从机有相同的地址，只有被主机寻址的从机会通过在第 9 个 SCL 时钟周期将 SDA 置为低电平作为应答。

### 23.3-4 数据传输

当从机地址被成功识别，就可以根据 RW 所决定的方向，开始一字节一字节的数据传输，每个传输字节最后带一个第 9 时钟周期上的响应信号，如果从机上产生无响应信号(NACK)，主机可以产生停止信号来退出数据传输，或者产生重复起始信号开始新一轮的数据传输。

当主机作为接收器件时，发生无响应信号(NACK)，从机释放 SDA 线，使主机产生停止信号或重复起始信号。

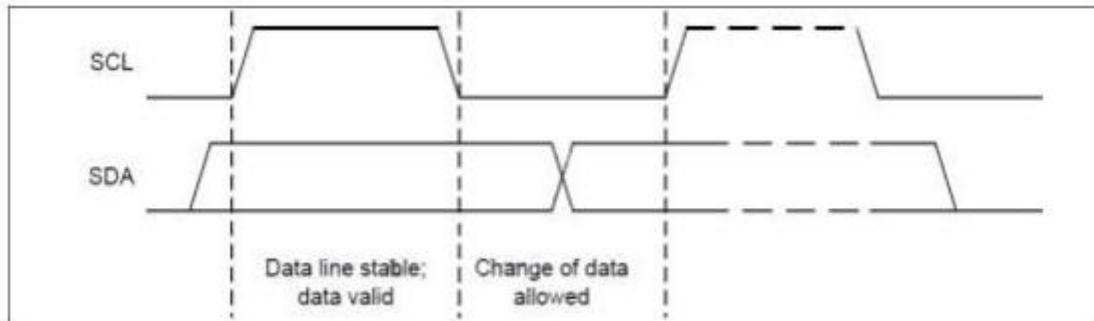


Figure 23-5 I2C 总线上位传输

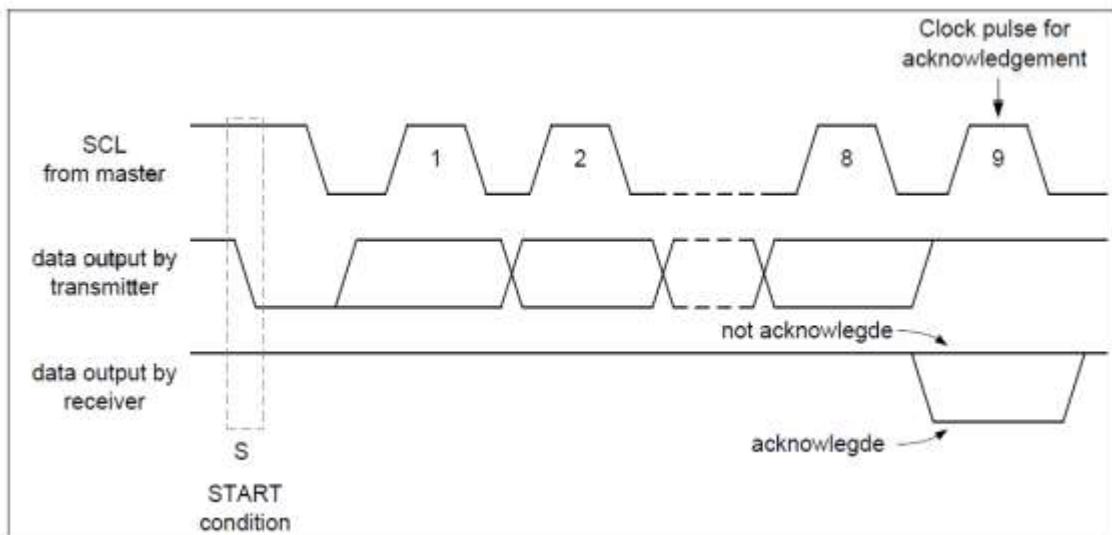


Figure 23-6 I2C 总线上应答信号



### 23.4 I2C 功能描述

I2C 总线使用双线在连接到总线“SCL”(串行时钟线)和“SDA”(串行数据线)的设备间传送信息。由于只有无方向端口，I2C 组件需要使用到引脚的漏端开路缓冲器。每个连接到总线的设备都能使用软件通过特定地址寻址。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时开始传输数据时发生数据冲突。滤波逻辑可以过滤数据总线上的毛刺来保护数据的完整性。

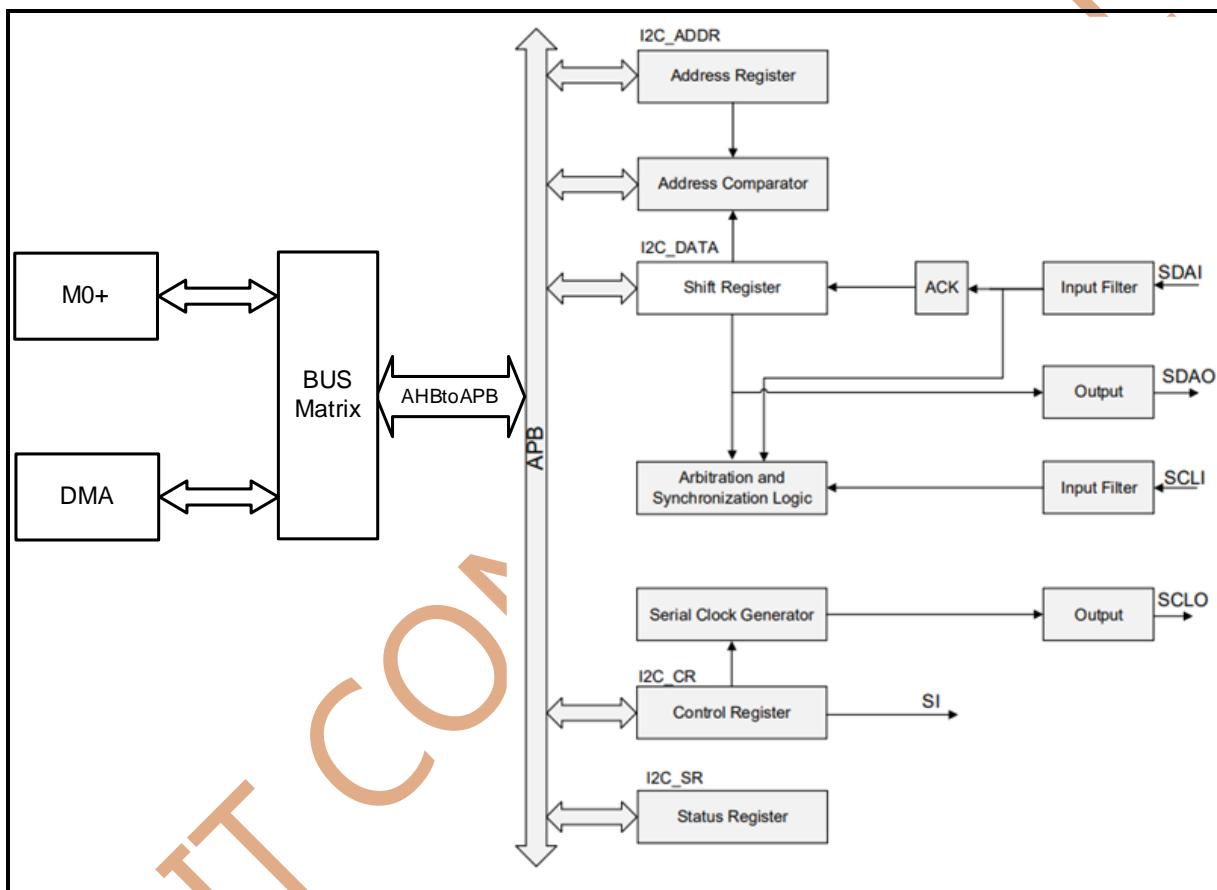


Figure 23-7 I2C 功能模块图



### 23.5 I2C 工作模式

I2C 模块可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100Kbits/s 而在高速模式下可达 400Kbits/s，在超高速模式下可达 1Mbits/s，并且可以在以下四种模式下工作：

1. 主机发送模式：当“SCL”输出串行时钟信号时“SDA”输出串行数据。
2. 主机接收模式：当“SCL”输出串行时钟信号时串行数据通过“SDA”接收。
3. 从机接收模式：串行数据和串行时钟分别通过“SDA”和“SCL”接收。
4. 从机发送模式：当串行时钟从“SCL”口输入时串行数据通过“SDA”口发送

#### 23.5-1 仲裁与同步逻辑

在主机发送模式中，仲裁逻辑检查每个发送的逻辑 1 是否真正出现在总线上。如果总线的另一个器件撤消了一个逻辑 1 并将 SDA 线拉低，仲裁丢失，I2C 模块立刻由主发送器变为从接收器。I2C 模块将继续输出时钟脉冲(在 SCL 上)，直至发送完当前的串行字节。

仲裁也可能在主接收模式中丢失。这种情况只在 I2C 模块正在向总线返回一个“非应答(逻辑 1)”时出现。当总线的另一个器件将信号拉低时仲裁丢失。由于它只在串行字节结束时出现，因此 I2C 模块不会再产生时钟脉冲。

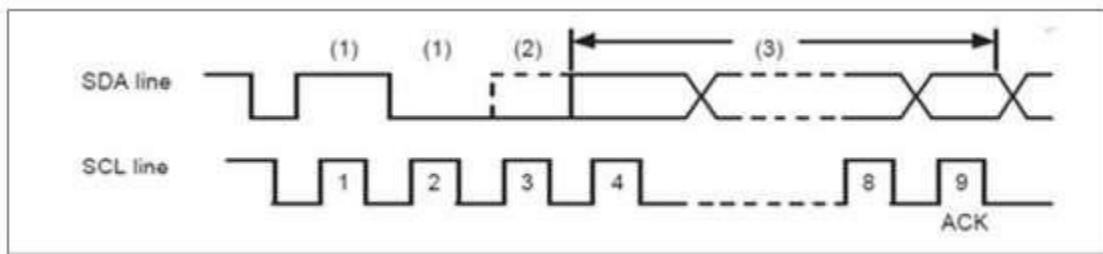


Figure 23-8 总线上的仲裁

1. 另一器件发送串行数据；
2. 另一器件通过拉低 SDA 先撤消了该 I2C 主机发送的一个逻辑 1(虚线)。仲裁丢失，I2C 进入 从接收模式；
3. 此时 I2C 处于从接收模式，但仍产生时钟脉冲，直至发送完当前字节。I2C 将不为下个字节 的传输产生时钟脉冲。一旦赢得仲裁，SDA 上的数据传输由新的主机来启动。

同步逻辑使得串行时钟发生器与另一个器件 SCL 线上的时钟脉冲同步。如果 2 个或更多主器件产生时钟脉冲，则高电平周期取决于产生最短高电平时间的器件；低电平周期取决于产生最长低电平 时间的器件。



### 23.5-2 串行时钟发生器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 PCLK 信号的频率关系为  $FSCL=FPCLK/8*(N+1)$

下面的表格表示 PCLK 为各种频率时，分频系数为 1-7 时，SCL 信号的频率值。

频率(KHz)	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

### 23.5-3 输入滤波器

输入信号与时钟信号(clk)同步，低于 3 个时钟周期的尖峰脉冲信号会被滤除。每个滤波器由 3 个触发器组成。第一个触发器用来直接锁存输入信号，并将数据载入由另外两个构成的移位寄存器中。

当第二和第三个触发器的状态是"11"或"00"时，内部的滤除信号会各自被置 1 或置 0。

### 23.5-4 地址比较器

I2C 比较器将自己的从机地址与接收到的 7 位从机地址做比较。它可使用"I2C\_ADDR"寄存器对自己的从机地址进行编程。并且它会根据"I2C\_ADDR"寄存器的"GC"位，与首次接收到的 8 位字节或与通用调用地址(0x00)相比较。如果任何一者相同，"I2C\_CR"寄存器的"SI"位会被置 1 并产生一个中断请求。

### 23.5-5 中断产生器

I2C 模块的所有四种模式都被使用时，则有 26 种可能的总线状态。当 I2C 进入 26 种状态的 25 种状态时，"I2C\_CR"寄存器的"SI"标志位会被硬件置 1。"SI"位唯一不会被置 1 的状态是 0xF8，这表明没有有效的相关状态信息。"SI"标志位必须通过软件清零。为了清除"SI"位，必须把 0 写入此位。若在"SI"里写 1 不会改变"SI"的值。为了确定中断的实际中断源，中断服务程序在清除"SI"标志位之前，会对 I2C 状态寄存器进行查询。



### 23.5-6 I2C 主机发送模式

必须将 ENS 置 “1” 来使能 I2C 模块。如果 AA 位复位，当另一个器件正变成总线主机时，I2C 模块将不会应答其自身的从机地址或通用调用地址。换句话说，如果 AA 位复位，I2C 接口就不能进入 从机模式。STA、STO 和 SI 必须复位。

此时，可通过置位 STA 位进入主发送模式。一旦总线空闲，I2C 逻辑会马上测试 I2C 总线并产生一个起始条件。当发送起始条件时，串行中断标志(SI)置位，状态寄存器(I2C\_SR)中的状态代码为 0x08。中断服务程序利用该状态代码进入相应 的状态服务程序，将从机地址和数据方向位(SLA+W)装入 I2C\_DATA。I2C\_CR 的 SI 位必须在串行传输继续之前复位。当发送完从机地址和方向位且 接收到一个应答位时，串行中断标志(SI)再次置位，I2C\_SR 中可能是一系列不同的状态代码。主 机模式下为 0x18、0x20 或 0x38，从机模式(AA=1)下为 0x68、0x78 或 0xB0。每个状态代码对应 的操作在下表中详细介绍。在发送完重复起始条件(状态 0x10)后，I2C 模块通过将 SLA+R 装入 I2C\_DATA 切换到主接收模 式。



状态代码	I2C 总线和硬件状态	应用软件响应					I2C 硬件执行的下一个动作	
		读/写	写 I2C CR					
			STA	STO	SI	AA		
08H	已发送起始条件	装入SLA+W	X	0	0	X	将发送SLA+W, 接收ACK	
10H	已发送重复起始条件	装入SLA+W	X	0	0	X	同上	
		装入SLA+R	X	0	0	X	将发送SLA+R, I2C 自动切换到主接收模式	
18H	已发送 SLA+W 已接收 ACK	装入数据字节	0	0	0	X	将发送数据字节, 将接收ACK	
		无 I2C DATA 动作	1	0	0	X	将发送重复起始条件	
		无 I2C DATA 动作	0	1	0	X	将发送停止条件, STO 标志位复位	
		无 I2C DATA 动作	1	1	0	X	将先发送停止条件, 随后发送起始条件, STO 标志位复位	
20H	已发送 SLA+W 已接收非ACK	装入数据字节	0	0	0	X	将发送数据字节, 将接收ACK	
		无 I2C DATA 动作	1	0	0	X	将发送重复起始条件	
		无 I2C DATA 动作	0	1	0	X	将发送停止条件, STO 标志位复位	
		无 I2C DATA 动作	1	1	0	X	将先发送停止条件, 随后发送起始条件, STO 标志位复位	
28H	已发送 I2C DATA 中的数据, 已接收ACK	装入数据字节	0	0	0	X	将发送数据字节, 将接收ACK	
		无 I2C DATA 动作	1	0	0	X	将发送重复起始条件	
		无 I2C DATA 动作	0	1	0	X	将发送停止条件, STO 标志位复位	
		无 I2C DATA 动作	1	1	0	X	将先发送停止条件, 随后发送起始条件, STO 标志位复位	
		装入数据字节	0	0	0	X	将发送数据字节, 将接收ACK	
		无 I2C DATA 动作	1	0	0	X	将发送重复起始条件	



30H	已发送 I2C_DATA 中 的数据	无 I2C_DATA 动作	0	1	0	X	将发送停止条件, STO 标志位复 位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件, 随后发送起 始 条件, STO 标志位复位
38H	在SLA+R/W 或写数据字 节时丢失 仲裁	无 I2C_DATA 动作	0	0	0	X	I2C 总线被释放, 进入不可寻址 从模 式
		无 I2C_DATA 动作	1	0	0	X	当I2C 总线空闲时发送起始条件

ZBT CONFIDENTIAL

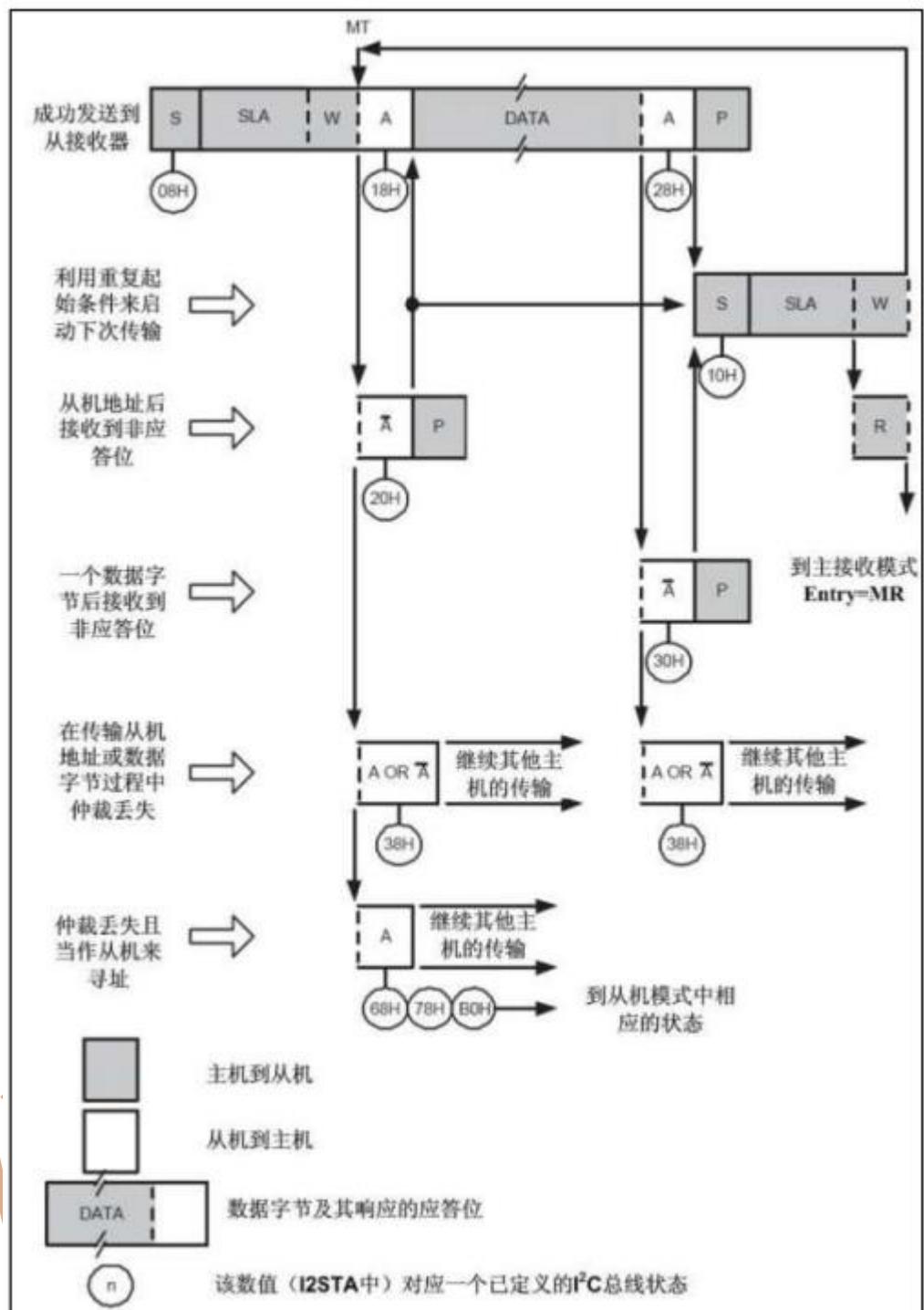


Figure 23-9 I<sup>2</sup>C 主机发送状态图



### 23.5.7 I2C 主机接收模式

在主机接收模式中，主机所接收的数据字节来自从发送器。按主机发送模式中的方法初始化传输。当发送完起始条件后，中断服务程序必须把 7 位从机地址和数据方向位(SLA+R)装入 I2C\_DATA。必须先清除 I2C\_CR 中的 SI 位，再继续执行串行传输。当发送完从机地址和数据方向位且接收到一个应答位时，串行中断标志 SI 再次置位，这时，I2C\_SR 中可能是一系列不同的状态代码。主机模式下为 0x40、0x48 或 0x38，从机模式(AA=1)下为 0x68、0x78 或 0xB0。每个状态代码对应的操作详见下表。在发送完重复起始条件(状态 0x10)后，I2C 模块通过将 SLA+W 装入 I2C\_DATA 切换到主发送模式。

状态代码	I2C 总线和硬件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写	写 I2C_CR				
			STA	STO	SI	AA	
08H	已发送起始条件	装入SLA+R	X	0	0	X	将发送SLA+R，接收ACK
10H	已发送重复起始条件	装入SLA+R	X	0	0	X	同上
		装入SLA+W	X	0	0	X	将发送SLA+W，I2C 自动切换到主发送模式
38H	在非ACK中丢失仲裁	无 I2C_DATA 动作	0	0	0	X	I2C 总线将被释放；进入从模式
		无 I2C_DATA 动作	1	0	0	X	当总线空闲时发起起始条件
40H	已发送 SLA+R 已接收 ACK	无 I2C_DATA 动作	0	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	0	0	0	1	将接收数据字节，将返回ACK
48H	已发送 SLA+R 已接收非ACK	无 I2C_DATA 动作	1	0	0	X	将发送重复起始条件
		无 I2C_DATA 动作	0	1	0	X	将发送停止条件，STO 标志位复位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件，STO 标志位复位
	已接收数据	读取数据字节	0	0	0	0	将接收数据字节，将返回非 ACK



50H	字节, ACK 已 返 回	读取数据字 节	0	0	0	1	将接收数据字节, 将返回ACK
58H	已接收数据 字节, 非 ACK 已返回	读取数据字 节	1	0	0	X	将发送重复起始条件
		读取数据字 节	0	1	0	X	将发送停止条件, STO 标志 位复 位

ZBT CONFIDENTIAL

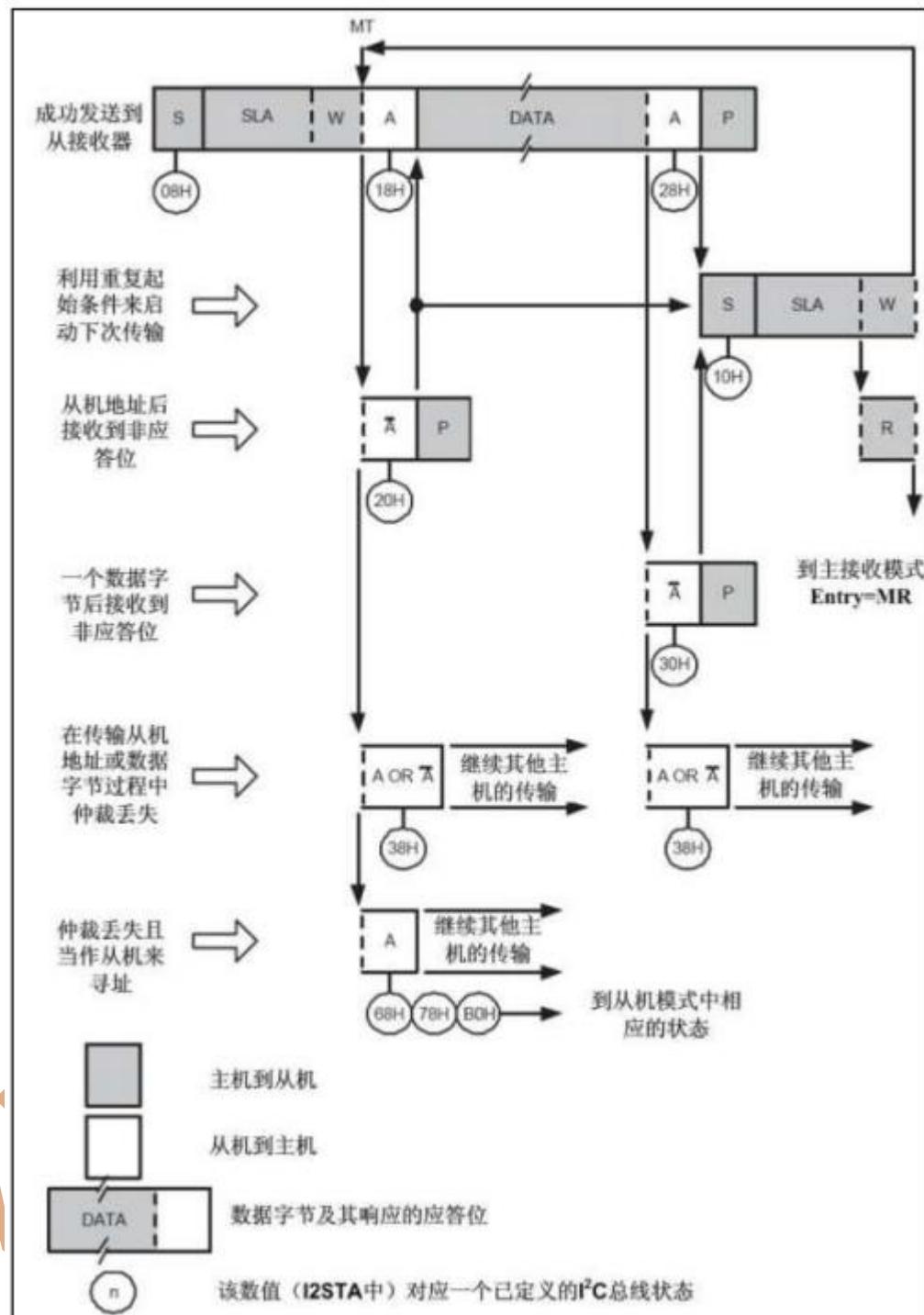


Figure 23-10 I<sup>2</sup>C 主机接收状态图



### 23.5-8 I2C 从机接收模式

在从机接收模式中，从机接收的数据字节来自主发送器。高 7 位是主机寻址时 I2C 模块响应的地址。如果 LSB(GC)被置位，I2C 模块将响应通用调用地址(0x00)；否则忽略通用调用地址。

I2C 总线速率的设置不影响从机模式中的 I2C 模块。必须置位 ENS 来使能 I2C 模块。AA 位必须置位以使能 I2C 模块来应答其自身从机地址或通用调用地址。STA, STO 和 SI 必须复位。

当 I2C\_ADDR 和 I2C\_CR 完成初始化后，I2C 模块一直等待，直至被从机地址寻址，之后是数据方向位寻址，为了工作在从接收模式中，数据方向位必须为“0”(W)。接收完其自身的从机地址和 W 位后，串行中断标志(SI)置位，可从 I2C\_SR 中读出一个有效的状态代码。该状态代码用作状态服务程序的向量。每个状态代码的对应操作见下表。如果 I2C 模块在主机模式中仲裁丢失，也可进入从接收模式(请参考状态 0x68 和 0x78 的描述)。

如果 AA 位在传输过程中复位，则在接收完下一个数据字节后 I2C 模块将向 SDA 返回一个非应答(逻辑 1)。当 AA 复位时，I2C 模块不响应其自身的从机地址或通用调用地址。但是，I2C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可临时将 I2C 模块从 I2C 总线上分离出来。

状态代码	I2C 总线和硬件状态	应用软件响应					I2C 硬件执行的下一个动作
		写 I2C_CR					
读/写	I2C_DATA	STA	STO	SI	AA		
60H	已接收自身的 SLA+W；已接收 ACK	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK
68H	主控时在 SLA+R/W 丢失仲裁；已接收自身的 SLA+W；已返回 ACK；	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK
70H	已接收通用调用地址(0x00)；已返回ACK；	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK



状态代码	I2C 总线和硬件状态	应用软件响应					I2C 硬件执行的下一个动作	
		读/写	写I2C_CR					
			STA	STO	SI	AA		
78H	主控时在 SLA+R/W 中丢失仲裁；已接收通用调用地址；已返回ACK；	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非ACK	
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK	
80H	前一次寻址使用自身从地址；已接收数据字节；已返回ACK；	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非ACK	
		无 I2C_DATA 动作	X	1	0	1	将接收数据字节，将返回ACK	
88H	前一次寻址使用自身从地址；已接收数据字节；已返回非ACK；	读取数据字节	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		读取数据字节	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		读取数据字节	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		读取数据字节	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；	
90H	前一次寻址使用通用调用地址；已接收数据；已返回ACK；	读取数据字节	1	0	0	X	将接收数据字节，将返回非ACK	
		读取数据字节	0	1	0	X	将接收数据字节，将返回ACK	
98H	前一次寻址使用通用调用地址；已接收数据；已返回非ACK；	读取数据字节	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		读取数据字节	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		读取数据字节	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线	



状态代码	I2C 总线和硬件状态	应用软件响应					I2C 硬件执行的下一个动作	
		读/写	写I2C_CR					
			STA	STO	SI	AA		
A 0 H	当使用从接收 / 从发送模式中静态寻址时，接收到停止条件或重复起始条件	读/写 I2C_DATA					空闲后发送起始条件；	
		读取数据字节	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；	
A 0 H	当使用从接收 / 从发送模式中静态寻址时，接收到停止条件或重复起始条件	无 I2C_DATA 动作	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		无 I2C_DATA 动作	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；	
		无 I2C_DATA 动作	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；	
		无 I2C_DATA 动作	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；	

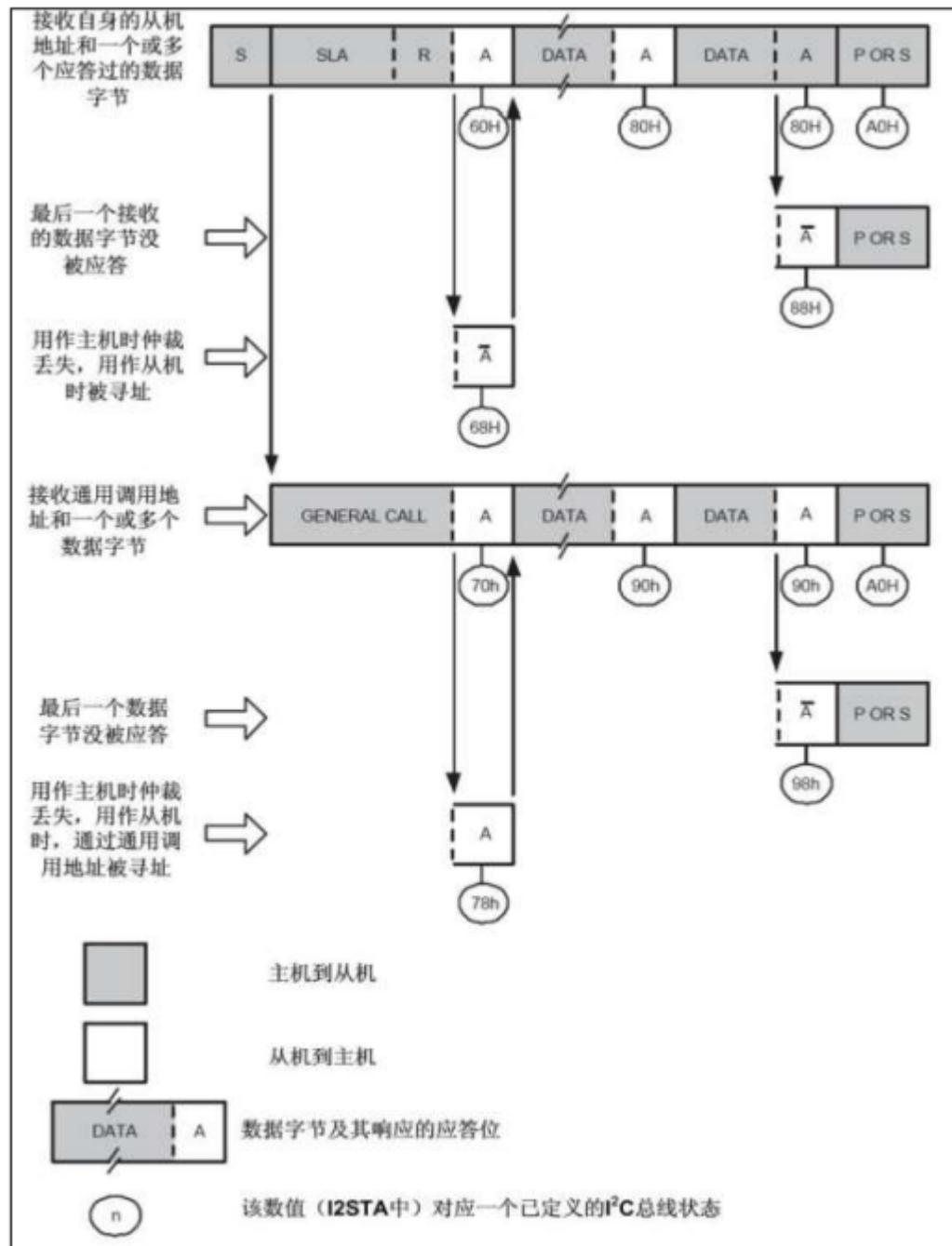


Figure 23-11 I<sup>2</sup>C 从机接收状态图



### 23.5.9 I2C 从机发送模式

在从机发送模式中，向主接收器发送数据字节。数据传输按照从机接收模式中的情况初始化。当初始化 I2C\_ADDR 和 I2C\_CR 后，I2C 模块一直等待，直至被自身的从机地址寻址，之后是数据方向位，该数据方向位必须为“1”(R)，以便 I2C 模块工作在从机发送模式下。接收完其自身的从机地址和 R 位后，串行中断标志(SI)置位，并且可从 I2C\_SR 中读取一个有效的状态代码。该状态代码用作状态服务程序的向量，每个状态代码的对应操作见下表所示。如果 I2C 模块在主机模式下时仲裁丢失，则可进入从机发送模式(见状态 0xB0)。

如果 AA 位在传输过程中复位，则 I2C 模块将发送最后一个字节并进入状态 0xC0 或 0xC8。I2C 模块切换到非寻址的从机模式，如果继续传输，它将忽略主接收器。因此主接收器接收所有 1 作为串行数据。当 AA 复位时，I2C 模块不响应其自身的从机地址或通用调用地址。但是，I2C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可用来暂时将 I2C 模块从 I2C 总线上分离出来。

状态代码	I2C 总线和硬件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写 I2C_DATA	写 I2C_CR				
			STA	STO	SI	AA	
A8H	已接收自身的 SLA+R；已返回 ACK	装入数据字节	X	0	0	0	将发送最后一个数据字节；将接收ACK；
		装入数据字节	X	0	0	1	将发送一个数据字节；将接收ACK；
B0H	当主控时在 SLA+R/W 中丢失仲裁；已接收自身 SLA+R；已返回 ACK；	装入数据字节	X	0	0	0	将发送最后一个数据字节；将接收ACK；
		装入数据字节	X	0	0	1	将发送一个数据字节；将接收ACK；
B8H	已发送数据；已接收 ACK；	装入数据字节	X	0	0	0	将发送最后一个数据字节；将接收ACK；
		装入数据字节	X	0	0	1	将发送一个数据字节；将接收ACK；
		无 I2C_DATA 动作	0	0	0	0	切换到不可寻址从机模式；不识别自身从地址或通用地址；



状态代码	I2C 总线和硬件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写 I2C_DATA	写 I2C_CR				
			STA	STO	SI	AA	
C0H	已发送数据字节；已接收非 ACK；	无 I2C_DATA 动作	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；
		无 I2C_DATA 动作	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；
		无 I2C_DATA 动作	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；
C8H	装入的数据字节已被发送；已接收 ACK；	无 I2C_DATA 动作	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；
		无 I2C_DATA 动作	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；
		无 I2C_DATA 动作	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；
		无 I2C_DATA 动作	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件；

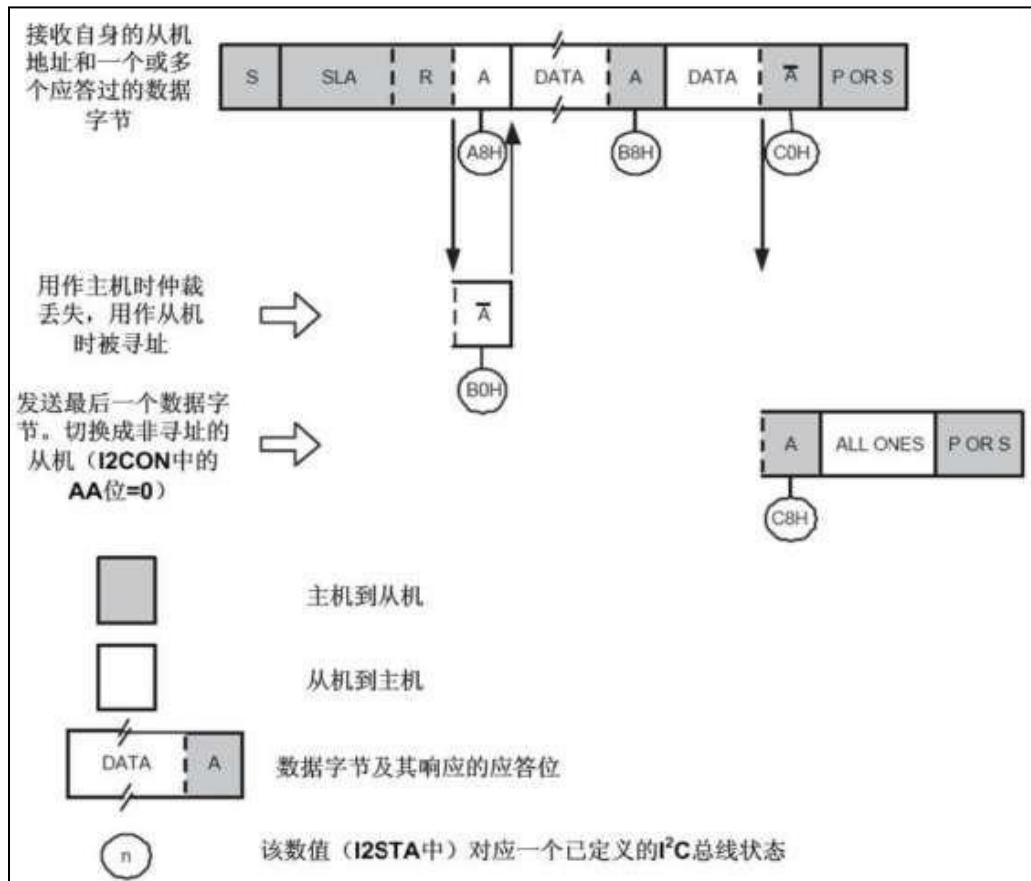


Figure 23-12 I<sup>2</sup>C 从机发送状态图



### 23.5-10 I2C 其他杂项状态

#### I2C\_SR = 0xF8

这个状态码表示没有任何可用的相关信息，因为串行中断标志 SI 还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

#### I2C\_SR = 0x00

该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时 SI 置位。要从总线错误中恢复，STO 标志必须置位，SI 必须被清除。这使得 I2C 模块进入“非寻址”的从机模式(已定义的状态)并清除 STO 标志(I2C\_CR 中的其它位不受影响)。SDA 和 SCL 线被释放(不发送停止条件)。

状态代码	I2C 总线和硬件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写	写 I2C_CR				
			STA	STO	SI	AA	
F8H	无可用的相关状态信息；SI=0；	无 I2C_DATA 动作	无 I2C_DATA 动作			等待或执行当前传输	
00H	由于非法的起始或停止条件的出现，在主机或被选中的从机将出现总线错误；当外部干扰使 I2C 进入未定义的状态时也会出 0x00 状态	无 I2C_DATA 动作	0	1	0	X	只有在主机或被寻址的从机模式中，内部硬件受影响。一般情况下，总线被释放，I2C 模块切换到非寻址的从机模式。STO 复位。

## 23.6 I2C 操作模式

### 23.6-1 初始化程序

将 I2C 接口初始化用作从机和/或主机的例子。

1. 将自身的从机地址装入 I2C\_ADDR，使能通用调用识别(如果需要的话)；
2. 使能 I2C 中断；
3. 向寄存器 I2C\_CR 写入 0x44 来置位 ENS 和 AA 位，并使能从机功能。对于主机功能，可向寄存器 I2C\_CR 写入 0x40。



### 23.6-2 端口配置程序

I2C 接口信号 SCL, SDA 映射到芯片引脚 PB4, PB5 的例子。

1. 配置 PB4, PB5 为开漏输出模式: P2OD[6], P2OD[5]配置为 0x1
2. 配置 PB4, PB5 的功能配置寄存器: PBAFR4, PBAFR5 配置为 0x4
3. 配置 PB4, PB5 的上拉使能配置寄存器: PBPUPD4, PBPUPD5 配置为 0x1

### 23.6-3 启动主机发送功能

#### 23.6-3.1 正常发送

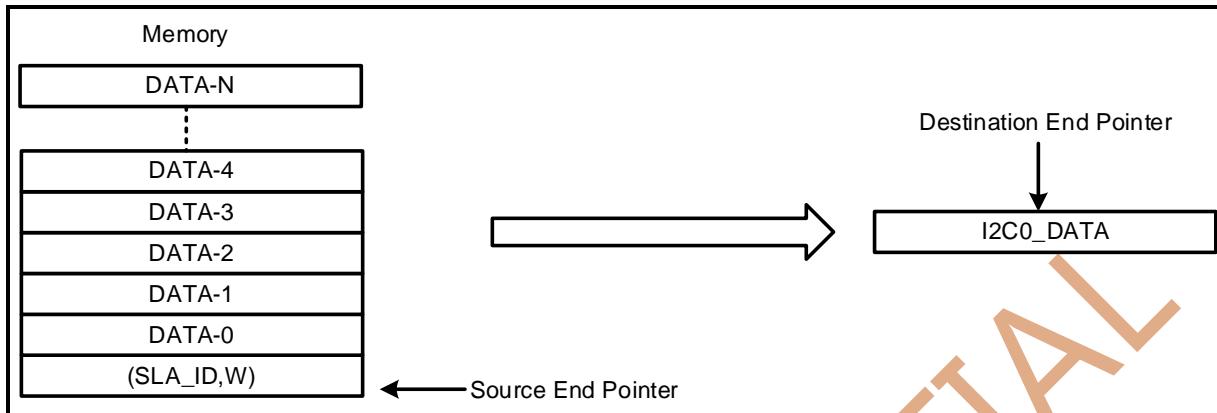
通过建立缓冲区、指针和数据计数然后发启起始条件便可执行主发送操作。

1. 初始化主机数据计数器;
2. 建立数据将被发送到的从机地址, 并且添加写位;
3. 向 I2C\_CR 写入 0x20 来置位 STA 位;
4. 在主发送缓冲区内建立要发送的数据;
5. 初始化主机数据计数器来匹配正在发送的信息长度;
6. 退出

#### 23.6-3.2 使用 DMA 发送

通过建立 DMA 配置(来源端指针, 目标端指针, 控制数据配置)后发启起始条件便可执行主发送操.

1. 初始化主机数据计数器;
2. 建立数据将被发送到的从机地址, 并且添加写位在来源端指针的储存器(Source End Pointer);
3. 建立要发送的 N 笔数据在来源端指针的储存器(Source End Pointer+1, Source End Pointer+2...N)
4. 设置 I2CX\*\_TIMDMA.DMATEN=1;
5. 向 I2C\_CR 写入 0x20 来置位 STA 位;
6. 等待 DMA 通道(11)完成 flag 为 1 或发生 DMA 中断时检查通道(11)完成 flag 位是否为 1.



**Figure 23-13 I2C1 Master DMA TX**

Note: 如为从机发送, 就不需 SLA\_ID 数据.



### 23.6-4 启动主机接收功能

#### 23.6-4.1 正常接收

通过建立缓冲区、指针和数据计数然后发启起始条件便可执行主接收操作。

1. 初始化主机数据计数器；
2. 建立数据将被发送到的从机地址，并且添加读位；
3. 向 I2C\_CR 写入 0x20 来置位 STA 位；
4. 在主接收缓冲区内建立要发送的数据；
5. 初始化主机数据计数器来匹配正在发送的信息长度；
6. 退出。

ZBT CONFIDENTIAL



### 23.6-4.2 使用 DMA 接收

通过建立 DMA 配置(来源端指针, 目标端指针, 控制数据配置)后发启起始条件便可执行主接收操作.

1. 初始化主机数据计数器;
2. 建立数据将被发送到的从机地址, 并且添加读位;
3. 设置 I2CX\*\_TIMDMA.DMAREN=1;
4. 向 I2C\_CR 写入 0x20 来置位 STA 位;
5. 等待 DMA 通道(10)完成 flag 为 1 或发生 DMA 中断时检查通道(10)完成 flag 位是否为 1.
6. 如 DMA\_CHTCIF CHTCIF[10] =1 , 目标端指针储存器内的值即为接收的数据

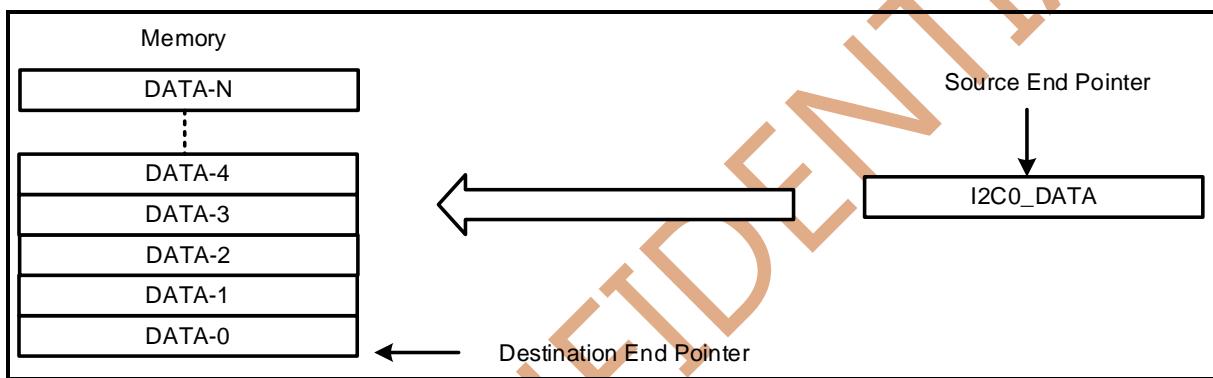


Figure 23-14 I2C0 Master DMA RX

### 23.6-5 I2C 中断程序

确定 I2C 的状态和处理该状态的状态程序。

1. 从 I2C\_SR 中读出 I2C 的状态;
2. 使用状态值跳转到 26 个可能状态程序中的一个。

### 23.6-6 无指定模式状态

1. 状态: 0x00 总线错误。进入非寻址的从机模式并释放总线。
  - a) 向 I2C\_CR 写入 0x14 来置位 STO 和 AA 位;
  - b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
  - c) 退出。
2. 主机状态  
状态 08 和 10 适用于主发送模式和主接收模式。R/W 位决定了下一个状态是在主发送模式中还是在 主接收模式中。
3. 状态: 0x08



已发送起始条件。即将发送从机地址+R/W 位和接收 ACK 位。

- a) 向 I2C\_DATA 写入从机地址和 R/W 位;
- b) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 建立主发送模式数据缓冲区;
- e) 建立主接收模式数据缓冲区;
- f) 初始化主机数据计数器;
- g) 退出。

4. 状态: 0x10

已发送重复起始条件。即将发送从机地址+R/W 位和接收 ACK 位。

- a) 向 I2C\_DATA 写入从机地址和 R/W 位;
- b) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 建立主发送模式数据缓冲区;
- e) 建立主接收模式数据缓冲区;
- f) 初始化主机数据计数器;
- g) 退出。



### 23.6.7 主发送状态

#### 1. 状态: 0x18

之前状态为 8 或 10 表示已发送从机地址和写操作位，并接收了应答。即将发送第一个数据字节和接收 ACK 位。

- a) 将主发送缓冲区的第一个数据字节装入 I2C\_DATA;
- b) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 主发送缓冲区指针加 1;
- e) 退出。

#### 2. 状态: 0x20 已发送从机地址和写操作位并接收了非应答。即将发送停止条件。

- a) 向 I2C\_CR 写入 0x14 来置位 STO 和 AA 位;
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- c) 退出。

#### 3. 状态: 0x28

已发送数据并接收了 ACK。如果发送的数据是最后一个数据字节则发送一个停止条件，否则发送下一个数据字节。

- a) 主机数据计数器减 1，如果发送的不是最后一个数据字节就跳至第 e) 步;
- b) 向 I2C\_CR 写入 0x14 来置位 STO 和 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 退出;
- e) 将主发送缓冲区的下一个数据字节装入 I2C\_DATA;
- f) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- g) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- h) 主机发送缓冲区指针加 1;
- i) 退出。

#### 4. 状态: 0x30 已发送数据并接收到非应答。即将发送停止条件;

- a) 向 I2C\_CR 写入 0x14 来置位 STO 和 AA 位;
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- c) 退出。

#### 5. 状态: 0x38 仲裁已在发送从机地址和写操作位或数据的过程中丢失。总线已被释放且进入 非寻址的从机模式。当总线再次空闲时将发送一个新的起始条件。

- a) 向 I2C\_CR 写入 0x24 来置位 STA 和 AA 位;
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- c) 退出。



### 23.6-8 主接收状态

#### 1. 状态: 0x40

前面的状态是 08 或 10 表示已发送从机地址和读操作位，并接收到 ACK。将接收数据并返回 ACK。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 退出。

#### 2. 状态: 0x48

已发送从机地址和读操作位，并接收到非应答。将发送停止条件。

- a) 向 I2C\_CR 写入 0x14 来置位 STO 和 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 退出。

#### 3. 状态: 0x50

已接收到数据，并返回 ACK。将从 I2C\_DATA 读取数据。将接收其它的数据。如果这是最后一个数据字节，则返回非应答，否则返回 ACK。

- a) 读取 I2C\_DATA 中的数据字节，存放到主机接收缓冲区；
- b) 主机数据计数器减 1，如果不是最后一个数据字节就跳到第 e) 步；
- c) 向 I2C\_CR 写入 0xF3 来清除 SI 标志和 AA 位；
- d) 退出；
- e) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- f) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- g) 主机接收缓冲区指针加 1；
- h) 退出。

#### 4. 状态: 0x58

已接收到数据，已返回非应答。将从 I2C\_DATA 中读取数据并发送停止条件。

- a) 读取 I2C\_DATA 中的数据字节，存放到主机接收缓冲区；
- b) I2C\_CR 写入 0x14 来置位 STO 和 AA 位；
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- d) 退出。



### 23.6.9 从接收状态

#### 1. 状态: 0x60

已接收到自身从机地址和写操作位，已返回 ACK。将接收数据和返回 ACK。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 建立从接收模式数据缓冲区；
- d) 初始化从机数据计数器；
- e) 退出。

#### 2. 状态: 0x68

用作总线主机时仲裁已在传输从机地址和 R/W 位时丢失。已接收到自身从机地址和写操作位，并已返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 STA 来重启主机模式。

- a) 向 I2C\_CR 写入 0x24 来置位 STA 和 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 建立从接收模式数据缓冲区；
- d) 初始化从机数据计数器；
- e) 退出。

#### 3. 状态: 0x70

已接收到通用调用和返回 ACK。将接收数据和返回 ACK。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 建立从接收模式数据缓冲区；
- d) 初始化从机数据计数器；
- e) 退出。

#### 4. 状态: 0x78

用作总线主机时仲裁已在传输从机地址和 R/W 位时丢失。已接收到通用调用和返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 STA 来重启主机模式。

- a) 向 I2C\_CR 写入 0x24 来置位 STA 和 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 建立从接收模式数据缓冲区；
- d) 初始化从机数据计数器；
- e) 退出。

#### 5. 状态: 0x80

之前寻址自身从机地址。已接收到数据并返回 ACK。将读取其它数据。

- a) 读取 I2C\_DATA 的数据字节，存放到从机接收缓冲区。



- b) 从机数据计数器减 1，如果不是最后一个数据字节就跳到第 e)步；
- c) 向 I2C\_CR 写入 0xF3 来清除 SI 标志和 AA 位；
- d) 退出；
- e) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- f) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- g) 从机接收缓冲区指针加 1；
- h) 退出。

6. 状态: 0x88

之前寻址自身从机地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机 模式。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 退出。

7. 状态: 0x90

之前寻址通用调用地址。已接收到数据并返回 ACK。将保存接收到的数据。只接收第一个数据字节并返回 ACK。接收其它数据字节后返回非应答。

- a) 读取 I2C\_DATA 的数据字节，并放入从机接收缓冲区；
- b) 向 I2C\_CR 写入 0xF3 来清除 SI 标志和 AA 位；
- c) 退出。

8. 状态: 0x98

之前寻址通用调用地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机 模式。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 退出。

9. 状态: 0xA0

已接收停止条件或重复起始条件，但仍作为从机寻址。不保存接收到的数据。进入非寻址的从机 模式。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位；
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志；
- c) 退出。



### 23.6-10 从发送状态

#### 1. 状态: 0xA8

已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。

- a) 将从机发送缓冲区的第一个数据字节装入 I2C\_DATA;
- b) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 建立从发送模式数据缓冲区;
- e) 从机发送缓冲区指针加 1;
- f) 退出。

#### 2. 状态: 0xB0

用作总线主机时，在传输从机地址和 R/W 位时丢失仲裁。已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。当总线再次空闲后置位 STA 来重启主机模式。

- a) 将从机发送缓冲区的第一个数据字节装入 I2C\_DATA;
- b) 向 I2C\_CR 写入 0x24 来置位 STA 和 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 建立从发送模式数据缓冲区;
- e) 从机发送缓冲区指针加 1;
- f) 退出。

#### 3. 状态: 0xB8

已发送数据并接收到 ACK。将发送数据和接收 ACK 位。

- a) 将从机发送缓冲区的数据字节装入 I2C\_DATA;
- b) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- c) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- d) 从机发送缓冲区指针加 1;
- e) 退出。

#### 4. 状态: 0xC0

已发送数据并接收到非应答。进入非寻址的从机模式。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;
- c) 退出。

#### 5. 状态: 0xC8

已发送最后一个数据字节并接收到 ACK。进入非寻址的从机模式。

- a) 向 I2C\_CR 写入 0x04 来置位 AA 位;
- b) 向 I2C\_CR 写入 0xF7 来清除 SI 标志;



c) 退出。

### 23.7 I2CX\* 寄存器列表

I2C0 基地址: 0x4000\_0C00

I2C1 基地址: 0x40000E00

偏移地址	名称	寄存器描述	复位值
0x00	I2CX*_CR	I2C 配置寄存器。	0x00000000
0x04	I2CX*_DATA	I2C 数据寄存器。	0x00000000
0x08	I2CX*_ADDR	I2C 地址寄存器。	0x00000000
0x0c	I2CX*_SR	I2C 状态寄存器。	0x000000F8
0x10	I2CX*_TIMRUN	I2C 波特率计数器使能寄存器。	0x00000000
0x14	I2CX*_BAUDCR	I2C 波特率计数器配置寄存器。	0x00000000

Table 23-1 I2CX\* 寄存器列表

Note : I2CX\* : I2C0,I2C1



## 23.8 I2C 寄存器说明

### 23.8-1 I2C 配置寄存器(I2CX\*\_CR)

偏移地址: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								ENS	STA	STO	SI	AA	INTEN	H1M	
保留								R/W							

位	标记	功能描述	复位值	读写
31:7	-	保留	0x0	-
6	ENS	I2C 模块使能。 0: 禁止 1: 使能	0	R/W
5	STA	开始标志使能。 0: 禁止 1: 使能	0	R/W
4	STO	停止标志使能。 0: 禁止 1: 使能	0	R/W
3	SI	I2C 中断标志位。	0	R/W
2	AA	应答标志使能。 0: 禁止 1: 使能	0	R/W
1	INTEN	I2CX 中断使能 0: 使能 I2CX 中断發生 1: 禁止 I2CX 中断發生	0	R/W
0	H1M	I2C 高速 1Mbps 模式使能。 0: 禁止 1: 使能	0	R/W



### 23.8-2 I2C 数据寄存器(I2CX\*\_DATA)

偏移地址: 0x04 复位值: 0x0000 0000

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	I2CDAT[7:0] ]	I2C 数据寄存器。 在 I2C 发送模式下，写发送数据到这个寄存器。在 I2C 接收模式下，读接收数据从这个寄存器。	0x00	R/W

### 23.8-3 I2C 地址寄存器(I2CX\*\_ADDR)

偏移地址: 0x08 复位值: 0x0000 0000

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:1	I2CADR[6:0]	I2C 从机模式地址。	0x0	R/ W
0	GC	广播地址应答使能。 0: 禁止 1: 使能	0	R/ W



### 23.8-4 I2C 状态寄存器(I2C\_SR)

偏移地址: 0x0C 复位值: 0x0000 00F8



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	I2CSTA[7:0]	I2C 状态寄存器。	0xF8	RO



### 23.8-5 I2C 计数器 DMA 使能寄存器(I2CX\*\_TIMDMA)

偏移地址: 0x10 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7	DMATEN	I2C1 DMA TX enable (Only I2C0) 1 : Enable 0 : Disable	0	R/W
6	DMAREN	I2C1 DMA RX enable (Only I2C0) 1 : Enable 0 : Disable	0	R/W
5:1	-	保留	0x0	-
0	TME	波特率计数器使能寄存器。 0: 禁止 1: 使能	0	R/W



### 23.8-6 I2C 波特率计数器配置寄存器(I2CX\*\_BAUDCR)

偏移地址: 0x14 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	TM	TM: 波特率计数器配置值。 $F_o = F_i / 8 * (N + 1)$ , N=TM, N>0	0x0	R/W



## 【24】 串行外设接口(SPI/I2S)

### 24.1 SPI 简介

SPI(Serial Peripheral Interface)总线是一种同步串行外设接口，它可以使 MCU 与各种外围设备以串行的方向进行信息交换。SPI 接口使用 4 条线：串行时钟线(SCLK)、主机输出/从机输入线(MOSI)、主机输入/从机输出线(MISO)、低电平有效从机选择线(SSN)。

在大容量产品和互联型产品上，SPI 接口可以配置为支持 SPI 协议或者支持 I2S 音频协议。SPI 接口默认工作在 SPI 方式，可以通过软件把功能从 SPI 模式切换到 I2S 模式。

在小容量和中容量产品上，不支持 I2S 音频协议。

串行外设接口(SPI)允许芯片与外部设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为外部从设备提供通信时钟(SCK)。接口还能以多主配置方式工作。

它可用于多种用途，包括使用一条双向数据线的双线单工同步传输，还可使用 CRC 校验的可靠通信。

I2S 也是一种 3 引脚的同步串行接口通讯协议。它支持四种音频标准，包括飞利浦 I2S 标准，MSB 和 LSB 对齐标准，以及 PCM 标准。它在半双工通讯中，可以工作在主和从 2 种模式下。当它作为主设备时，通过接口向外部的从设备提供时钟信号。

### 24.2 SPI/I2S 主要特性

**SPI 控制器支持以下特性：**

- 3 线全双工同步传输
- 带或不带第三根双向数据线的双线单工同步传输
- 8 或 16 位传输帧格式选择
- 主或从操作
- 支持多主模式
- 8 个主模式波特率预分频系数(最大为 fPCLK/2)
- 从模式频率：最慢频率：fPCLK/128 ( $SCK \geq fPCLK/128$ )
- 主模式和从模式的快速通信
- 主模式和从模式下均可以由软件或硬件进行 NSS 管理：主/从操作模式的动态改变
- 可编程的时钟极性和相位
- 可编程的数据顺序，MSB 在前或 LSB 在前
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持可靠通信的硬件 CRC
  - 在发送模式下，CRC 值可以被作为最后一个字节发送
  - 在全双工模式中对接收到的最后一个字节自动进行 CRC 校验



- 可触发中断的主模式故障、过载以及 CRC 错误标志
- 支持 DMA 功能的 1 字节发送和接收缓冲器：产生发送和接受请求

#### I2S 功能

- 单工通信(仅发送或接收)
- 主或者从操作
- 8 位线性可编程预分频器，获得精确的音频采样频率(8KHz 到 96kHz)
- 数据格式可以是 16 位，24 位或者 32 位
- 音频信道固定数据包帧为 16 位(16 位数据帧)或 32 位(16、24 或 32 位数据帧)
- 可编程的时钟极性(稳定态)
- 从发送模式下的下溢标志位和主/从接收模式下的溢出标志位
- 16 位数据寄存器用来发送和接收，在通道两端各有一个寄存器
- 支持的 I2S 协议：
  - I2S 飞利浦标准
  - MSB 对齐标准(左对齐)
  - LSB 对齐标准(右对齐)
  - PCM 标准(16 位通道帧上带长或短帧同步或者 16 位数据帧扩展为 32 位通道帧)
- 数据方向总是 MSB 在先
- 发送和接收都具有 DMA 能力
- 主时钟可以输出到外部音频设备，比率固定为  $256 \times F_s$ ( $F_s$  为音频采样频率)
- 在互联型产品中，两个 I2S 模块(I2S1 和 I2S2)可用 PLL 产生更加精准得时钟



## 24.3 功能描述

### 24.3-1 概述

SPI 的方框图见下图。

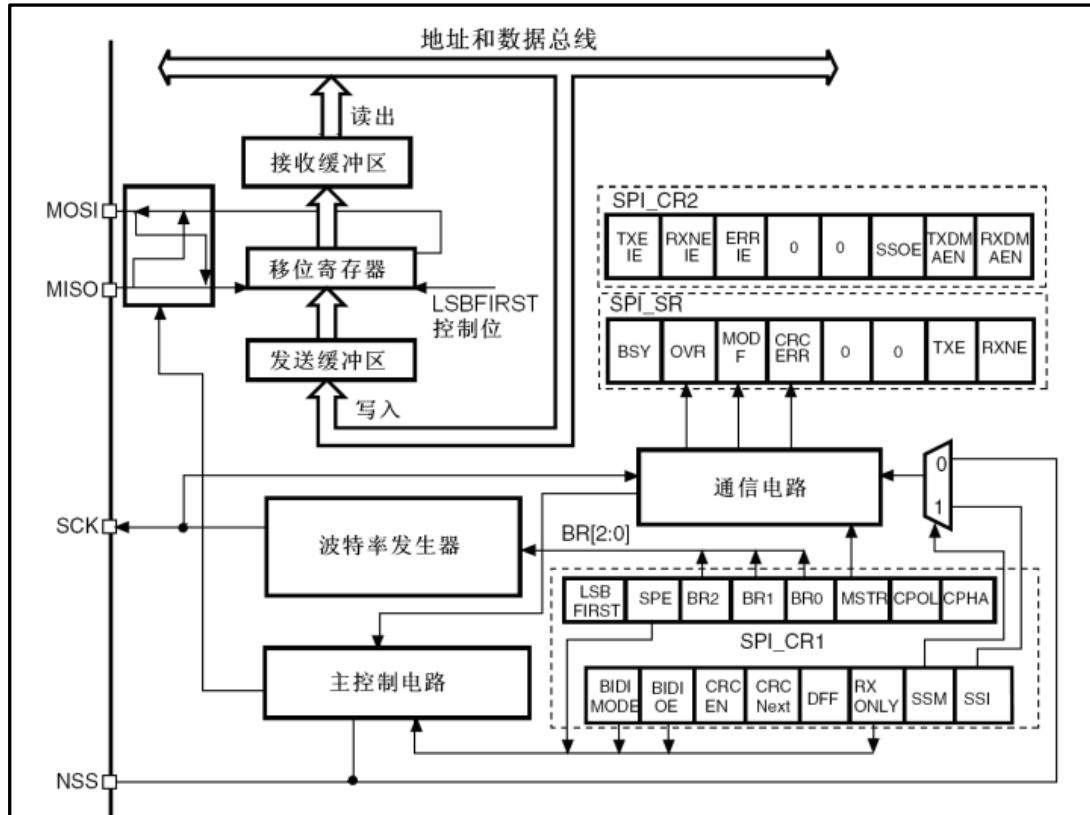


Figure 24-1 SPI 的方框图

通常 SPI 通过 4 个引脚与外部器件相连：

- **MISO:** 主设备输入/从设备输出引脚。该引脚在从模式下发送数据，在主模式下接收数据。
- **MOSI:** 主设备输出/从设备输入引脚。该引脚在主模式下发送数据，在从模式下接收数据。
- **SCK:** 串口时钟，作为主设备的输出，从设备的输入
- **NSS:** 从设备选择。这是一个可选的引脚，用来选择主/从设备。它的功能是用来作为“片选引脚”，让主设备可以单独地与特定从设备通讯，避免数据线上的冲突。从设备的 NSS 引脚可以由主设备的一个标准 I/O 引脚来驱动。一旦被使能(SSOE 位)，NSS 引脚也可以作为输出引脚，并在 SPI 处于主模式时拉低；此时，所有的 SPI 设备，如果它们的 NSS 引脚连接到主设备的 NSS 引脚，则会检测到低电平，如果它们被设置为 NSS 硬件模式，就会自动进入从设备状态。当配置为主设备、NSS 配置为输入引脚(MSTR=1, SSOE=0)时，如果 NSS 被拉低，则这个 SPI 设备进入主模式失败状态：即 MSTR 位被自动清除，此设备进入从模式(参见)



下图是一个单主和单从设备互连的例子。

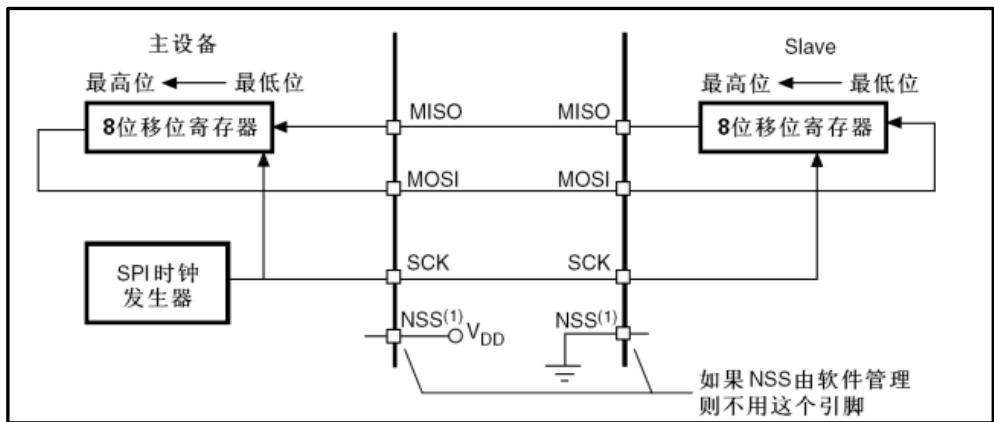


Figure 24-2 单主和单从应用

### 1. 这里 NSS 引脚设置为输入

MOSI 脚相互连接，MISO 脚相互连接。这样，数据在主和从之间串行地传输(MSB 位在前)。

通信总是由主设备发起。主设备通过 MOSI 脚把数据发送给从设备，从设备通过 MISO 引脚回传数据。这意味着全双工通信的数据输出和数据输入是用同一个时钟信号同步的；时钟信号由主设备通过 SCK 脚提供。

从选择(NSS)脚管理

有 2 种 NSS 模式：

- 软件 NSS 模式：可以通过设置 SPI\_CR1 寄存器的 SSM 位来使能这种模式(见 )。在这种模式下 NSS 引脚可以用作它用，而内部 NSS 信号电平可以通过写 SPI\_CR1 的 SSI 位来驱动
- 硬件 NSS 模式，分两种情况：(Fig, 25-3)
  - NSS 输出被使能：当工作为主 SPI，并且 NSS 输出已经通过 SPI\_CR2 寄存器的 SSOE 位使能，这时 NSS 引脚被拉低，所有 NSS 引脚与这个主 SPI 的 NSS 引脚相连并配置为硬件 NSS 的 SPI 设备，将自动变成从 SPI 设备。当一个 SPI 设备需要发送广播数据，它必须拉低 NSS，这意味着总线上有另外一个主设备在通信，这时将产生一个硬件失败错误(Hard Fault)。
  - NSS 输出被关闭：允许操作于多主环境。

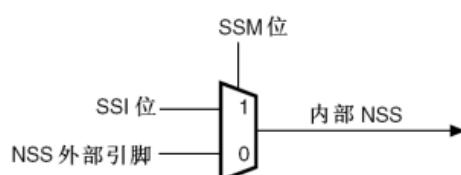


Figure 24-3 硬件/软件的从选择管理



### 时钟信号的相位和极性

SPI\_CR 寄存器的 CPOL 和 CPHA 位，能够组合成四种可能的时序关系。CPOL(时钟极性)位控制在没有数据传输时时钟的空闲状态电平，此位对主模式和从模式下的设备都有效。如果 CPOL 被清' 0'，SCK 引脚在空闲状态保持低电平；如果 CPOL 被置' 1'，SCK 引脚在空闲状态保持高电平。如果 CPHA(时钟相位)位被置' 1'，SCK 时钟的第二个边沿(CPOL 位为 0 时就是下降沿，CPOL 位为' 1' 时就是上升沿)进行数据位的采样，数据在第二个时钟边沿被锁存。如果 CPHA 位被清' 0'，SCK 时钟的第一个边沿(CPOL 位为' 0' 时就是下降沿，CPOL 位为' 1' 时就是上升沿)进行数据位采样，数据在第一个时钟边沿被锁存。

CPOL 时钟极性和 CPHA 时钟相位的组合选择数据捕捉的时钟边沿。

图 212 显示了 SPI 传输的 4 种 CPHA 和 CPOL 位组合。此图可以解释为主设备和从设备的 SCK 脚、MISO 脚、MOSI 脚直接连接的主或从时序图。

- 注意：
1. 在改变 CPOL/CPHA 位之前，必须清除 SPE 位将 SPI 禁止。
  2. 主和从必须配置成相同的时序模式。
  3. SCK 的空闲状态必须和 SPI\_CR1 寄存器指定的极性一致(CPOL 为' 1' 时，空闲时应上拉 SCK 为高电平；CPOL 为' 0' 时，空闲时应下拉 SCK 为低电平)。
  4. 数据帧格式(8 位或 16 位)由 SPI\_CR1 寄存器的 DFF 位选择，并且决定发送/接收的数据长度。

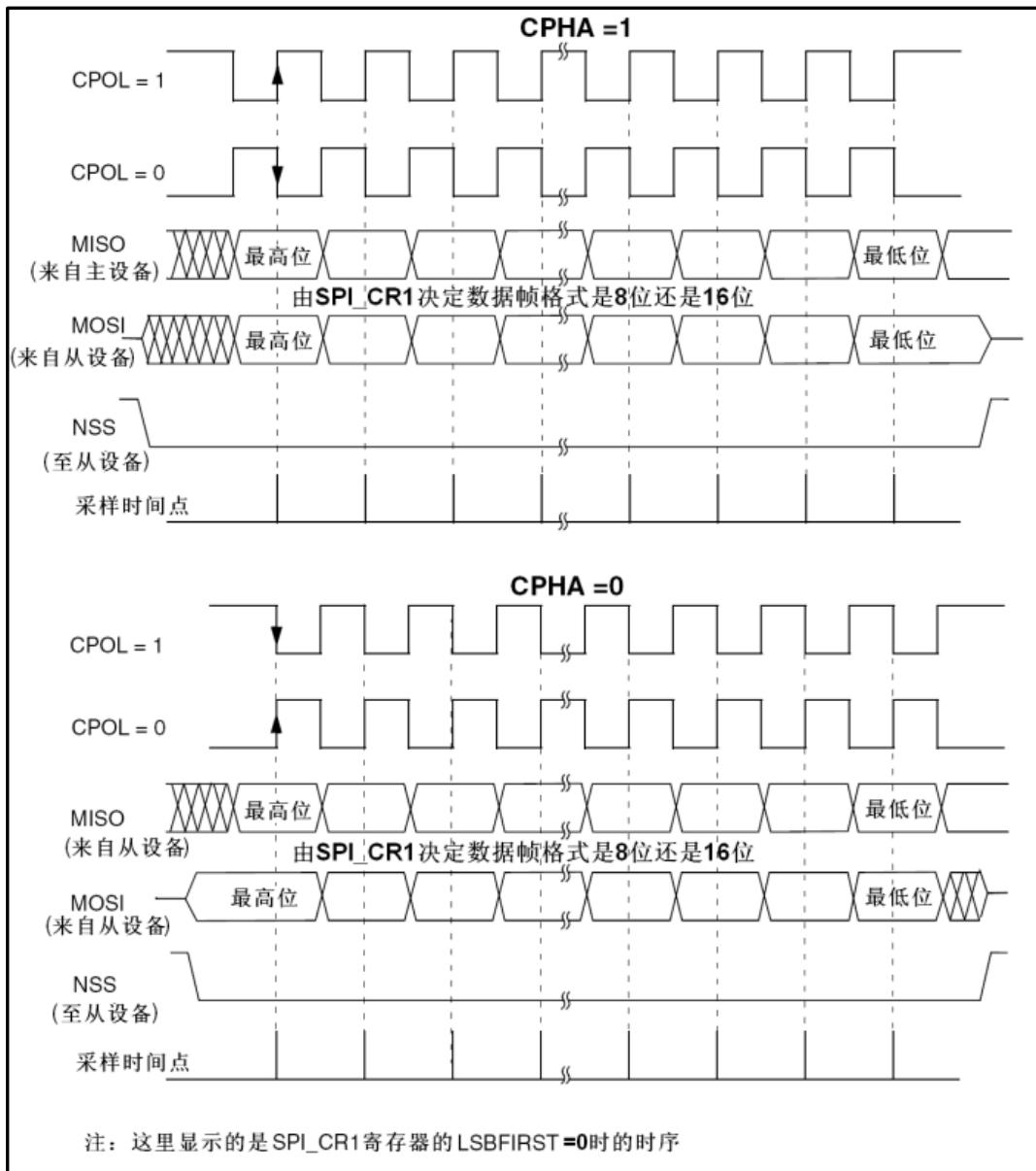


Figure 24-4 数据时钟时序图

#### 数据帧格式

根据 SPI\_CR1 寄存器中的 LSBFIRST 位，输出数据位时可以 MSB 在先也可以 LSB 在先。

根据 SPI\_CR1 寄存器的 DFF 位，每个数据帧可以是 8 位或是 16 位。所选择的数据帧格式对发送和 / 或接收都有效。



### 24.3-2 配置 SPI 为从模式

在从模式下，SCK 引脚用于接收从主设备来的串行时钟。SPI\_CR1 寄存器中 BR[2:0]的设置不影响数据传输速率。

**注: (1)** 主设备来的频率： $SCK \geq f_{PCLK}/128$

**(2)** 建议在主设备发送时钟之前使能 SPI 从设备，否则可能会发生意外的数据传输。在通信时钟的第一个边沿到来之前或正在进行的通信结束之前，从设备的数据寄存器必须就绪。在使能从设备和主设备之前，通信时钟的极性必须处于稳定的数值。

请按照以下步骤配置 SPI 为从模式：

配置步骤

1. 设置 DFF 位以定义数据帧格式为 8 位或 16 位。
2. 选择 CPOL 和 CPHA 位来定义数据传输和串行时钟之间的相位关系(见图 212)。为保证正确的数据传输，从设备和主设备的 CPOL 和 CPHA 位必须配置成相同的方式。
3. 帧格式(SPI\_CR1 寄存器中的 LSBFIRST 位定义的“ MSB 在前” 还是“ LSB 在前” )必须与主设备相同。
4. 硬件模式下(参考从选择(NSS)脚管理部分)，在完整的数据帧(8 位或 16 位)传输过程中，NSS 引脚必须为低电平。在 NSS 软件模式下，设置 SPI\_CR1 寄存器中的 SSM 位并清除 SSI 位。
5. 清除 MSTR 位、设置 SPE 位(SPI\_CR1 寄存器)，使相应引脚工作于 SPI 模式下。

在这个配置中，MOSI 引脚是数据输入，MISO 引脚是数据输出。

数据发送过程

在写操作中，数据字被并行地写入发送缓冲器。

当从设备收到时钟信号，并且在 MOSI 引脚上出现第一个数据位时，发送过程开始(译注：此时第一个位被发送出去)。余下的位(对于 8 位数据帧格式，还有 7 位；对于 16 位数据帧格式，还有 15 位)被装进移位寄存器。当发送缓冲器中的数据传输到移位寄存器时，SPI\_SP 寄存器的 TXE 标志被设置，如果设置了 SPI\_CR2 寄存器的 TXEIE 位，将会产生中断。

数据接收过程

对于接收器，当数据接收完成时：

- 移位寄存器中的数据传送到接收缓冲器，SPI\_SR 寄存器中的 RXNE 标志被设置。
- 如果设置了 SPI\_CR2 寄存器中的 RXNEIE 位，则产生中断。

在最后一个采样时钟边沿后，RXNE 位被置‘ 1’，移位寄存器中接收到的数据字节被传送到接收缓冲器。当读 SPI\_DR 寄存器时，SPI 设备返回这个接收缓冲器的数值。

读 SPI\_DR 寄存器时，RXNE 位被清除。



### 24.3-3 配置 SPI 为主模式

在主配置时，在 SCK 脚产生串行时钟。配置步骤：

1. 通过 SPI\_CR1 寄存器的 BR[2:0]位定义串行时钟波特率。
2. 选择 CPOL 和 CPHA 位，定义数据传输和串行时钟间的相位关系(见图 212)。
3. 设置 DFF 位来定义 8 位或 16 位数据帧格式。
4. 配置 SPI\_CR1 寄存器的 LSBFIRST 位定义帧格式。
5. 如果需要 NSS 引脚工作在输入模式，硬件模式下，在整个数据帧传输期间应把 NSS 脚连接到高电平；在软件模式下，需设置 SPI\_CR1 寄存器的 SSM 位和 SSI 位。如果 NSS 引脚工作在输出模式，则只需设置 SSOE 位。
6. 必须设置 MSTR 位和 SPE 位(只当 NSS 脚被连到高电平，这些位才能保持置位)。

在这个配置中，MOSI 引脚是数据输出，而 MISO 引脚是数据输入。

配置 SPI 为主模式

#### 数据发送过程

当写入数据至发送缓冲器时，发送过程开始。

在发送第一个数据位时，数据字被并行地(通过内部总线)传入移位寄存器，而后串行地移出到 MOSI 脚上；MSB 在先还是 LSB 在先，取决于 SPI\_CR1 寄存器中的 LSBFIRST 位的设置。数据从发送缓冲器传输到移位寄存器时 TXE 标志将被置位，如果设置了 SPI\_CR1 寄存器中的 TXEIE 位，将产生中断。

#### 数据接收过程

对于接收器来说，当数据传输完成时：

- 传送移位寄存器里的数据到接收缓冲器，并且 RXNE 标志被置位。
- 如果设置了 SPI\_CR2 寄存器中的 RXNEIE 位，则产生中断。

在最后采样时钟沿，RXNE 位被设置，在移位寄存器中接收到的数据字被传送到接收缓冲器。读 SPI\_DR 寄存器时，SPI 设备返回接收缓冲器中的数据。读 SPI\_DR 寄存器将清除 RXNE 位。

一旦传输开始，如果下一个将发送的数据被放进了发送缓冲器，就可以维持一个连续的传输流。在试图写发送缓冲器之前，需确认 TXE 标志应该为‘1’。

注： 在 NSS 硬件模式下，从设备的 NSS 输入由 NSS 引脚控制或另一个由软件驱动的 GPIO 引脚控制。



#### 24.3-4 配置 SPI 为单工通信

SPI 模块能够以两种配置工作于单工方式:

- 1 条时钟线和 1 条双向数据线;
- 1 条时钟线和 1 条数据线(只接收或只发送);  
1 条时钟线和 1 条双向数据线(BIDIMODE=1)

设置 SPI\_CR1 寄存器中的 BIDIMODE 位而启用此模式。在这个模式下, SCK 引脚作为时钟, 主设备使用 MOSI 引脚而从设备使用 MISO 引脚作为数据通信。传输的方向由 SPI\_CR1 寄存器里的 BIDIOE 控制, 当这个位是' 1' 的时候, 数据线是输出, 否则是输入。

1 条时钟和 1 条单向数据线(BIDIMODE=0)

在这个模式下, SPI 模块可以或者作为只发送, 或者作为只接收。

- 只发送模式类似于全双工模式(BIDIMODE=0, RXONLY=0): 数据在发送引脚(主模式时是 MOSI、从模式时是 MISO)上传输, 而接收引脚(主模式时是 MISO、从模式时是 MOSI)可以作为通用的 I/O 使用。此时, 软件不必理会接收缓冲器中的数据(如果读出数据寄存器, 它不包含任何接收数据)。
- 在只接收模式, 可以通过设置 SPI\_CR2 寄存器的 RXONLY 位而关闭 SPI 的输出功能; 此时, 发送引脚(主模式时是 MOSI、从模式时是 MISO)被释放, 可以作为其它功能使用。

配置并使能 SPI 模块为只接收模式的方式是:

- 在主模式时, 一旦使能 SPI, 通信立即启动, 当清除 SPE 位时立即停止当前的接收。在此模式下, 不必读取 BSY 标志, 在 SPI 通信期间这个标志始终为' 1' 。
- 在从模式时, 只要 NSS 被拉低(或在 NSS 软件模式时, SSI 位为' 0' )同时 SCK 有时钟脉冲, SPI 就一直在接收。



#### 24.3.5 数据发送与接收过程

##### 接收与发送缓冲器

在接收时，接收到的数据被存放在一个内部的接收缓冲器中；在发送时，在被发送之前，数据将首先被存放在一个内部的发送缓冲器中。

对 SPI\_DR 寄存器的读操作，将返回接收缓冲器的内容；写入 SPI\_DR 寄存器的数据将被写入发送缓冲器中。

##### 主模式下开始传输

- 全双工模式(BIDIMODE=0 并且 RXONLY=0)
  - 当写入数据到 SPI\_DR 寄存器(发送缓冲器)后，传输开始；
  - 在传送第一位数据的同时，数据被并行地从发送缓冲器传送到 8 位的移位寄存器中，然后按顺序被串行地移位送到 MOSI 引脚上；
  - 与此同时，在 MISO 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI\_DR 寄存器(接收缓冲器)中。
- 单向的只接收模式(BIDIMODE=0 并且 RXONLY=1)
  - SPE=1 时，传输开始；
  - 只有接收器被激活，在 MISO 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI\_DR 寄存器(接收缓冲器)中。
- 双向模式，发送时(BIDIMODE=1 并且 BIDIOE=1)
  - 当写入数据到 SPI\_DR 寄存器(发送缓冲器)后，传输开始；
  - 在传送第一位数据的同时，数据被并行地从发送缓冲器传送到 8 位的移位寄存器中，然后按顺序被串行地移位送到 MOSI 引脚上；
  - 不接收数据。
- 双向模式，接收时(BIDIMODE=1 并且 BIDIOE=0)
  - SPE=1 并且 BIDIOE=0 时，传输开始；
  - 在 MOSI 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI\_DR 寄存器(接收缓冲器)中。
  - 不激活发送器，没有数据被串行地送到 MOSI 引脚上。

##### 从模式下开始传输

- 全双工模式(BIDIMODE=0 并且 RXONLY=0)
  - 当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始，随后的数据位依次移动进入移位寄存器；
  - 与此同时，在传输第一个数据位时，发送缓冲器中的数据被并行地传送到 8 位的移位寄存器，随后被串行地发送到 MISO 引脚上。软件必须保证在 SPI 主设备开始数据传输之前在发送寄存器中写入要发送的数据。



- 单向的只接收模式(BIDIMODE=0 并且 RXONLY=1)
  - 当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始，随后数据位依次移动进入移位寄存器；
  - 不启动发送器，没有数据被串行地传送到 MISO 引脚上。
- 双向模式，发送时(BIDIMODE=1 并且 BIDIOE=1)
  - 当从设备接收到时钟信号并且发送缓冲器中的第一个数据位被传送到 MISO 引脚上的时候，数据传输开始；
  - 在第一个数据位被传送到 MISO 引脚上的同时，发送缓冲器中要发送的数据被平行地传送到 8 位的移位寄存器中，随后被串行地传送到 MISO 引脚上。软件必须保证在 SPI 主设备开始数据传输之前在发送寄存器中写入要发送的数据；
  - 不接收数据。
- 双向模式，接收时(BIDIMODE=1 并且 BIDIOE=0)
  - 当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始；
  - 从 MISO 引脚上接收到的数据被串行地传送到 8 位的移位寄存器中，然后被平行地传送到 SPI\_DR 寄存器(接收缓冲器)；
  - 不启动发送器，没有数据被串行地传送到 MISO 引脚上。

#### 处理数据的发送与接收

当数据从发送缓冲器传送到移位寄存器时，设置 TXE 标志(发送缓冲器空)，它表示内部的发送缓冲器可以接收下一个数据；如果在 SPI\_CR2 寄存器中设置了 TXEIE 位，则此时会产生一个中断；写入 SPI\_DR 寄存器即可清除 TXE 位。

注：在写入发送缓冲器之前，软件必须确认 TXE 标志为‘1’，否则新的数据会覆盖已经在发送缓冲器中的数据。

在采样时钟的最后一个边沿，当数据被从移位寄存器传送到接收缓冲器时，设置 RXNE 标志(接收缓冲器非空)；它表示数据已经就绪，可以从 SPI\_DR 寄存器读出；如果在 SPI\_CR2 寄存器中设置了 RXNEIE 位，则此时会产生一个中断；读出 SPI\_DR 寄存器即可清除 RXNE 标志位。

在一些配置中，传输最后一个数据时，可以使用 BSY 标志等待数据传输的结束。

#### 主或从模式下(BIDIMODE=0 并且 RXONLY=0)全双工发送和接收过程模式

软件必须遵循下述过程，发送和接收数据(见图 25.5,6)：

1. 设置 SPE 位为‘1’，使能 SPI 模块；
2. 在 SPI\_DR 寄存器中写入第一个要发送的数据，这个操作会清除 TXE 标志；
3. 等待 TXE=1，然后写入第二个要发送的数据。等待 RXNE=1，然后读出 SPI\_DR 寄存器并获得第一个接收到的数据，读 SPI\_DR 的同时清除了 RXNE 位。重复这些操作，发送后续的数据同时接收 n-1 个数据；
4. 等待 RXNE=1，然后接收最后一个数据；



5. 等待 TXE=1，在 BSY=0 之后关闭 SPI 模块。也可以在响应 RXNE 或 TXE 标志的上升沿产生的中断的处理程序中实现这个过程。

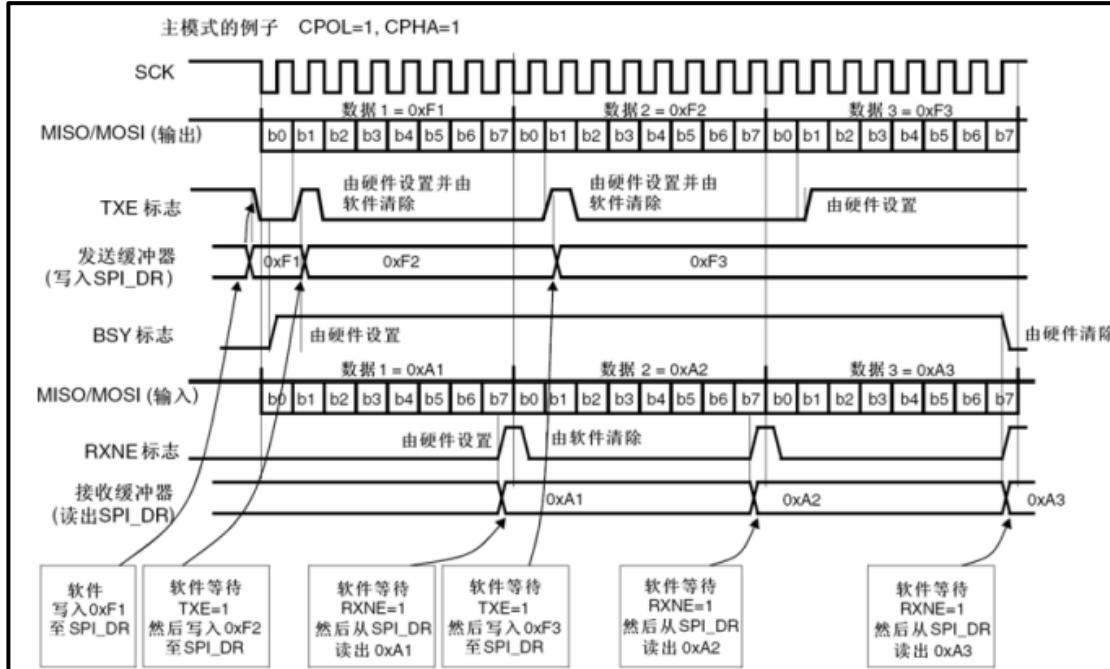


Figure 24-5 主模式、全双工模式下(BIDIMODE=0 并且 RXONLY=0)连续传输时, TXE/RXNE/BSY 的变化示意图

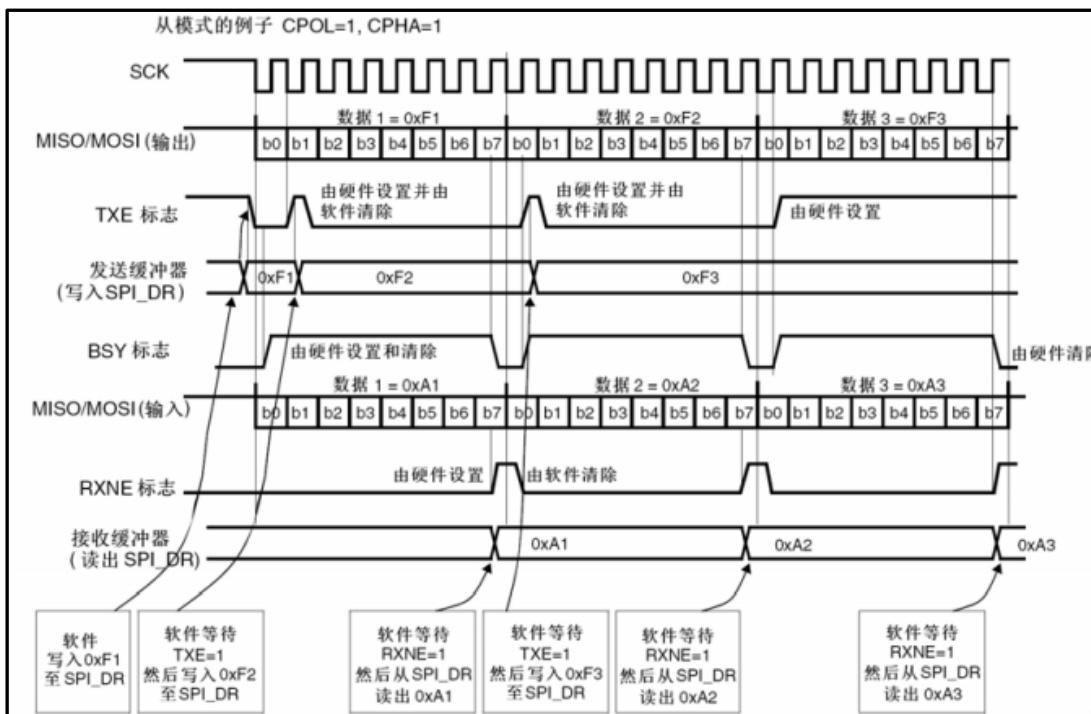


Figure 24-6 从模式、全双工模式下(**BIDIMODE=0** 并且 **RXONLY=0**)连续传输时, **TXE/RXNE/BSY** 的变化示意图

#### 只发送过程(**BIDIMODE=0** 并且 **RXONLY=0**)

在此模式下, 传输过程可以简要说明如下, 使用 BSY 位等待传输的结束(见图 25-7,8):

1. 设置 SPE 位为' 1' , 使能 SPI 模块;
2. 在 SPI\_DR 寄存器中写入第一个要发送的数据, 这个操作会清除 TXE 标志;
3. 等待 TXE=1, 然后写入第二个要发送的数据。重复这个操作, 发送后续的数据;
4. 写入最后一个数据到 SPI\_DR 寄存器之后, 等待 TXE=1; 然后等待 BSY=0, 这表示最后一个数据的传输已经完成。也可以在响应 TXE 标志的上升沿产生的中断的处理程序中实现这个过程。

注: 1. 对于不连续的传输, 在写入 SPI\_DR 寄存器的操作与设置 BSY 位之间有 2 个 APB 时钟周期延迟, 因此在只发送模式下, 写入最后一个数据后, 最好先等待 TXE=1, 然后再等待 BSY=0。  
2. 只发送模式下, 在传输 2 个数据之后, 由于不会读出接收到的数据, SPI\_SR 寄存器中的 OVR 位会变为' 1' 。(译注: 软件不必理会这个 OVR 标志位)

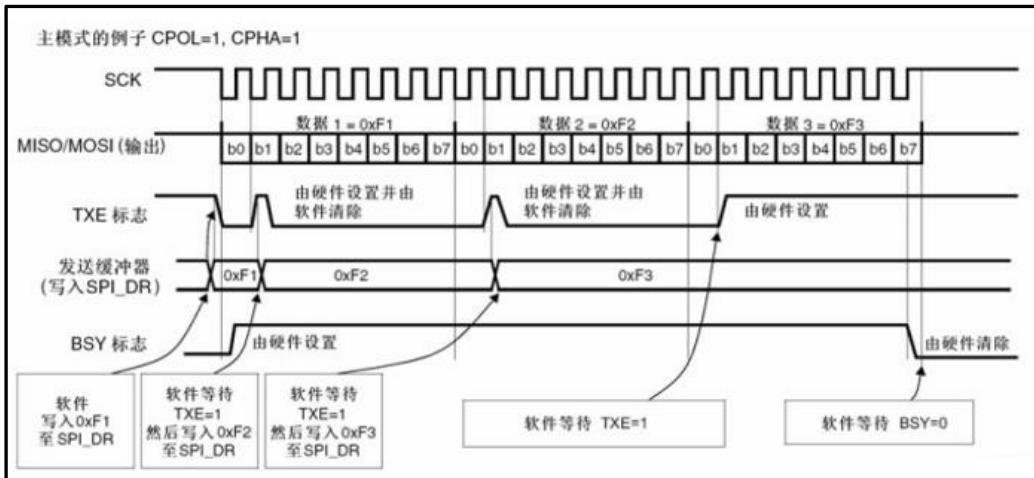


Figure 24-7 主设备只发送模式(BIDIMODE=0 并且 RXONLY=0)下连续传输时， TXE/BSY 变化示意图

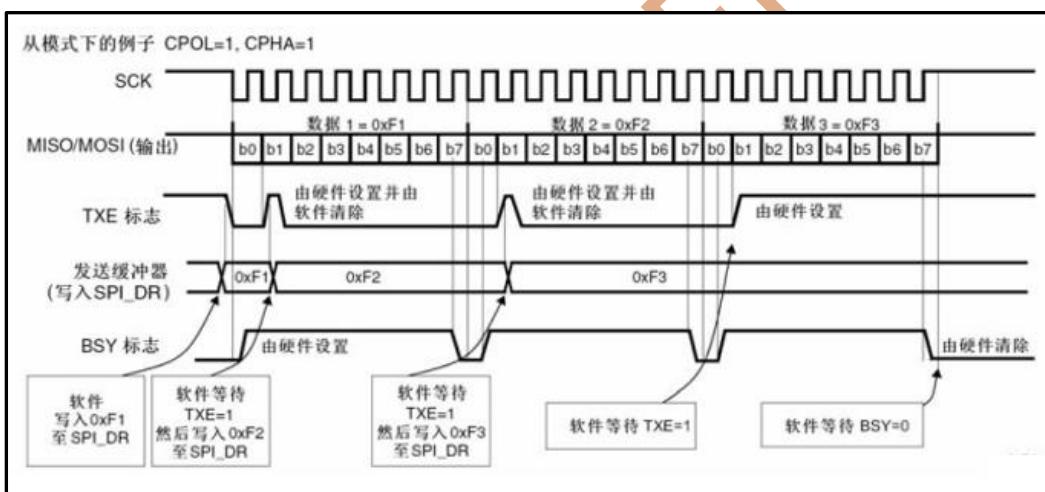


Figure 24-8 从设备只发送模式(BIDIMODE=0 并且 RXONLY=0)下连续传输时， TXE/BSY 变化示意

### 双向发送过程(BIDIMODE=1 并且 BIDIOE=1)

在此模式下，操作过程类似于只发送模式，不同的是：在使能 SPI 模块之前，需要在 SPI\_CR2 寄存器中同时设置 BIDIMODE 和 BIDIOE 位为‘1’。

### 单向只接收模式(BIDIMODE=0 并且 RXONLY=1)

在此模式下，传输过程可以简要说明如下(见图 25-9)：

1. 在 SPI\_CR2 寄存器中，设置 RXONLY=1；
2. 设置 SPE=1，使能 SPI 模块：
  - a) 主模式下，立刻产生 SCK 时钟信号，在关闭 SPI(SPE=0)之前，不断地接收串行数据；
  - b) 从模式下，当 SPI 主设备拉低 NSS 信号并产生 SCK 时钟时，接收串行数据。



3. 等待 RXNE=1，然后读出 SPI\_DR 寄存器以获得收到的数据(同时会清除 RXNE 位)。重复这个操作接收所有数据。

也可以在响应 RXNE 标志的上升沿产生的中断的处理程序中实现这个过程。

注： 如果在最后一个数据传输结束后关闭 SPI 模块，请按照第 25.3.8 节的建议操作。

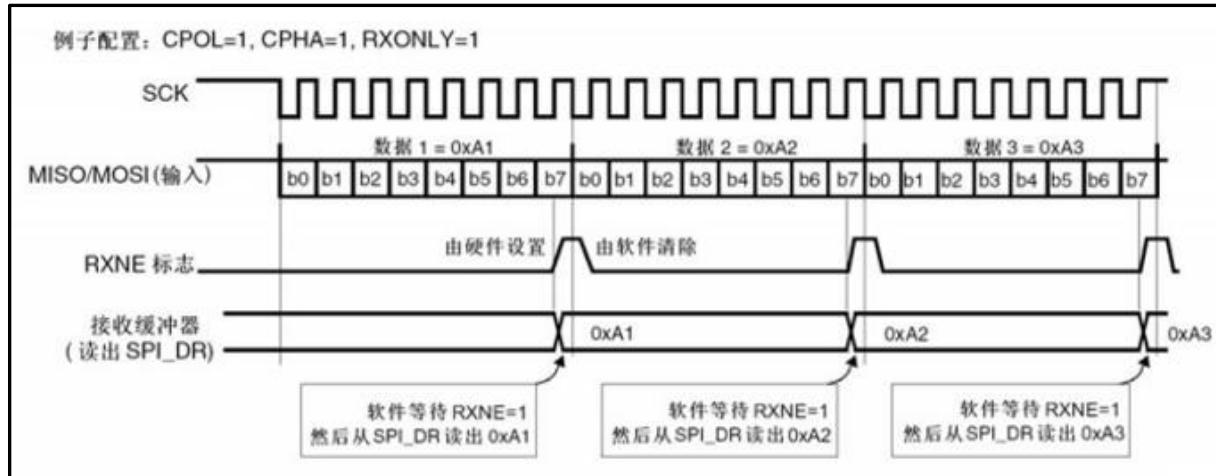


Figure 24-9 只接收模式(BIDIMODE=0 并且 RXONLY=1)下连续传输时，RXNE 变化示意图

#### 单向接收过程(BIDIMODE=1 并且 BIDIOE=0)

在此模式下，操作过程类似于只接收模式，不同的是：在使能 SPI 模块之前，需要在 SPI\_CR2 寄存器中设置 BIDIMODE 为‘1’并清除 BIDIOE 位为‘0’。

#### 连续和非连续传输

当在主模式下发送数据时，如果软件足够快，能够在检测到每次 TXE 的上升沿(或 TXE 中断)，并立即在正在进行的传输结束之前写入 SPI\_DR 寄存器，则能够实现连续的通信；此时，在每个数据项的传输之间的 SPI 时钟保持连续，同时 BSY 位不会被清除。



如果软件不够快，则会导致不连续的通信；这时，在每个数据传输之间会被清除(见图 25-10)。

在主模式的只接收模式下(**RXONLY=1**)，通信总是连续的，而且 **BSY** 标志始终为‘1’。

在从模式下，通信的连续性由 SPI 主设备决定。不管怎样，即使通信是连续的，**BSY** 标志会在每个数据项之间至少有一个 SPI 时钟周期为低(见图 25-8)。

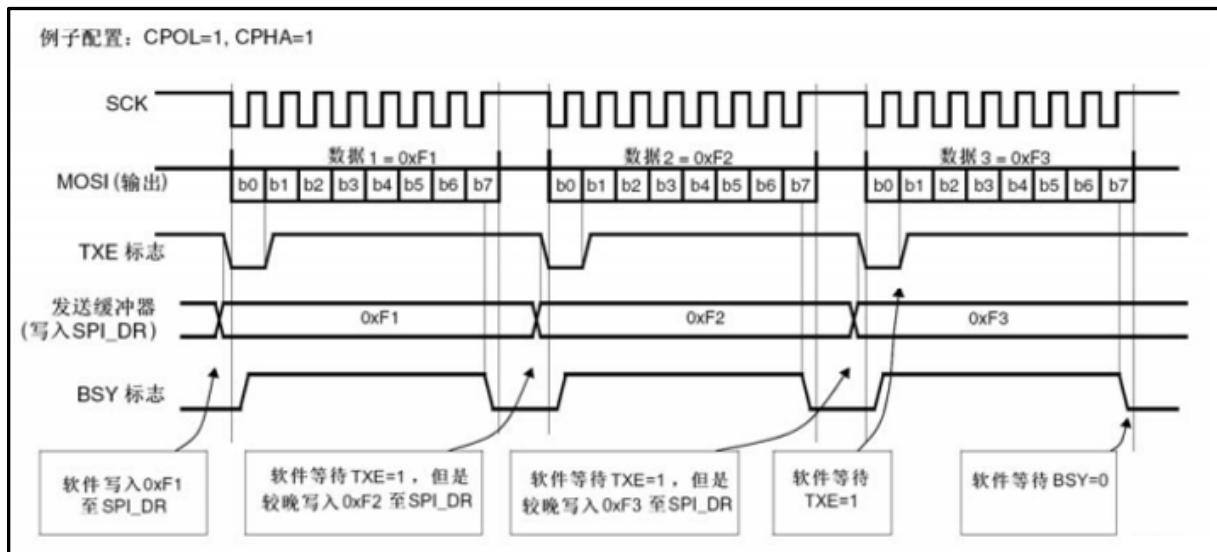


Figure 24-10 非连续传输发送(**BIDIMODE=0** 并且 **RXONLY=0**)时，**TXE/BSY** 变化示意图



### 24.3-6 CRC 计算

CRC 校验用于保证全双工通信的可靠性。数据发送和数据接收分别使用单独的 CRC 计算器。通过对每一个接收位进行可编程的多项式运算来计算 CRC。CRC 的计算是在由 SPI\_CR1 寄存器中 CPHA 和 CPOL 位定义的采样时钟边沿进行的。

注意：该 SPI 接口提供了两种 CRC 计算方法，取决于所选的发送和/或接收的数据帧格式：8 位数据帧采用 CR8；16 位数据帧采用 CRC16。

CRC 计算是通过设置 SPI\_CR1 寄存器中的 CRCEN 位启用的。设置 CRCEN 位时同时复位 CRC 寄存器(SPI\_RXCRCR 和 SPI\_TXCRCR)。当设置了 SPI\_CR1 的 CRCNEXT 位，SPI\_TXCRCR 的内容将在当前字节发送之后发出。

在传输 SPI\_TXCRCR 的内容时，如果在移位寄存器中收到的数值与 SPI\_RXCRCR 的内容不匹配，则 SPI\_SR 寄存器的 CRCERR 标志位被置 1。

如果在 TX 缓冲器中还有数据，CRC 的数值仅在数据字节传输结束后传送。在传输 CRC 期间，CRC 计算器关闭，寄存器的数值保持不变。

注意：请参考产品说明书，以确认有此功能(不是所有型号都有此功能)。

SPI 通信可以通过以下步骤使用 CRC：

- 设置 CPOL、CPHA、LSBFirst、BR、SSM、SSI 和 MSTR 的值；
- 在 SPI\_CRCPR 寄存器输入多项式；
- 通过设置 SPI\_CR1 寄存器 CRCEN 位使能 CRC 计算，该操作也会清除寄存器 SPI\_RXCRCR 和 SPI\_TXCRC；
- 设置 SPI\_CR1 寄存器的 SPE 位启动 SPI 功能；
- 启动通信并且维持通信，直到只剩最后一个字节或者半字；
- 在把最后一个字节或半字写进发送缓冲器时，设置 SPI\_CR1 的 CRCNext 位，指示硬件在发送完成最后一个数据之后，发送 CRC 的数值。在发送 CRC 数值期间，停止 CRC 计算；
- 当最后一个字节或半字被发送后，SPI 发送 CRC 数值，CRCNext 位被清除。同样，接收到的 CRC 与 SPI\_RXCRCR 值进行比较，如果比较不相配，则设置 SPI\_SR 上的 CRCERR 标志位，当设置了 SPI\_CR2 寄存器的 ERRIE 时，则产生中断。

注意：当 SPI 模块处于从设备模式时，请注意在时钟稳定之后再使能 CRC 计算，否则可能会得到误的 CRC 计算结果。事实上，只要设置了 CRCEN 位，只要在 SCK 引脚上有输入时钟，不管 SPE 位的状态，都会进行 CRC 的计算。

当 SPI 时钟频率较高时，用户在发送 CRC 时必须小心。在 CRC 传输期间，使用 CPU 的时间应尽可能少；为了避免在接收最后的数据和 CRC 时出错，在发送 CRC 过程中应禁止函数调用。必须在发送/接收最后一个数据之前完成设置 CRCNEXT 位的操作。

当 SPI 时钟频率较高时，因为 CPU 的操作会影响 SPI 的带宽，建议采用 DMA 模式以避免 SPI 降低



的速度。

当配置 SPI 为从模式并且使用 CRC 的功能，即使 NSS 引脚为高时仍然会执行 CRC 的计算(译注：当 NSS 信号为高时，如果 SCK 引脚上有时钟脉冲，则 CRC 计算会继续执行)。例如：当主设备交替地与多个从设备进行通信时，将会出现这种情况(译注：此时要想办法避免 CRC 的误操作)。在不选中一个从设备(NSS 信号为高)转换到选中一个新的从设备(NSS 信号为低)的时候，为了保持主从设备端下次 CRC 计算结果的同步，应该清除主从两端的 CRC 数值。

按照下述步骤清除 CRC 数值：

1. 关闭 SPI 模块(SPE=0);
2. 清除 CRCEN 位为' 0'；
3. 设置 CRCEN 位为' 1'；
4. 使能 SPI 模块(SPE=1)。

#### 24.3-7 状态标志

应用程序通过 3 个状态标志可以完全监控 SPI 总线的状态。

##### 发送缓冲器空闲标志(TXE)

此标志为' 1' 时表明发送缓冲器为空，可以写下一个待发送的数据进入缓冲器中。当写入 SPI\_DR 时，TXE 标志被清除。

##### 接收缓冲器非空(RXNE)

此标志为' 1' 时表明在接收缓冲器中包含有效的接收数据。读 SPI 数据寄存器可以清除此标志。

##### 忙(Busy)标志

BSY 标志由硬件设置与清除(写入此位无效果)，此标志表明 SPI 通信层的状态。

当它被设置为' 1' 时，表明 SPI 正忙于通信，但有一个例外：在主模式的双向接收模式下(MSTR=1、BDM=1 并且 BDOE=0)，在接收期间 BSY 标志保持为低。

在软件要关闭 SPI 模块并进入停机模式(或关闭设备时钟)之前，可以使用 BSY 标志检测传输是否结束，这样可以避免破坏最后一次传输，因此需要严格按照下述过程执行。BSY 标志还可以用于在多主系统中避免写冲突。除了主模式的双向接收模式(MSTR=1、BDM=1 并且 BDOE=0)，当传输开始时，BSY 标志被置' 1'。

以下情况时此标志将被清除为' 0'：

- 当传输结束(主模式下，如果是连续通信的情况例外)；
- 当关闭 SPI 模块；
- 当产生主模式失效(MODF=1)。

如果通信不是连续的，则在每个数据项的传输之间，BSY 标志为低。

当通信是连续时：

- 主模式下：在整个传输过程中，BSY 标志保持为高；
- 从模式下：在每个数据项的传输之间，BSY 标志在一个 SPI 时钟周期中为低。



**注:** 不要使用 BSY 标志处理每一个数据项的发送和接收, 最好使用 TXE 和 RXNE 标志。

### 24.3-8 关闭 SPI

当通讯结束, 可以通过关闭 SPI 模块来终止通讯。清除 SPE 位即可关闭 SPI。

在某些配置下, 如果再传输还未完成时, 就关闭 SPI 模块并进入停机模式, 则可能导致当前的传输被破坏, 而且 BSY 标志也变得不可信。

为了避免发生这种情况, 关闭 SPI 模块时, 建议按照下述步骤操作:

**在主或从模式下的全双工模式(BIDIMODE=0, RXONLY=0)**

1. 等待 RXNE=1 并接收最后一个数据;
2. 等待 TXE=1;
3. 等待 BSY=0;
4. 关闭 SPI(SPE=0), 最后进入停机模式(或关闭该模块的时钟)。

**在主或从模式下的单向只发送模式(BIDIMODE=0, RXONLY=0)或双向的发送模式(BIDIMODE=1, BIDIOE=1)**

在 SPI\_DR 寄存器中写入最后一个数据后:

1. 等待 TXE=1;
2. 等待 BSY=0;
3. 关闭 SPI(SPE=0), 最后进入停机模式(或关闭该模块的时钟)。

**在主或从模式下的单向只接收模式(MSTR=1, BIDIMODE=0, RXONLY=1)或双向的接收模式(MSTR=1, BIDIMODE=1, BIDIOE=0)**

这种情况需要特别地处理, 以保证 SPI 不会开始一次新的传输:

1. 等待倒数第二个(第 n-1 个)RXNE=1;
2. 在关闭 SPI(SPE=0)之前等待一个 SPI 时钟周期(使用软件延迟);
3. 在进入停机模式(或关闭该模块的时钟)之前等待最后一个 RXNE=1。

**注:** 在主模式下的单向只发送模式(MSTR=1, BDM=1, BDOE=0)时, 传输过程中 BSY 标志始终为低。

**在从模式下的只接收模式(MSTR=0, BIDIMODE=0, RXONLY=1)或双向的接收模式(MSTR=0, BIDIMODE=1, BIDIOE=0)**

1. 可以在任何时候关闭 SPI(SPE=0), SPI 会在当前的传输结束后被关闭;
2. 如果希望进入停机模式, 在进入停机模式(或关闭该模块的时钟)之前必须首先等待 BSY=0。



### 24.3-9 使用 DMA 的 SPI 通信

为了达到最大通信速度，需要及时往 SPI 发送缓冲器填数据，同样接收缓冲器中的数据也必须及时读走以防止溢出。为了方便高速率的数据传输，SPI 实现了一种采用简单的请求/应答的 DMA 机制。

当 SPI\_CR2 寄存器上的对应使能位被设置时，SPI 模块可以发出 DMA 传输请求。发送缓冲器和接收缓冲器亦有各自的 DMA 请求(见 25-11)。

- 发送时，在每次 TXE 被设置为' 1' 时发出 DMA 请求，DMA 控制器则写数据至 SPI\_DR 寄存器，TXE 标志因此而被清除。
- 接收时，在每次 RXNE 被设置为' 1' 时发出 DMA 请求，DMA 控制器则从 SPI\_DR 寄存器读出数据，RXNE 标志因此而被清除。

当只使用 SPI 发送数据时，只需使能 SPI 的发送 DMA 通道。此时，因为没有读取收到的数据，OVR 被置为' 1' (译注：软件不必理会这个标志)。

当只使用 SPI 接收数据时，只需使能 SPI 的接收 DMA 通道。在发送模式下，当 DMA 已经传输了所有要发送的数据(DMA\_ISR 寄存器的 TCIF 标志变为' 1' )后，可以通过监视 BSY 标志以确认 SPI 通信结束，这样可以避免在关闭 SPI 或进入停止模式时，破坏最后一个数据的传输。因此软件需要先等待 TXE=1，然后等待 BSY=0。

注： 在不连续的通信中，在写数据到 SPI\_DR 的操作与 BSY 位被置为' 1' 之间，有 2 个 APB 时钟周期的延迟，因此，在写完最后一个数据后需要先等待 TXE=1 再等待 BSY=0。

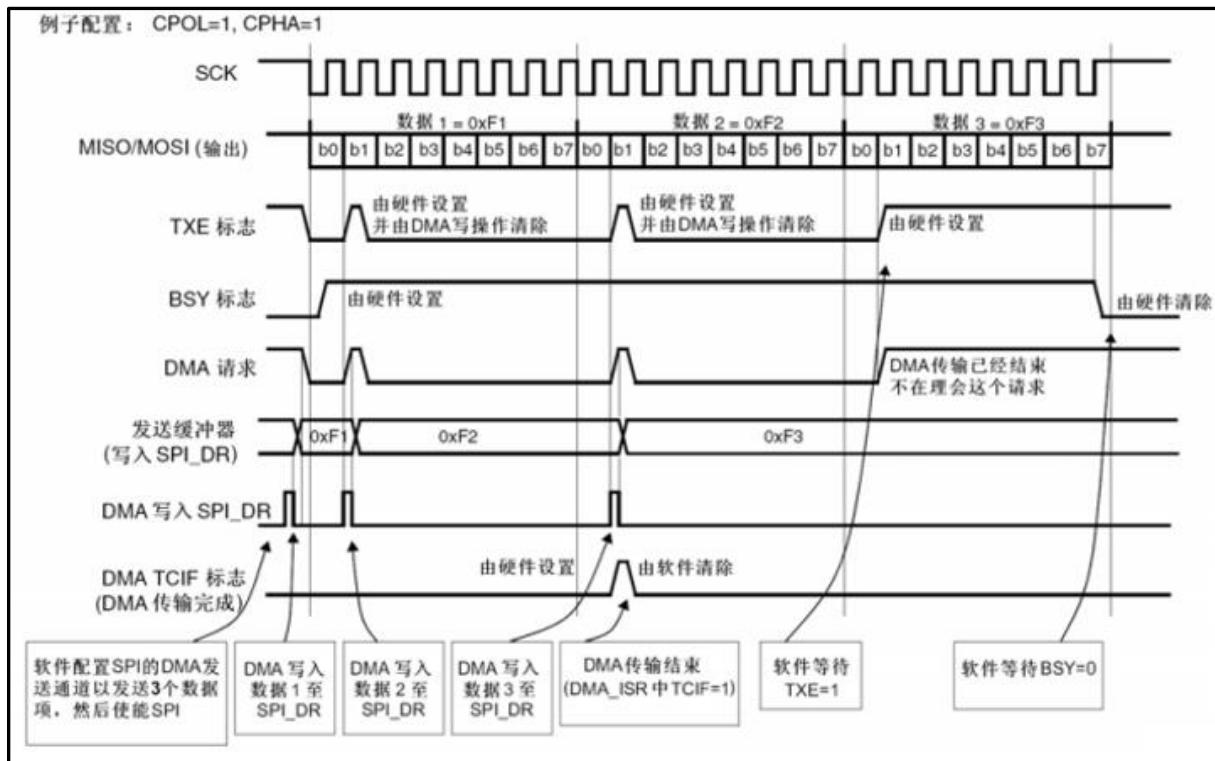


Figure 24-11 使用 DMA 发送

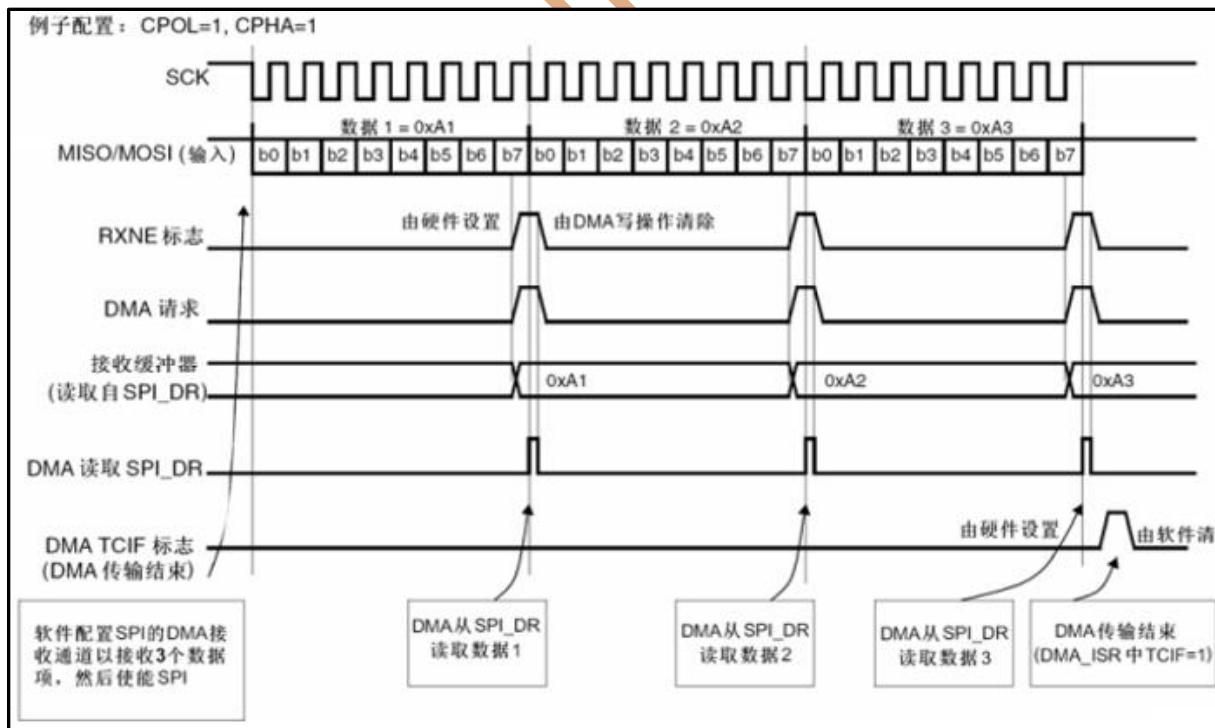


Figure 24-12 使用 DMA 接收



**带 CRC 的 DMA 功能**

当使能 SPI 使用 CRC 检验并且启用 DMA 模式时，在通信结束时，CRC 字节的发送和接收是自动完成的。

数据和 CRC 传输结束时，SPI\_SR 寄存器的 CRCERR 标志为‘1’表示在传输期间发生错误。

ZBT CONFIDENTIAL



### 24.3.10 错误标志

#### 主模式失效错误(MODF)

主模式失效仅发生在： NSS 引脚硬件模式管理下，主设备的 NSS 脚被拉低；或者在 NSS 引脚软件模式管理下， SSI 位被置为‘ 0 ’时； MODF 位被自动置位。主模式失效对 SPI 设备有以下影响：

- MODF 位被置为‘ 1 ’，如果设置了 ERRIE 位，则产生 SPI 中断；
- SPE 位被清为‘ 0 ’。这将停止一切输出，并且关闭 SPI 接口；
- MSTR 位被清为‘ 0 ’，因此强迫此设备进入从模式。

下面的步骤用于清除 MODF 位：

1. 当 MODF 位被置为‘ 1 ’时，执行一次对 SPI\_SR 寄存器的读或写操作；
2. 然后写 SPI\_CR1 寄存器。

在有多个 MCU 的系统中，为了避免出现多个从设备的冲突，必须先拉高该主设备的 NSS 脚，再对 MODF 位进行清零。在完成清零之后， SPE 和 MSTR 位可以恢复到它们的原始状态。出于安全的考虑，当 MODF 位为‘ 1 ’时，硬件不允许设置 SPE 和 MSTR 位。通常配置下，从设备的 MODF 位不能被置为‘ 1 ’。然而，在多主配置里，一个设备可以在设置了 MODF 位的情况下，处于从设备模式；此时， MODF 位表示可能出现了多主冲突。中断程序可以执行一个复位或返回到默认状态来从错误状态中恢复。

#### 溢出错误

当主设备已经发送了数据字节，而从设备还没有清除前一个数据字节产生的 RXNE 时，即为溢出错误。当产生溢出错误时：

- OVR 位被置为‘ 1 ’；当设置了 ERRIE 位时，则产生中断。

此时，接收器缓冲器的数据不是主设备发送的新数据，读 SPI\_DR 寄存器返回的是之前未读的数据，所有随后传送的数据都被丢弃。

依次读出 SPI\_DR 寄存器和 SPI\_SR 寄存器可将 OVR 清除。

#### CRC 错误

当设置了 SPI\_CR 寄存器上的 CRCEN 位时， CRC 错误标志用来核对接收数据的有效性。如果移位寄存器中接收到的值(发送方发送的 SPI\_TXCRCR 数值)与接收方 SPI\_RXCRCR 寄存器中的数值不匹配，则 SPI\_SR 寄存器上的 CRCERR 标志被置位为‘ 1 ’。



### 24.3-11 SPI 中断

中断事件	事件标志	使能控制位
发送缓冲器空标志	TXE	TXEIE
接收缓冲器非空标志	RXNE	RXNEIE
主模式失效事件	MODF	
溢出错误	OVR	ERRIE
CRC错误标志	CRCERR	

Table 24-1 SPI 中断请求

## 24.4 I2S 功能描述

### 24.4-1 I2S 功能描述

I2S 的框图如下图所示:

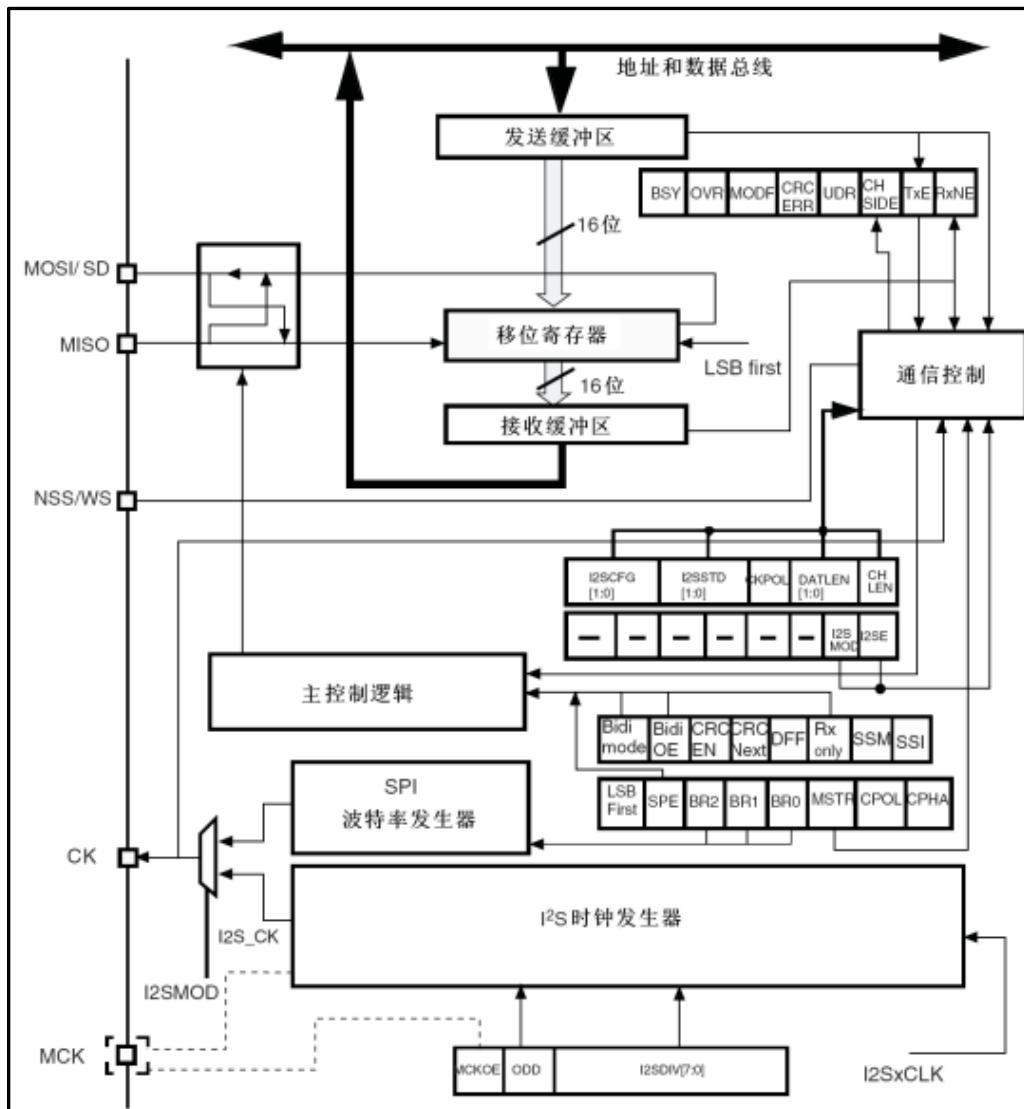


Figure 24-13 I2S 框图

通过将寄存器 SPI\_I2SCFGR 的 I2SMOD 位置为‘1’，即可使能 I2S 功能。此时，可以把 SPI 模块用作 I2S 音频接口。I2S 接口与 SPI 接口使用大致相同的引脚、标志和中断。

I2S 与 SPI 共享 3 个引脚：

- SD：串行数据(映射至 MOSI 引脚)，用来发送和接收 2 路时分复用通道的数据；
- WS：字选(映像至 NSS 引脚)，主模式下作为数据控制信号输出，从模式下作为输入；
- CK：串行时钟(映射至 SCK 引脚)，主模式下作为时钟信号输出，从模式下作为输入。

在某些外部音频设备需要主时钟时，可以另有一个附加引脚输出时钟：

- MCK：主时钟(独立映射)，在 I2S 配置为主模式，寄存器 SPI\_I2SPR 的 MCKOE 位为‘1’时，作为输出额外的时钟信号引脚使用。输出时钟信号的频率预先设置为  $256 \times F_s$ ，其中  $F_s$  是音



频信号的采样频率。

设置成主模式时，I2S 使用自身的时钟发生器来产生通信用的时钟信号。这个时钟发生器也是主时钟输出的时钟源。I2S 模式下有 2 个额外的寄存器，一个是与时钟发生器配置相关的寄存器 SPI\_I2SPR，另一个是 I2S 通用配置寄存器 SPI\_I2SCFGR(可设置音频标准、从/主模式、数据格式、数据包帧、时钟极性等参数)。

在 I2S 模式下不使用寄存器 SPI\_CR1 和所有的 CRC 寄存器。同样，I2S 模式下也不使用寄存器 SPI\_CR2 的 SSOE 位，和寄存器 SPI\_SR 的 MODF 位和 CRCERR 位。

I2S 使用与 SPI 相同的寄存器 SPI\_DR 用作 16 位宽模式数据传输。

#### 24.4.2 支持的音频协议

三线总线支持 2 个声道上音频数据的时分复用：左声道和右声道，但是只有一个 16 位寄存器用作发送或接收。因此，软件必须在对数据寄存器写入数据时，根据当前传输中的声道写入相应数据；同样，在读取寄存器数据时，通过检查寄存器 SPI\_SR 的 CHSIDE 位来判明接收到的数据属于哪个声道。左声道总是先于右声道发送数据(CHSIDE 位在 PCM 协议下无意义)。

有四种可用的数据和包帧组合。可以通过以下四种数据格式发送数据：

- 16 位数据打包进 16 位帧
- 16 位数据打包进 32 位帧
- 24 位数据打包进 32 位帧
- 32 位数据打包进 32 位帧

在使用 16 位数据扩展到 32 位帧时，前 16 位(MSB)是有意义的数据，后 16 位(LSB)被强制为 0，该操作不需要软件干预，也不需要有 DMA 请求(仅需要一次读/写操作)。24 位和 32 位数据帧需要 CPU 对寄存器 SPI\_DR 进行 2 次读或写操作，在使用 DMA 时，需要 2 次 DMA 传输。对于 24 位数据，扩展到 32 位后，最低 8 位由硬件置 0。对于所有的数据格式和通讯标准，总是先发送最高位(MSB)。

I2S 接口支持四种音频标准，可以通过设置寄存器 SPI\_I2SCFGR 的 I2SSTD[1:0]位和 PCMSYNC 位来选择。

#### I2S 飞利浦标准

在此标准下，引脚 WS 用来指示正在发送的数据属于哪个声道。在发送第一位数据(MSB)前 1 个时钟周期，该引脚即为有效。

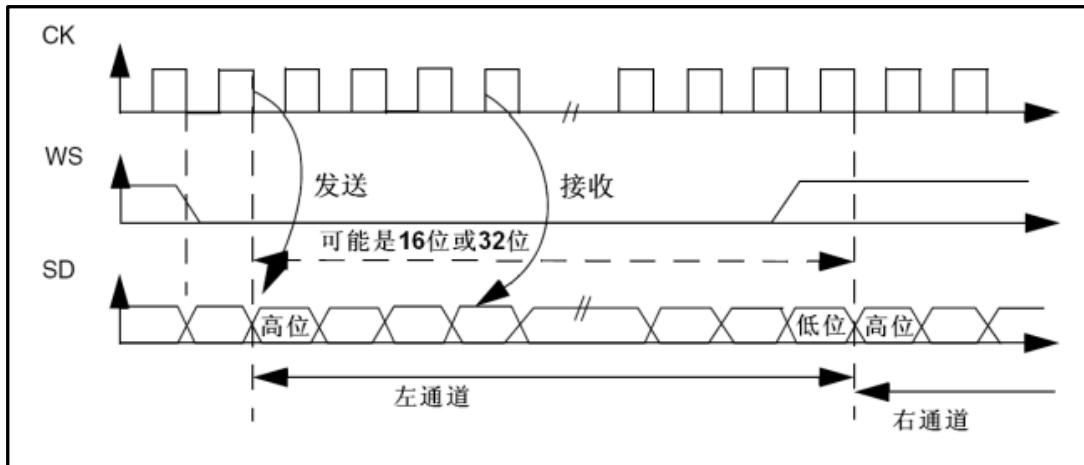


Figure 24-14 I2S 飞利浦协议波形(16/32 位全精度, CPOL = 0)

发送方在时钟信号(CK)的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在时钟信号的下降沿变化。

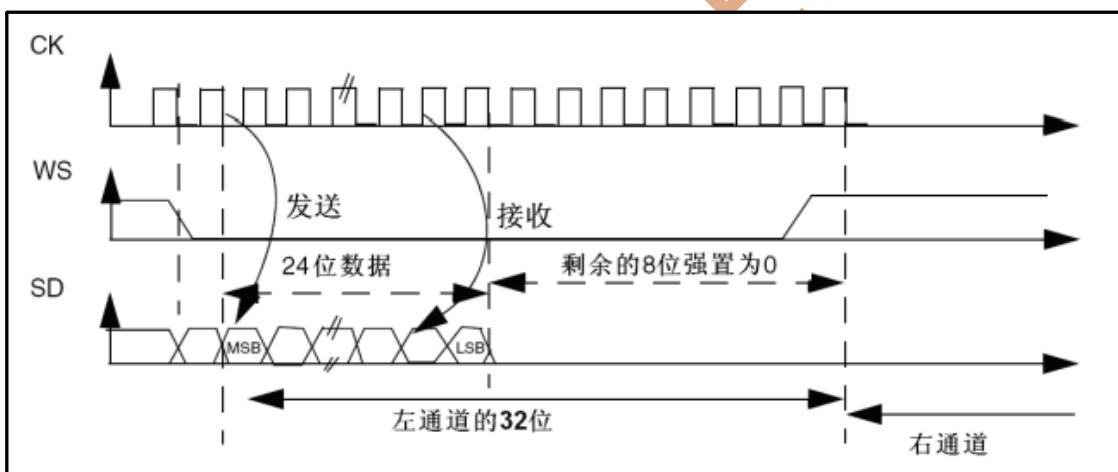


Figure 24-15 I2S 飞利浦协议标准波形(24 位帧, CPOL = 0)

此模式需要对寄存器 SPI\_DR 进行 2 次读或写操作。

- 在发送模式下：如果需要发送 0x8EAA33(24 位)：

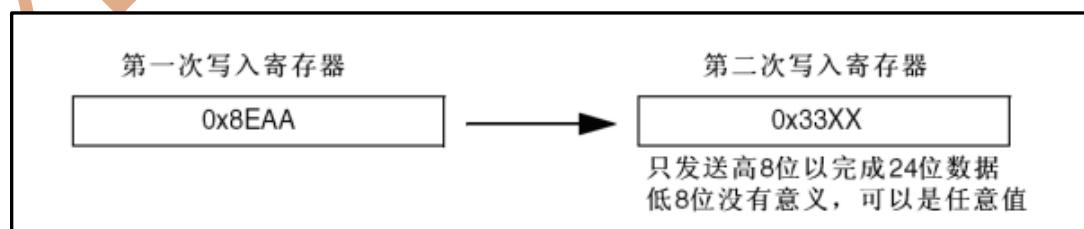


Figure 24-16 发送 0x8EAA33



- 在接收模式下：如果接收 0x8EAA33：

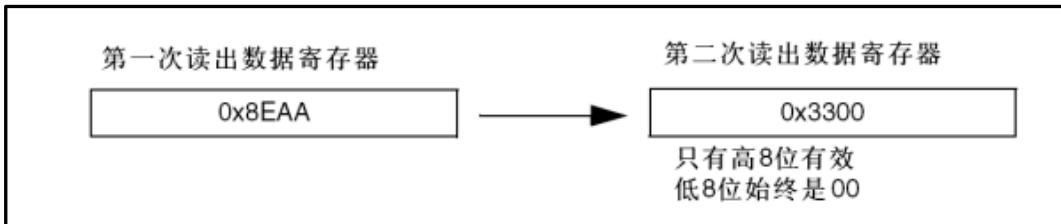


Figure 24-17 接收 0x8EAA33

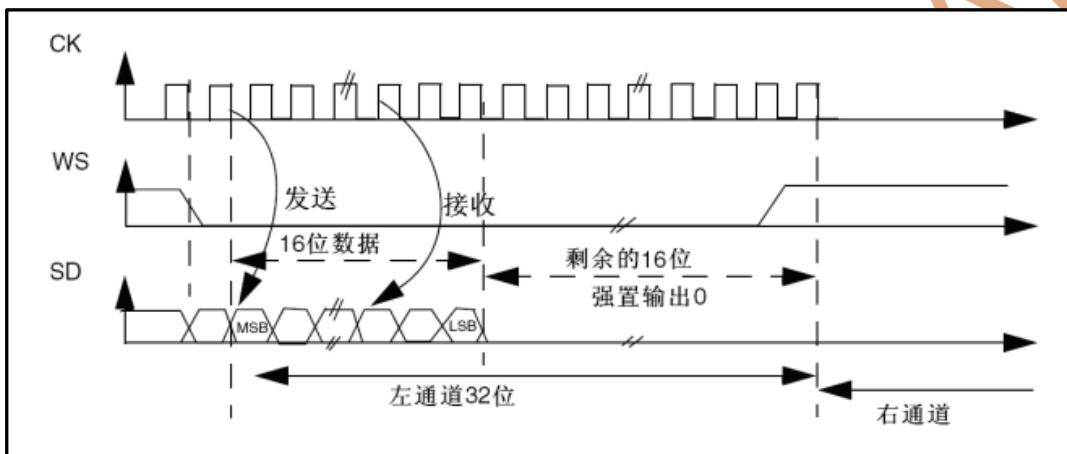


Figure 24-18 I2S 飞利浦协议标准波形(16 位扩展至 32 位声道帧, CPOL = 0)

在 I2S 配置阶段，如果选择将 16 位数据扩展到 32 位声道帧，只需要访问一次寄存器 SPI\_DR。用来扩展到 32 位的低 16 位被硬件置为 0x0000。

如果待传输或者接收的数据是 0x76A3(扩展到 32 位是 0x76A30000)，需要的操作如下图所示。

只需要操作一次 SPI\_DR

0X76A3

在发送时需要将 MSB 写入寄存器 SPI\_DR；标志位 TXE 为‘1’表示可以写入新的数据，如果允许了相应的中断，则可以产生中断。发送是由硬件完成的，即使还未发送出后 16 位的 0x0000，也会设置 TXE 并产生相应的中断。接收时，每次收到高 16 位半字(MSB)后，标志位 RXNE 置‘1’，如果允许了相应的中断，则可以产生中断。这样，在 2 次读和写之间有更多的时间，可以防止下溢或者上溢的情况发生。



### MSB 对齐标准

在此标准下，WS 信号和第一个数据位，即最高位(MSB)同时产生。

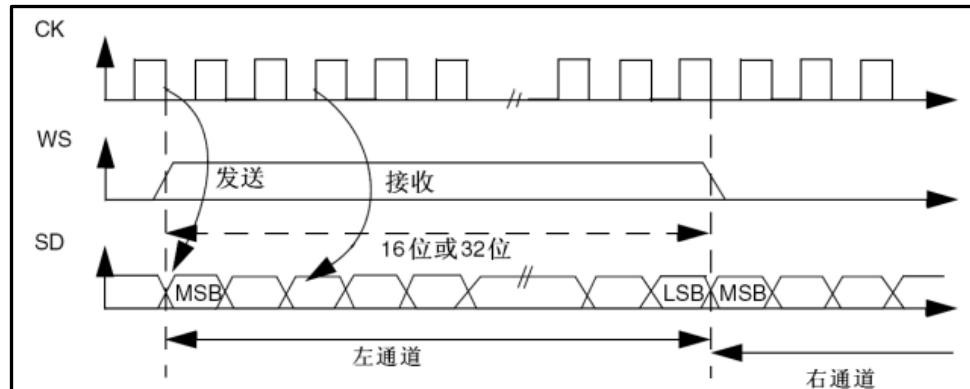


Figure 24-19 MSB 对齐 16 位或 32 位全精度, CPOL = 0

发送方在时钟信号的下降沿改变数据；接收方是在上升沿读取数据。

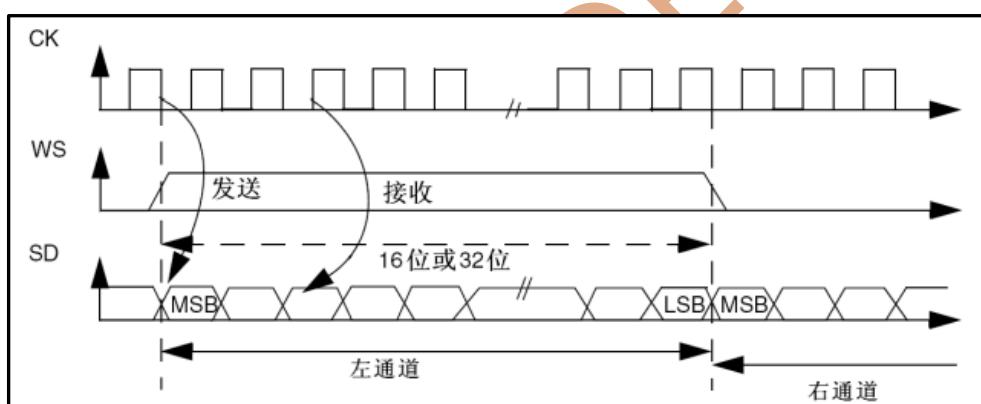


Figure 24-20 MSB 对齐 24 位数据, CPOL = 0

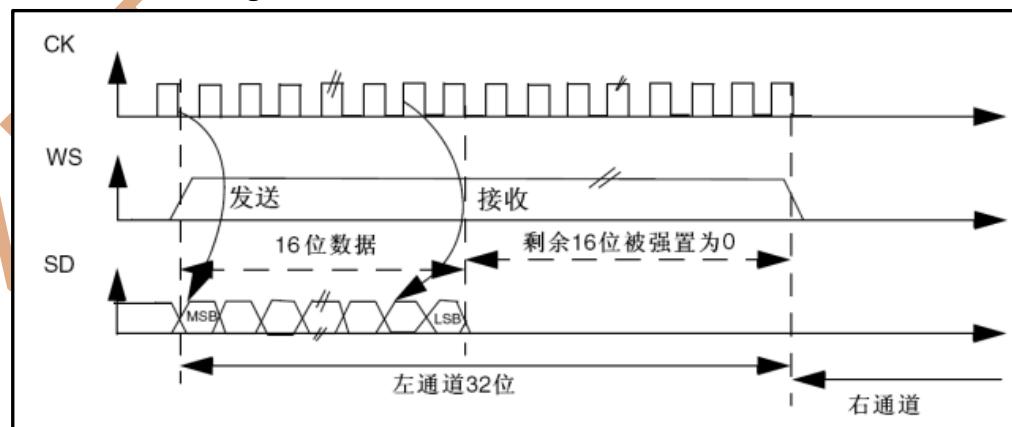


Figure 24-21 MSB 对齐 16 位数据扩展到 32 位包帧, CPOL = 0



### LSB 对齐标准

此标准与 MSB 对齐标准类似(在 16 位或 32 位全精度帧格式下无区别)。

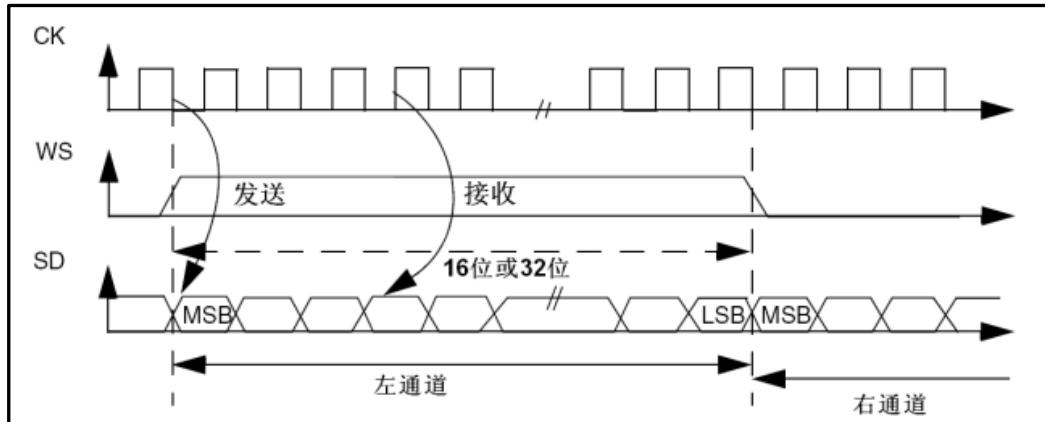


Figure 24-22 LSB 对齐 16 位或 32 位全精度, CPOL = 0

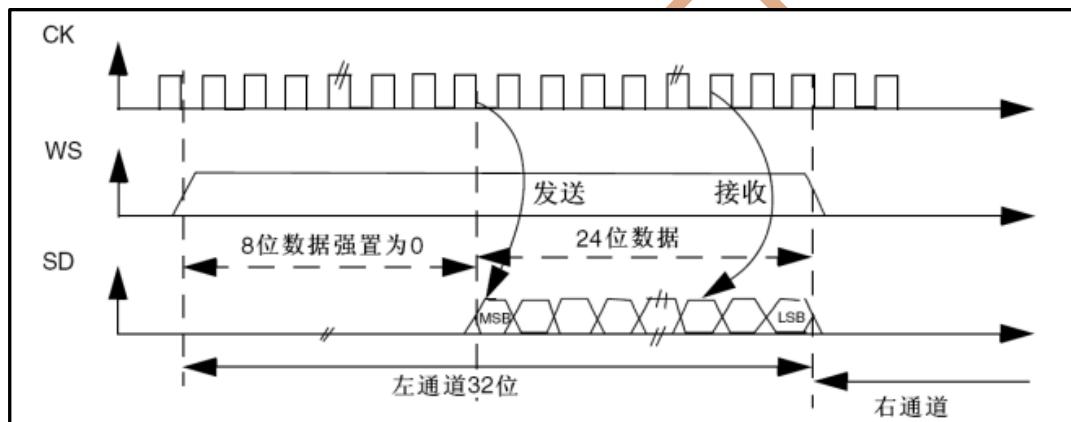


Figure 24-23 LSB 对齐 24 位数据, CPOL = 0

- 在发送模式下

如果要发送数据 0x3478AE，需要通过软件或者 DMA 对寄存器 SPI\_DR 进行 2 次写操作。操作流程如下图所示。

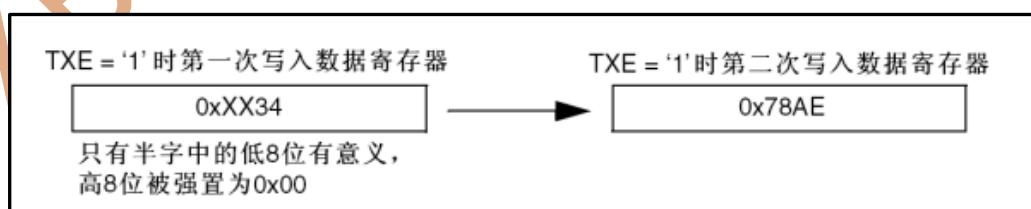


Figure 24-24 要求发送 0x3478AE 的操作



- 在接收模式下

如果要接收数据 0x3478AE，需要在 2 个连续的 RXNE 事件发生时，分别对寄存器 SPI\_DR 进行 1 次读操作。

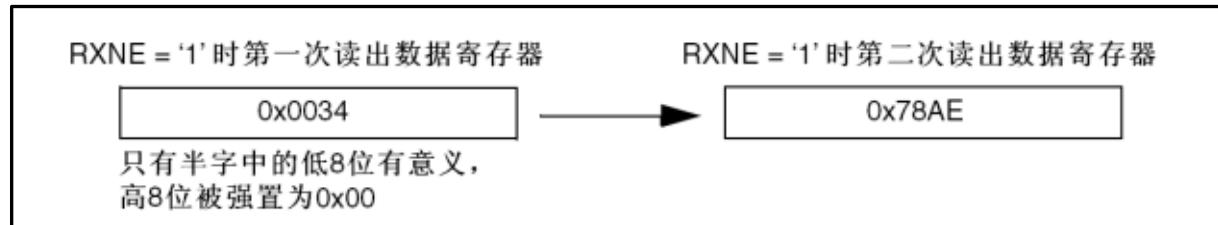


Figure 24-25 要求接收 0x3478AE 的操作

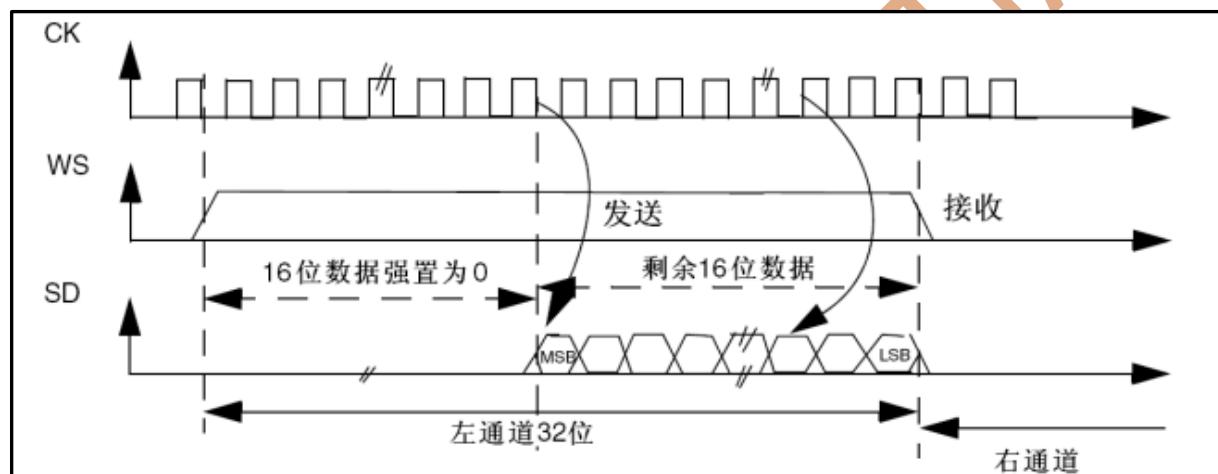


Figure 24-26 LSB 对齐 16 位数据扩展到 32 位包帧，CPOL =0

在 I2S 配置阶段，如果选择将 16 位数据扩展到 32 声道帧，只需要访问一次寄存器 SPI\_DR。此时，扩展到 32 位后的高半字(16 位 MSB)被硬件置为 0x0000。

如果待传输或者接收的数据是 0x76A3(扩展到 32 位是 0x0000 76A3)，需要的操作如下图所示。

只需操作一次 SPI\_DR

0X76A3

在发送时，如果 TXE 为‘1’，用户需要写入待发送的数据(即 0x76A3)。用来扩展到 32 位的 0x0000 部分由硬件首先发送出去，一旦有效数据开始从 SD 引脚送出，即发生下一次 TXE 事件。在接收时，一旦接收到有效数据(而不是 0x0000 部分)，即发生 RXNE 事件。这样，在 2 次读和写之间有更多的时间，可以防止下溢或者上溢的情况发生。



### PCM 标准

在 PCM 标准下，不存在声道选择的信息。PCM 标准有 2 种可用的帧结构，短帧或者长帧，可以通过设置寄存器 SPI\_I2SCFGR 的 PCMSYNC 位来选择。

对于长帧，主模式下，用来同步的 WS 信号有效的时间固定为 13 位。

对于短帧，用来同步的 WS 信号长度只有 1 位。

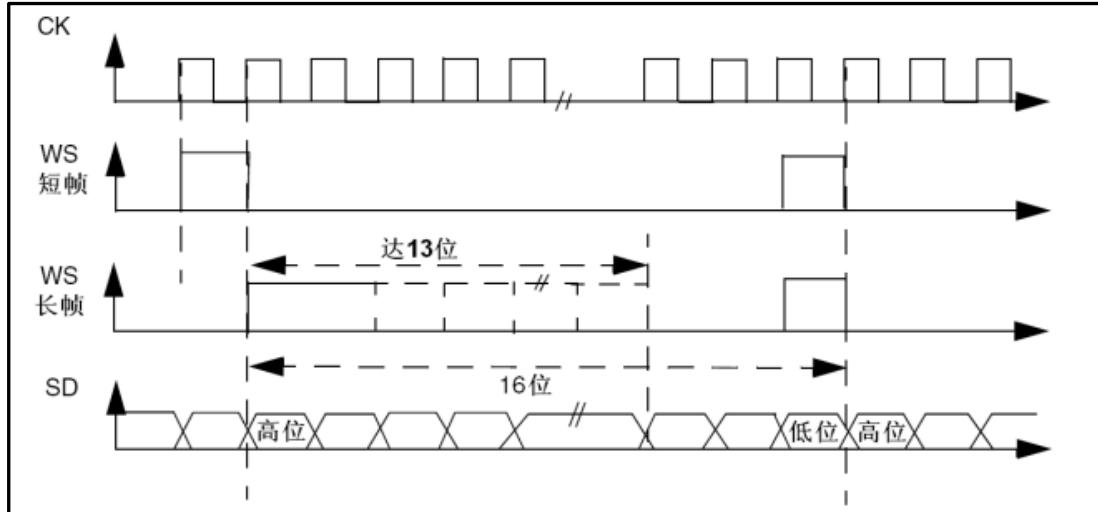


Figure 24-27 PCM 标准波形(16 位)

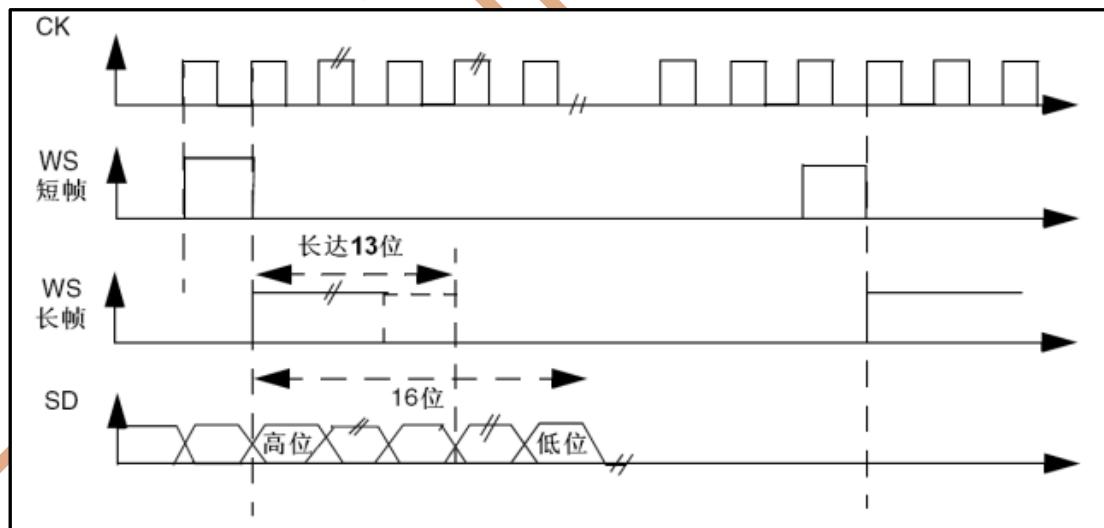


Figure 24-28 PCM 标准波形(16 位扩展到 32 位包帧)

注意：无论哪种模式(主或从)、哪种同步方式(短帧或长帧)，连续的 2 帧数据之间和 2 个同步信号间的时间差，(即使是从模式)需要通过设置 SPI\_I2SCFGR 寄存器的 DATLEN 位和 CHLEN 位来确定。



#### 24.4-3 时钟发生器

I2S 的比特率即确定了在 I2S 数据线上的数据流和 I2S 的时钟信号频率。

I2S 比特率 = 每个声道的比特数 × 声道数目 × 音频采样频率

对于一个具有左右声道和 16 位音频信号, I2S 比特率计算如下

I2S 比特率 =  $16 \times 2 \times F_s$

如果包长为 32 位, 则有: I2S 比特率 =  $32 \times 2 \times F_s$

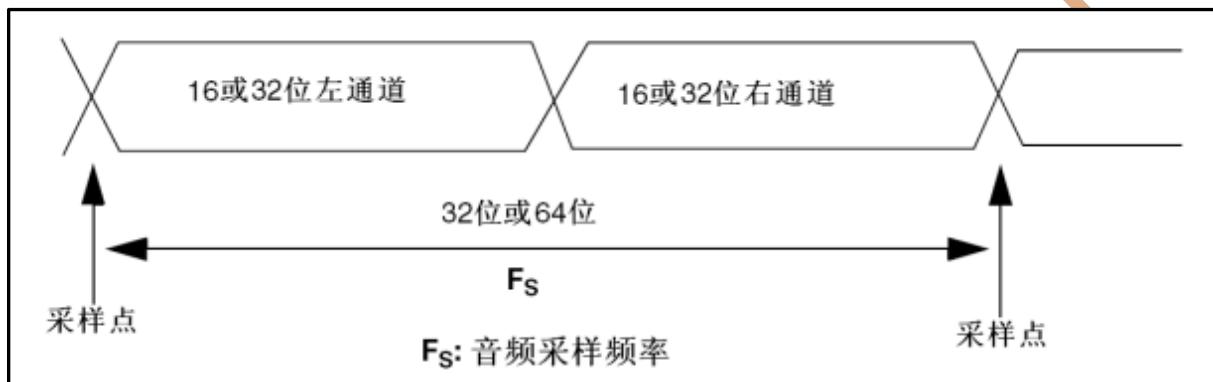


Figure 24-29 音频采样频率定义

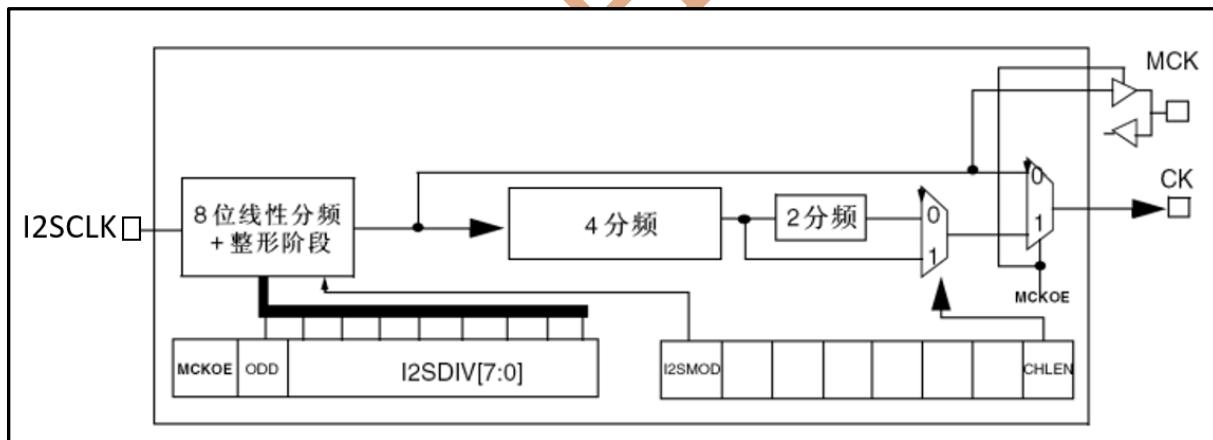


Figure 24-30 I2S 时钟发生器结构

上图中 I2SxCLK 的时钟源是系统时钟(即驱动 AHB 时钟的 HSI、HXT 或 PLL)。对于互联型产品，

I2SxCLK 可以来自 SYSCLK 时钟从而得到最精确的时钟(可使用 PLL 微调频率 Error~0)，可以通过 RCC\_CFGR2 寄存器的 I2S2SRC 和 I2S3SRC 位选择。

音频的采样频率可以是 96kHz、48kHz、44.1kHz、32kHz、22.05kHz、16kHz、11.025kHz 或者 8kHz(或任何此范围内的数值)。为了获得需要的频率, 需按照以下公式设置线性分频器:

当需要生成主时钟时(寄存器 SPI\_I2SPR 的 MCKOE 位为' 1' ):

声道的帧长为 16 位时,  $F_s = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)*8]$



声道的帧长为 32 位时,  $F_s = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)*4]$

当关闭主时钟时(MCKOE 位为' 0' ):

声道的帧长为 16 位时,  $F_s = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)]$

声道的帧长为 32 位时,  $F_s = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)]$

下面 2 张表给出了不同时钟配置时, 精确参数的例子。

注: 可以使用其它配置以达到优化时钟精确度的目的。

SYSCLK (MHz)	I2S_DIV		I2S_ODD		MCLK	Target fs (Hz)	Real f <sub>s</sub> (KHz)		Error	
	16-bit	32-bit	16-bit	32-bit			16-bit	32-bit	16-bit	32-bit
64	31	15	0	1	No	32000	32258.06	32258.06	0.81%	0.81%
64	45	22	0	1	No	22050	22222.22	22222.22	0.78%	0.78%
64	62	31	1	0	No	16000	16000.00	16129.03	0.00%	0.81%
64	91	45	0	0	No	11025	10989.01	11111.11	- 0.33%	0.78%
64	124	62	1	1	No	8000	8032.13	8000	0.40%	0.00%
64	2	2	1	0	yes	96000	100000.00	62500	4.17%	34.90%
64	5	2	0	1	yes	48000	50000.00	50000	4.17%	4.17%
64	8	4	0	0	yes	32000	31250.00	31250	- 2.34%	-2.34%
64	11	5	1	1	yes	22050	21739.13	22727.27273	- 1.41%	3.07%
64	16	8	0	0	yes	16000	15625.00	15625	- 2.34%	-2.34%
64	23	11	0	1	yes	11025	10869.57	10869.56522	- 1.41%	-1.41%
64	31	15	1	1	yes	8000	7936.51	8064.516129	- 0.79%	0.81%

Table 24-2 使用标准的 8MHz HXT 时钟得到精确的音频频率



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBT CONFIDENTIAL



#### 24.4-4 I2S 主模式

设置 I2S 工作在主模式，串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。可以通过设置寄存器 SPI\_I2SPR 的 MCKOE 位来选择输出或者不输出主时钟(MCK)。

##### 流程

1. 设置寄存器 SPI\_I2SPR 的 I2SDIV[7:0]定义与音频采样频率相符的串行时钟波特率。同时也要定义寄存器 SPI\_I2SPR 的 ODD 位。
2. 设置 CKPOL 位定义通信用时钟在空闲时的电平状态。如果需要向外部的 DAC/ADC 音频器件提供主时钟 MCK，将寄存器 SPI\_I2SPR 的 MCKOE 位置为‘1’。(按照不同的 MCK 输出状态，计算 I2SDIV 和 ODD 的值，详见 25.4.3 节)。
3. 设置寄存器 SPI\_I2SCFGR 的 I2SMOD 位为‘1’激活 I2S 功能，设置 I2SSTD[1:0]和 PCMSYNC 位选择所用的 I2S 标准，设置 CHLEN 选择每个声道的数据位数。还要设置寄存器 SPI\_I2SCFGR 的 I2SCFG[1:0]选择 I2S 主模式和方向(发送端还是接收端)。
4. 如果需要，可以通过设置寄存器 SPI\_CR2 来打开所需的中断功能和 DMA 功能。
5. 必须将寄存器 SPI\_I2SCFGR 的 I2SE 位置为‘1’。
6. 引脚 WS 和 CK 需要配置为输出模式。如果寄存器 SPI\_I2SPR 的 MCKOE 位为‘1’，引脚 MCK 也要配置成输出模式。

##### 发送流程

当写入 1 个半字(16 位)的数据至发送缓存，发送流程开始。

假设第一个写入发送缓存的数据对应的是左声道数据。当数据从发送缓存移到移位寄存器时，标志位 TXE 置‘1’，这时，要把对应右声道的数据写入发送缓存。标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。标志位 CHSIDE 的值在 TXE 为‘1’ 时更新，因此它在 TXE 为‘1’ 时有意义。在先左声道后右声道的数据都传输完成后，才能被认为是一个完整的数据帧。不可以只传输部分数据帧，如仅有左声道的数据。

当发出第一位数据的同时，半字数据被并行地传送至 16 位移位寄存器，然后后面的位依次按高位在先的顺序从引脚 MOSI/SD 发出。每次数据从发送缓存移至移位寄存器时，标志位 TXE 置为‘1’，如果寄存器 SPI\_CR2 的 TXIE 位为‘1’，则产生中断。

写入数据的操作取决于所选择的 I2S 标准，详见 23.4.2 节。

为了保证连续的音频数据传输，建议在当前传输完成之前，对寄存器 SPI\_DR 写入下一个要传输的数据。

建议在要关闭 I2S 功能时，等待标志位 TXE=1 及 BSY=0，再将 I2SE 位清‘0’。



## 接收流程

接收流程的配置步骤除了第 3 点外，与发送流程的一致(参见前述的“发送流程” )，需要通过配置 I2SCFG[1:0]来选择主接收模式。

无论何种数据和声道长度，音频数据总是以 16 位包的形式接收。即每次填满接收缓存后，标志位 RXNE 置‘ 1’，如果寄存器 SPI\_CR2 的 RXNEIE 位为‘ 1’，则产生中断。根据配置的数据和声道长

度，收到左声道或右声道的数据会需要 1 次或者 2 次把数据传送到接收缓存的过程。

对寄存器 SPI\_DR 进行读操作即可清除 RXNE 标志位。

每次接收以后即更新 CHSIDE。它的值取决于 I2S 单元产生的 WS 信号。

读取数据的操作取决于所选择的 I2S 标准，详见 23.4.2 节。

如果前一个接收到的数据还没有被读取，又接收到新数据，即发生上溢，标志位 OVR 被置

为‘ 1’，如果寄存器 SPI\_CR2 的 ERRIE 位为‘ 1’，则产生中断，表示发生了错误。

若要关闭 I2S 功能，需要执行特别的操作，以保证 I2S 模块可以正常地完成传输周期而不会开始新的数据传输。操作过程与数据配置和信道长度、以及音频协议的模式相关：

- 16 位数据扩展到 32 位通道长度(DATLEN=00 并且 CHLEN=1)，使用 LSB(低位)对齐模式(I2SSTD=10)
  - a) 等待倒数第二个(n-1)RXNE=1;
  - b) 等待 17 个 I2S 时钟周期(使用软件延迟);
  - c) 关闭 I2S(I2SE=0)。
- 16 位数据扩展到 32 位通道长度(DATLEN=00 并且 CHLEN=1)，使用 MSB(高位)对齐、I2S 或 PCM 模式(分别为 I2SSTD=00, I2SSTD=01 或 I2SSTD=11)
  - a) 等待最后一个 RXNE=1;
  - b) 等待 1 个 I2S 时钟周期(使用软件延迟);
  - c) 关闭 I2S(I2SE=0)。
- 所有其它 DATLEN 和 CHLEN 的组合，I2SSTD 选择的任意音频模式，使用下述方式关闭 I2S:
  - a) 等待倒数第二个(n-1)RXNE=1;
  - b) 等待一个 I2S 时钟周期(使用软件延迟);
  - c) 关闭 I2S(I2SE=0)。

注： 在传输期间 BSY 标志始终为低。



#### 24.4-5 I2S 从模式

在从模式下，I2S 可以设置成发送和接收模式。从模式的配置方式基本遵循和配置主模式一样的流程。在从模式下，不需要 I2S 接口提供时钟。时钟信号和 WS 信号都由外部主 I2S 设备提供，连接到相应的引脚上。因此用户无需配置时钟。

配置步骤列举如下：

1. 设置寄存器 SPI\_I2SCFGR 的 I2SMOD 位激活 I2S 功能；设置 I2SSTD[1:0]来选择所用的 I2S 标准；设置 DATLEN[1:0]选择数据的比特数；设置 CHLEN 选择每个声道的数据位数。设置寄存器 SPI\_I2SCFGR 的 I2SCFG[1:0]选择 I2S 从模式的数据方向(发送端还是接收端)。
2. 根据需要，设置寄存器 SPI\_CR2 打开所需的中断功能和 DMA 功能。
3. 必须设置寄存器 SPI\_I2SCFGR 的 I2SE 位为' 1' 。

##### 发送流程

当外部主设备发送时钟信号，并且当 NSS\_WS 信号请求传输数据时，发送流程开始。必须先使能从设备，并且写入 I2S 数据寄存器之后，外部主设备才能开始通信。

对于 I2S 的 MSB 对齐和 LSB 对齐模式，第一个写入数据寄存器的数据项对应左声道的数据。当开始通信时，数据从发送缓冲器传送到移位寄存器，然后标志位 TXE 置为' 1' ；这时，要把对应右声道的数据项写入 I2S 数据寄存器。

标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。与主模式的发送流程相比，在从模式中，CHSIDE 取决于来自外部主 I2S 的 WS 信号。这意味着从 I2S 在接收到主端生成的时钟信号之前，就要准备好第一个要发送的数据。WS 信号为' 1' 表示先发送左声道。

**注意：** 设置 I2SE 位为' 1' 的时间，应当比 CK 引脚上的主 I2S 时钟信号早至少 2 个 PCLK 时钟周期。

当发出第一位数据的时候，半字数据并行地通过 I2S 内部总线传输至 16 位移位寄存器，然后其它位依次按高位在先的顺序从引脚 MOSI/SD 发出。每次数据从发送缓冲器传送至移位寄存器时，标志位 TXE 置' 1' ，如果寄存器 SPI\_CR2 的 TXIE 位为' 1' ，则产生中断。

**注意：** 在对发送缓冲器写入数据前，要确认标志位 TXE 为' 1' 。写入数据的操作取决于所选中的 I2S 标准，详见 23.4.2 节。

为了保证连续的音频数据传输，建议在当前传输完成之前，对寄存器 SPI\_DR 写入下一个要传输的数据。如果在代表下一个数据传输的第一个时钟边沿到达之前，新的数据仍然没有写入寄存器 SPI\_DR，下溢标志位会置' 1' ，并可能产生中断；它指示软件发送数据错误。如果寄存器 SPI\_CR2 的 ERRIE 位为' 1' ，在寄存器 SPI\_SR 的标志位 UDR 为高是，就会产生中断。建议在这时关闭 I2S，然后重新从左声道开始发送数据。

建议在清除 I2SE 位关闭 I2S 之前，先等待 TXE=1 并且 BSY=0。



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBT CONFIDENTIAL



### 接收流程

配置步骤除了第 1 点外，与发送流程一致。需要通过配置 I2SCFG[1:0]来选择主接收模式。

无论何种数据和声道长度，音频数据总是以 16 位包的形式接收，即每次填满接收缓存，标志位

RXNE 置‘ 1’，如果寄存器 SPI\_CR2 的 RXNEIE 位为‘ 1’，则产生中断。按照不同的数据和声道长

度

设置，收到左声道或者右声道数据会需要 1 次或者 2 次传输数据至接收缓冲器的过程。

每次接收到数据(将要从 SPI\_DR 读出)以后即更新 CHSIDE，它对应 I2S 单元产生的 WS 信号。

读取 SPI\_DR 寄存器，将清除 RXNE 位。

读取数据的操作取决于所选中的 I2S 标准，详见 25.4.2 节。

在还没有读出前一个接收到的数据，又接收到新数据时，即产生上溢，并设置标志位 OVR

为‘ 1’；如果寄存器 SPI\_CR2 的 ERRIE 位为‘ 1’，则产生中断，指示发生了错误。

要关闭 I2S 功能时，需要在接收到最后一次 RXNE=1 时将 I2SE 位清‘ 0’。

**注意：** 外部主 I2S 器件需要有通过音频声道发送/接收 16 位或 32 位数据包的功能。



#### 24.4.6 状态标志位

有 3 个状态标志位供用户监控 I2S 总线的状态。

##### 忙标志位(BSY)

BSY 标志由硬件设置与清除(写入此位无效果)，该标志位指示 I2S 通信层的状态。

该位为' 1' 时表明 I2S 通讯正在进行中，但有一个例外：主接收模式(I2SCFG=11)下，在接收期间 BSY 标志始终为低。

在软件要关闭 SPI 模块之前，可以使用 BSY 标志检测传输是否结束，这样可以避免破坏最后一次传输，因此需要严格按照下述过程执行。

当传输开始时，BSY 标志被置为' 1'，除非 I2S 模块处于主接收模式。

下述情况时，该标志位被清除：

- 当传输结束时(除了主发送模式，这种模式下通信是连续的);
- 当关闭 I2S 模块时。

当通信是连续的时候：

- 在主发送模式时，整个传输期间，BSY 标志始终为高;
- 在从模式时，每个数据项传输之间，BSY 标志在 1 个 I2S 时钟周期内变低。

注： 不要使用 BSY 标志处理每一个数据项的发送和接收，最好使用 TXE 和 RXNE 标志。

##### 发送缓存空标志位(TXE)

该标志位为' 1' 表示发送缓冲器为空，可以对发送缓冲器写入新的待发送数据。在发送缓冲器中已有数据时，标志位清' 0'。在 I2S 被关闭时(I2SE 位为' 0' )，该标志位也为' 0'。

##### 接收缓存非空标志位(RXNE)

该标志位置' 1' 表示在接收缓存里有接收到的有效数据。在读取寄存器 SPI\_DR 时，该位清' 0'。

##### 声道标志位(CHSIDE)

在发送模式下，该标志位在 TXE 为高时刷新，指示从 SD 引脚上发送的数据所在的声音。如果在从发送模式下发生了下溢错误，该标志位的值无效，在重新开始通讯前需要把 I2S 关闭再打开。

在接收模式下，该标志位在寄存器 SPI\_DR 接收到数据时刷新，指示接收到的数据所在的声音。

注意，如果发生错误(如上溢 OVR)，该标志位无意义，需要将 I2S 关闭再打开(同时，如果必要修改 I2S 的配置)。

在 PCM 标准下，无论短帧格式还是长帧格式，这个标志位都没有意义。

如果寄存器 SPI\_SR 的标志位 OVR 或 UDR 为' 1'，且寄存器 SPI\_CR2 的 ERRIE 位为' 1'，则会产生

中断。(中断源已经被清除后)可以通过读寄存器 SPI\_SR 来清除中断标志。



#### 24.4-7 错误标志位

I2S 单元有 2 个错误标志位。

##### 下溢标志位(UDR)

在从发送模式下，如果数据传输的第一个时钟边沿到达时，新的数据仍然没有写入 SPI\_DR 寄存器，该标志位会被置' 1'。在寄存器 SPI\_I2SCFGR 的 I2SMOD 位置' 1' 后，该标志位才有效。如果寄存器 SPI\_CR2 的 ERRIE 位为' 1'，就会产生中断。

通过对寄存器 SPI\_SR 进行读操作来清除该标志位。

##### 上溢标志位(OVR)

如果还没有读出前一个接收到的数据时，又接收到新的数据，即产生上溢，该标志位置' 1'，如果寄存器 SPI\_CR2 的 ERRIE 位为' 1'，则产生中断指示发生了错误。

这时，接收缓存的内容，不会刷新为从发送设备送来的新数据。对寄存器 SPI\_DR 的读操作返回最后一个正确接收到的数据。其他所有在上溢发生后由发送设备发出的 16 位数据都会丢失。

通过先读寄存器 SPI\_SR 再读寄存器 SPI\_DR，来清除该标志位。

#### 24.4-8 I2S 中断

下表列举了全部 I2S 中断

中断事件	事件标志位	使能标志位
发送缓冲器空标志位	TXE	TXEIE
接收缓冲器非空标志位	RXNE	RXNEIE
下溢标志位	OVR	ERRIE
上溢标志位	UDR	

Table 24-3 I2S 中断请求

#### 24.4-9 DMA 功能

DMA 的工作方式在 I2S 模式除了 CRC 功能不可用以外，与在 SPI 模式完全相同。因为在 I2S 模式下没有数据传输保护系统。



#### 24.5 SPI/I2S 寄存器列表

SPI/I2S 基地址: 0x4000 0800

偏移地址	名称	描述	复位值
0x00	SPI_CR1	SPI控制寄存器.(I2S模式下不使用)	0x0000 0000
0x04	SPI_CR2	SPI控制寄存器 2	0x0000 0000
0x08	SPI_SR	SPI 状态寄存器。	0x0000 0000
0x0C	SPI*_DATA	SPI 数据寄存器。	0x0000 0000
0x10	SPI_CRCPR	SPI CRC多项式寄存器(I2S模式下不使用)	0x0000 0007
0x14	SPI_RXCRCR	SPI RX CRC寄存器( I2S模式下不使用)	0x0000 0000
0x18	SPI_TXCRCR	SPI TX CRC寄存器	0x0000 0002
0x1C	SPI_I2S_CFGR	SPI_I2S配置寄存器	0x0000 0000
0x20	SPI_I2SPR	SPI_I2S预分频寄存器	0x0000 0007
0x24	SPI_RXCRCR	SPI RX CRC寄存器( I2S模式下不使用)	0x0000 0000

Table 24-4 SPI/I2S 寄存器列表和复位值



## 24.6 SPI/I2S 寄存器说明

### 24.6-1 SPI 控制寄存器 1(SPI\_CR1)(I2S 模式下不使用)

偏移地址: 0x00 复位值: 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E BIDIMODE	BIDIOE	CRCEN	T CRCNEX	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]	MSTR	CPOL	CPHA		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r w	r w	r w	rw	rw	rw

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15	BIDIMODE E	BIDIMODE: 双向数据模式使能 (Bidirectional data mode enable)  0: 选择“双线双向”模式;  1: 选择“单线双向”模式。  注: I2S模式下不使用。	0	R/W
14	BIDIOE 向	BIDIOE: 双向模式下的输出使能 (Output enable in bidirectional mode) 和BIDIMODE位一起决定在“单线双向”模式下数据的输出方向 0: 输出禁止(只收模式); 1: 输出使能(只发模式)。 这个“单线”数据线在主设备端为MOSI引脚, 在从设备端为MISO引脚。 注: I2S模式下不使用。	0	R/W
13	CRCEN	CRCEN: 硬件CRC校验使能 (Hardware CRC calculation enable)  0: 禁止CRC计算;  1: 启动CRC计算。  注: 只有在禁止SPI时(SPE=0), 才能写该位, 否则出错。	0	R/W



位	标记	功能描述	复位值	读写
		该位只能在全双工模式下使用。  注：I2S模式下不使用。		
12	CRCNEXT	CRCNEXT: 下一个发送CRC (Transmit CRC next) 0: 下一个发送的值来自发送缓冲区。 1: 下一个发送的值来自发送CRC寄存器。 注：在SPI DR寄存器写入最后一个数据后应马上设置该位。 注：I2S模式下不使用。	0	R/W
11	DFF	DFF: 数据帧格式 (Data frame format)  0: 使用8位数据帧格式进行发送/接收； 1: 使用16位数据帧格式进行发送/接收。 注：只有当SPI禁止(SPE=0)时，才能写该位，否则出错。 注：I2S模式下不使用。	0	R/W
10	RXONLY	RXONLY: 只接收 (Receive only)  该位和BIDIMODE位一起决定在“双线双向”模式下的传输方向。在多个从设备的配置中，在未被访问的从设备上该位被置1，使得只有被访问的从设备有输出，从而不会造成数据线上数据冲突。 0: 全双工(发送和接收); 1: 禁止输出(只接收模式)。 注：I2S模式下不使用。	0	R/W
9	SSM	SSM: 软件从设备管理 (Software slave management)  当SSM被置位时，NSS引脚上的电平由SSI位的值决定。 0: 禁止软件从设备管理； 1: 启用软件从设备管理。 注：I2S模式下不使用。	0	R/W
8	SSI	SSI: 内部从设备选择 (Internal slave select)  该位只在SSM位为‘1’时有意义。它决定了NSS上的电平，在NSS引脚上的I/O操作无效。 注：I2S模式下不使用。	0	R/W
7	LSBFIRST	LSBFIRST: 帧格式 (Frame format)  0: 先发送MSB;	0	R/W



位	标记	功能描述	复位值	读写
		1: 先发送LSB。  注: 当通信在进行时不能改变该位的值。  注: I2S模式下不使用。		
6	SPE	SPE: SPI使能 (SPI enable)  0: 禁止SPI设备;  1: 开启SPI设备。  注: I2S模式下不使用。  注: 当关闭SPI设备时, 请按照第23.3.8节的过程操作。。	0	R/W
5:3	BR[2:0]	BR[2:0]: 波特率控制 (Baud rate control)  000: fPCLK/2 001: fPCLK/4 010: fPCLK/8 011: fPCLK/16 100: fPCLK/32 101: fPCLK/64 110: fPCLK/128 111: fPCLK/256  当通信正在进行的时候, 不能修改这些位。  注意: I2S模式下不使用。	0	R/W
2	MSTR	MSTR: 主设备选择 (Master selection)  0: 配置为从设备;  1: 配置为主设备。  注: 当通信正在进行的时候, 不能修改该位。  注: I2S模式下不使用。	0	R/W
1	CPOL	CPOL: 时钟极性 (Clock polarity)  0: 空闲状态时, SCK保持低电平;	0	R/W



位	标记	功能描述	复位值	读写
		<p>1: 空闲状态时, SCK保持高电平。 注: 当通信正在进行的时候, 不能修改该位。 注: I2S模式下不使用。</p>		
0	CPHA	<p>CPHA: 时钟相位 (Clock phase) 0: 数据采样从第一个时钟边沿开始; 1: 数据采样从第二个时钟边沿开始。 注: 当通信正在进行的时候, 不能修改该位。 注: I2S模式下不使用。</p>	0	R/W



## 24.6-2 SPI 控制寄存器 2(SPI\_CR2)

偏移地址: 0x04 复位值: 0x0000 0000

31~8	7	6	5	4	3	2	1	0
Reserved	TXEIE	RXNEIE	ERRIE	Res.	Res.	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw			rw	rw	rw

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7	TXEIE	TXEIE: 发送缓冲区空中断使能 (Tx buffer empty interrupt enable) 0: 禁止TXE中断; 1: 允许TXE中断, 当TXE标志置位为' 1' 时产生中断请求。	0	R/W
6	RXNEIE	RXNEIE: 接收缓冲区非空中断使能 (RX buffer not empty interrupt enable) 0: 禁止RXNE中断; 1: 允许RXNE中断, 当RXNE标志置位时产生中断请求。	0	R/W
5	ERRIE	ERRIE: 错误中断使能 (Error interrupt enable) 当错误(CRCERR、OVR、MODF)产生时, 该位控制是否产生中断 0: 禁止错误中断; 1: 允许错误中断。	0	R/W
4:3	Revered	---	0x0	--
2	SSOE	SSOE: SS输出使能 (SS output enable) 0: 禁止在主模式下SS输出, 该设备可以工作在多主设备模式; 1: 设备开启时, 开启主模式下SS输出, 该设备不能工作在多主设备模式。 注: I2S模式下不使用。	0x0	R/W



1	TXDMAEN	TXDMAEN: 发送缓冲区DMA使能 (Tx buffer DMA enable) 当该位被设置时, TXE标志一旦被置位就发出DMA请求 0: 禁止发送缓冲区DMA; 1: 启动发送缓冲区DMA。	0	R/W
0	RXDMAEN	RXDMAEN: 接收缓冲区DMA使能 (Rx buffer DMA enable) 当该位被设置时, RXNE标志一旦被置位就发出DMA请求 0: 禁止接收缓冲区DMA; 1: 启动接收缓冲区DMA。	0	R/W

#### 24.6-3 SPI 状态寄存器(SPI\_SR)

偏移地址: 0x08 复位值: 0x0000 0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7	BSY	BSY: 忙标志 (Busy flag) 0: SPI不忙; 1: SPI正忙于通信, 或者发送缓冲非空。 该位由硬件置位或者复位。 注: 使用这个标志时需要特别注意, 详见第 25.7,8 节	0	R/W
6	OVR	OVR: 溢出标志 (Overrun flag) 0: 没有出现溢出错误; 1: 出现溢出错误。 该位由硬件置位, 由软件序列复位。关于软件序列的详细信	0	R/W



位	标记	功能描述	复位值	读写
		息，参考25.4.7节。		
5	MODF	<p>MODF: 模式错误 (Mode fault)</p> <p>0: 没有出现模式错误;</p> <p>1: 出现模式错误。</p> <p>该位由硬件置位，由软件序列复位。关于软件序列的详细信息，参考25.3.10节。</p> <p>注：I2S模式下不使用。。</p>	0	R/W
4	CRCERR	<p>CRCERR: CRC错误标志 (CRC error flag)</p> <p>0: 收到的CRC值和SPI_RXCRCR寄存器中的值匹配;</p> <p>1: 收到的CRC值和SPI_RXCRCR寄存器中的值不匹配。</p> <p>该位由硬件置位，由软件写' 0' 而复位。</p> <p>注：I2S模式下不使用。</p>	0x0	--
3	UDR	<p>UDR: 下溢标志位 (Underrun flag)</p> <p>0: 未发生下溢;</p> <p>1: 发生下溢。</p> <p>该标志位由硬件置' 1'，由一个软件序列清' 0'，详见25.4.7节。</p> <p>注：在SPI模式下不使用。。</p>	0x0	R/W
2	CHSIDE	<p>CHSIDE: 声道 (Channel side)</p> <p>0: 需要传输或者接收左声道;</p> <p>1: 需要传输或者接收右声道。</p> <p>注：在SPI模式下不使用。在PCM模式下无意义。</p>	0	R/W
1	TXE	<p>TXE: 发送缓冲为空 (Transmit buffer empty)</p> <p>0: 发送缓冲非空;</p> <p>1: 发送缓冲为空。</p>	1	R/W
0	RXNE	<p>RXNE: 接收缓冲非空 (Receive buffer not empty)</p>	0	R/W



位	标记	功能描述	复位值	读写
		0: 接收缓冲为空; 1: 接收缓冲非空。		

ZBT CONFIDENTIAL



#### 24.6-4 SPI 数据寄存器(SPI\_DR)

偏移地址: 0x0C 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留位	0x0	-
15:0	DR[15:0]	DR[15:0]: 数据寄存器 (Data register) 位15:0 待发送或者已经收到的数据 数据寄存器对应两个缓冲区：一个用于写(发送缓冲)；另外 一个用于读(接收缓冲)。写操作将 数据写到发送缓冲区；读操作将返回接收缓冲区里的数 据。 对SPI模式的注释：根据SPI_CR1的DFF位对数据帧格式的 选择，数据的发送和接收可以是8位 或者16位的。为保证正确的操作，需要在启用SPI之前就确 定好数据帧格式。 对于8位的数据，缓冲器是8位的，发送和接收时只会用到 SPI_DR[7:0]。在接收时， SPI_DR[15:8]被强制为0。 对于16位的数据，缓冲器是16位的，发送和接收时会用到 整个数据寄存器，即SPI_DR[15:0]。	0x0	R/W



#### 24.6-5 SPI CRC 多项式寄存器(SPI\_CRCPR)(I2S 模式下不使用)

偏移地址: 0x10 复位值: 0x0000 0007

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留位	0x0	-
15:0	CRCPOLY[15:0]	<p>CRCPOLY[15:0]: CRC多项式寄存器 (CRC polynomial register)</p> <p>该寄存器包含了CRC计算时用到的多项式。</p> <p>其复位值为0x0007，根据应用可以设置其他数值。</p> <p>注：在I2S模式下不使用。</p>	0x0007	R/W

#### 24.6-6 SPI Rx CRC 寄存器(SPI\_RXCRCR)(I2S 模式下不使用)

偏移地址: 0x14 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留位	0x0	-
15:0	RXCRC	RXCRC[15:0]: 接收CRC寄存器	0x0000	R



	[15:0]	<p>位15:0</p> <p>在启用CRC计算时，RXCRC[15:0]中包含了依据收到的字节计算的CRC数值。当在SPI_CR1的CRCEN位写入‘1’时，该寄存器被复位。CRC计算使用SPI_CRCPR中的多项式。</p> <p>当数据帧格式被设置为8位时，仅低8位参与计算，并且按照CRC8的方法进行；当数据帧格式为16位时，寄存器中的所有16位都参与计算，并且按照CRC16的标准。</p> <p>注：当BSY标志为‘1’时读该寄存器，将可能读到不正确的数值。</p> <p>注：在I2S模式下不使用。</p>		
--	--------	--	--	--

#### 24.6-7 SPI Tx CRC 寄存器(SPI\_TXCRCR)

偏移地址: 0x18 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留位	0x0	-
15:0	TXCRC [15:0]	<p>TXCRC[15:0]: 发送CRC寄存器位15:0</p> <p>在启用CRC计算时，TXCRC[15:0]中包含了依据将要发送的字节计算的CRC数值。当在SPI_CR1中的CRCEN位写入‘1’时，该寄存器被复位。CRC计算使用SPI_CRCPR中的多项式。</p> <p>当数据帧格式被设置为8位时，仅低8位参与计算，并且按照CRC8的方法进行；当数据帧格式为16位时，寄存器中的所有16个位都参与计算，并且按照</p>	0x0000	R



	CRC16的标准。 注：当BSY标志为' 1' 时读该寄存器，将可能读到不正确的数值。 注：在I2S模式下不使用。	
--	---	--

ZBT CONFIDENTIAL



#### 24.6-8 SPI\_I2S 配置寄存器(SPI\_I2S\_CFRG)

偏移地址: 0x1C 复位值: 0x0000 0000

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

1 5	1 4	1 3	1 2	11	10	9	8	7	6	5	4	3 D	2 L	1 N	0 N
Reserved				I2SMOD	I2S	I2SCF	C	PCMSYN	Re	I2SST	CKPO	DATLE	CHLE		
				D	E	G	C		S	D	L	N	N		
				rw	rw	r w	r w	rw		r w	r w	rw	rw	rw	

位	标记	功能描述	复位值	读写
31:12	-	保留位	0x0	-
11	I2SMOD	I2SMOD: I2S模式选择 (I2S mode selection) 0: 选择SPI模式; 1: 选择I2S模式。 注: 该位只有在关闭了SPI或者I2S时才能设置。。	0	R/W
10	I2SE	I2SE: I2S使能 (I2S enable) 0: 关闭I2S; 1: I2S使能。 注: 在SPI模式下不使用。。	0	R/W
9:8	I2SCFG	I2SCFG: I2S模式设置 (I2S configuration mode) 00: 从设备发送; 01: 从设备接收; 10: 主设备发送; 11: 主设备接受。 注: 该位只有在关闭了I2S时才能设置。 在SPI模式下不使用。。	0	R/W
7	PCMSYNC	PCMSYNC: PCM帧同步 (PCM frame synchronization) 0: 短帧同步;	0	R/W



位	标记	功能描述	复位值	读写
		1: 长帧同步。 注: 该位只在I2SSTD = 11 (使用PCM标准)时有意义。 在SPI模式下不使用。		
6:5	Revered	--	0	R/W
4:5	I2SSTD	I2SSTD: I2S标准选择 (I2S standard selection) 00: I2S飞利浦标准; 01: 高字节对齐标准 (左对齐); 10: 低字节对齐标准(右对齐); 11: PCM 标准。 关于I2S标准的细节, 详见23.4.2节。 注: 为了正确操作, 只有在关闭了I2S时才能设置该位。 在SPI模式下不使用。。	0	R/W
3	CKPOL	CKPOL: 静止态时钟极性 (Steady state clock polarity) 0: I2S时钟静止态为低电平; 1: I2S时钟静止态为高电平。 注: 为了正确操作, 该位只有在关闭了I2S时才能设置。 在SPI模式下不使用。	0	R/W
2:1	DATLEN	DATLEN: 待传输数据长度 (Data length to be transferred) 00: 16位数据长度; 01: 24位数据长度; 10: 32位数据长度; 11: 不允许。 注: 为了正确操作, 该位只有在关闭了I2S时才能设置。 在SPI模式下不使用。	0	R/W
0	CHLEN	CHLEN: 声道长度 (每个音频声道的数据位数) (Channel length (number of bits per audio channel)) 0: 16位宽; 1: 32位宽。 只有在 DATLEN = 00 时该位的写操作才有意义, 否则声道长度都由硬件固定为32位。		



位	标记	功能描述	复位值	读写
		注：为了正确操作，该位只有在关闭了I2S时才能设置。 在SPI模式下不使用。		

ZBT CONFIDENTIAL



#### 24.6-9 SPI\_I2S 预分频寄存器(SPI\_I2SPR)

偏移地址: 0x18 复位值: 0x0000 0002

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCKOE	ODD	I2SDIV							
						rw	rw	rw							

位	标记	功能描述	复位值	读写
31:10	-	保留位	0x0	-
9	MCKOE	MCKOE: 主设备时钟输出使能 (Master clock output enable) 0: 关闭主设备时钟输出; 1: 主设备时钟输出使能。 注: 为了正确操作, 该位只有在关闭了I2S时才能设置。仅在I2S主设备模式下使用该位。 在SPI模式下不使用	0	R/W
8	ODD	ODD: 奇系数预分频 (Odd factor for the prescaler) 0: 实际分频系数 = I2SDIV *2; 1: 实际分频系数 = (I2SDIV * 2)+1。 参见 25.4.3节。 注: 为了正确操作, 该位只有在关闭了I2S时才能设置。仅在I2S主设备模式下使用该位。 在SPI模式下不使用。	0	R/W
7:0	I2SDIV[7:0]	I2SDIV: I2S线性预分频 (I2S linear prescaler) 禁止设置I2SDIV [7:0] = 0或者I2SDIV [7:0] = 1 参见 25.4.3节。 注: 为了正确操作, 该位只有在关闭了I2S时才能设置。仅在I2S主设备模式下使用该位。	0x02	R/W



	在SPI模式下不使用。	
--	-------------	--

## 【25】 时钟校准/监测模块(CLKTRIM)

### 25.1 简介

CLKTRIM (Clock Trimming)模块是一个专门用来校准/监测时钟的电路。在校准模式下选择精准的时钟源来校准不精准的时钟源，反复校准，调节不精准时钟源的参数，直到被校准时钟源的频率达到精度要求。校准模式下计数值会有一定的误差，但是在允许的精度误差范围内。在监测模式下选择稳定的时钟源来监测系统工作时钟，在设定的监测周期下，监测系统工作时钟是否有失效的情况发生并产生中断。在校准模式和监控模式下，所需的时钟源都必须初始化和使能，具体的配置过程请参考第6章系统复位与时钟(RCC)。

### 25.2 主要特性

CLKTRIM 支持以下特性：

- (1) 校准模式
- (2) 监测模式
- (3) 32 位参考时钟计数器可加载初值
- (4) 32 位待校准时钟计数器可配置溢出值
- (5) 6 种参考时钟源
- (6) 4 种待校准时钟源
- (7) 支持中断方式



## 25.3 CLKTRIM 功能描述

### 25.3-1 CLKTRIM 校准模式

校准模式主要用于选择一个精准的时钟源作为参考时钟来校准一个不精准的待校准时钟源。由软件按照下面的操作流程反复校准，调节待校准时钟源的参数，直到待校准时钟源满足频率精度要求。

#### 25.3-1.1 操作流程

1. 设置 CLKTRIM\_CR.REFCLK\_SEL 寄存器选择参考时钟。
2. 设置 CLKTRIM\_CR.CALCLK\_SEL 寄存器选择被校准时钟。
3. 设置 CLKTRIM\_CR.CLKEN 使能校准和参考时钟。
4. 设置 CLKTRIM\_REFCON.RCNTVAL 寄存器为校准时间。
5. 设置 CLKTRIM\_CR.IE 寄存器使能中断。
6. 设置 CLKTRIM\_CR.TRIM\_START 寄存器开始校准。
7. 参考时钟计数器和待校准时钟计数器开始计数。
8. 当参考时钟计数器从初始值减计数到 0 时，CLKTRIM\_IFR.STOP 置 1，触发中断。
9. 中断服务子程序判断 CLKTRIM\_IFR.STOP 为 1，读取寄存器 CLKTRIM\_REFCNT 和 CLKTRIM\_CALCNT 的值，清零 CLKTRIM\_CR.TRIM\_START 寄存器结束校准。

注意，校准模式在校准过程中有可能因为校准时间设置过长，发生待校准时钟计数器在 CLKTRIM\_IFR.STOP 置 1 之前溢出的情况，CLKTRIM\_IFR.CALCNT\_OVF 置 1，触发中断。中断服务子程序发现 CLKTRIM\_IFR.CALCNT\_OVF 置 1 时，清零 CLKTRIM\_CR.TRIM\_START 寄存器结束校准。这种情况下校准是无法正确进行的，必须调整校准时间，重新校准。具体步骤是：

- (1) 设置 CLKTRIM\_REFCON.RCNTVAL 寄存器调整校准时间。
- (2) 设置 CLKTRIM\_CR.TRIM\_START 寄存器重新开始校准。



### 25.3-2 CLKTRIM 监测模式

监测模式主要用于选择一个稳定的时钟源作为参考时钟，在设定的时间周期下监测系统工作时钟的异常状态。在监测模式下只能选择外部 HXT 时钟或者外部 LXT 时钟作为被监测时钟。

#### 25.3-2.1 操作流程

1. 设置 CLKTRIM\_CR.REFCLK\_SEL 寄存器选择参考时钟。
2. 设置 CLKTRIM\_CR.CALCLK\_SEL 寄存器选择被监控时钟。
3. 设置 CLKTRIM\_CR.CLKEN 使能被监控和参考时钟。
4. 设置 CLKTRIM\_REFCON.RCNTVAL 寄存器为监控间隔时间。
5. 设置 CLKTRIM\_CALCON.CALOVCNT 寄存器为被监控时钟计数器溢出时间。
6. 设置 CLKTRIM\_CR.MON\_EN 寄存器使能监控功能。
7. 设置 CLKTRIM\_CR.IE 寄存器使能中断。
8. 设置 CLKTRIM\_CR.TRIM\_START 寄存器开始监控。
9. 参考时钟计数器和被监控时钟计数器开始计数。
10. 当参考时钟计数器计数到达监控间隔时间时，判断被监控时钟计数器是否溢出。如果溢出表示被监控时钟工作正常。如果没有溢出表示被监控时钟失效，CLKTRIM\_IFR.HXT\_FAULT 或 CLKTRIM\_IFR.LXT\_FAULT 置 1，触发中断。
11. 如果配置 RCC\_SYSCLKCR.CLKFAILEN 为 1(参考 6.4.8 系统时钟源配置寄存器 (RCC\_SYSCLKCR))，在中断发生后，会自动切换系统时钟源到内部高速 RC 时钟(HIRC)，处理中断服务子程序，清除中断标志位 CLKTRIM\_IFR.HXT\_FAULT 或 CLKTRIM\_IFR.LXT\_FAULT，清零 CLKTRIM\_CR.TRIM\_START 寄存器结束监测。



#### 25.4 CLKTRIM 寄存器列表

基址地址: 0x4000 3400

地址偏移	名称	寄存器描述	复位值
0x00	CLKTRIM_CR	配置寄存器。	0x0000 0000
0x04	CLKTRIM_REFCON	参考计数器初值配置寄存器。	0x0000 0000
0x08	CLKTRIM_REFCNT	参考计数器值寄存器。	0x0000 0000
0x0C	CLKTRIM_CALCNT	校准计数器值寄存器。	0x0000 0000
0x10	CLKTRIM_IFR	中断标志位寄存器。	0x0000 0000
0x14	CLKTRIM_ICLR	中断标志位清除寄存器	0x0000 0000
0x18	CLKTRIM_CALCON	校准计数器溢出值配置寄存器	0xFFFF FFFF

Table 25-1 CLKTRIM 寄存器列表和复位值



## 25.5 CLKTRIM 寄存器说明

### 25.5-1 配置寄存器(CLKTRIM\_CR)

地址偏移: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								RDM ODE	MSB F IRST	EN	SIZE	保留		CLKDIV[1:0]	
								R/W	R/W	R/W	R/W			R/W	

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
8	CLKEN	参考时钟和校准时钟使能 0: 禁止 1: 时钟使能	0x0	R/W
7	IE	中断使能寄存器 0: 禁止 1: 使能	0x0	R/W
6	MON_EN	监视模式使能寄存器 0: 禁止 1: 使能	0x0	R/W
5:4	CALCLK_SEL[1:0]	待校准/监测时钟选择寄存器 00: HIRC 01: HXT 10: SIRC 11: LXT	0x0	R/W
3:1	REFCLK_SEL[2:0]	参考时钟选择寄存器: 000: HIRC 001: HXT 010: SIRC 011: LXT 100: HXT 旁路时钟	0x0	R/W
0	TRIM_START	校准/监测开始寄存器: 0: 停止 1: 开始	0x0	R/W



### 25.5-2 参考计数器处置配置寄存器(CLKTRIM\_REFCON)

偏移地址: 0x04 复位值: 0xFFFF FFFF

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

位	标记	功能描述	复位值	读写
31:0	RCNTVAL	参考计数器初始值	0x0	R/W



### 25.5-3 参考计数器值寄存器(CLKTRIM\_REFCNT)

偏移地址: 0x08 复位值: 0xFFFF FFFF

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

位	标记	功能描述	复位值	读写
31:0	REFCNT	参考计数器值 读该寄存器需要先打开时钟使能, 当 TRIM_START 有效后, 写入的初始值就会更新到该寄存器	0x0	RO

### 25.5-4 校准计数器值寄存器(CLKTRIM\_CALCNT)

偏移地址: 0x0C 复位值: 0x0000 0000

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

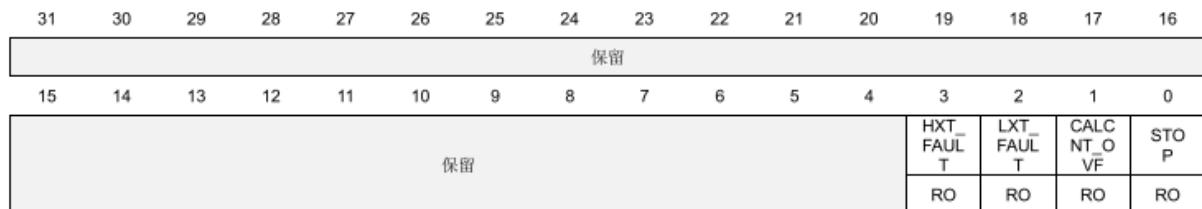
位	标记	功能描述	复位值	读写
31:0	CALCNT	校准计数器值	0x0	RO



### 25.5-5 中断标志位寄存器(CLKTRIM\_IFR)

偏移地址: 0x10

复位值: 0x0000 0000



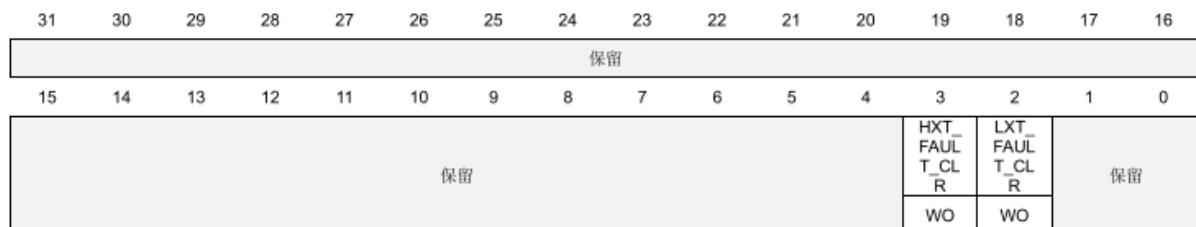
位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3	HXT_FAULT	HXT 失效标志 1: HXT 失效 0: HXT 未失效	0	RO
2	LXT_FAULT	LXT 失效标志 1: LXT 失效 0: LXT 未失效	0	RO
1	CALCNT_OVF	校准计数器溢出标志 CLKTRIM_CR.TRIM_START 写零清除此标志位	0	RO
0	STOP	参考计数器停止标志 CLKTRIM_CR.TRIM_START 写零清除此标志位	0	RO



### 25.5-6 中断标志位清除寄存器(CLKTRIM\_ICLR)

偏移地址: 0x14

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3	HXT_FAULT_CLR	清除HXT 失效标志, 写 1 清除。	0x0	WO
2	LXT_FAULT_CLR	清除LXT 失效标志, 写 1 清除。	0x0	WO
1:0	-	保留	0x0	-



### 25.5-7 校准计数器溢出值配置寄存器(CLKTRIM\_CALCON)

地址偏移: 0x18 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												RXD ONE CLR	TXD ONE CLR	INITD ONE CLR	ACK ERR CLR
												WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15:0	CALOVCNT	校准计数器溢出值比较值 在监控模式，在监控周期内如果被监控时钟计数到 CALOVCNT 值时，表示被监控时钟工作正常；如果小于该设定值表示监控时钟停止，会置起被监控时钟的失效位。	0x0	R/W



## 【26】实时时钟(RTC-L032)

### 26.1 简介

实时时钟(RTC)是一个独立的 BCD 定时器/计数器，提供秒、分、时(12/24 小时制)、周、日、月和年信息。

RTC 模块拥有自动唤醒功能，用于管理所有的低功耗模式。

两个 32 位寄存器以 BCD 格式存储秒、分、时(12/24 小时制)、周、日、月和年。

RTC 具有自动月份天数补偿功能，每月的天数和闰年的天数可自动调整。

使用两个 32 位寄存器存储可编程报警信息，包括秒、分、时、周、日、月和年。

对由晶体本身的频偏、温度漂移及其他原因引起任何误差，可以利用 RTC 本身的数字校准功能进行修正。

上电复位后，所有 RTC 寄存器将被禁止访问，以防止意外的写操作。

当设备处于运行模式、低功耗模式或复位状态(上电复位(POR 复位)除外)，只要电压在工作范围内，RTC 将保持正常运行。

### 26.2 主要特性

RTC 模块的主要特性如下：

- (1) 日历功能，可显示秒、分、时(12/24 小时制)、周、日、月和年。
- (2) 可进行自动闰年调整。
- (3) 具有闹钟中断和周期中断功能。
- (4) 数字校准电路(计数器定期修正)：来自一个几秒钟的校准窗口。
- (5) 时钟源可选：外部低速时钟(LXT)、内部低速时钟(SIRC)、外部高速时钟(HXT)
- (6) 1Hz 方波输出



### 26.3 RTC 功能描述

#### 26.3-1 RTC 结构框图

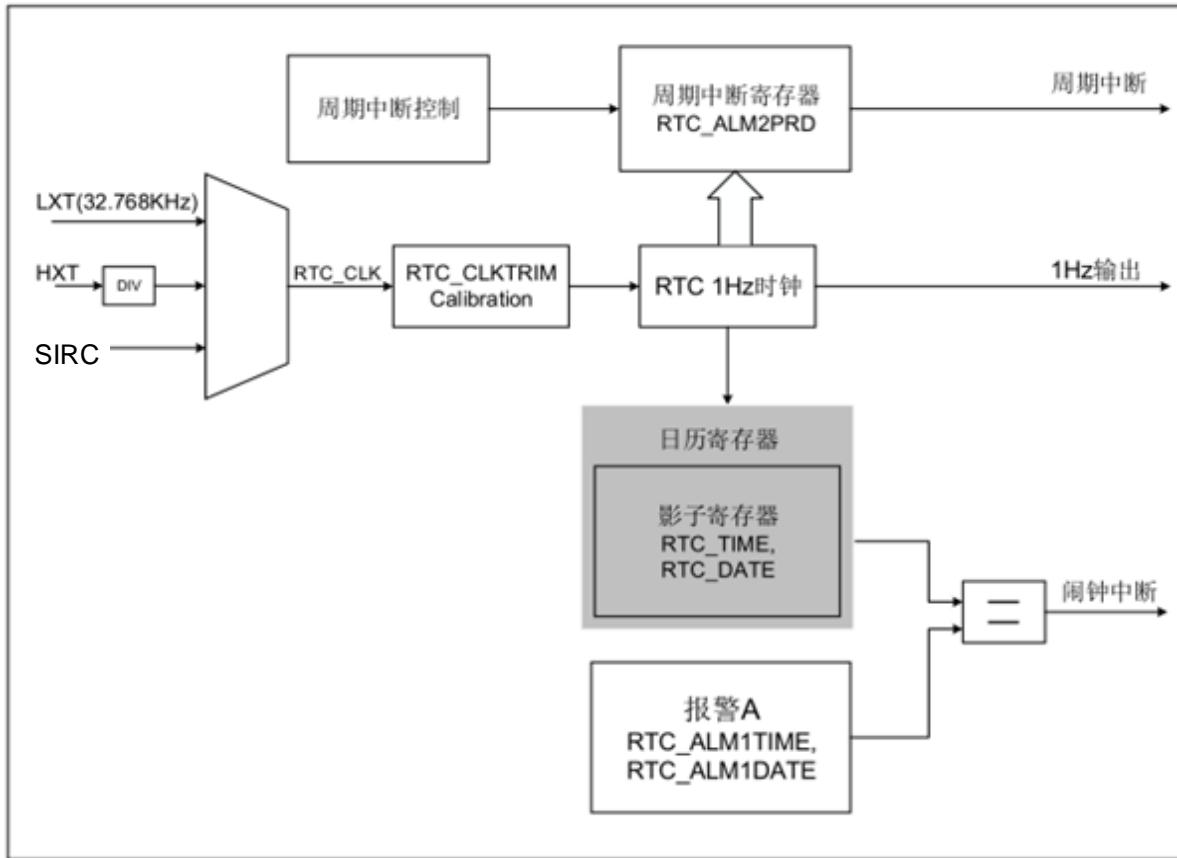


Figure 26-1 RTC 框图

RTC 模块包括：

- (1) 一个闹钟报警中断
- (2) 一个周期中断
- (3) 校准后的 1Hz 时钟输出
- (4) 万年历寄存器



### 26.3-2 RTC 时钟

由时钟控制器从以下 3 种时钟中选择 RTC 时钟源(RTC\_CLK):

- (1) LXT 时钟作为 RTC 时钟;
- (2) SIRC 时钟作为 RTC 时钟;
- (3) HXT 时钟作为 RTC 时钟。

更多有关 RTC 时钟源的配置信息, 请参考第 7 章系统复位与时钟(RCC)

### 26.3-3 复位过程

任何可用的系统复位源都将导致日历影子寄存器和 RTC 初始化和状态寄存器(RTC\_ISR)复位至默认值。

然而, 下列寄存器的复位与系统复位无任何关联, 只与上电复位(POR 复位)有关: RTC 控制寄存器(RTC\_CR)、RTC 时钟控制寄存器(RTC\_CLKCR)、RTC 校准寄存器(RTC\_CLKTRIM)、ALM 寄存器(RTC\_ALM1DATE/RTC\_ALM1TIME/RTC\_ALM2PRD)、RTC 当前日历寄存器(RTC\_TIME/RTC\_DATA)。

除上电复位外, 发生任何系统复位时 RTC 将维持运行状态。发生上电复位后, RTC 停止运行, 所有 RTC 寄存器复位至默认值。

### 26.3-4 寄存器的写保护

上电复位后, 所有 RTC 寄存器将处于写保护状态。通过向写保护寄存器 RTC\_WPR 写入指定关键字来启动 RTC 寄存器的写权限。

通过以下操作解除所有 RTC 寄存器的写保护。

1. 向 RTC\_WPR 寄存器写入'0xCA';
2. 向 RTC\_WPR 寄存器写入'0x53'。

注意: 保护解除后, 任何对该寄存器的再一次写将重新激活写保护。



### 26.3-5 日历初始化及配置

按照以下顺序完成时间和日期值的初始化，包括时间格式和预分频器的配置：

1. 通过 RTC\_CLKCR.CKSEL 选择 RTC 计时时钟源，如果选择 HXT 时钟还要先设定预分频器。
2. 设定 RTC\_CLKCR.RTCCKEN 使能 RTC 计时时钟。
3. RTC\_ISR 寄存器的 WAIT 位置 “1”，进入初始化模式。
4. 等待 RTC\_ISR 寄存器的 WAITF 位置 “1”，确保已经正式进入初始化模式。由于时钟同步的延迟，该过程大约需要 2 个 RTC\_CLK 时钟周期。在该模式下日历计数器暂停运行，此时可更新时间和日期计数器的值。
5. 通过 RTC\_CR 寄存器的 FMT 位设置时间格式(12 小时制/24 小时制)。
6. 将初始时间和日期值加载到时间寄存器(RTC\_TIME 与 RTC\_DATE)。
7. 需要进行时钟误差补偿时，设定时钟补偿寄存器 RTC\_CLKTRIM。
8. 清除 RTC\_ISR.WAIT 位的值退出初始化模式。日历计数器的实际值将会自动加载，并在 4 个 RTCCLK 时钟周期后重新启动。

在完成上述一系列初始化操作后，日历将开始计时。

### 26.3-6 读出计数寄存器

当 RTC\_CR 寄存器的 BYPSHAD 控制位被清除时：

为确保在安全同步机制下正常读 RTC 日历寄存器(RTC\_TIME 和 RTC\_DATE)，APB 时钟频率(fPCLK) 应至少为 RTC 时钟频率(fRTCCLK) 的 7 倍以上。当 APB 时钟频率低于 7 倍 RTC 时钟频率时，软件必须两次读取日历时间和日期寄存器。如果第二次读取的值与第一次读取的值相同，说明返回值是正确的，否则需再次读取。任何情况下，APB 时钟频率都必须大于 RTC 时钟频率。

日历寄存器的内容被复制到 RTC\_TIME 和 RTC\_DATE 影子寄存器中时，RTC\_ISR 寄存器的 RSF 位被置位。复制操作每两个 RTC\_CLK 周期执行一次。为确保两者的值保持一致，读 RTC\_TIME 时硬件会锁定 RTC\_DATE 影子寄存器的值，直到 RTC\_DATE 的值被读取。

为避免软件在时间间隔少于 2 个 RTC\_CLK 周期的情况下多次访问日历，每次读日历 RSF 位应由软件清零，软件必须等待 RSF 位被置位后才能读 RTC\_TIME 和 RTC\_DATE 寄存器。

从低功耗模式唤醒后，RSF 位应由软件清零，软件必须等待 RSF 位被再次置位后才能读 RTC\_TIME 和 RTC\_DATE 寄存器。RSF 位应在唤醒后被清除，而不是进入低功耗模式前。

系统复位后，软件必须等待 RSF 位被置位后才能读 RTC\_TIME 和 RTC\_DATE 寄存器。事实上系统复位将导致影子寄存器复位至其默认值。

当 RTC\_CR 寄存器的 BYPSHAD 控制位被置位时(无需考虑影子寄存器)：

读日历寄存器，直接从日历计数器获取值，无需等待 RSF 位被置位。此功能在刚退出低功耗模式时非常有用，因为影子寄存器在低功耗模式下不会自动更新。



在 BYPSHAD 为 “1” 时，如果两次读寄存器之间出现 RTC\_CLK 边沿，不同寄存器中的结果可能 会互不相关。此外，如果在读操作过程中遇到 RTC\_CLK 边沿，则某个寄存器的值可能不正确。软 件必须读取所有的寄存器两次，并比较两次读取的结果；或者通过比较两组最低有效日历寄存器的 结果，以检验数据是否正确且有一定关联。

### 26.3-7 写入计数寄存器

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

### 26.3-8 闹钟设定

设置 RTC\_CR 寄存器的 ALM1EN 位，启动闹钟功能。当日历中秒、分、时、周、日、月、年与报 警寄存器 RTC\_ALM1TIME 和 RTC\_ALM1DATE 中设定的值匹配，RTC\_ISR.ALM1\_F 由硬件置 1。所有日历字段都可以通过 RTC\_ALM1DATE 寄存器中的 ALMxEN 位选择为报警源。设置 RTC\_CR 寄存器的 ALM1\_INTEN 位，会产生报警中断。

### 26.3-9 校准 1Hz 输出

RTC 可选择输出校准后的 1Hz 时钟。通过 RTC\_CR.RTC1HZOE 设定输出使能，通过 RTC\_CLKTRIM 来设定校准值。

### 26.3-10 RTC 时钟校准

RTC 模块通过每隔一个固定的时间周期屏蔽指定的 RTC 时钟周期数来补偿 RTC 时钟的频率，通 过 RTC\_CLKTRIM.MODE[1:0]来选择调整的时间间隔：

- 0b00：每 60 秒(SEC=00)校准一次
  - 0b01：每 30 秒(SEC=00, 30)校准一次
  - 0b10：每 15 秒(SEC=00, 15, 30, 45)校准一次
  - 0b11：每 6 秒(SEC=00, 06, 12, 18, 24, 30, 36, 42, 48, 54)校准一次
- 通过 RTC\_CLKTRIM.TRIM[7:0]来指定屏蔽的 RTC 时钟周期数。

注意 RTC\_CLKTRIM.TRIM[7:0]的值时有符号整数，范围为-128~+127。



## 26.4 RTC 中断

RTC 支持两种中断类型。闹钟中断、定周期中断。闹钟中断与定周期中断共享一个中断信号，通过标志寄存器位来区分中断源。

### 26.4-1 RTC 闹钟中断

1. 设定 RTC\_CR.ALM1EN=0，禁止闹钟功能；
2. 设定时间闹钟寄存器 RTC\_ALM1TIME、日期闹钟寄存器 RTC\_ALM1DATE；
3. 设定 RTC\_CR.ALM1EN=1，闹钟许可；
4. 清除中断标志位 RTC\_ISR.ALM1\_F；
5. 设定 RTC\_CR.ALM1\_INTEN=1，闹钟中断许可，若当前日历时间与闹钟寄存器相等时，触发闹钟中断；
6. 等待发生中断；

### 26.4-2 RTC 周期中断

控制寄存器 RTC\_CR 的 RTC\_CR.ALM2\_INTEN=1 时，选择的周期发生后，触发定周期唤醒中断，由于闹钟和定周期共享中断，通过标志寄存器位来区分。

1. 设定 RTC\_CR.ALM2\_INTEN=0，禁止周期中断功能；
2. 设定周期闹钟寄存器 RTC\_ALM2PRD；
3. 清除中断标志位 RTC\_ISR.ALM2\_F；
4. 设定 RTC\_CR.ALM2\_INTEN=1，周期中断许可，选择的周期发生后，触发定周期唤醒中断；
5. 等待发生中断；



## 26.5 RTC 寄存器列表

基址: 0x4000 2800

地址偏移	名称	描述	默认值
0x00	RTC_CR	RTC 控制寄存器	0x00000000
0x04	RTC_CLKCR	RTC 时钟控制寄存器	0x00000000
0x08	RTC_TIME	RTC 时间寄存器	0x00000000
0x0C	RTC_DATE	RTC 日期寄存器	0x00000000
0x10	RTC_ALM1TIME	RTC 时间闹钟寄存器	0x00000000
0x14	RTC_ALM1DATE	RTC 日期闹钟寄存器	0x00000000
0x18	RTC_ALM2PRD	RTC 周期闹钟寄存器	0x00000000
0x1C	RTC_CLKTRIM	RTC 时钟调校寄存器	0x00000000
0x20	RTC_ISR	初始化和状态寄存器	0x00000000
0x24	RTC_INTCLR	RTC 状态清除寄存器	0x00000000
0x28	RTC_WPR	RTC 写保护寄存器	0x00000000

Table 26-1 RTC 寄存器列表和复位值



## 26.6 RTC 寄存器说明

### 26.6-1 RTC 控制寄存器(RTC\_CR)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							STAR T	保留	ALM1 EN	ALM2 _INT EN	ALM1 _INT EN	保留	FMT	RTC1 HZO E	BYP S HAD
							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:10	-	保留	0x0	-
8	START	0: 停止RTC 计数器 1: 使能RTC 计数器	0	R/W
7	-	保留	0	-
6	ALM1EN	ALM1 闹钟功能使能。 0: 禁止ALM1 闹钟功能 1: 使能ALM1 闹钟功能  注意: 在日历计数 RTC_CLKCR.RTCCKEN=1) 过程中并且 ALM1 闹钟中断许可使能(ALM1_INTEN=1) 的情况下使能ALM1EN 时, 为防止误动作, 请将系统中断关闭。 使能后请将 ALM1_F 标志位清除。	0	R/W
5	ALM2_INTEN	ALM2 周期中断使能。 0: 禁止ALM2 周期中断 1: 使能ALM2 周期中断	0	R/W
4	ALM1_INTEN	ALM1 闹钟中断使能。 0: 禁止ALM1 闹钟中断 1: 使能ALM1 闹钟中断	0	R/W
3	-	保留	0	-
2	FMT	时间格式。 0: 12 小时制(AM/PM 时间格式) 1: 24 小时制	0	R/W
1	RTC1HZOE	RTC 1Hz 输出时能 0: 禁止	0	R/W

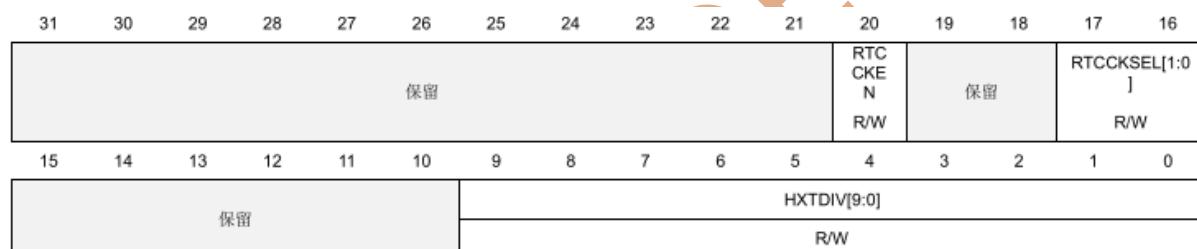


		1: 使能		
0	BYPSHAD	绕过影子寄存器 0: 从影子寄存器读取日历值, 影子寄存器每两个 RTC_CLK 周期更新一次 1: 直接从日历计数器读取日历值 注: 如果 APB 时钟频率低于 RTCCLK 频率的 7 倍, BYPSHAD 必须置 "1"。	0	R/W

### 26.6-2 RTC 时钟控制寄存器(RTC\_CLKCR)

偏移地址: 0x04

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:21	-	保留	0x0	-
20	RTCCKEN	RTC 计数时钟使能: 由软件置 1 或清 0 写: 0: RTC 时钟关闭 1: RTC 时钟开启 启读: 0: RTC 时钟关闭 1: RTC 时钟开启	0	R/W
19:18	-	保留	0x0	-



17:16	RTCKSEL[1:0]	RTC 时钟源选择 由软件设置来选择 RTC 时钟源。一旦 RTC 时钟源被选定，这些位值不能被改变，除非 RTC 被复位。可通过设置 RCC_RTCRST.RTCRST 位来复位 RTC 域。 00: LXT 振荡器作为RTC 时钟 01: SIRC 振荡器作为RTC 时钟 10: FHXT/(HXTDIV[9:0]) 11: 保留	0x0	R/W
15:10	-	保留	0x0	-
9:0	HXTDIV[9:0]	外部高速晶振时钟分频 0: 停止 其它值: $F=F_{HXT}/(HXTDIV[9:0])$	0x0	R/W



### 26.6-3 RTC 时间寄存器(RTC\_TIME)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				WEEK[2:0]			保留		H20_PA	HOUR19[4:0]					
				R/W					R/W	R/W					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		MIN[6:0]				保留		SEC[6:0]				R/W			
		R/W						R/W							

位	标记	功能描述	复位值	读写
31:27	-	保留	0x0	-
26:24	WEEK[2:0]	星期计数器。星期计数器值为二进制计数，计数区间从 0 到 6 (7 不被使用，除非不使用星期计数)。星期和星期计数器值得对应关系由用户定义。 (比如星期日=0, 星期一=1.....星期六=6)	0x0	R/W
23:22	-	保留	0x0	-
21	H20_PA	这两位表示小时计数器。HOUR19 的值为BCD 编码。时间格式是由时钟系统决定的。	0	R/W
20:16	HOUR19[4:0]	12 小时制模式，H20_PA 指上午或者下午。24 小时制模式，H20_PA 决定了计数器的十位是否为 2。 12 小时制模式，当[H20_PA,HOUR19]从[1,11](11PM)数到[0,12](13M) 的时候，日期计数器增加一天。24 小时制模式，当[H20_PA,HOUR19]从[1,3](23H)数到[0,0](0H)的时候，日期计数器增加一天。	0x0	R/W
15	-	保留	0x0	-
14:8	MIN[6:0]	分钟计数器。分钟计数器的值为BCD 编码，计数区间从 0 到 59。当分钟计数器从 59 数到 0 的时候，小时计数器增长 1。当这个计数器被写入时，小于一秒的时间将被忽略掉。	0x0	R/W
7	-	保留	0	-



6:0	SEC[6:0]	秒计数器。秒计数器的值为BCD 编码，计数区间从 0 到 59。当秒计数器从 59 数到 0 的时候，分钟计数器增长 1。当这个计数器被写入时，小于一秒的时间将被忽略掉。	0x0	R/W
-----	----------	---	-----	-----

ZBT CONFIDENTIAL



#### 26.6-4 RTC 日期寄存器(RTC\_DATE)

偏移地址: 0x0C

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:24	-	保留	0x0	-
23:16	YEAR[7:0]	年计数器。年计数器代表了十进制年的十位和个位。年计数器为BCD编码，计数区间从00到99。当年计数器从99数到00时，世纪计数器增长1。 当世纪计数器为0时，04, 08, ..., 92, 96为闰年。 当世纪计数器为1时，00, 04, 08, ..., 92, 96为闰年。	0x0	R/W
15	CEN	世纪计数器。0代表20世纪，1代表21世纪	0	R/W
14:13	-	保留	0x0	-
12:8	MONTH[4:0]	月计数器。月计数器为BCD编码，计数区间从01到12。当年计数器从12数到01时，年计数器增长1。	0x0	R/W
7:6	-	保留	0x0	-
5:0	DAY[5:0]	天计数器。天计数器为BCD编码，计数区间如下： 01到31：一月，三月，五月，七月，八月，十月，十二月； 01到30：四月，六月，九月，十一月； 01到29：闰年的二月 01到28：非闰年的二月 当年计数器从99数到00时，世纪计数器增长1。	0x0	R/W



### 26.6-5 RTC 时间闹钟寄存器(RTC\_ALM1TIME)

偏移地址: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留				ALWEEK[2:0]			保留		ALH20_PA		ALHOUR19[4:0]					
				R/W					R/W		R/W					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留		ALMIN[6:0]				保留		ALSEC[6:0]				R/W				
		R/W						R/W				R/W				

位	标记	功能描述	复位值	读写
31:27	-	保留	0x0	-
26:24	ALWEEK[2:0]	闹钟星期设定	0x0	R/W
23:22	-	保留	0x0	-
21	ALH20_PA	闹钟小时设定。参考小时计数寄存器。	0x0	R/W
20:16	ALHOUR19[4:0] ]			
15	-	保留	0	-
14:8	ALMIN[6:0]	闹钟分钟设定	0x0	R/W
7	-	保留	0	-
6:0	ALSEC[6:0]	闹钟秒设定	0x0	R/W



### 26.6-6 RTC 日期闹钟寄存器(RTC\_ALM1DATE)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ALMYEAR EAREN	ALMMONEN	ALMDAYEN	ALMWEEKEN	ALMHOUREN	ALMMINEN	ALMSECEN	ALYEAR[7:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALCEN	保留	ALMONTH[4:0]				保留	ALDAY[5:0]				R/W				R/W
R/W		R/W					R/W				R/W				R/W

位	标记	功能描述	复位值	读写
31	-	保留	0	-
30	ALMYEAREN	闹钟年设定使能	0	R/W
29	ALMMONEN	闹钟月设定使能	0	R/W
28	ALMDAYEN	闹钟日设定使能	0	R/W
27	ALMWEEKEN	闹钟星期设定使能	0	R/W
26	ALMHOUREN	闹钟小时设定使能	0	R/W
25	ALMMINEN	闹钟分钟设定使能	0	R/W
24	ALMSECEN	闹钟秒设定使能	0	R/W
23:16	ALYEAR[7:0]	闹钟年设定	0x0	R/W
15	ALCEN	闹钟世纪设定	0	R/W
14:13	-	保留	0x0	-
12:8	ALMONTH[4:0]	闹钟月设定	0x0	R/W
7:6	-	保留	0x0	-
5:0	ALDAY[5:0]	闹钟日设定	0x0	R/W



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

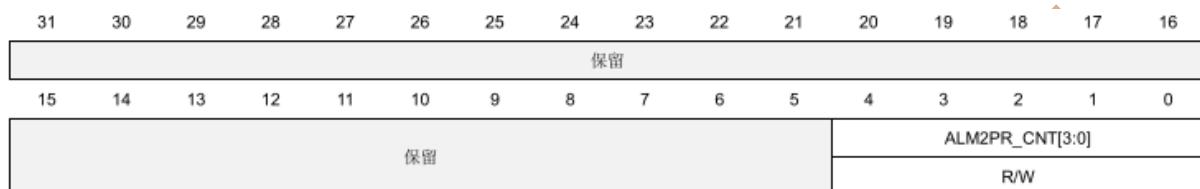
ZBT CONFIDENTIAL



### 26.6-7 RTC 周期闹钟寄存器(RTC\_ALM2PRD)

偏移地址: 0x18

复位值: 0x0000 0000



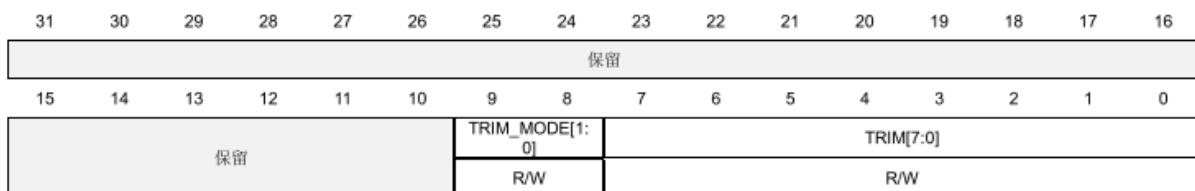
位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3:0	ALM2PR_CNT	周期闹钟 2 计数周期设定。 0x0:关闭周期闹钟 2 0x1:1 秒 0x2:1/2 秒 0x3:1/4 秒 0x4:1/8 秒 0x5:1/16 秒 0x6:1/32 秒 0x7:1/64 秒 0x8:1/128 秒 0x9:10 秒 0xA:30 秒 0xB:1 分钟 0xC:30 分钟 0xD:60 分钟 0xE:12 小时 0xF:24 小时	0x0	R/W



### 26.6-8 RTC 时钟调校寄存器(RTC\_CLKTRIM)

偏移地址: 0x1C

复位值: 0x0000 0000



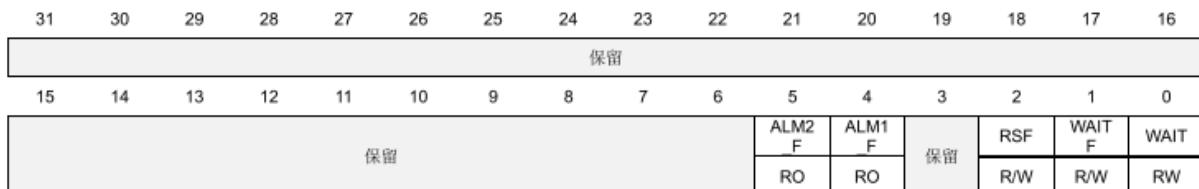
位	标记	功能描述	复位值	读写
31:10	-	保留	0x0	-
9:8	MODE[1:0]	时钟调节寄存器。决定了时钟调节的频率。 0x0: 每 60 秒(SEC=00) 0x1: 每 30 秒(SEC=00, 30) 0x2: 每 15 秒(SEC=00, 15, 30, 45) 0x3: 每 6 秒(SEC=00, 06, 12, 18, 24, 30, 36, 42, 48, 54)	0x0	R/W
7:0	TRIM[7:0]	时钟补偿时间寄存器。此寄存器为有符号整数。(-128~+127)	0x0	R/W



### 26.6-9 RTC 初始化和状态寄存器(RTC\_ISR)

偏移地址: 0x20

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:6	-	保留	0x0	-
5	ALM2_F	周期闹钟 2 中断原始状态寄存器。 当此寄存器被读出时，状态值被返回： 0: 周期闹钟 2 中断没有被激活。 1: 周期闹钟 2 中断被激活。	0	RO
4	ALM1_F	闹钟中断原始状态寄存器。 当此寄存器被读出时，状态值被返回： 0: 闹钟中断没有被激活。 1: 闹钟中断被激活。	0	RO
3	保留		0	-
2	RSF	寄存器同步标志 每当日历寄存器中的内容复制到影子寄存器 (RTC_TIME、RTC_DATE) 中时，该位由硬件置位。当处于忽略影子寄存器模式 (BYPSHAD=1) 下时，该位由硬件在初始化模式下清除。该位也可由软件清除。 在初始化模式下，该位可由硬件/软件清除。 0: 日历影子寄存器尚未同步； 1: 日历影子寄存器已经同步。注意不能软件写 1	0	R/W
1	WAITF	0: 非写入/配置状态 1: 写入/配置状态  注意：WAITF 是 WAIT 位设定是否有效标志。在写入/配置前请确认该位是否为“1”。计数过程中，在WAIT 位清“0”后等待写入完成后该位才清“0”。	0	R/W



0	WAIT	0: 正常计数模式 1: 写入/配置模式  注意：在写入/配置时请将该位置“1”，由于计数器在连续计数，请在1秒的时间内完成写入/配置操作并将该位清“0”。	0	R/W
---	------	---	---	-----

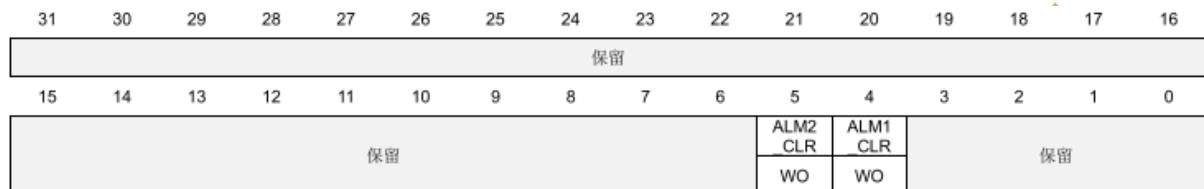
ZBT CONFIDENTIAL



### 26.6-10 RTC 状态清除寄存器(RTC\_INTCLR)

偏移地址: 0x24

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:6	-	保留	0x0	-
5	ALM2_CLR	周期闹钟 2 中断原始状态清除寄存器。 当此寄存器被写入时，中断原始状态被要求清除： 0: 没有操作。 1: 周期闹钟 2 中断原始状态被清除。	0	WO
4	ALM1_CLR	闹钟中断原始状态清除寄存器。 当此寄存器被写入时，中断原始状态被要求清除： 0: 没有操作。 1: 闹钟中断原始状态被清除。	0	WO
3:0	-	保留	0x0	-



### 26.6-11 RTC 写保护寄存器(RTC\_WPR)

偏移地址: 0x28

复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	WPR	写入指定关键字来启动RTC寄存器的写权限。向RTC_WPR寄存器写入'0xCA'; 向RTC_WPR寄存器写入'0x53'。 注意：保护解除后，任何对该寄存器的再一次写将重新激活写保护。	0x0	WO



## 【27】实时时钟(RTC-F103)

### 27.1 简介

实时时钟是一个独立的定时器。RTC 模块拥有一组连续计数的计数器，在相应软件配置下，可提供时钟日历的功能。修改计数器的值可以重新设置系统当前的时间和日期。

RTC 模块和时钟配置系统(RCC\_BDCR 寄存器)处于后备区域，即在系统复位或从待机模式唤醒后，RTC 的设置和时间维持不变。

系统复位后，对后备寄存器和 RTC 的访问被禁止，这是为了防止对后备区域(BKP)的意外写操作。执行以下操作将使能对后备寄存器和 RTC 的访问：

- (1) 设置寄存器 RCC\_APB1ENR 的 PWREN 和 BKREN 位，使能电源和后备接口时钟
- (2) 设置寄存器 PWR\_CR 的 DBP 位，使能对后备寄存器和 RTC 的访问。

### 27.2 主要特性

RTC 模块的主要特性如下：

- 可编程的预分频系数：分频系数高为  $2^{20}$ 。
- 32 位的可编程计数器，可用于较长时间段的测量。
- 2 个分离的时钟：用于 APB1 接口的 PCLK1 和 RTC 时钟(RTC 时钟的频率必须小于 PCLK1 时钟频率的四分之一以上)。
- 可以选择以下三种 RTC 的时钟源：
  - HXT 时钟除以 128；
  - LSE 振荡器时钟；
  - SIRC 振荡器时钟(详见 6.2.8 节 RTC 时钟)。
- 2 个独立的复位类型：
  - APB1 接口由系统复位；
  - RTC 核心(预分频器、闹钟、计数器和分频器)只能由后备域复位(详见 6.1.3 节)。
- 3 个专门的可屏蔽中断：
  - 闹钟中断，用来产生一个软件可编程的闹钟中断。
  - 一秒中断，用来产生一个可编程的周期性中断信号(长可达 1 秒)。
- 溢出中断，指示内部可编程计数器溢出并回转为 0 的状态



### 27.3 RTC 功能描述

#### 27.3-1 RTC 概述与结构框图

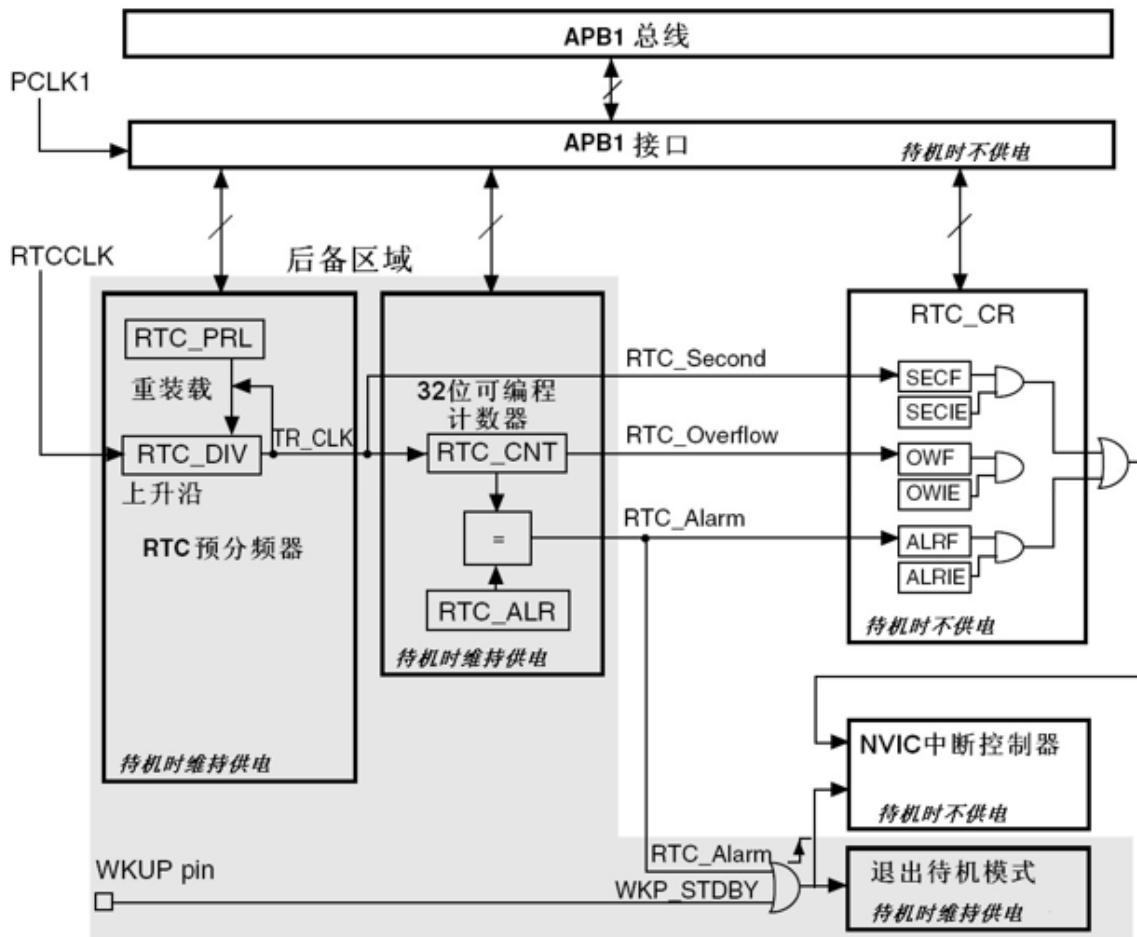


Figure 27-1 RTC 框图

RTC 由两个主要部分组成(参见上图)。第一部分(APB1 接口)用来和 APB1 总线相连。此单元还包含一组 16 位寄存器，可通过 APB1 总线对其进行读写操作(参见 28.4 节)。APB1 接口由 APB1 总线时钟驱动，用来与 APB1 总线接口。

另一部分[RTC 核心]由一组可编程计数器组成，分成两个主要模块。第一个模块是 RTC 的预分频模块，它可编程产生长为 1 秒的 RTC 时间基准 TR\_CLK。RTC 的预分频模块包含了一个 20 位的可编程分频器(RTC 预分频器)。如果在 RTC\_CR 寄存器中设置了相应的允许位，则在每个 TR\_CLK 周期中 RTC 产生一个中断(秒中断)。第二个模块是一个 32 位的可编程计数器，可被初始化为当前的系统时间。系统时间按 TR\_CLK 周期累加并与存储在 RTC\_ALR 寄存器中的可编程时间相比较，如果 RTC\_CR 控制寄存器中设置了相应允许位，比较匹配时将产生一个闹钟中断。



### 27.3-2 复位过程

除了 RTC\_PRL、RTC\_ALR、RTC\_CNT 和 RTC\_DIV 寄存器外，所有的系统寄存器都由系统复位或电源复位进行异步复位。

RTC\_PRL、RTC\_ALR、RTC\_CNT 和 RTC\_DIV 寄存器仅能通过备份域复位信号复位，详见第 8.1-2 节

### 27.3-3 读 RTC 寄存器。

RTC 核完全独立于 RTC APB1 接口。

软件通过 APB1 接口访问 RTC 的预分频值、计数器值和闹钟值。但是，相关的可读寄存器只在与

RTC APB1 时钟进行重新同步的 RTC 时钟的上升沿被更新。RTC 标志也是如此的。

这意味着，如果 APB1 接口曾经被关闭，而读操作又是在刚刚重新开启 APB1 之后，则在第一次的内部寄存器更新之前，从 APB1 上读出的 RTC 寄存器数值可能被破坏了(通常读到 0)。下述几种情况下能够发生这种情形：

- (1) 发生系统复位或电源复位
- (2) 系统刚从待机模式唤醒(参见第 6.2-2 节：低功耗模式)。
- (3) 系统刚从停机模式唤醒(参见第 6.2-2 节：低功耗模式)

所有以上情况下，APB1 接口被禁止时(复位、无时钟或断电)RTC 核仍保持运行状态。

因此，若在读取 RTC 寄存器时，RTC 的 APB1 接口曾经处于禁止状态，则软件首先必须等待 RTC\_CRL 寄存器中的 RSF 位(寄存器同步标志)被硬件置'1'。

注： RTC 的 APB1 接口不受 WFI 和 WFE 等低功耗模式的影响。

### 27.3-4 配置 RTC 寄存器

必须设置 RTC\_CRL 寄存器中的 CNF 位，使 RTC 进入配置模式后，才能写入 RTC\_PRL、RTC\_CNT、RTC\_ALR 寄存器。

另外，对 RTC 任何寄存器的写操作，都必须在前一次写操作结束后进行。可以通过查询 RTC\_CR 寄存器中的 RTOFF 状态位，判断 RTC 寄存器是否处于更新中。仅当 RTOFF 状态位是'1'时，才可以写入 RTC 寄存器。

配置过程：

1. 查询 RTOFF 位，直到 RTOFF 的值变为'1'
2. 置 CNF 值为 1，进入配置模式
3. 对一个或多个 RTC 寄存器进行写操作
4. 清除 CNF 标志位，退出配置模式



5. 查询 RTOFF，直至 RTOFF 位变为'1'以确认写操作已经完成。

仅当 CNF 标志位被清除时，写操作才能进行，这个过程至少需要 3 个 RTCCLK 周期。

注意：保护解除后，任何对该寄存器的再一次写将重新激活写保护。

### 27.3-5 RTC 标志的设置

按照以下顺序完成时间和日期值的初始化，包括时间格式和预分频器的配置：

在每一个 RTC 核心的时钟周期中，更改 RTC 计数器之前设置 RTC 秒标志(SECF)。

在计数器到达 0x0000 之前的后一个 RTC 时钟周期中，设置 RTC 溢出标志(OWF)。

在计数器的值到达闹钟寄存器的值加 1(RTC\_ALR+1)之前的 RTC 时钟周期中，设置 RTC\_Alarm 和 RTC 闹钟标志(ALRF)。对 RTC 闹钟的写操作必须使用下述过程之一与 RTC 秒标志同步：

- 使用 RTC 闹钟中断，并在中断处理程序中修改 RTC 闹钟和/或 RTC 计数器。
- 等待 RTC 控制寄存器中的 SECF 位被设置，再更改 RTC 闹钟和/或 RTC 计数器。

下图是 PR=0003, ALARM=00004 秒和闹钟波形图示例，PR=0003, ALARM=00004

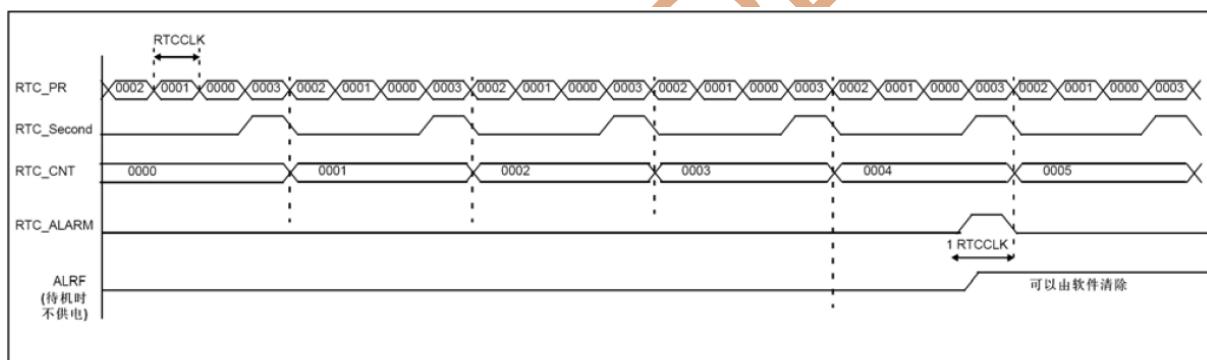


Figure 27-2 RTC 秒和闹钟波形示例图

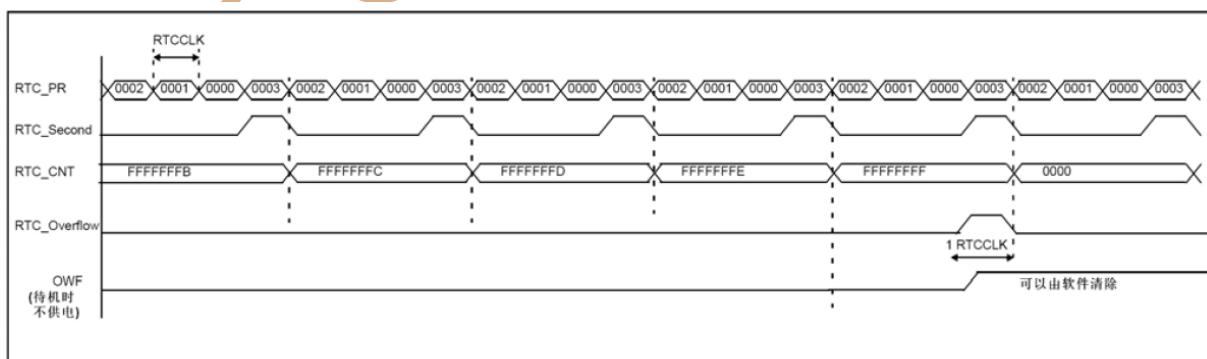


Figure 27-3 RTC 溢出波形图示例，PR=0003 示意图



#### 27.4 RTC 寄存器列表

基址: 0x4000 3800

地址偏移	名称	描述	默认值
0x00	RTC_CRH	RTC 控制寄存器高位	0x00000000
0x04	RTC_CRL	RTC 控制寄存器低位	0x00000000
0x08	RTC_PRLH	RTC 预分频装载寄存器高位	0x00000000
0x0C	RTC_PRLL	RTC 预分频装载寄存器低位	0x00000000
0x10	RTC_DIVH	RTC 预分频器余数寄存器高位	0x00000000
0x14	RTC_DIVL	RTC 预分频器余数寄存器低位	0x00000000
0x18	RTC_CNTH	RTC 计数器寄存器高位	0x00000000
0x1C	RTC_CNTL	RTC 计数器寄存器低位	0x00000000
0x20	RTC_ALRH	RTC 闹钟寄存器高位	0x00000000
0x24	RTC_ALRL	RTC 闹钟寄存器低位	0x00000000

Table 27-1 RTC 寄存器列表和复位值



## 27.5 RTC 寄存器说明

### 27.5-1 控制寄存器高位(RTC\_CRH)

偏移地址: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OWIE	ALRIE	SECIE												
保留	R/W	R/W	R/W												

位	标记	功能描述	复位值	读写
31:3	-	保留	0x0	-
2	<b>OWIE</b>	<b>OWIE</b> : 允许溢出中断位 (Overflow interrupt enable) 0 : 屏蔽(不允许)溢出中断  1 : 允许溢出中断	0	-
1	<b>ALRIE</b>	<b>ALRIE</b> : 允许闹钟中断 (Alarm interrupt enable) 0 : 屏蔽(不允许)闹钟中断 1 : 允许闹钟中断	0	R/W
0	<b>SECIE</b>	<b>SECIE</b> : 允许秒中断 (Second interrupt enable) 0 : 屏蔽(不允许)秒中断 1 : 允许秒中断	0	R/W



### 27.5-2 RTC 控制寄存器低位(RTC\_CRL)

偏移地址: 0x04 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RTOFF	CNF	RSF	OWF	ALRF										
保留	R	R/W	W	W	W										

位	标记	功能描述	复位值	读写
31:6	-	保留	-	-
5	<b>RTOFF</b>	<b>RTOFF</b> : RTC操作关闭 (RTC operation OFF) RTC模块利用这位来指示对其寄存器进行的后一次操作的状态，指示操作是否完成。若此位为'0'，则表示无法对任何的RTC寄存器进行写操作。此位为只读位。 0 : 上一次对RTC寄存器的写操作仍在进行； 1 : 上一次对RTC寄存器的写操作已经完成。	0x0	R
4	<b>CNF</b>	<b>CNF</b> : 配置标志 (Configuration flag) 此位必须由软件置'1'以进入配置模式，从而允许向RTC_CNT、RTC_ALR或RTC_PRL寄存器写入数据。只有当此位在被置'1'并重新由软件清'0'后，才会执行写操作。 0 : 退出配置模式(开始更新RTC寄存器)； 1 : 进入配置模式。	0x0	R/W
3	<b>RSF</b>	<b>RSF</b> : 寄存器同步标志 (Registers synchronized flag) 每当RTC_CNT寄存器和RTC_DIV寄存器由软件更新或清'0'时，此位由硬件置'1'。在APB1复位后，或APB1时钟停止后，此位必须由软件清'0'。要进行任何的读操作之前，用户程序必须等待这位被硬件置'1'，以确保RTC_CNT、RTC_ALR或RTC_PRL已经被同步。 0 : 寄存器尚未被同步； 1 : 寄存器已经被同步。	0x0	W
2	<b>OWF</b>	<b>OWF</b> : 溢出标志 (Overflow flag) 当32位可编程计数器溢出时，此位由硬件置'1'。如果RTC_CRH寄存器中OWIE=1，则产生中断。此位只能由软件清'0'。对此位写'1'是无效的。 0 : 无溢出； 1 : 32位可编程计数器溢出。	0x0	W
1	<b>ALRF</b>	<b>ALRF</b> : 闹钟标志 (Alarm flag) 当32位可编程计数器达到RTC_ALR寄存器所设置的预定值，此位由硬件		W



		置'1'。如果RTC_CRH 寄存器中ALRIE=1，则产生中断。此位只能由软件清'0'。对此位写'1'是无效的。 0：无闹钟； 1：有闹钟。		
0	<b>SECF</b>	<b>SECF</b> : 秒标志 (Second flag) 当32位可编程预分频器溢出时，此位由硬件置'1'同时RTC计数器加1。因此，此标志为分辨率可编程的RTC计数器提供一个周期性的信号(通常为1秒)。如果RTC_CRH寄存器中SECIE=1，则产生中断。此位只能由软件清除。对此位写'1'是无效的。 0：秒标志条件不成立； 1：秒标志条件成立。		W

ZBT CONFIDENTIAL



### 27.5.3 RTC 预分频装载寄存器高位(RTC\_PRLH)

偏移地址: 0x08 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PRL[19:16]														
保留	R/W														

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3:0	PRL[19:16]	<b>PRL[19:16] :</b> RTC预分频装载值高位 (RTC prescaler reload value high) 根据以下公式，这些位用来定义计数器的时钟频率： $f_{TR\_CLK} = f_{RTCCLK}/(PRL[19:0]+1)$ 注：不推荐使用0值，否则无法正确的产生RTC中断和标志位。	0	R/W



#### 27.5.4 RTC 预分频装载寄存器高位(RTC\_PRLH)

偏移地址: 0x0C 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15	PRL[15:0]	PRL[15:0] : RTC预分频装载值低位 根据以下公式，这些位用来定义计数器的时钟频率： $f_{RTC\_CLK} = f_{RTCCLK}/(PRL[19:0]+1)$	0	R/W



### 27.5.5 RTC 预分频器余数寄存器高位(RTC\_DIVH)

偏移地址: 0x10 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	<b>RTC_DIV[19:16]</b>														
保留	R/W														

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3:0	<b>RTC_DIV [19:16]</b>	<b>RTC_DIV[19:16] :</b> RTC时钟分频器余数高位 (RTC clock divider high)	0	R/W



### 27.5.6 RTC 预分频器余数寄存器高位(RTC\_DIVH)

偏移地址: 0x14 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15	RTC_DIV[15:0]	RTC_DIV[15:0] : RTC时钟器余数低位 (RTC clock divider low)	0x0	R/W



### 27.5-7 RTC 计数器寄存器高位(RTC\_CNTH)

偏移地址: 0x18 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RTC_CNT [19:16]														
保留		R/W													

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3:0	RTC_CNT [19:16]	RTC_CNT[31:16] : RTC计数器高位 (RTC counter high)  可通过读RTC_CNTH寄存器来获得RTC计数器当前值的高位部分。要对此寄存器进行写操作前，必须先进入配置模式(参见28.3-4节)。	0	R/W



### 27.5-8 RTC 计数器寄存器低位(RTC\_CNTL)

偏移地址: 0x1C 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15	RTC_CNT[15:0]	RTC_CNT[15:0] : RTC计数器低位。 可通过读RTC_CNTL寄存器来获得RTC计数器当前值的低位部分。要对此寄存器进行写操作，必须先进入配置模式(参见16.3.4节)	0x0	R/W



### 27.5.9 RTC 闹钟寄存器高位(RTC\_ALRH)

偏移地址: 0x20 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	RTC_ALR[19:16]														
保留		R/W													

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3:0	RTC_ALR [19:16]	RTC_ALR[31:16] : RTC闹钟值高位 (RTC alarm high)  此寄存器用来保存由软件写入的闹钟时间的高位部分。要对此寄存器进行写操作，必须先进入配置模式(参见18..4节)。	0	R/W



### 27.5.10 RTC 阔钟寄存器低位(RTC\_ALRL)

偏移地址: 0x24 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15	RTC_ALR[15:0]	RTC_ALR[15:0] : RTC阔钟值低位 (RTC alarm low) 此寄存器用来保存由软件写入的阔钟时间的低位部分。要对此寄存器进行写操作，必须先进入配置模式(参见18.3-4节)	0x0	R/W



## 【28】模拟/数字转换器(ADC1,2,3)

### 28.1 模块简介

ZB32F103 内部集成了一个 12 位高精度、高转换速率的逐次逼近(SAR)型模数转换器(ADC)模块。

具有以下特性：

- 12 位转换精度
- 1Msps 转换速度
- 24 路转换通道：  
16 个引脚通道、1 个 V18 , 1 个 V12 校准通道 , 1 个 DACO 通道,3 个 OPA 输出通道.
- 四个参考电压(Reference Voltage VREF):  
(1) 电源电压(AVDD), (2)PB01 EXT VREF (3)IN-Vref1 (1.5V) (4)IN-Vref2(2.048V)
- ADC 的电压输入范围: 0-VREF
- 3 种转换模式：单次转换、连续转换、累加转换 并 支持 DMA 请求
- ADC 的转换速率软件可配
- 支持片内及外设中断自动触发 ADC 转换启动，有效降低芯片功耗、提高转换实时性



## 28.2 ADC 框图

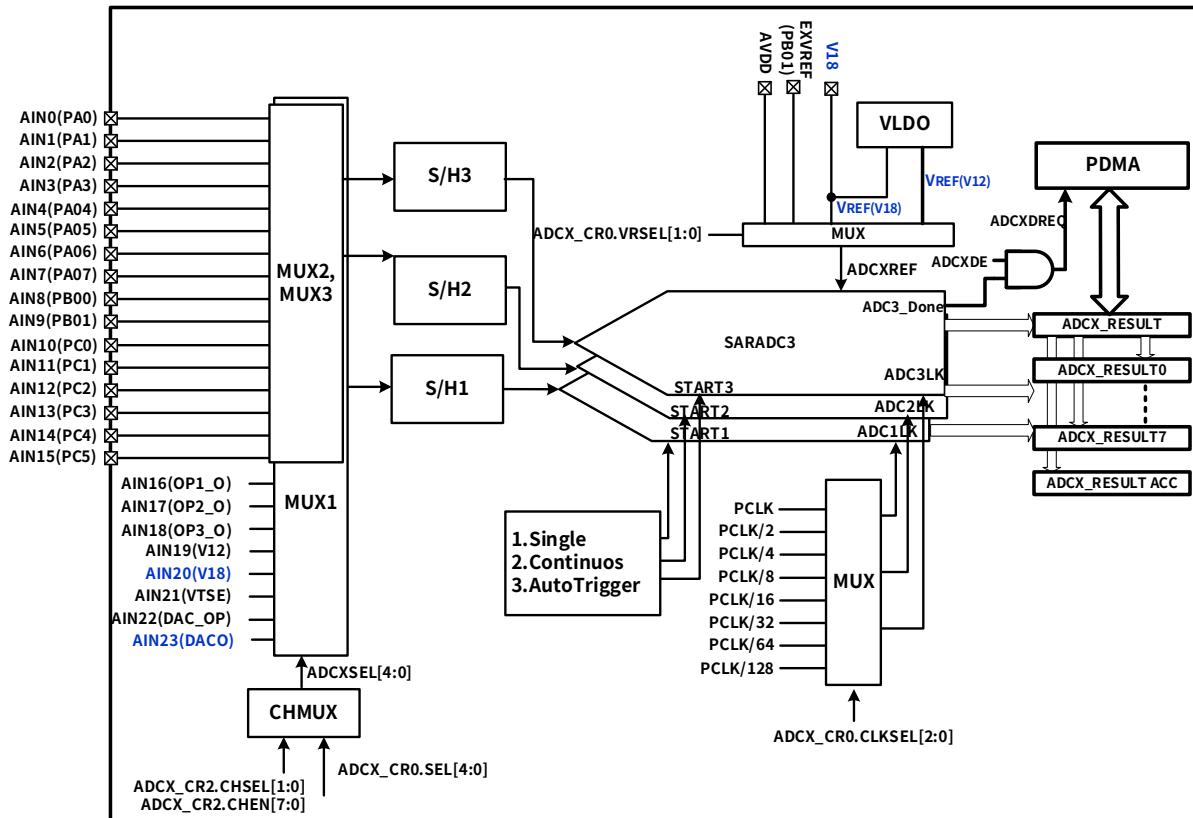
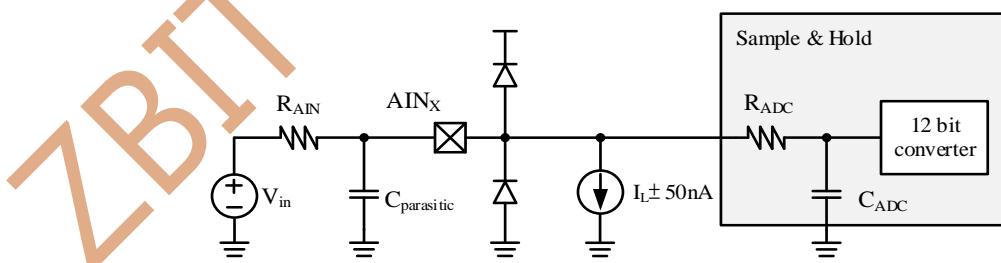


Figure 28-1 ADC 架构方块图

### 28.2-1 ADC 输入阻抗

ADC 典型应用图请参考如图(A)。



图(A).ADC 典型应用图

1.  $C_{parasitic}$  为 PCB 上的电容，其电容值大小取决于 PCB 线路配置(大约  $7\text{pF}$ )。若电容值过大将会降低 ADC 精准度，或需降低 ADC clock 频率来维持 ADC 精准度。
2. 表二. $R_{AIN}$  为参考表(B)与图 A 中  $C_{ADC}$  与  $R_{ADC}$  所得



表(B).  $R_{AIN}$  对应  $f_{ADCCLK}$

$t_s(\mu s)$	$f_{ADCCLK}(\text{Hz})$	$SAM$	$R_{AIN}(k\Omega)$
0.167	24M	4	0.05
0.333	12M	4	0.5
0.667	6M	4	2.0
2.67	3M	8	10
5.33	1.5M	8	20
10.7	0.75M	8	40
21.3	0.375M	8	50

ZBT CONFIDENTIAL



### 28.3 转换时序及速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由采样过程及逐次比较过程组成。其中采样过程需要 4~8 个 ADC 时钟，由 ADCX\_CR0.SAM 配置；逐次比较过程需要 12 个 ADC 时钟。所以，一次 ADC 转换共需要 16~20 个 ADC 时钟。

ADC 转换速度的单位为 sps(samples per second)，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：ADC 时钟的频率/一次 ADC 转换所需要的 ADC 时钟的个数。

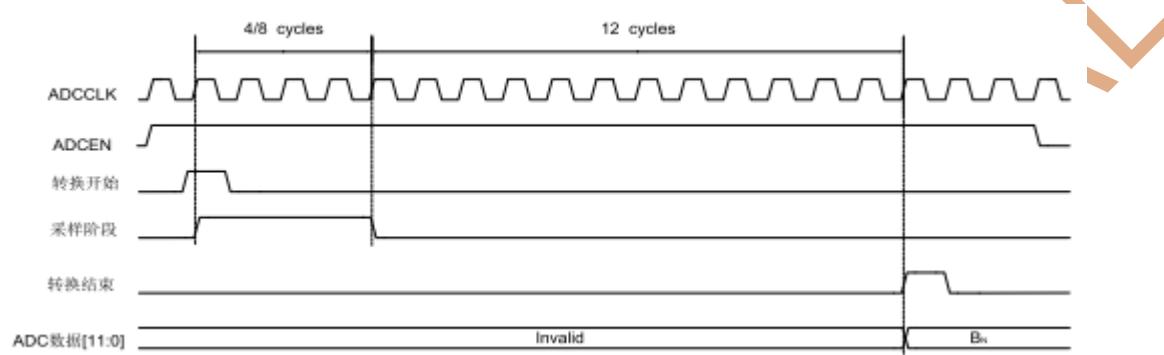


Figure 28-2 ADC 转换时序图



#### 28.4 单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 21 路 ADC 通道进行转换。该模式既可通过设置 ADCX\_CR0.START 位启动也可通过设置 ADCX\_CR1[9:0]的外部触发启动。一旦选定通道的 ADC 转换完成，ADCX\_CR0.START 位自动清零，转换结果保存在 ADCX\_RESULT 寄存器中。通过 START 位启动 ADC 单次转换操作流程：

- 根据 pin 配置对应的 GPIO，将待转换的 ADC 信道配置为模拟端口。
- 配置 ADCX\_CR2.CIRCLE\_MODE 为 0，选择非循环模式。
- 配置 ADCX\_CR1.CT 为 0，选择单次转换模式。
- 配置 ADCX\_CR0.SAM 及 ADCX\_CR0.CLKSEL，设置 ADC 的转换速度。
- 配置 ADCX\_CR0.SEL，选择待转换的通道(注意需要和步骤 1 保持一致)。
- 配置 ADCX\_CR0.ADCEN 为 1，使能 ADC 模块。
- 配置 ADCX\_CR0.START 为 1，启动 ADC 单次转换。
- 等待 ADCX\_CR0.START 为 0，读取 ADCX\_RESULT 寄存器以获取 ADC 转换结果。
- 如需对其他通道进行转换，重复执行步骤 4~7。
- 配置 ADCX\_CR0.ADCEN，关闭 ADC 模块。

注：内部信号触发 ADC 转换的流程配置类似，额外需增加 trigger 的选择。

#### 28.5 连续转换模式

在连续转换模式下，启动一次 ADC 可对多个通道依次进行多次转换；可转换的 ADC 7 路通道为 AIN0~AIN7/AIN8~AIN15/AIN16~AIN23。ADC 转换的总次数由 ADCX\_CR2.ADCCNT[7:0]进行配置；待进行转换的通道由 ADCX\_CR2.CHEN[7:0], ADCX\_CR2.CHSEL[1:0] 进行配置,如下表所示。

CHSEL[1:0]	2' b00	2' b01	2' b10	ADC Convert Result
CHEN[7:0]				
8' b0000_0001	AIN0	AIN8	AIN16	ADCX_Result0
8' b0000_0010	AIN1	AIN9	AIN17	ADCX_Result1
8' b0000_0100	AIN2	AIN10	AIN18	ADCX_Result2
8' b0000_1000	AIN3	AIN11	AIN19	ADCX_Result3
8' b0001_0000	AIN4	AIN12	AIN20	ADCX_Result4
8' b0010_0000	AIN5	AIN13	AIN21	ADCX_Result5
8' b0100_0000	AIN6	AIN14	AIN22	ADCX_Result6
8' b1000_0000	AIN7	AIN15	AIN20	ADCX_Result7

该模式既可通过设置 ADCX\_CR0.START 位启动，也可通过设置 ADCX\_CR1[9:0]的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成



总转换次数后，`ADCX_RAWINTSR.CONT_INTF` 位会自动置 1，转换结果保存在转换通道所对应的 `ADCX_RESULT0~ADCX_RESULT7` 寄存器中。如果总转换次数大于待转换的 ADC 通道的数量，则 `ADCX_RESULT0~ADCX_RESULT7` 寄存器中只保存最后一次的转换结果。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换的过程。通过寄存器将 `START` 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 `ADCCNT` 的计数值变为 0。

EX : `ADCX_CR2.CHSEL[1:0] = 2' b00` :

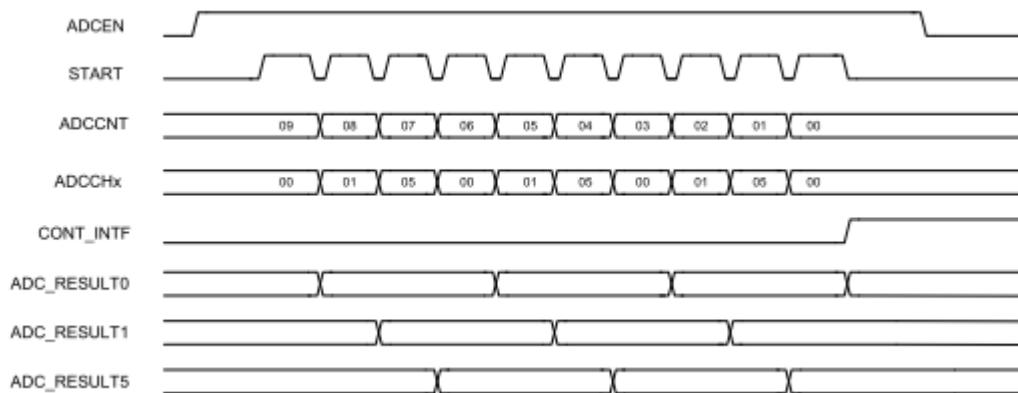


Figure 28-3 ADC 连续转换过程示例



配置步骤：

1. 根据 pin 配置对应的 GPIO，将待转换的 ADC 信道配置为模拟端口。
2. 配置 ADCX\_CR2.circle\_mode 为 0，选择非循环模式。
3. 配置 ADCX\_CR1.ct 为 1，选择连续转换模式。
4. 配置 ADCX\_CR2.adccnt[7:0]，选择连续转换的总转换次数。
5. 配置 ADCX\_CR0.sam 及 ADCX\_CR0.clksel，设置 ADC 的转换速度。
6. 配置 ADCX\_CR2.chen[7:0]，使能待转换的通道。
7. 配置 ADCX\_INTCLR.cont\_intc 为 1，清除 ADCX\_RAWINTSR.cont\_intf 标志。
8. 配置 ADCX\_INTEN.CONT\_IEN 为 1，打开连续转换完成中断掩码使能。
9. 配置 ADCX\_CR0.staterst 为 1，复位连续转换状态。
10. 配置 ADCX\_CR0.adcen 为 1，使能 ADC 模块。
11. 配置 ADCX\_CR0.start 为 1，启动 ADC 连续转换。
12. 等待 ADCX\_RAWINTSR.cont\_intf 变为 1，读取 ADCX\_result0~ADCX\_result7 寄存器以获取相应通道的转换结果。
13. 如需对其他通道进行转换，重复执行步骤 6~11。
14. 配置 ADCX\_CR0.adcen，关闭 ADC 模块。



## 28.6 连续转换累加模式

在连续转换累加模式下，启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加；可转换的 ADC 通道为 AIN0~AIN7。ADC 转换的总次数通过 ADCX\_CR2.ADCCNT[7:0]进行配置；待进行转换的通道由 ADCX\_CR2.CHEN[7:0]进行配置。该模式既可通过设置 ADCX\_CR0.START 位启动，也可通过设置 ADCX\_CR1[9:0]的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADCX\_RAWINTSR.CONT\_INTF 位会自动置 1，转换结果的累加值保存在 ADCX\_RESULT\_ACC 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。通过寄存器将 START 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。每次转换完成时，ADCX\_RESULT\_ACC 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

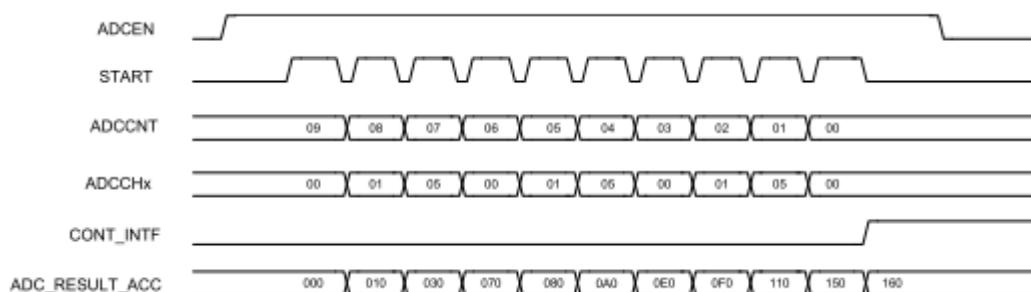


Figure 28-4 连续转换累加过程示例

配置步骤：

1. 根据 pin 配置对应的 GPIO，将待转换的 ADC 信道配置为模拟端口。
2. 配置 ADCX\_CR2.circle\_mode 为 0，选择非循环模式。
3. 配置 ADCX\_CR1.ct 为 1，选择连续转换模式。
4. 配置 ADCX\_CR1.racc\_en 为 1，选择 ADC 转换自动累加功能。
5. 配置 ADCX\_CR2.adccnt[7:0]，选择连续转换的总转换次数。
6. 配置 ADCX\_CR0.sam 及 ADCX\_CR0.clksel，设置 ADC 的转换速度。
7. 配置 ADCX\_CR2.chen[7:0]，选择待转换的通道。
8. 配置 ADCX\_INTCLR.cont\_intc 为 1，清除 ADCX\_RAWINTSR.cont\_intf 标志。
9. 配置 ADCX\_INTEN.CONT\_IEN 为 1，打开连续转换完成中断掩码使能。
10. 设置 ADCX\_CR1.racc\_clr 为 1，清除 ADCX\_result\_acc 寄存器。
11. 配置 ADCX\_CR0.staterst 为 1，复位连续转换状态。
12. 配置 ADCX\_CR0.adcen 为 1，使能 ADC 模块。
13. 配置 ADCX\_CR0.start 为 1，启动 ADC 连续转换。



14. 等待 ADCX\_RAWINTSR.cont\_intf 变为 1，读取 ADCX\_result\_ACC 寄存器以获取连续转换累加结果。
15. 如需对其他通道进行转换，重复执行步骤 6~11。
16. 配置 ADCX\_CR0.adcen，关闭 ADC 模块。

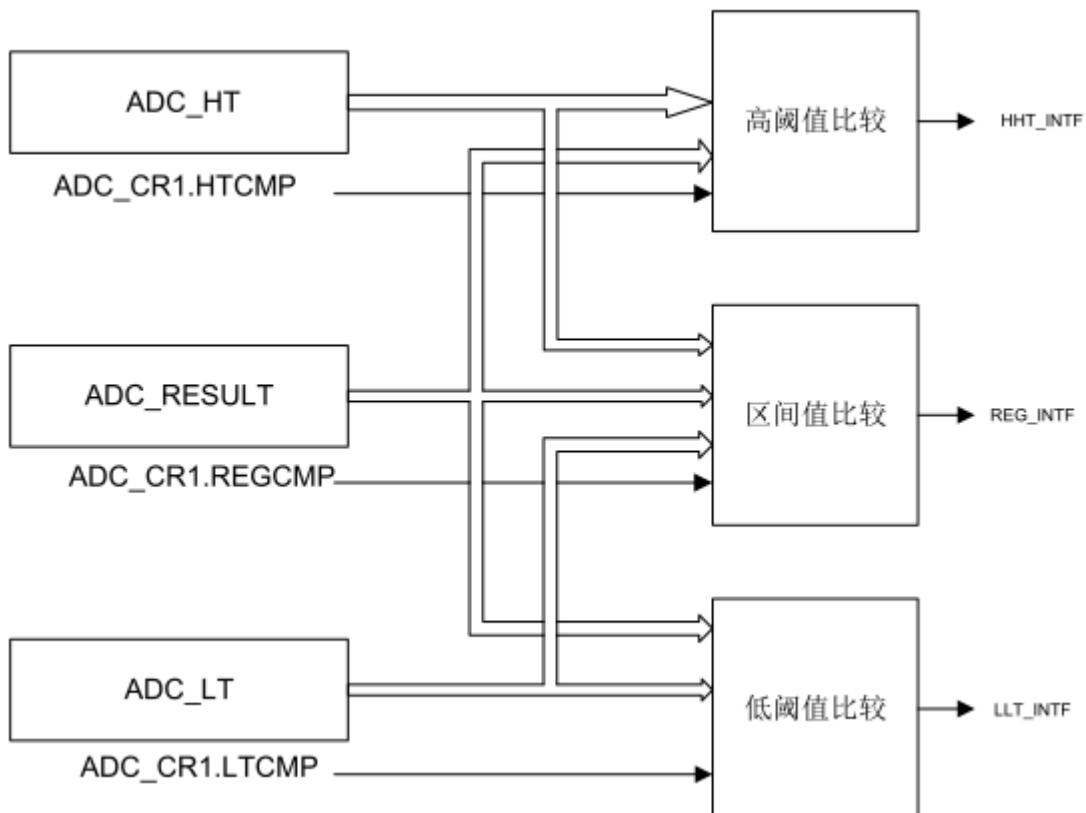
ZBT CONFIDENTIAL



## 28.7 ADC 转换结果比较

ADC 转换完成时，ADC 转换结果可以与用户设定的阈值进行比较，支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 `ADCX_CR1.HTCMP`、`ADCX_CR1.LTCMP`、`ADCX_CR1.REGCMP` 置 1。该功能可实现对模拟量的自动监测，直到 ADC 转换结果符合用户预期时才产生中断申请用户程序界入。

- 上阈值比较：当 ADC 转换结果位于 [ADCX\_LT, ADCX\_HT] 区间内，则 `ADCX_RAWINTSR.HHT_INTF` 置 1；向 `ADCX_INTCLR.HHT_INTC` 写入 1 则清零 `ADCX_RAWINTSR.HHT_INTF`。
- 下阈值比较：当 ADC 转换结果位于 [0, ADCX\_LT) 区间内，则 `ADCX_RAWINTSR.LLT_INTF` 置 1；向 `ADCX_INTCLR.LLT_INTC` 写入 1 则清零 `ADCX_RAWINTSR.LLT_INTF`。
- 区间值比较：当 ADC 转换结果位于 [ADCX\_LT, ADCX\_HT) 区间内，则 `ADCX_RAWINTSR.REG_INTF` 置 1；向 `ADCX_INTCLR.REG_INTC` 写入 1 则清零 `ADCX_RAWINTSR.REG_INTF`。





## 28.8 ADC 中断

ADC 中断请求如下表所示：

中断源	中断标志	中断使能掩码
ADC 连续转换完成	ADCX_MSKINTSR.CONT_MIF	ADCX_INTEN.CONT_IEN
ADC 转换结果位于区间值区域	ADCX_MSKINTSR.REG_MIF	ADCX_INTEN.REG_IEN
ADC 转换结果位于上阈值区域	ADCX_MSKINTSR.HHT_MIF	ADCX_INTEN.HHT_IEN
ADC 转换结果比较下阈值区域	ADCX_MSKINTSR.LLT_MIF	ADCX_INTEN.LLT_IEN

## 28.9 ADCX 寄存器列表

ADC1 基地址：0x4001 2400

ADC2 基地址：0x4001 2800

ADC3 基地址：0x4001 3C00

偏移地址	名称	描述	复位值
0x00	ADCX_CR0	ADCX 配置寄存器 0	0x0000 0000
0x04	ADCX_CR1	ADCX 配置寄存器 1	0x0000 7000
0x08	ADCX_CR2	ADCX 配置寄存器 2	0x0000 0000
0x0C	ADCX_RESULT0	ADCX 通道 0 转换结果	0x0000 0000
0x10	ADCX_RESULT1	ADCX 通道 1 转换结果	0x0000 0000
0x14	ADCX_RESULT2	ADCX 通道 2 转换结果	0x0000 0000
0x18	ADCX_RESULT3	ADCX 通道 3 转换结果	0x0000 0000
0x1C	ADCX_RESULT4	ADCX 通道 4 转换结果	0x0000 0000
0x20	ADCX_RESULT5	ADCX 通道 5 转换结果	0x0000 0000
0x24	ADCX_RESULT6	ADCX 通道 6 转换结果	0x0000 0000
0x28	ADCX_RESULT7	ADCX 通道 7 转换结果	0x0000 0000
0x2C	ADCX_RESULT	ADCX 转换结果	0x0000 0000
0x30	ADCX_RESULT_ACC	ADCX 转换结果累加值	0x0000 0000
0x34	ADCX_LT	ADCX 比较上阈值	0x0000 0FFF
0x38	ADCX_LT	ADCX 比较下阈值	0x0000 0000
0x44	ADCX_INTEN	ADCX 中断使能寄存器	0x0000 0000
0x48	ADCX_INTCLR	ADCX 中断清除寄存器	0x0000 0000
0x4C	ADCX_RAWINTSR	ADCX 掩码前中断状态寄存器	0x0000 0000
0x50	ADCX_MSKINTSR	ADCX 掩码后中断状态寄存器	0x0000 0000
0x54	ADCVTSETRIM	ADC/VTSE TRIM寄存器	0x0000 030F

Table 28-1 ADCX 寄存器列表和复位值

Note :ADCX : ADC1,ADC2,ADC3



## 28.10 寄存器说明

### 28.10-1 ADCX 配置寄存器 0(ADCX\_CR0)

地址偏移: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	SEL12V	VTSEN	CT_STATE	ADCDE	ADCK_SF[2:0]										
保留	R/W	R/W	R/W	R/W	R/W	R/W	R/W								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STERST	SAM	ADCDE	CLKDIV3	SAM	SEL[2:0]			保留	CLKSEL[2:0]		ADCDE		STAR_T	ADC EN	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:21	-	保留，始终读为 0。	0	-
22	SEL12V	ADC AIN13 转换通道选择1.2V 1: Ain13选择通道 為PLL LDO 1.2V 0: Ain13选择通道 為V12(CPU) LDO 1.2V	0	R/W
21	VTSEN	VTSE致能(温度感测模块) 0:关闭VTSE 1:致能VTSE	0	R/W
20	CT_STATE	0: 非ADC 连续转换状态 1: ADC 连续转换状态	0	RO
19	ADCDE	始能DMA请求(ADCDREQ) 0: 无DMA请求 1:始能触发DMA请求(ADCDREQ)	0	RW



位	标记	功能描述	复位值	读写
18:16	ADCK_SF[2:0]	000: ADC CLK= no shift with SYSCLK. 001: ADC CLK= shift 4ns with SYSCLK. 010: ADC CLK= shift 8ns with SYSCLK. 011: ADC CLK= shift 12ns with SYSCLK. 100: ADC CLK= no shift with SYSCLK negative edge. 101: ADC CLK= shift 4ns with SYSCLK negative edge. 110: ADC CLK= shift 8ns with SYSCLK negative edge. 111: ADC CLK= shift 12ns with SYSCLK negative edge.	0x4	R/W
15	STATERST	ADC 连续转换状态控制 0: 无效 1: 复位ADC 连续转换状态	0x0	R/W
14	SAM	ADC 采样周期选择 0: 4 个采样周期 1: 8 个采样周期	0x0	R/W
13	VREFEN	VREF(2.048V) LDO 致能 0:关闭 LDO 1: 致能 LDO	0x0	R/W
12	CLKDIV3	ADC时钟选择 0: 选择CLKSEL[2:0]的时钟 1: PCLK 时钟 3 分频	0	R/W
11:7	SEL[4:0]	ADC 转换通道选择(单次转换模式):  00000: 选择通道 AIN0 00001: 选择通道 AIN1 00010: 选择通道 AIN2 00011: 选择通道 AIN3 00100: 选择通道 AIN4 00101: 选择通道 AIN 5 00110: 选择通道 AIN 6 00111: 选择通道 AIN7 01000: 选择通道 AIN8 01001: 选择通道 AIN9	0x0	R/W



位	标记	功能描述	复位值	读写
		01010: 选择通道 AIN10 01011: 选择通道 AIN11 01100: 选择通道 AIN12 <b>01101: 选择通道 AIN 13</b> 01110: 选择通道 AIN 14 01111: 选择通道 AIN 15 10000: 选择通道 AIN16 10001: 选择通道 AIN17 10010: 选择通道 AIN18 10011: 选择通道 AIN19 10100: 选择通道 AIN10 10101: 选择通道 AIN 21 10110: 选择通道 AIN 22 10111: 选择通道 AIN 23		
6:4	CLKSEL[2:0]	ADC 时钟选择 000: PCLK 时钟 001: PCLK 时钟 2 分频 010: PCLK 时钟 4 分频 011: PCLK 时钟 8 分频 100: PCLK 时钟 16 分频 101: PCLK 时钟 32 分频 110: PCLK 时钟 64 分频 111: PCLK 时钟 128 分频	0x0	R/W
3:2	VRSEL[1:0]	ADC 参考电压选择 00: AVDD 01: VREF(1.2V) 10: EX_VREF PB1 11: VREF(1.8V)	0	R/W
1	START	ADC 转换控制 0: 停止 ADC 转换 1: 启动 ADC 转换, 需要在 <b>ADCEN=1</b> 时之后设定。 注: 该位域软件写 1, 硬件清 0。	0x0	R/W
0	ADCEN	ADC 使能控制 0: 禁止ADC.(当 <b>ADCEN</b> 清除为0时,同时也需清 <b>START bit</b> 为 0) 1: 使能ADC	0x0	R/W



### 28.10-2 ADCX 配置寄存器 1(ADCX\_CR1)

地址偏移: 0x04 复位值: 0x0000 7000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留															
RACC_CLR	REGCMP	HTCMP	LTCMP	RACC_EN	CT	TRIGS1[4:0]						TRIGS0[4:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W				R/W			

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15	RACC_CLR	ADC 转换结果累加寄存器清零 0: 无作用; 1: ADC 转换结果累加寄存器(ADCX_RESULT_ACC)清零。 注: 该 bit 读出为 0, 故操作本寄存器时需要特别注意该 bit 的值以防误动作。	0x0	R/W
14	REGCMP	ADC 区间比较控制 0: 禁止区间比较 1: 使能区间比较	0x1	R/W
13	HTCMP	ADC 高阈值比较控制 0: 禁止高阈值比较 1: 使能高阈值比较	0x1	R/W
12	LTCMP	ADC 低阈值比较控制 0: 禁止低阈值比较 1: 使能低阈值比较	0x1	R/W
11	RACC_EN	ADC 转换结果自动累加控制 0: 禁止ADC 转换结果自动累加功能 1: 使能ADC 转换结果自动累加功能	0x0	R/W
10	CT	ADC 转换模式选择 0: 单次转换模式 1: 连续转换模式	0x0	R/W
		ADC 转换自动触发选择 1: 00000: 禁用自动触发 ADC 转换		



位	标记	功能描述	复位值	读写
9:5	TRIGS1[4:0]	<p>00001: Timer10 中断, 自动触发ADC 转换 00010: Timer11 中断, 自动触发ADC 转换 00011: TIM1 中断, 自动触发ADC 转换 00100: LPTIM 中断, 自动触发ADC 转换 00101: TIM1 TRGO, 自动触发ADC 转换 00110: TIM2 TRGO, 自动触发ADC 转换 00111: TIM2 中断, 自动触发ADC 转换 01000: UART0 中断, 自动触发ADC 转换 01001: UART1 中断, 自动触发ADC 转换 01010: LPUART 中断, 自动触发ADC 转换 01011: VC0 中断, 自动触发ADC 转换 01100: VC1 中断, 自动触发ADC 转换 01101: RTC 中断, 自动触发ADC 转换 01110: PCA 中断, 自动触发ADC 转换 01111: SPI 中断, 自动触发ADC 转换 10000: PA1 中断, 自动触发ADC 转换 10001: PA2 中断, 自动触发ADC 转换 10010: PA3 中断, 自动触发ADC 转换 10011: PB4 中断, 自动触发ADC 转换 10100: PB5 中断, 自动触发ADC 转换 10101: PA5 中断, 自动触发ADC 转换 10110: PA6 中断, 自动触发ADC 转换 10111: PC5 中断, 自动触发ADC 转换 11000: PC6 中断, 自动触发ADC 转换 11001: PA7 中断, 自动触发ADC 转换 11010: PA8 中断, 自动触发ADC 转换 11011: PA9 中断, 自动触发ADC 转换 11100: PA10 中断, 自动触发ADC 转换 11101: PB00 中断, 自动触发ADC 转换 11110: PB01 中断, 自动触发ADC 转换 11111: PD00中断, 自动触发ADC 转换</p> <p>Note: 触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发, 需要清除中断标志。如果不进入中断服务程序, 请不要使能 NVIC 的中断使能</p> <p>ZBIT</p>	0x0	R/W
		ADC 转换自动触发选择 0: 00000: 禁用自动触发 ADC 转换		



位	标记	功能描述	复位值	读写
	TRIGS0[4:0]	<p>00001: Timer10 中断, 自动触发ADC 转换 00010: Timer11 中断, 自动触发ADC 转换 00011: TIM1 中断, 自动触发ADC 转换 00100: LPTIM 中断, 自动触发ADC 转换 00101: TIM1 TRGO, 自动触发ADC 转换 00110: TIM2 TRGO, 自动触发ADC 转换 00111: TIM2 中断, 自动触发ADC 转换 01000: UART0 中断, 自动触发ADC 转换 01001: UART1 中断, 自动触发ADC 转换 01010: LPUART 中断, 自动触发ADC 转换 01011: VC0 中断, 自动触发ADC 转换 01100: VC1 中断, 自动触发ADC 转换 01101: RTC 中断, 自动触发ADC 转换 01110: PCA 中断, 自动触发ADC 转换 01111: SPI 中断, 自动触发ADC 转换 10000: PA1 中断, 自动触发ADC 转换 10001: PA2 中断, 自动触发ADC 转换 10010: PA3 中断, 自动触发ADC 转换 10011: PB4 中断, 自动触发ADC 转换 10100: PB5 中断, 自动触发ADC 转换 10101: PA5 中断, 自动触发ADC 转换 10110: PA6 中断, 自动触发ADC 转换 10111: PC5 中断, 自动触发ADC 转换 11000: PC6 中断, 自动触发ADC 转换 11001: PA7 中断, 自动触发ADC 转换 11010: PA8 中断, 自动触发ADC 转换 11011: PA9 中断, 自动触发ADC 转换 11100: PA10 中断, 自动触发ADC 转换 11101: PB00 中断, 自动触发ADC 转换 11110: PB01 中断, 自动触发ADC 转换 11111: PD00中断, 自动触发ADC 转换</p> <p><b>Note:</b> 触发 ADC 使用的是各中断标志位的上升沿。如果需要重 复触发, 需要清除中断标志。 如果不希望进入中断服务程 序, 请不要使能 NVIC 的中断 使能。</p> <p style="text-align: right;">ZBIT</p> <p style="text-align: right;">CONFIDENTIAL</p>	0x0	R/W



### 28.10-3 ADCX 配置寄存器 2(ADCX\_CR2)

偏移地址 0x08 复位值 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														CHSEL[1:0]	CIRC LE_M ODE
保留														R/W	R/W

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

位	标记	功能描述	复位值	读写
31:19	-	保留, 始终读为 0。	0	-
18:17	CHSEL[1:0]	ADC 连续转换通道 选择 Group Channel 00 : AIN7 ~AIN0 ( CHEN[7:0] 使能) 01 : AIN15 ~AIN8 ( CHEN[7:0] 使能) 10 : AIN23 ~AIN16 ( CHEN[7:0] 使能)	0	R/W
16	CIRCLE_MODE	ADC 转换循环模式选择 0: 非循环模式 1: 循环模式	0x0	R/W
15:8	ADCCNT[7:0]	ADC 连续转换次数配置 0: 连续转换 1 次 1: 连续转换 2 次 ..... 255: 连续转换 256 次	0x0	R/W
7:0	CHEN[7:0]	ADC 连续转换通道 7~0 使能 0: 禁止 1: 使能	0x0	R/W



#### 28.10-4 ADCX 通道 0 转换结果(ADCX\_RESULT0)

地址偏移: 0x0C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT0											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT0[11:0]	ADC 通道 0/8/16 转换结果	0x0	RO

#### 28.10-5 ADCX 通道 1 转换结果(ADCX\_RESULT1)

地址偏移: 0x10 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT1											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT1[11:0]	ADCX 通道 1/9/17 转换结果	0x0	RO



### 28.10-6 ADCX 通道 2 转换结果(ADCX\_RESULT2)

地址偏移: 0x14 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT2											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT2[11:0]	ADCX 通道 2/10/18 转换结果	0x0	RO

### 28.10-7 ADCX 通道 3 转换结果(ADCX\_RESULT3)

地址偏移: 0x18 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT3											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT3[11:0]	ADCX 通道 3/11/19 转换结果	0x0	RO



### 28.10-8 ADCX 通道 4 转换结果(ADCX\_RESULT4)

地址偏移: 0x1C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT4											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT4[11:0]	ADCX 通道 4/12/20 转换结果	0x0	RO

### 28.10-9 ADCX 通道 5 转换结果(ADCX\_RESULT5)

地址偏移: 0x20 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT5											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT5[11:0]	ADCX 通道 5/13/21 转换结果	0x0	RO



### 28.10-10 ADCX 通道 6 转换结果(ADCX\_RESULT6)

地址偏移: 0x24 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT6											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT6[11:0]	ADCX 通道 6/14/22 转换结果	0x0	RO

### 28.10-11 ADCX 通道 7 转换结果(ADCX\_RESULT7)

地址偏移: 0x28 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT7											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT7[11:0]	ADCX 通道 7/15/23 转换结果	0x0	RO



### 28.10-12 ADCX 转换结果(ADCX\_RESULT)

地址偏移: 0x2C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT											
保留				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT[11:0]	ADCX 转换结果	0x0	RO

### 28.10-13 ADCX 转换结果累加值(ADCX\_RESULT\_ACC)

地址偏移: 0x30 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														RESULT_ACC[19:16]	
保留														RO	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT_ACC[15:0]															
RO															

位	标记	功能描述	复位值	读写
31:20	-	保留, 始终读为 0。	0	-
19:0	RESUL_ACC[19:0]	ADCX 转换结果累加值	0x0	RO



### 28.10-14 ADCX 比较上阈值(ADCX\_LT)

地址偏移: 0x34 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	HT[11:0]	ADCX 转换结果比较上阈值	0xFFFF	R/W

### 28.10-15 ADCX 比较下阈值(ADCX\_LT)

地址偏移: 0x38 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	LT[11:0]	ADCX 转换结果比较下阈值	0x0	R/W



### 28.10-16 ADCX 中断使能寄存器(ADCX\_INTEN)

地址偏移: 0x44 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCXXIEN[23:16]								ADCXXIEN[15:8]							
R/W								R/W							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CONT_IEN	REG_IEN	HHT_IEN	LLT_IEN	ADCXXIEN[7:0]							
保留				R/W								R/W			

位	标记	功能描述	复位值	读写
31:24	ADCXXIEN[23:16]	ADCX 通道 23~16 中断掩码配置(AIN23~AIN16) 0: 禁止 1: 使能	0x0	R/W
23:16	ADCXXIEN[15:8]	ADCX 通道 15~8 中断掩码配置(AIN15~AIN8) 0: 禁止 1: 使能	0x0	R/W
15:12	保留	始终读为 0。	0x0	R/W
11	CONT_IEN	连续转换完成中断掩码配置 0: 禁止中断 1: 使能中断	0x0	R/W
10	REG_IEN	ADCX 转换结果比较区间中断掩码配置 0: 禁止 1: 使能	0x0	R/W
9	HHT_IEN	ADCX 转换结果比较上阈值中断掩码配置 0: 禁止 1: 使能	0x0	R/W
8	LLT_IEN	ADCX 转换结果比较下阈值中断掩码配置 0: 禁止 1: 使能	0x0	R/W
7:0	ADCXXIEN[7:0]	ADCX 通道 7~0 中断掩码配置(AIN7~AIN0) 0: 禁止 1: 使能	0x0	R/W



### 28.10-17 ADCX 中断清除寄存器(ADCX\_INTCLR)

地址偏移: 0x48 复位值: 0x0000 0000

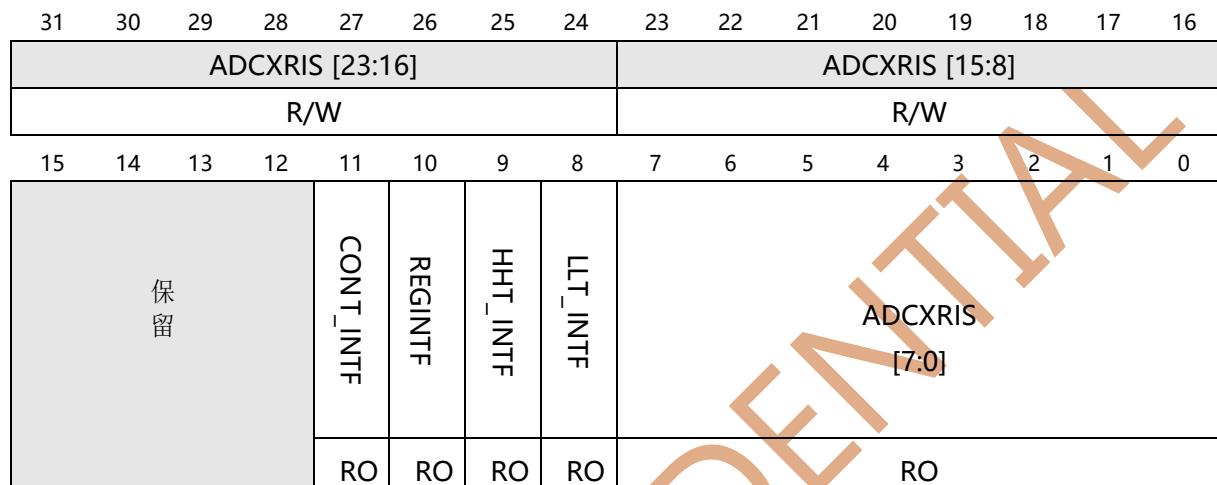
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCXICLR[23:16]								ADCXICLR[15:8]							
R/W								R/W							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CONT_INTC	REG_INTC	HHT_INTC	LLT_INTC	ADCXICL R[7:0]							
				WO	WO	WO	WO	WO							

位	标记	功能描述	复位值	读写
31:24	ADCXICLR[23:16]	写1 清除 ADCX 通道 23~16 中断状态 写0 无作用	0x0	WO
23:16	ADCXICLR[15:8]	写1 清除 ADCX 通道 15~8中断状态 写0 无作用	0x0	WO
15:12	-	保留, 始终读为 0。	0	-
11	CONT_INTC	写1 清除连续转换完成标志 写0 无作用	0x0	WO
10	REG_INTC	写1 清除 ADCX 转换结果比较区间标志 写0 无作用	0x0	WO
9	HHT_INTC	写1 清除 ADCX 转换结果比较上阈值 写0 无作用	0x0	WO
8	LLT_INTC	写1 清除 ADCX 转换结果比较下阈值标志 写0 无作用	0x0	WO
7:0	ADCXICL R[7:0]	写1 清除 ADCX 通道 7~0 中断状态 写0 无作用	0x0	WO



### 28.10-18 ADCX 掩码前中断状态寄存器(ADCX\_RAWINTSR)

地址偏移: 0x4C 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:24	ADCXRIS [23:16]	ADCX 通道 23~16 转换完成中断状态 (掩码前)	0x0	RO
23:16	ADCXRIS[15:8]	ADCX 通道 15~8 转换完成中断状态 (掩码前)	0x0	RO
15:12	-	保留, 始终读为 0。	0	
11	CONT_INTF	连续转换完成标志 0: ADCX 连续转换未完成 1: ADCX 连续转换完成	0x0	RO
10	REG_INTF	ADCX 转换结果比较区间标志 0: ADCX 转换结果位于[ADCX_LT, ADCX_HT)区间外 1: ADCX 转换结果位于[ADCX_LT, ADCX_HT)区间内	0x0	RO
9	HHT_INTF	ADCX 转换结果比较上阈值标志 0: ADCX 转换结果位于[ADCX_HT, 4095]区间外 1: ADCX 转换结果位于[ADCX_HT, 4095]区间内	0x0	RO
8	LLT_INTF	ADCX 转换结果比较下阈值标志 0: ADCX 转换结果位于[0, ADCX_LT)区间外 1: ADCX 转换结果位于[0, ADCX_LT)区间内	0x0	RO
7:0	ADCXRIS[7:0]	ADCX 通道 7~0 转换完成中断状态 (掩码前)	0x0	RO



### 28.10-19 ADCX 掩码后中断状态寄存器(ADCX\_MSKINTSR)

地址偏移: 0x50 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:24	ADCXMIS[23:16]	ADCX 通道 23~16 转换完成中断状态（掩码后）	0x0	RO
23:16	ADCXMIS[15:8]	ADCX 通道 15~8 转换完成中断状态（掩码后）	0x0	RO
15:12	-	保留，始终读为 0。	0	-
11	CONT_MIF	连续转换完成掩码后中断 0: ADCX 连续转换未完成 1: ADCX 连续转换完成	0x0	RO
10	REG_MIF	ADCX 转换结果比较区间掩码后中断 0: ADCX 转换结果位于[ADCX_LT, ADCX_HT]区间外 1: ADCX 转换结果位于[ADCX_LT, ADCX_HT]区间内	0x0	RO
9	HHT_MIF	ADCX 转换结果比较上阈值掩码后中断 0: ADCX 转换结果位于[ADCX_HT, 4095]区间外 1: ADCX 转换结果位于[ADCX_HT, 4095]区间内	0x0	RO
8	LLT_MIF	ADCX 转换结果比较下阈值掩码后中断 0: ADCX 转换结果位于[0, ADCX_LT)区间外 1: ADCX 转换结果位于[0, ADCX_LT)区间内	0x0	RO
7:0	ADCXMIS[7:0]	ADCX 通道 7~0 转换完成中断状态（掩码后）	0x0	RO



### 28.10-20 ADC/VTSE TRIM 寄存器 2(ADCVTSETRIM)

地址偏移: 0x54

复位值: 0x0000 0027

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PLL	PL	PLLSTART	PLL	PLLTRIM			
								PW	LS	UP [1:0]	RD	[2:0]			
								M	YS		Y				
								R/ W	R/ W	R/W	R/W	R/W			

位	标记	功能描述	复位值	读写
31:1 6	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:1 1	Revered	Revered	-	-
10:8	VTSETRIM[2:0] ]	内部TSE 调整 <ul style="list-style-type: none"><li>● VTSE参考电压 校准值地址: 0x1800_008E 对于裸片 (KGD):</li><li>● VTSE参考电压 校准值地址: 0x1800_008C</li></ul>	0b011	R/ W
7:5	Revered	Revered	-	-
4:0	ADCXTRIM[4: 0]	内部ADCX Vref=2.048V 调整 <ul style="list-style-type: none"><li>● ADCX参考电压 校准值地址: 0x1800_0096 对于裸片 (KGD):</li><li>● ADCX参考电压 校准值地址: 0x1800_0094</li></ul>	0b0111 1	R/ W

Note: 只有 ADCX\_UNLOCK 寄存器保护解除后，才能写该寄存器。



## 【29】低电压检测器(LVD)

### 29.1 LVD 简介

LVD 可用于监测工作电压，当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号。中断或复位信号只能被中断或复位清零信号清除。只有当中断或复位信号被清零后，才会在触发条件下，再次产生中断或复位信号。

采样滤波时钟可配置，可配置为 APB 时钟或者 SIRC。滤波计数值可配置。采样达到滤波计数值次数时结果一致输出。

3 种触发条件：高电平、上升沿、下降沿组合。

2 种触发结果：中断、复位信号(禁止在滤波时钟选择 PCLK 时选择产生复位信号)。中断与复位信号不能同时产生。

### 29.2 LVD 框图

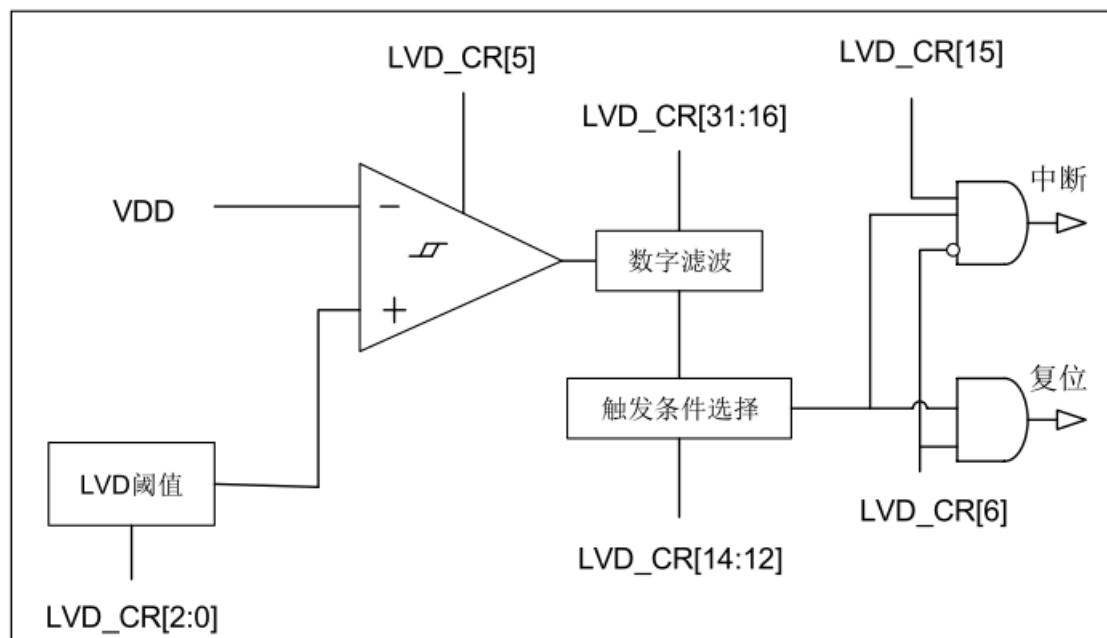


Figure 29-1 LVD 结构框图



### 29.3 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD\_CR.FLT\_NUM[15:0]设定时间的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。使能数字滤波模块，滤波示意如下所示：

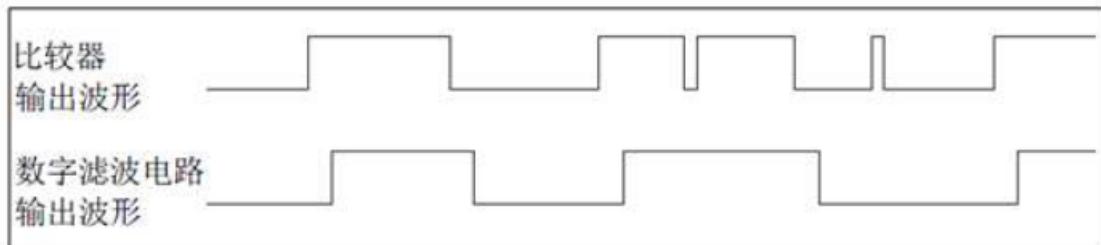


Figure 29-2 LVD 滤波输出

### 29.4 配置示例

#### 29.4-1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。配置方法如下所示：

- Step1：配置 LVD\_CR.DIV\_SEL，选择待监测的电压分压。
- Step2：配置 LVD\_CR.FLT\_NUM，选择 LVD 滤波时间。
- Step3：配置 LVD\_CR.FLTCLK\_SEL，选择滤波时钟。
- Step4：配置 LVD\_CR.FLTEN，使能 LVD 滤波。
- Step5：设置 LVD\_CR.HIGHINTEN 为 1，选择高电平触发 LVD 动作。
- Step6：设置 LVD\_CR.ACT 为 1，选择 LVD 动作为复位。
- Step7：设置 LVD\_CR.LVDEN 为 1，使能 LVD。

#### 29.4-2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。配置方法如下所示：

- Step1：配置 LVD\_CR.DIV\_SEL 选择待监测的电压来源。
- Step2：配置 LVD\_CR.FLT\_NUM，选择 LVD 滤波时间。
- Step3：配置 LVD\_CR.FLTCLK\_SEL，选择滤波时钟。
- Step4：配置 LVD\_CR.FLTEN，使能 LVD 滤波。
- Step5：设置 LVD\_CR.RISEINTEN 为 1、或 LVD\_CR.FALLINTEN 为 1、或两者都为 1，选择电平变化触发 LVD 动作。
- Step6：设置 LVD\_CR.ACT 为 0，选择 LVD 动作为中断。
- Step7：设置 LVD\_CR.INT\_EN 为 1，使能 LVD 中断。



Step8：设置 LVD\_CR.LVDEN 为 1，使能 LVD。

Step9：在中断服务程序中向 LVD\_SR.INTF 写入 0 以清除中断标志。

## 29.5 LVD 寄存器列表

基址地址：0x4001 0800

偏移地址	名称	描述	默认值
0x00	LVD_CR	LVD 控制寄存器	0x000 0007
0x04	LVD_SR	LVD 状态寄存器	0x000 0000

Table 29-1 LVD 寄存器列表



## 29.6 寄存器说明

### 29.6-1 LVD 控制寄存器(LVD\_CR)

地址偏移: 0x00 复位值: 0x0000 0007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT_NUM[15:0]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_EN	HIGHINTEN	NTEN	RISEI	保留	保留	FLTCLK SEL[1:0]	FLTEN	ACT	LVDEN	保留	保留	保留	保留	保留	DIV_SEL[2:0]
R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W						R/W

位	标记	功能描述	复位值	读写
31:16	FLT_NUM[15:0]	LVD 采样滤波计数值 采样时钟为 ABP 时钟或者 SIRC。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLT_NUM[15:0]	0x0	R/W
15	INT_EN	LVD 中断使能 0: 禁止 1: 使能	0	R/W
14	HIGHINTEN	高电平触发使能(VDD 低于阈值电压)0: 禁止 1: 使能	0	R/W
13	RISEINTEN	上升沿触发使能(VDD 从高于阈值电压变为低于阈值电压)0: 禁止 1: 使能	0	R/W
12	FALLINTEN	下降沿触发使能(VDD 从低于阈值电压变为高于阈值电压)0: 禁止 1: 使能	0	R/W
11:10	-	保留	0x0	-



位	标记	功能描述	复位值	读写
9:8	FLTCLK_SEL[1:0]	滤波时钟选择 00: 滤波时钟无效 01: 滤波时钟选择为 PCLK(只能配置成中断模式) 10: 滤波时钟选择为 SIRC(只能配置成中断模式) 11: 保留	0x0	R/W
7	FLTEN	数字滤波功能配置 0: 禁止数字滤波 1: 使能数字滤波	0	R/W
6	ACT	LVD 中断复位选择位 0: 产生中断 1: 产生复位	0	R/W
5	LVDEN	LVD 使能 0: 禁止LVD 1: 使能LVD	0	R/W
4:3	-	保留	0x0	-
2:0	DIV_SEL[2:0]	LVD 分压配置 000: 3.1V 001: 3.0V 010: 2.8V 011: 2.6V 100: 2.4V 101: 2.2V 110: 2.0V 111: 1.8V	0x7	R/W



## 29.6-2 LVD 状态寄存器(LVD\_SR)

地址偏移: 0x04 复位值: 0x0000 0000



位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	INTF	LVD 中断标志: 0: 未发生LVD 中断 1: 发生LVD 中断 写0 清除中断标志, 写1 无效。	0	W0C



## 【30】电压比较器(VC)

### 30.1 VC 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“+”输入端电压高于“-”输入端电压时，电压比较器输出高电平；当“+”输入端电压低于“-”输入端电压时，

电压比较器输出低电平。

“+”输入端与“-”输入端均支持 4 路电源输入选择。

3 种触发条件：高电平、上升沿、下降沿组合。

2 种触发结果：中断、复位信号。中断与复位信号不能同时产生。

### 30.2 VC 框图

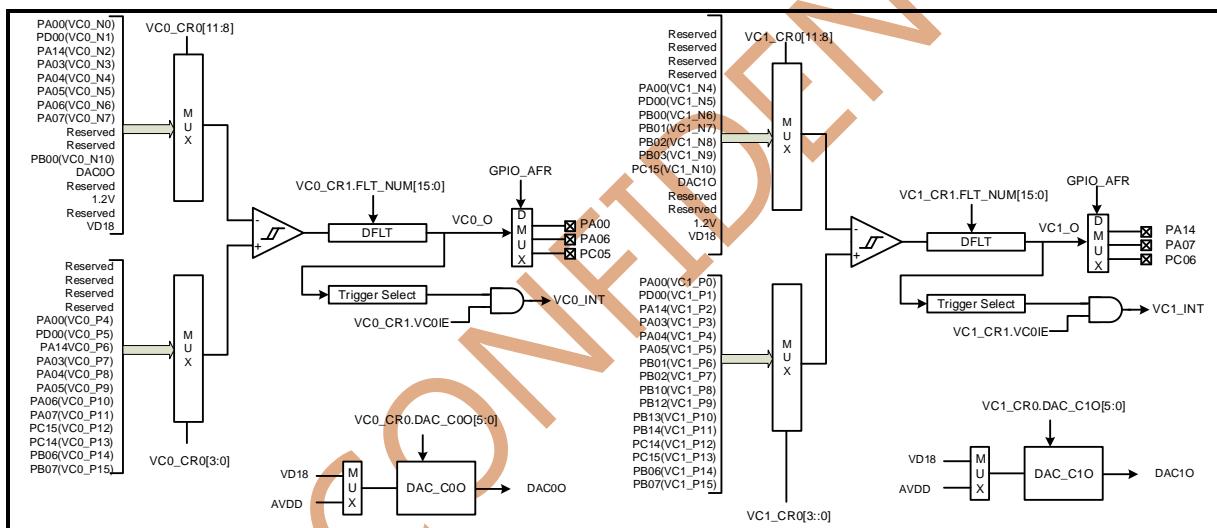


Figure 30-1 VC0,VC1 结构框图



### 30.3 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 VC\_CR1.FLT\_NUM[15:0] 设定时间的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。使能数字滤波模块，滤波示意如下所示：

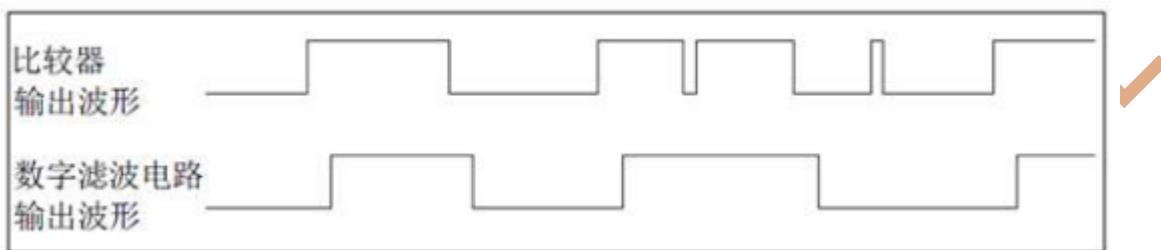


Figure 30-2 VC 滤波输出

### 30.4 配置示例

在本模式下，监测电压高于或低于阈值电压时产生中断。配置方法如下所示：

- Step1: 配置 VC\_CR0.V25DIV\_EN 使能分压。
- Step2: 配置 VC\_CR0.V25DIV，设置分压系数。
- Step3: 配置 VC\_CR0.NINSEL，选择 “-” 端待监测的电压来源。
- Step4: 配置 VC\_CR0.PINSEL，选择 “+” 端待监测的电压来源。
- Step5: 配置 VC\_CR1.FLT\_NUM[15:0]，选择 VC 滤波时间。
- Step6: 配置 VC\_CR1.FLTCLK\_SEL，选择滤波时钟。
- Step7: 配置 VC\_CR1.FLTEN，使能 VC 滤波。
- Step8: 设置 VC\_CR1 的 HIGHINTEN、RISEINTEN、FALLINTEN，选择触发模式。
- Step9: 设置 VC\_CR1.INT\_EN 为 1，使能 VC 中断。
- Step10: 设置 VC\_CR1.VCEN 为 1，使能 VC。
- Step11: 在中断服务程序中向 VC\_SR.INTF 写入 0 以清除中断标志。



### 30.5 VC 寄存器列表

VC 基地址: 0x4001 2000

偏移地址	名称	描述	默认值
0x080	VC0_CR0	VC0 控制寄存器 0	0x00000000
0x084	VC0_CR1	VC0 控制寄存器 1	0x00000000
0x088	VC0_OUTCFG	VC0 输出配置寄存器	0x00000000
0x08C	VC0_SR	VC0 状态寄存器	0x00000000
0x100	VC1_CR0	VC1 控制寄存器 0	0x00000000
0x104	VC1_CR1	VC1 控制寄存器 1	0x00000000
0x108	VC1_OUTCFG	VC1 输出配置寄存器	0x00000000
0x10C	VC1_SR	VC1 状态寄存器	0x00000000

Table 30-1 VC 寄存器列表和复位值



### 30.6 VC 寄存器说明

#### 30.6-1 VCO 电压控制寄存器(VC0\_CR0)

地址偏移: 0x80 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保 留	DAC_EN	DAC_C0O[5:0]													
保 留	R/W	r/w													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留	保 留	保 留	DACVRE	NINSEL[3:0]				保 留	保 留	保 留	保 留	PINSEL[3:0]			
保 留	保 留	保 留	R/ W	R/W				保 留	保 留	保 留	保 留	R/W			

位	标记	功能描述	复位值	读写
31:23	-	保留	0x0	-
22	VCDAC_EN	DAC0, DAC1使能控制 0: 禁止 1: 使能	0x0	R/W
21:16	DAC_C0O[5:0]	DAC_C0O 电压输出始能	0x0	R/W
15:13	-	保留	0x0	-
12	DACVREF	VC0,VC1 DAC 参考电压选择 1:AVDD 0:VCAP(2.5V)	0x0	R/W
11:8	VN0INSEL[3:0]	VCO“ -” 输入端电压电压选择。	0x0	R/W



位	标记	功能描述	复位值	读写
		VN0INSEL[3:0] VN0 input 4' b0000 PA00(VC0_N0) 4' b0001 PD00(VC0_N1) 4' b0010 PA14(VC0_N2) 4' b0011 PA03(VC0_N3) 4' b0100 PA04(VC0_N4) 4' b0101 PA05(VC0_N5) 4' b0110 PA06(VC0_N6) 4' b0111 PA07(VC0_N7) 4' b1000 Reserved 4' b1001 Reserved 4' b1010 PB00(VC0_N10) 4' b1011 DAC0O 4' b1100 Reserved 4' b1101 1.2V 4' b1110 ADCXVREF 4' b1111 VCAP		
7:4	保留	保留	0x0	
3:0	VP1INSEL[3:0]	VCO" + " 输入端电压电压 选择  VP0INSEL[3:0] VP0input 4' b0000 Reserved 4' b0001 Reserved 4' b0010 Reserved 4' b0011 Reserved 4' b0100 PA00(VC0_P4) 4' b0101 PD00(VC0_P5) 4' b0110 PA14(VC0_P6)	0x0	R/W



位	标记	功能描述			复位值	读写
		4' b0111	PA03(VC0_P7)			
		4' b1000	PA04(VC0_P8)			
		4' b1001	PA05(VC0_P9)			
		4' b1010	PA06(VC0_P10)			
		4' b1011	PA07(VC0_P11)			
		4' b1100	PC15(VC0_P12)			
		4' b1101	PC14(VC0_P13)			
		4' b1110	PB06(VC0_P14)			
		4' b1111	PB07(VC0_P15)			

ZBT CONFIDENTIAL



### 30.6-2 VCO 控制寄存器(VCO\_CR1)

地址偏移: 0x84 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT_NUM[15:0]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC0INT_EN	VC0HIGHINTEN	VC0RISEINTEN	VC0FALLINTEN	保留				VC0FLTEN	保留				VCO_FLTCLKS	VC0EN	
R/W	R/W	R/W	R/W					R/W					R/W	R/W	

位	标记	功能描述	复位值	读写
31:16	VC0FLT_NUM[15:0]	VC0 采样滤波计数值 采样时钟为 APB 时钟或者 SIRC。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLT_NUM[15:0]	0x0	R/W
15	VC0INT_EN	VC0 中断使能 0: 禁止 1: 使能	0	R/W
14	VC0HIGHINTEN	VC0 输出信号高电平触发使能 0: 禁止 1: 使能	0	R/W
13	VC0RISEINTEN	VC0 输出信号上升沿触发使能 0: 禁止 1: 使能	0	R/W
12	VC0FALLINTEN	VC0 输出信号下降沿触发使能 0: 禁止 1: 使能	0	R/W
11:9	-	保留	0x0	-
8	VC0_FLTEN	数字滤波功能配置 0: 禁止数字滤波 1: 使能数字滤波	0	R/W



7:4	-	保留	0x0	-
3:2	VC0_FLTCLK_SEL[1:0]	VC0 滤波时钟选择 00: 滤波时钟无效 01: 滤波时钟选择为 PCLK 10: 滤波时钟选择为 SIRC 11: 保留	0x0	R/W
1	VC0HY_EN	VC0 输入迟滞使能 0: 禁止 1: 使能	0	-
0	VCOEN	VC0 Enable 0: 电压比较功能禁止 1: 电压比较功能使能	0	R/W



### 30.6.3 VCO 输出配置寄存器(VC0\_OUTCFG)

地址偏移: 0x88 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	TM1 CH4_EN
													INV_PAD	TM1BKE		

保  
留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INV_TIMX
TM1CH4	INV_TM1CH4	TM1CH3	INV_TM1CH3	TM1CH2	INV_TM1CH2	TM1CH1	INV_TM1CH1	PCA_CPO	PCA_CPA	INV_PCA	MEXT_EN	LPT1_M_EN	TIM1_EN	TIM0_EN		

位	标记	功能描述	复位值	读写
31:19	-	保留	0x0	-
18	INV_PAD	VCO filter 结果输出到PAD 反向使能 0: 禁止; 1: 使能	0	R/W
17	TM1BKE	VCO 中断作为TIM1 刹车控制 0: 禁止; 1: 使能	0	R/W
16	TM1CH4_EN	VCO filter 结果输出到TIM1 CH4 门控使能 0: 禁止; 1: 使能	0	R/W
15	INV_TM1CH4	VCO filter 结果输出到TIM1 CH4 门控反向使能 0: 禁止; 1: 使能	0	R/W
14	TM1CH3_EN	VCO filter 结果输出到TIM1 CH3 门控使能 0: 禁止; 1: 使能	0	R/W
13	INV_TM1CH3	VCO filter 结果输出到TIM1 CH3 门控反向使能 0: 禁止; 1: 使能	0	R/W
12	TM1CH2_EN	VCO filter 结果输出到TIM1 CH2 门控使能 0: 禁止; 1: 使能	0	R/W
11	INV_TM1CH2	VCO filter 结果输出到TIM1 CH2 门控反向使能 0: 禁止; 1: 使能	0	R/W



位	标记	功能描述	复位值	读写
10	TM1CH1_EN	VC0 filter 结果输出到TIM1 CH1 门控使能 0: 禁止; 1: 使能	0	R/W
9	INV_TM1CH1	VC0 filter 结果输出到TIM1 CH1 门控反向使能 0: 禁止; 1: 使能	0	R/W
8	PCAECI_EN	VC0 filter 结果输出到PCA 外部时钟使能 0: 禁止; 1: 使能	0	R/W
7	PCACAP0_EN	VC0 filter 结果输出到PCA 捕获0 使能 0: 禁止; 1: 使能	0	R/W
6	INV_PCA	VC0 filter 结果输出到PCA 反向使能 0: 禁止; 1: 使能	0	R/W
5	LPTIMEXT_EN	VC0 filter 结果输出到LPTIM 外部时钟使能控制 0: 禁止; 1: 使能	0	R/W
4	LPTIM_EN	VC0 filter 结果输出到LPTIM 门控使能 0: 禁止; 1: 使能	0	R/W
3	-	保留	0	-
2	TIM1_EN	VC0 filter 结果输出到TIM11 门控使能 0: 禁止; 1: 使能	0	R/W
1	TIM0_EN	VC0 filter 结果输出到TIM10 门控使能 0: 禁止; 1: 使能	0	R/W
0	INV_TIMX	VC0 filter 结果输出到TIM10, TIM11, LPTIM 门控反向使能 0: 禁止; 1: 使能	0	R/W



### 30.6-4 VC0 状态寄存器(VC0\_SR)

地址偏移: 0x8C 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:2	-	保留	0x0	-
1	VC0_FLOUT	VC0 的 Filter 后状态 0: VC0 滤波结果为 0 1: VC0 滤波结果为 1	0	RO
0	INTF	VC0 中断标志 0: 未发生 VC 中断 1: 发生 V1C 中断 写 0 清除中断标志, 写 1 无效	0	WOC



### 30.6-5 VC1 电压控制寄存器(VC1\_CR0)

地址偏移: 0x100 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保 留	保留	DAC_C1O[5:0]													
保 留	保留	r/w													

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

位	标记	功能描述	复位值	读写														
31:23	-	保留	0x0	-														
22			0x0	R/W														
21:16	DAC_C1O[5:0]	DAC1 电压输出始能	0x0	R/W														
15:12	-	保留	0x0	-														
11:8	VN1INSEL[3:0]	VC1" -" 输入端电压电压选择。  <table border="1"><tr><td>VN1INSEL[3:0]</td><td>VN1 input</td></tr><tr><td>4' b0000</td><td>Reserved</td></tr><tr><td>4' b0001</td><td>Reserved</td></tr><tr><td>4' b0010</td><td>Reserved</td></tr><tr><td>4' b0011</td><td>Reserved</td></tr><tr><td>4' b0100</td><td>PA00(VC1_N4)</td></tr><tr><td>4' b0101</td><td>PD00(VC1_N5)</td></tr></table>	VN1INSEL[3:0]	VN1 input	4' b0000	Reserved	4' b0001	Reserved	4' b0010	Reserved	4' b0011	Reserved	4' b0100	PA00(VC1_N4)	4' b0101	PD00(VC1_N5)	0x0	R/W
VN1INSEL[3:0]	VN1 input																	
4' b0000	Reserved																	
4' b0001	Reserved																	
4' b0010	Reserved																	
4' b0011	Reserved																	
4' b0100	PA00(VC1_N4)																	
4' b0101	PD00(VC1_N5)																	



位	标记	功能描述	复位值	读写																																
		<table border="1"><tr><td>4' b0110</td><td>PB00(VC1_N6)</td></tr><tr><td>4' b0111</td><td>PB01(VC1_N7)</td></tr><tr><td>4' b1000</td><td>PB02(VC1_N8)</td></tr><tr><td>4' b1001</td><td>PB03(VC1_N9)</td></tr><tr><td>4' b1010</td><td>PC15(VC1_N10)</td></tr><tr><td>4' b1011</td><td>DAC1O</td></tr><tr><td>4' b1100</td><td>Reserved</td></tr><tr><td>4' b1101</td><td>Reserved</td></tr><tr><td>4' b1110</td><td>1.2V</td></tr><tr><td>4' b1111</td><td>VCAP</td></tr></table>	4' b0110	PB00(VC1_N6)	4' b0111	PB01(VC1_N7)	4' b1000	PB02(VC1_N8)	4' b1001	PB03(VC1_N9)	4' b1010	PC15(VC1_N10)	4' b1011	DAC1O	4' b1100	Reserved	4' b1101	Reserved	4' b1110	1.2V	4' b1111	VCAP														
4' b0110	PB00(VC1_N6)																																			
4' b0111	PB01(VC1_N7)																																			
4' b1000	PB02(VC1_N8)																																			
4' b1001	PB03(VC1_N9)																																			
4' b1010	PC15(VC1_N10)																																			
4' b1011	DAC1O																																			
4' b1100	Reserved																																			
4' b1101	Reserved																																			
4' b1110	1.2V																																			
4' b1111	VCAP																																			
7:4	保留	保留	0x0																																	
3:0	VP1INSEL[3:0]	<p>VC1" +" 输入端电压电压选择</p> <table border="1"><tr><td>VP1INSEL[3:0]</td><td>VP1 input</td></tr><tr><td>4' b0000</td><td>PA00(VC1_P0)</td></tr><tr><td>4' b0001</td><td>PD00(VC1_P1)</td></tr><tr><td>4' b0010</td><td>PA14(VC1_P2)</td></tr><tr><td>4' b0011</td><td>PA03(VC1_P3)</td></tr><tr><td>4' b0100</td><td>PA04(VC1_P4)</td></tr><tr><td>4' b0101</td><td>PA05(VC1_P5)</td></tr><tr><td>4' b0110</td><td>PB01(VC1_P6)</td></tr><tr><td>4' b0111</td><td>PB02(VC1_P7)</td></tr><tr><td>4' b1000</td><td>PB10(VC1_P8)</td></tr><tr><td>4' b1001</td><td>PB12(VC1_P9)</td></tr><tr><td>4' b1010</td><td>PB13(VC1_P10)</td></tr><tr><td>4' b1011</td><td>PB14(VC1_P11)</td></tr><tr><td>4' b1100</td><td>PC15(VC1_P12)</td></tr><tr><td>4' b1101</td><td>PC14(VC1_P13)</td></tr><tr><td>4' b1110</td><td>PB06(VC1_P14)</td></tr></table>	VP1INSEL[3:0]	VP1 input	4' b0000	PA00(VC1_P0)	4' b0001	PD00(VC1_P1)	4' b0010	PA14(VC1_P2)	4' b0011	PA03(VC1_P3)	4' b0100	PA04(VC1_P4)	4' b0101	PA05(VC1_P5)	4' b0110	PB01(VC1_P6)	4' b0111	PB02(VC1_P7)	4' b1000	PB10(VC1_P8)	4' b1001	PB12(VC1_P9)	4' b1010	PB13(VC1_P10)	4' b1011	PB14(VC1_P11)	4' b1100	PC15(VC1_P12)	4' b1101	PC14(VC1_P13)	4' b1110	PB06(VC1_P14)	0x0	R/W
VP1INSEL[3:0]	VP1 input																																			
4' b0000	PA00(VC1_P0)																																			
4' b0001	PD00(VC1_P1)																																			
4' b0010	PA14(VC1_P2)																																			
4' b0011	PA03(VC1_P3)																																			
4' b0100	PA04(VC1_P4)																																			
4' b0101	PA05(VC1_P5)																																			
4' b0110	PB01(VC1_P6)																																			
4' b0111	PB02(VC1_P7)																																			
4' b1000	PB10(VC1_P8)																																			
4' b1001	PB12(VC1_P9)																																			
4' b1010	PB13(VC1_P10)																																			
4' b1011	PB14(VC1_P11)																																			
4' b1100	PC15(VC1_P12)																																			
4' b1101	PC14(VC1_P13)																																			
4' b1110	PB06(VC1_P14)																																			



位	标记	功能描述		复位值	读写
		4' b1111	PB07(VC1_P15)		

ZBT CONFIDENTIAL



### 30.6.6 VC1 控制寄存器(VC1\_CR1)

地址偏移: 0x104 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT_NUM[15:0]															
R/W															

15	I <sup>4</sup>	I <sup>3</sup>	I <sup>2</sup>	I <sup>1</sup>	I <sup>10</sup>	I <sup>9</sup>	I <sup>8</sup>	I <sup>7</sup>	I <sup>6</sup>	I <sup>5</sup>	I <sup>4</sup>	I <sup>3</sup>	I <sup>2</sup>	I <sup>1</sup>	I <sup>0</sup>
	VC1HIGHINTEN	VC1RISEINTEN	VC1FALLINTEN		Revered				Revered				VC1_FLTCLK_S	VC1INT_EN	
R/ W	R/ W	R/ W	R/ W		R/W				R/W				R/ W		

位	标记	功能描述	复位值	读写
31:16	VC1FLT_NUM[15:0]	VC1 采样滤波计数值 采样时钟为 APB 时钟或者 SIRC。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLT_NUM[15:0]	0x0	R/W
15	VC1INT_EN	VC1 中断使能 0: 禁止 1: 使能	0	R/W
14	VC1HIGHINTEN	VC1 输出信号高电平触发使能 0: 禁止 1: 使能	0	R/W
13	VC1RISEINTEN	VC1 输出信号上升沿触发使能 0: 禁止 1: 使能	0	R/W
12	VC1FALLINTEN	VC1 输出信号下降沿触发使能 0: 禁止 1: 使能	0	R/W
11:9	-	保留	0x0	-
8	VC1_FLTEN	VC1 数字滤波功能配置 0: 禁止数字滤波 1: 使能数字滤波	0	R/W
7:4	-	保留	0x0	-



3:2	VC1_FLTCLK_SEL[1:0]	VC1 滤波时钟选择 00: 滤波时钟无效 01: 滤波时钟选择为 PCLK 10: 滤波时钟选择为 SIRC 11: 保留	0x0	R/W
1	VC1HY_EN	VC1 输入迟滞使能 0: 禁止 1: 使能	0	-
0	VC1EN	VC1 Enable 0: 电压比较功能禁止 1: 电压比较功能使能	0	R/W

### 30.6.7 VC1 输出配置寄存器(VC1\_OUTCFG)

地址偏移: 0x108 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													INV_PAD	TM1A_CH4_EN	TM1A_BKE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INV_TM1ACH4	TM1ACH3_EN	INV_TM1ACH3	TM1ACH2_EN	INV_TM1ACH2	TM1ACH1_EN	INV_TM1ACH1									保留



位	标记	功能描述	复位值	读写
31:19	-	保留	0x0	-
18	VC1INV_PAD	VC1 filter 结果输出到PAD 反向使能 0: 禁止; 1: 使能	0	R/W
17	VC1TM1ABKE	VC1 中断作为TIM1A 刹车控制 0: 禁止; 1: 使能	0	R/W
16	VC1TM1ACH4_EN	VC1 filter 结果输出到TIM1A CH4 门控使能 0: 禁止; 1: 使能	0	R/W
15	VC1INV_TM1ACH4	VC1 filter 结果输出到TIM1A CH4 门控反向使能 0: 禁止; 1: 使能	0	R/W
14	VC1TM1ACH3_EN	VC1 filter 结果输出到TIM1A CH3 门控使能 0: 禁止; 1: 使能	0	R/W
13	VC1INV_TM1ACH3	VC1 filter 结果输出到TIM1A CH3 门控反向使能 0: 禁止; 1: 使能	0	R/W
12	VC1TM1ACH2_EN	VC1 filter 结果输出到TIM1A CH2 门控使能 0: 禁止; 1: 使能	0	R/W
11	VC1INV_TM1ACH2	VC1 filter 结果输出到TIM1A CH2 门控反向使能 0: 禁止; 1: 使能	0	R/W
10	VC1TM1ACH1_EN	VC1 filter 结果输出到TIM1A CH1 门控使能 0: 禁止; 1: 使能	0	R/W
9	VC1INV_TM1ACH1	VC1 filter 结果输出到TIM1A CH1 门控反向使能 0: 禁止; 1: 使能	0	R/W
8:0	-	保留	0	R/W



### 30.6-8 VC1 状态寄存器(VC1\_SR)

地址偏移: 0x10C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
														VC_FLOUT	INTF
														RO	WOC

位	标记	功能描述	复位值	读写
31:2	-	保留	0x0	-
1	VC1_FLOUT	VC1 的 Filter 后状态 0: VC1 滤波结果为 0 1: VC1 滤波结果为 1	0	RO
0	VC1INTF	VC1 中断标志 0: 未发生 VC1 中断 1: 发生 VC1 中断 写 0 清除中断标志, 写 1 无效	0	WOC



## 【31】运算放大器（OPA）

### 31.1 OPA 简介

OPA 模块可以由用户弹性应用配置，适用于简易滤波器和 Buffer 应用。内部的两个运放可以配置为反向、同向具有不同增益的组合运放，也可以使用外部电阻进行级联。OPA 的输入范围是 0V 到 AVDD，输出范围是 0.1V 到 AVDD-0.1V。.

### 31.2 OPA 特性

- (3) 两个独立配置运放
- (4) OPA 的输入范围是 0 到 AVDD-1V，输出范围是 0.1 到 AVDD -0.1 可编程增益
- (5) OPA 通过外部电阻连接可配置为仪表放大器
- (6) 可配置为以下模式
  - 通用运放模式 (general purpose OPA)
  - 电压跟随器
  - 反相,输入 PGA
  - 同相输入 PGA
  - 级联反相同相 PGA
  - 可使用软件 Real Time OPA offset Cancel.
  - 互补输入 PGA (提供 DAC 的参考电圧,作 OPA offset Cancel)



### 31.3 OPA 框图

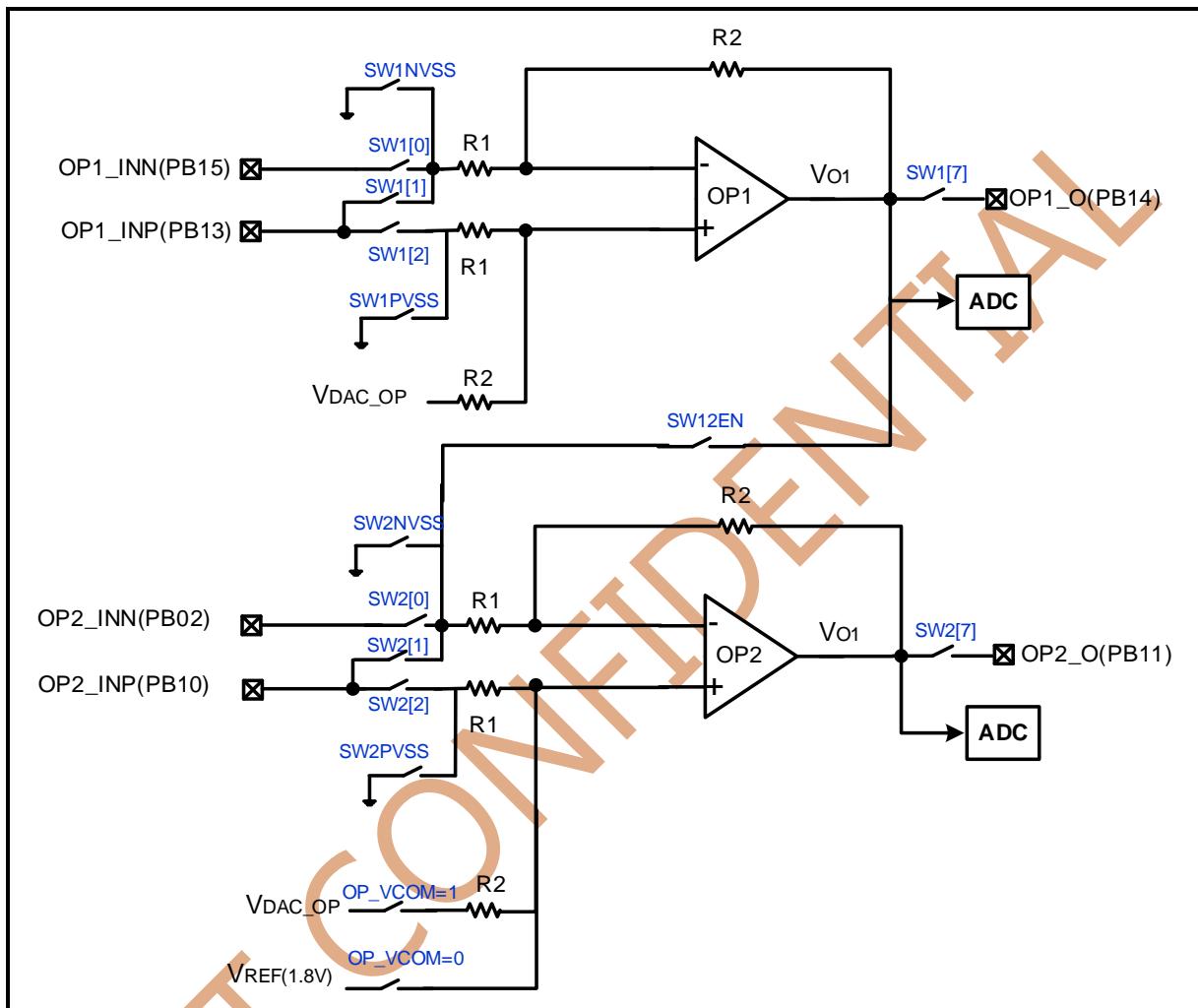


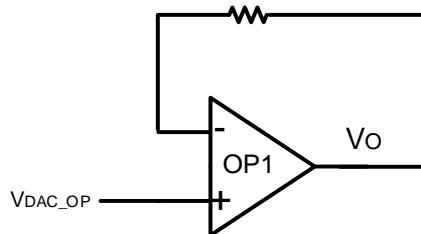
Figure 31-1 OPA1,OPA2 结构框图



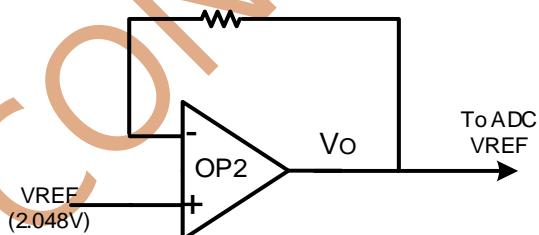
### 31.4 PGA 增益

OPA 放大倍率 PGA 的增益  $R_2/R_1=16$

#### 31.4-1 单位增益 PGA



OP2 Switch	设定值
SW1[0]	0
SW1[1]	0
SW1[2]	0
SW1[7]	0
SW1PVSS, SW1NVSS	0
SW2EN	0

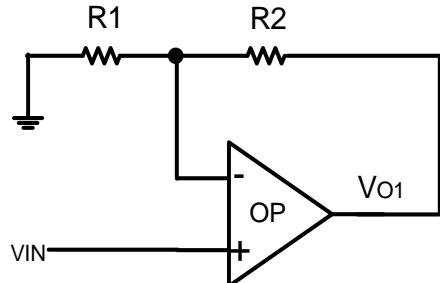


OP2 Switch	设定值
SW2[0]	0
SW2[1]	0
SW2[2]	0
SW2[7]	0
SW2PVSS, SW2NVSS	0
OP2_VCOMS	1
SW2EN	0

Note : 1 : short , 0 : Open



### 31.4-2 正向输入 PGA



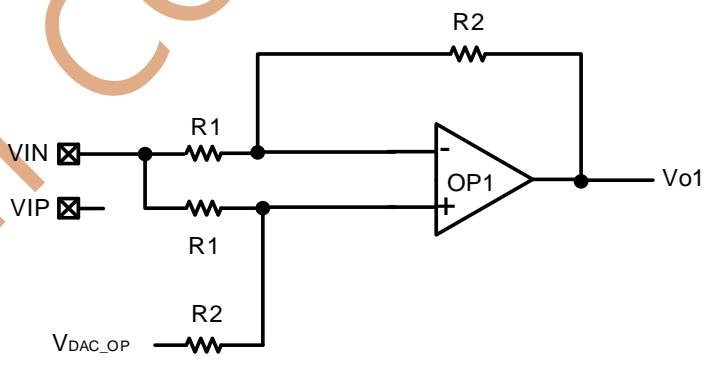
$$Vo1 = VIN * (1 + GAIN) \rightarrow GAIN = R2/R1 = 16$$

OP1 Switch	设定值
SW1[0]	0
SW1[1]	0
SW1[2]	0
SW1[7]	0
SW1PVSS	1
SW12EN	0

Note : 1 : short , 0 : Open

### 31.4-3 互补输入 PGA

Step 1 : OP1 Vos Cancelling



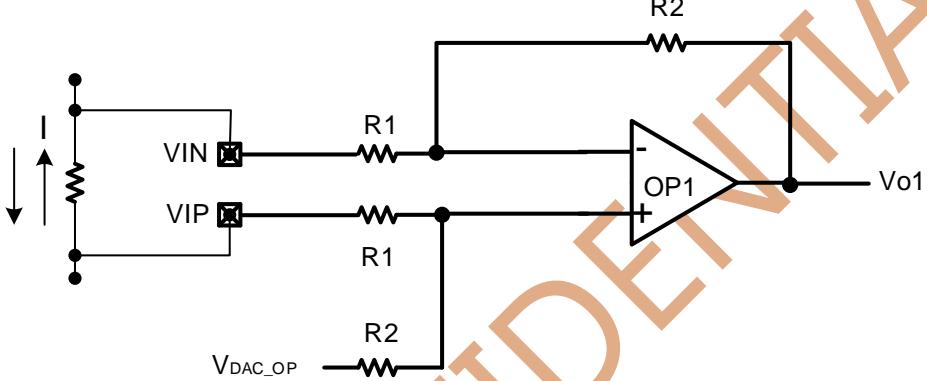
OP1 Switch	设定值
SW1[0]	1
SW1[1]	1
SW1[2]	0



SW1[7]	0
SW1PVSS	0
SW12EN	0

→ ADCX 采样 Vo1 的电压值 : Vcom\_ADCX

Step 1 : 量测(VIP-VIN)



$$VO_1 = V_{DAC\_OP} + GAIN2 * (VIP - VIN) \rightarrow GAIN2 = R_2 / R_1 = 16$$

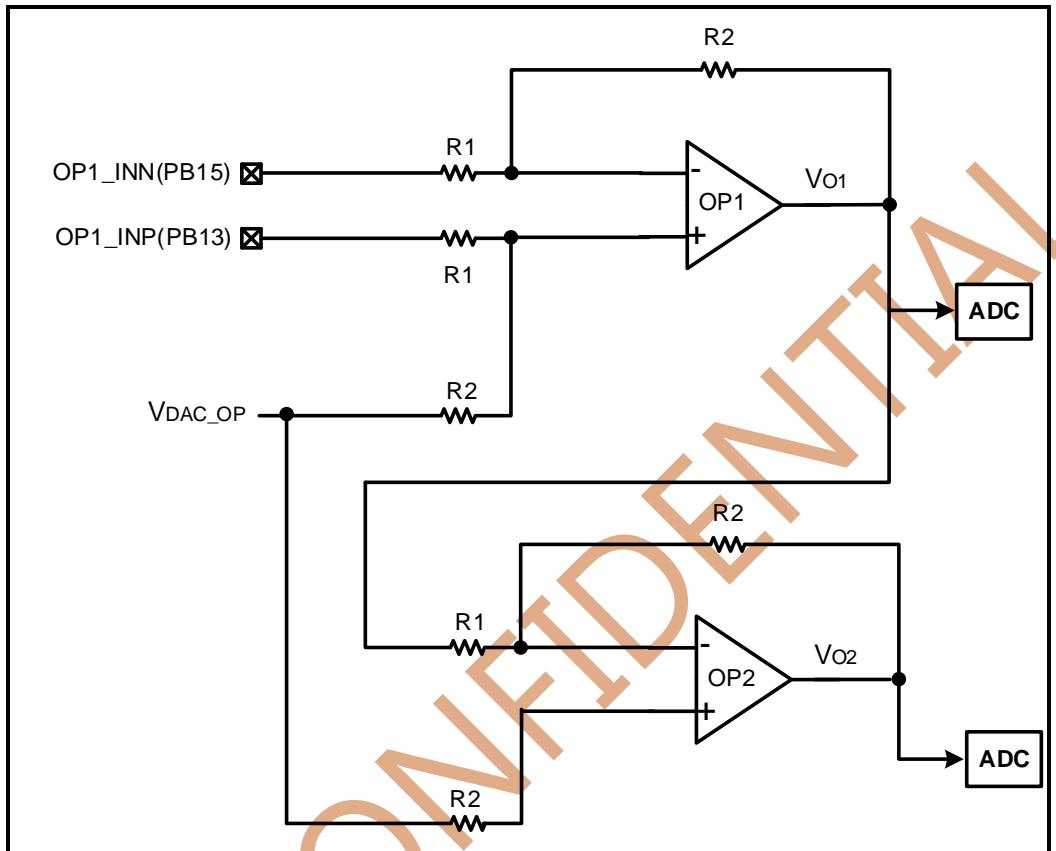
OP1 Switch	设定值
SW1[0]	1
SW1[1]	0
SW1[2]	1
SW1[7]	0
SW1VSS	0
SW12EN	0

→ ADCX 采样  $Vo_1$  的电压值 :  $Vo\_ADCX$

→  $Vo_1\_ADCX = Vo\_ADCX - Vcom\_ADCX$  (正值 :  $VIP > VIN$ , 负值 :  $VIP < VIN$ )



### 31.4-4 串級 PGA



$$V_{O2} = V_{DAC\_OP} + GAIN2 * (OP1\_INP - OP1\_INN) \rightarrow GAIN2 = 256$$

OP1 Switch	设定值
SW1[0]	1
SW1[1]	0
SW1[2]	1
SW1[7]	0
SW1PVSS, SW1NVSS	0
SW12EN	1
SW2[0]	0
SW2[1]	0
SW2[2]	0
SW2[3]	0
SW2[7]	0
SW2PVSS, SW2NVSS	0
OP_VCOM	1



### 31.5 OPA 寄存器列表

OPA 基地址: 0x4000 5400

偏移地址	名称	描述	默认值
0x000	OP1_CR0	OP1 控制寄存器 0	0x00000000
0x004	OP2_CR0	OP2 控制寄存器 0	0x00000000
0x008	OPDAC_CR	OP1,OP2的 DAC 控制寄存器	0x00000000

Table 31-1 OPA 寄存器列表和复位值



### 31.6 寄存器说明

#### 31.6-1 OP1 控制寄存器 0 (OP1\_CR0)

地址偏移: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SW12EN															SW1[7:0]
R/W															R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SW1EN														
保留	R/W														

位	标记	功能描述	复位值	读写
31	SW12EN	OP1 输出串接 OP2 输入 0: 禁止 1: 使能	0	R/W
30:24	-	Reserved	0	R/W
23:16	SW1[7:0]	OP1 选择配置 0: Open 1: Short	8' h00	R/W
15:6	-	Reserved	0	-
5:3	-	Reserved	3' b000	R/W
2	SW1NVSS	OP1 VN short to VSS 0: Open 1: Short	0	R/W
1	SW1PVSS	OP1 VP short to VSS 0: Open 1: Short	0	R/W
0	OP1EN	OP1使能控制 0: 禁止 1: 使能	0	R/W



### 31.6-2 OP2 控制寄存器 0 (OP2\_CR0)

地址偏移: 0x04 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	SW2[7:0]														
保留	R/W														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	Op2_VCO	保留	保留	保留	SW2NVSS	SW2PVSS	OP2EN								
保留	R/W	保留	保留	保留	R/W	R/W	R/W								

位	标记	功能描述	复位值	读写
31:24	-	Reserved	0	R/W
23:16	SW2[7:0]	OP2 选择配置 0: Open 1: Short	8' h00	R/W
15:7	--	Reserved	0	R/W
6	OP2_VCOMS	OP2 VCOM selection 0: 選擇 V18 電壓 1: 選擇 DAC output	0	R/W
5:3	--	Reserved	0	R/W
2	SW2NVSS	OP2 VN short to VSS 0: Open 1: Short	0	R/W
1	SW2PVSS	OP2 VP short to VSS 0: Open 1: Short	0	R/W
0	OP2EN	OP2使能控制 0: 禁止 1: 使能	0	R/W



### 31.6.3 OPDAC 控制寄存器 (OPDAC\_CR)

地址偏移: 0x08 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	OPDAC_VREF	OPDACBUF	OPDAC_EN	保留	保留	OPDAC[5:0]					
保留	保留	保留	保留	保留	R/W	R/W	R/W	保留	保留	R/W					

位	标记	功能描述	复位值	读写
31:11	-	Reserved	0	-
10	OPDAC_VREF	OPA DAC 参考电压选择 1:AVDD 0:V18	0	R/W
9	OPDAC_BUF_EN	OPA DAC Buffer 控制 0: 禁止 1: 始能	0	R/W
8	OPDAC_EN	OPA DAC 控制 0: 禁止 1: 始能	0	R/W
7:6	--	Reserved	0	-
5:0	OPDAC[5:0]	OPA DAC ( $V_{DAC\_OP}$ ) 输出电压选择: $V_{DAC\_OP}$ 可以使用 ADCX 的 AIN12 读出 电压值。 5' h0 : $V_{DAC\_OP} = 0$ 5' h1 : $V_{DAC\_OP} = VDD * 1/32$ 5' h2 : $V_{DAC\_OP} = VDD * 2/32$ : .5' h31 : $V_{DAC\_OP} = VDD * 31/32$	0	R/W



## 【32】PDMA 控制器

### 32.1 PDMA 概述

直接存储器存取(PDMA)用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输；传输过程中无须 CPU 任何干预，这有利于降低 CPU 的能耗和工作负载，并使系统能够保持低能耗模式，通过 DMA 数据可以快速地移动，此时 CPU 可同步进行其它操作。

### 32.2 PDMA 主要特性

- PDMA1,2 有 16 个通道与对应的请求。
  - 硬件请求：来自外设 的请求
  - 软件请求：通道对应的软件设置请求.
- 每个独立可配置的信道包括：
  - 传输模式
  - 优先等级设定(仲裁器来协调 DMA 请求的优先权)
  - 数据计数
  - 数据传输宽度：8-bit、16-bit、32-bit
  - 3 个事件标志: (1) DMA 传输中(2) DMA 传输完成 (3)DMA 传输出错
- 3 种传输型态：
  - 存储器到存储器： RAM/Flash to RAM
  - 存储器(RAM/Flash)到外设
  - 外设到存储器(RAM)
- 每个通道都有 1 个(主要[primary]) or 2 个(主要[primary], 备用 [alternate]) 描述元
- 闪存、SRAM、外设的 APB1 APB2 和 AHB 外设均可作为访问的源和目标。
- DMA 传输周期类型(CHNL\_CFG[2:0]):
  - 基本模式(Basic):个别使用主要(primary)或备用 (alternate) DMA 描述元, 适合外设使用.
  - 自动请求(Auto-request) : 这种模式对存储器到存储器复制大量应用非常有用, 不适合外设使用
  - 兵兵模式(ping-pong): 在主要或备用 DMA 描述元之间切换, 用于进出外设的连续数据流.
  - 离散-聚合模式(Scatter-gather) : 使用主要(primary) 描述元的数据结构复制到备用 (alternate) 描述元的数据结构.可分为 Peripheral or Memory Scatter-gather Mode
- 单个 DMA 周期中的传输次数可以编程为 1 到 1024
- DMA 通道可以由以下任一来外设的硬件请求
  - 通信串口: QSPI,SPI/I2S,I2C,USART,UART,SDIO
  - 定时器/计数器: TIM1,2,3,4,5,6,7,8,12 and LPTIM
  - 模拟模块: ADCX.DAC



### 32.3 功能框图

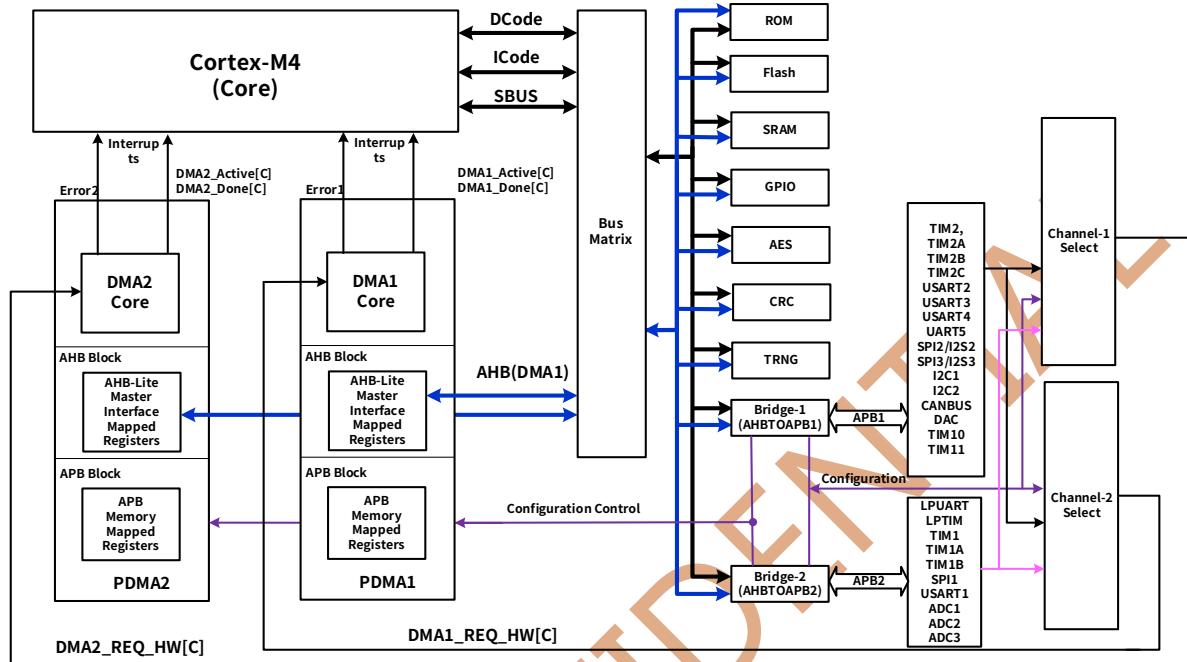


Figure 32-1 PDMA 框图

Note : DMA1\_Active[C]/DMA1\_Done[C] 和 DMA2\_Active[C]/DMA2\_Done[C] 其中 C 为通道数 =0~15

- PDMA1,PDMA2 : F103 PDMA1,2 各具有 16 条 DMA 信道，当信道之间发生冲突时，由优先级控制器进行仲裁
- Bus Matrix : CPU 和 DMAC 均连接到总线矩阵上。当 CPU 与 PDMAC 访问不同的 AHB 总线设备或 AHB 桥时，数据传输可以同时进行。当 CPU 与 DMAC 访问同一总线设备或 AHB 桥时，CPU 的优先级高于 PDMA。即只有 CPU 释放 AHB 总线设备或 AHB 桥时，PDMAC 才可访问该 AHB 总线设备或 AHB 桥。
- DMA requests : 如果外设支持硬件 DMA request，则该 DMA 信道可以配置为硬件触发否则只能配置为软件触发。



### 32.4 功能描述

PDMA 控制器和 Cortex®-M3 核心共享系统数据总线，执行直接存储器数据传输。当 CPU 和 PDMA

同时访问相同的目标(RAM 或外设)时，PDMA 请求会暂停 CPU 访问系统总线达若干个周期，总线仲裁器执行循环调度，以保证 CPU 至少可以得到一半的系统总线(存储器或外设)带宽。

PDMA 控制器非常灵活。它能够在外设和存储器之间传输数据，而无需 CPU 内核的参与.在传输大量数据时,利用 PDMA 的方式可大大提升系统的效能并将 CPU 停止可避免中断干扰.

PDMA 控制器包含 5 个独立的通道。这些通道中的每一个都可以连接到任何,通过写入配置寄存器，可用的外设触发源，请参见第 34.4.1 节.此外，每个通道都可以由软件触发（用于大内存传输或调试目的）。

PDMA 控制器应该做什么（当 其中的一个通道被触发时）是通过信道来配置的驻留在系统内存中的描述元。因此，在启用通道之前，软件必须小心地将此配置写入内存。当一个通道被触发时，PDMA 控制器将首先读取从系统内存中通道描述元，然后它会继续执行的内存传输描述元指定。描述符包含要读取的内存的内存地址、要写入地址、要传输的字节数等。通道描述元中有描述在第 33.4.3 节详细。

除了基本的传输模式，DMA 控制器还支持两种高级的传输模式：Ping-pong 和 Scatter-gather。Ping-pong 传输非常适合用于高速流数据，外设通信的因为 DMA 将准备好立即取回下一个传入的数据字节，而处理器内核仍在处理之前的数据字节（对于传出通信也是如此）。离散-聚合模式(Scatter-gather) 如在处理相关从内存执行一系列任务，而且为较复杂的方案可通过软件实现。

PDMA 控制器重新仲裁之前可使用不同通道的优先级以及 DMA 的字节数的设定，这样在时序要求较严格的传输就可确保按时提供字节完成传输。

PDMA 控制器使用 ARM 授权的 PL230 μDMA 控制器设计的.[1\*]  
.Note: [1\*] ARM PL230 homepage : [Link Path](#)



### 32.4.1 通道设置

通道选择模块允许选择哪个外设的请求线 (DMA\_HWREQ) 连接到每个 DMA 通道与对应的软件 请求。

请参考

外设请求 <b>CHSELREQ[C]</b>	外设请求 <b>CHSELREQ[C]</b>	通道(主要)	通道(备用)	DMA Request	
		[C=0,1,2..]	[C=0,1,2..]	<b>HWREQ</b>	<b>SWREQ</b>
<b>ADC1</b>	<b>TIM1</b>	0	0	V	V
<b>QSPI-RX(XIP)</b>	<b>TIM1B</b>	1	1		
<b>SPI1-TX</b>	<b>TIM1A</b>	2	2	V	V
<b>SPI1_RX</b>	<b>TIM2</b>	3	3	V	V
<b>USART2-RX</b>	<b>TIM2A</b>	4	4	V	V
<b>USART2-TX</b>	<b>TIM2B</b>	5	5	V	V
<b>SPI/I2S2-RX</b>	<b>TIM2C</b>	6	6	V	V
<b>SPI/I2S2-TX</b>	<b>LPTIM</b>	7	7	V	V
<b>USART3-RX</b>	<b>TIM10</b>	8	8	V	V
<b>USART3-TX</b>	<b>TIM11</b>	9	9	V	V
<b>UART4-RX</b>	<b>TIM2</b>	10	10	V	V
<b>UART4-TX</b>	<b>TIM2A</b>	11	11	V	V
<b>I2C1-RX</b>	<b>TIM2B</b>	12	12	V	V
<b>I2C1-TX</b>	<b>TIM2C</b>	13	13	V	V
<b>I2C2-RX</b>	<b>TIM10</b>	14	14	V	V
<b>I2C2-TX</b>	<b>TIM11</b>	15	15	V	V

Table 32-1 DMA1 使用的通道与请求列表



外设请求 CHSELREQ[C]	CHSELREQ[C] =0	通道(主要) [C=0,1,2..]		通道(备用) [C=0,1,2..]		DMA Request	
		HWREQ	SWREQ	HWREQ	SWREQ	HWREQ	SWREQ
ADC2	TIM1	0	0	V	V		
QSPI-TX(XIP)	TIM1B	1	1				
USB-DEV(H)	TIM1A	2	2	V	V		
SDIO	TIM2	3	3	V	V	V	V
QSPI-RX(EX)	TIM2A	4	4	V	V	V	V
QSPI-TX(EX)	TIM2B	5	5	V	V	V	V
SPI/I2S3-RX	TIM2C	6	6	V	V	V	V
SPI/I2S3-TX	LPTIM	7	7	V	V	V	V
USART1-RX	TIM10	8	8	V	V	V	V
USART1-TX	TIM11	9	9	V	V	V	V
DAC1	TIM2	10	10	V	V	V	V
DAC2	TRNG	11	11	V	V	V	V
LPUART-RX	TIM2B	12	12	V	V	V	V
LPUART-TX	TIM2C	13	13	V	V	V	V
AES-TX	TIM10	14	14	V	V	V	V
AES_RX	TIM11	15	15	V	V	V	V

Table 32-2 DMA2 使用的通道与请求列表

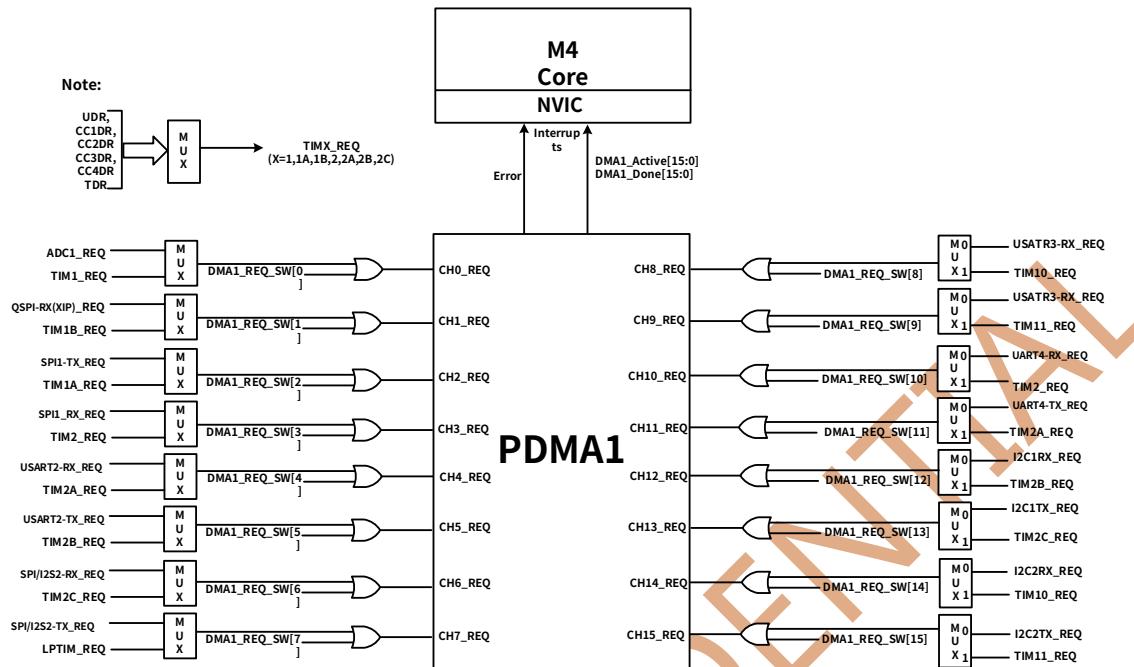


Figure 32-2 DMA1 通道请求框图

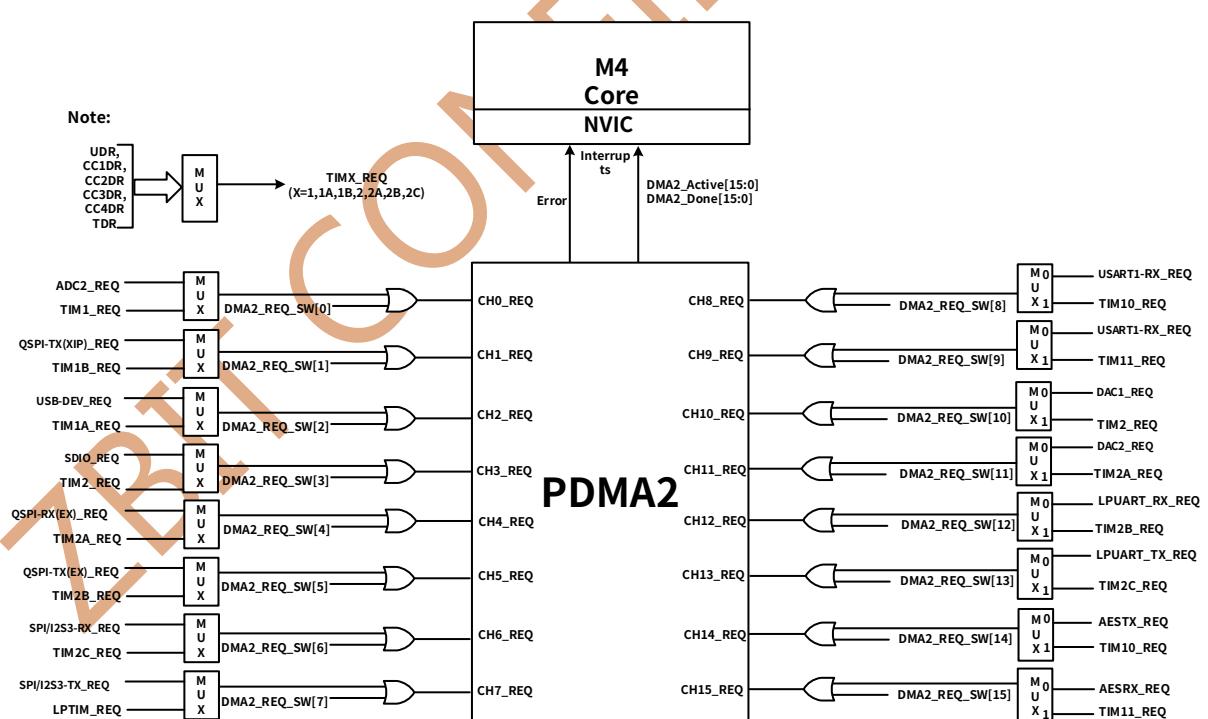


Figure 32-3 DMA2 通道请求框图



### 32.4-2 PDMA 仲裁率

您可以配置控制器在 DMA 传输期间进行仲裁的时间。这可使已致能的更高优先级的信道减少信道时间延迟。控制器提供四个位，用于配置在重新仲裁之前发生的 AHB 总线传输次数。

这些位被称为 **R\_power** 位，因为您输入的值 R 是 2 的幂，这决定了仲裁率，例如，如果  $R = 3$ ，则仲裁率为  $2^3$ ，即控制器每 8 次 DMA 传输进行一次仲裁。

R_power	重新仲裁的 DMA 传输次数
4' b0000	1
4' b0001	2
4' b0010	4
4' b0011	8
4' b0100	16
4' b0101	32
4' b0110	64
4' b0110	128
.	.
.	.
.	.
4' b1111	1024

Table 32-3 AHB bus transfer arbitration interval

需要完成的 dma 传输次数 N 由用户指定，当  $N > 2^R$  且不是  $2^R$  的整数倍时，控制器总是执行  $2^R$  次传输的序列直到  $N < 2^R$  仍有待传输，控制器在 DMA 周期结束时执行剩余的 N 次传输。

### 32.4-3 通道优先级

一个通道的优先性由其编号和优先级决定。各通道可以有两个优先级：默认或高。具有高优先级的所有通道优先于具有默认优先级的所有通道。优先级相同时，编号较低的通道优先于编号较高的通道。

通过写入 DMAPRISET 寄存器中的相应位，可以更改 DMA 通道优先级。

当控制器进行仲裁时，它通过使用以下确定下一个要服务的信道信息：

- 通道号
- 分配给通道的优先级，默认(Default)或高(High)。

可以通过设置 DMA\_CHPRIS 寄存器将每个信道配置为使用默认优先级或高优先级。通道编号 0 具有最高优先级，随着通道编号的增加，通道的优先级降低。



按优先级降序列出了 DMA 通道优先级如下表:

通道号	优先级设定	通道优先级降序
0	High	最高等级的 DMA 信道
1	High	-
.	High	
.	High	
14	High	-
15	High	-
0	Default	-
1	Default	-
.	Default	
.	Default	
.	Default	
14	Default	-
15	Default	最低等级的 DMA 信道

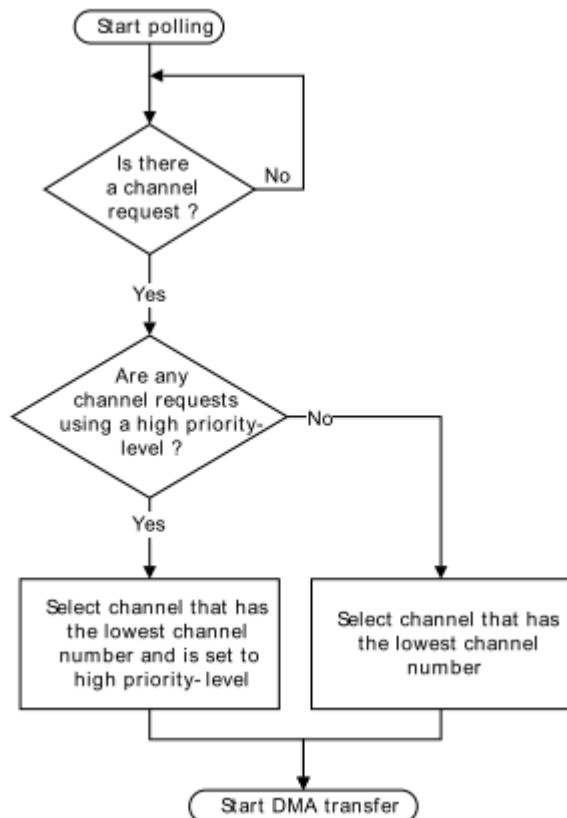


Figure 32-4 Polling flowchart



### 32.4-4 DMA 通道控制数据结构

各通道有两个与之相关的控制数据结构：主要数据结构和备选数据结构。对于简单传输模式，DMA 控制器既可使用主要数据结构，也可使用备选数据结构。对于较复杂的数据传输模式，如乒乓式或分散/聚集式，DMA 控制器同时使用主要数据结构和备选数据结构。各控制数据结构(主要或备选)在存储器中占用 4 个 32 位位置，如表 Table33-3 所示。整个通道控制数据结构如 Table 33-4 所示。

Table 32-4 通道控制数据结构表

偏移	名称	描述
0x00	SRC_END_PTR[31:0]	来源端指针
0x04	DST_END_PTR[31:0]	目标端指针
0x08	CHNL_CFG[31:0]	控制数据配置
0x0C	保留	保留

在控制器能够执行 DMA 传输之前，必须在系统存储器 SRAM 的指定位置设置 DMA 信道相关的数据结构。

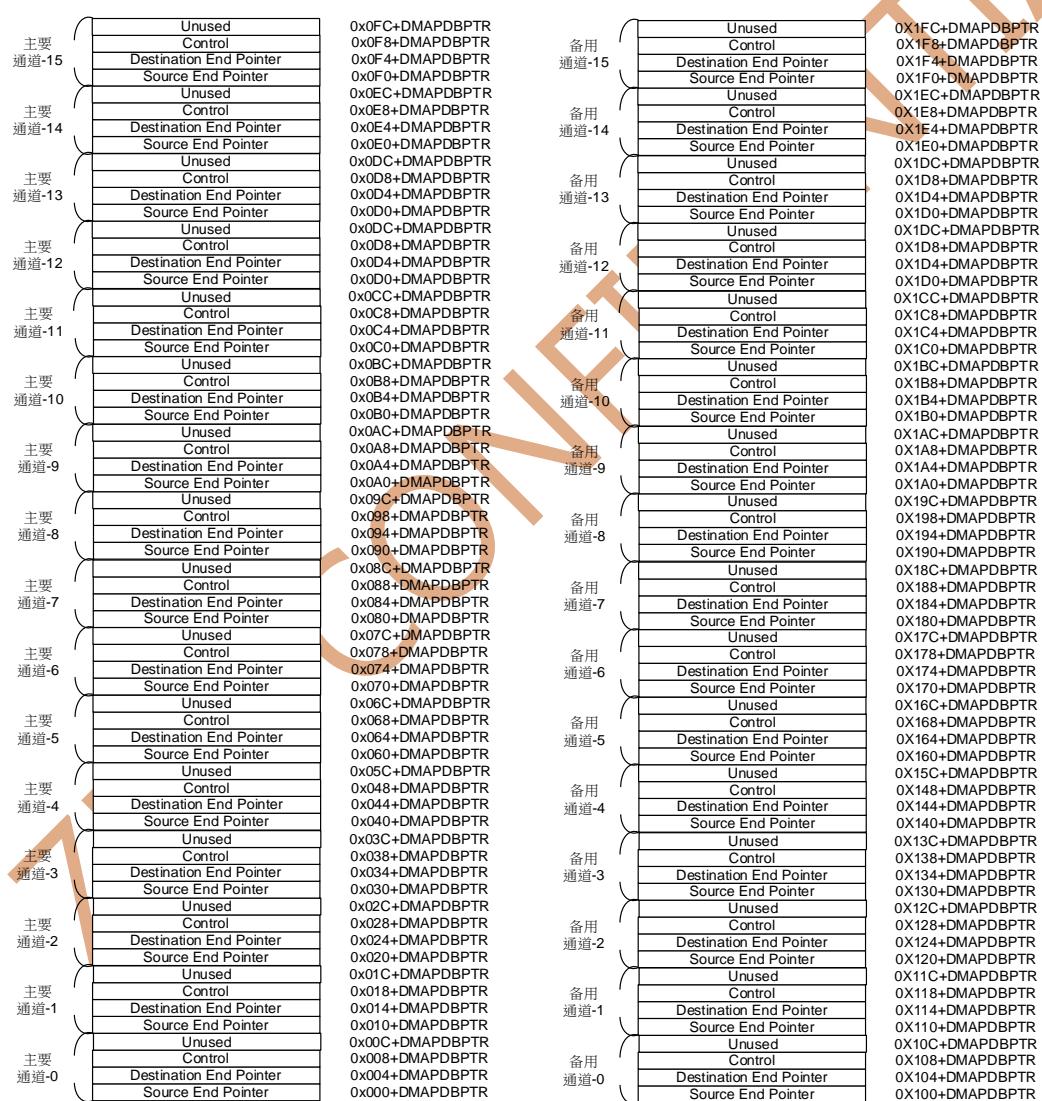
- (1) 来源端指针存储器位置包含来源数据的端地址
- (2) 目标端指针存储器位置包含目标数据的端地址
- (3) 控制数据配置存储器位置包含信道配置控制数据

编程确定来源和目标数据大小、传输次数和仲裁次数。

通道	主要结构		备用结构	
	内容描述	偏移 (储存器映像)	内容描述	偏移
通道 15	保留, 置 0	0x0FC	保留, 置 0	0x1FC
	控制数据配置	0x0F8	控制数据配置	0x1F8
	目标端指针	0x0F4	目标端指针	0x1F4
	来源端指针	0x0F0	来源端指针	0x1F0
...	...	...	...	...
通道 1	保留, 置 0	0x01C	保留, 置 0	0x11C
	控制数据配置	0x018	控制数据配置	0x118
	目标端指针	0x014	目标端指针	0x114
	来源端指针	0x010	来源端指针	0x110



通道	主要结构		备用结构	
	内容描述	偏移 (储存器映像)	内容描述	偏移
通道 0	保留, 置 0	0x00C	保留, 置 0	0x10C
	控制数据配置	0x008	控制数据配置	0x108
	目标端指针	0x004	目标端指针	0x104
	来源端指针	0x000	来源端指针	0x100



DMA\_ADBPTR=0X100+DMAPDBPTR

Figure 32-5 使用 16 通道的储存器映像含盖主要结构与备用结构



用户必须在源代码中定义 DMA 结构, 定义结构之后, 必须将其起始地址赋予给 DMA 基址指针寄存器 **DMAPDBPTR**。这样, 各 DMA 通道的各寄存器便处于偏移地址(如 Table34-4 之规定)加 DMAPDBPTR 寄存器值。

Note :

- (1)  $0x2000\_000 \leq \text{DMAPDBPTR} \leq 0x2000\_4000 \rightarrow$  储存器映像
- (2)  $0x100 + \text{DMAPDBPTR} \rightarrow \text{DMA\_ADB PTR}$

ZBT CONFIDENTIAL



### 32.4-4.1 控制数据配置( CHNL\_CFG)

Table 32-5 控制数据配置表

位	参数	说明		
[31:30]	dst_inc	目标地址增量。地址增量取决于来源数据宽度，如下所示：		
		来源数据宽度	DST_INC	目标地址增量
		字节	00 01 10 11	字节 半字 字 不递增。地址仍为 DST_END_PTR 存储器位置包含的值
		半字	00 01 10 11	Revered 半字 字 不递增。地址仍为 DST_END_PTR 存储器位置包含的值
		字	00 01 10 11	Revered Revered 字 不递增。地址仍为 DST_END_PTR 存储器位置包含的值
[29:28]	dst_size	目标数据大小。必须始终与SRC_SIZE相同。 00: 字节。 01: 半字。 10: 字。 11: 保留。		
[27:26]	src_inc	来源地址增量。地址增量取决于来源数据宽度，如下所示：		
		来源数据宽度	SRC_INC	目标地址增量
		字节	00 01 10 11	字节 半字 字 不递增。地址仍为 DST_END_PTR 存储器位置包含的值
		半字	00 01 10 11	Revered 半字 字 不递增。地址仍为 DST_END_PTR 存储器位置包含的值
		字	00	Revered



位	参数	说明
		01 10 11 Revered 字 不递增。地址仍为 DST_END_PTR 存储器位置包含的值
[25:24]	src_size	目标数据大小。必须始终与 SRC_SIZE 相同。 00: 字节。 01: 半字。 10: 字。 11: 保留
[23:18]	Revered	未定义。写入0
[17:14]	R_power	设置这些位以控制在控制器重新仲裁之前有多少 DMA 传输可以发生。 可能的仲裁率设置为： b0000 Arbitrates after each DMA transfer. b0001 Arbitrates after 2 DMA transfers. b0010 Arbitrates after 4 DMA transfers. b0011 Arbitrates after 8 DMA transfers. b0100 Arbitrates after 16 DMA transfers. b0101 Arbitrates after 32 DMA transfers. b0110 Arbitrates after 64 DMA transfers. b0111 Arbitrates after 128 DMA transfers. b1000 Arbitrates after 256 DMA transfers. b1001 Arbitrates after 512 DMA transfers. b1010-b1111 Arbitrates after 1024 DMA transfers. This means that no arbitration occurs during the DMA transfer because the maximum transfer size is 1024.
13:4	N_MINUS_1	针对该信道配置的传输次数减1。10位值表示DMA传输次数(非 总字节数)减1。可能的值有： 0x000: 1次DMA传输。 0x001: 2次DMA传输。 0x002: 3次DMA传输。 ... 0x3FF: 1024 次 DMA 传输
[3]	next_useburst	当控制器正在执行 外设分散收集并完成使用备用数据结构的 DMA 周期时， 需设置 CHX_CHBTSET [C] 位为 1。



位	参数	说明
		Changes the setting of single transfer 0: Do not change the value of . 1: Sets to "1".
[2:0]	cycle_ctrl	DMA 周期的传输类型 000: 停止(无效) 001: 基本。 010: 自动请求。 011: 乒乓。 100: 存储器分散/聚集式主要。 101: 存储器分散/聚集式备选。 110: 外设分散/聚集式主要。 111: 外设分散/聚集式备选。



### 32.4.5 DMA 传输周期的类型

cycle\_ctrl[2:0] 位控制如何执行 DMA 周期的型态, 设定 cycle\_ctrl[2:0] 位如下:

cycle_ctrl[2:0]	传输描述
3' b000	通道控制数据结构无效
3' b001	基本 DMA 传输(Basic)
3' b010	自动请求(Auto-request)
3' b011	Ping-pong
3' b100	存储器离散-聚合模式(scatter-gather) 使用主要的数据结构
3' b101	存储器离散-聚合模式(scatter-gather) 使用备用的数据结构
3' b110	外设离散-聚合模式(scatter-gather) 使用主要的数据结构
3' b111	外设离散-聚合模式(scatter-gather) 使用备用的数据结构

对于所有周期模型，控制器在 DMA 传输  $2^R$  后进行仲裁。如果将低优先级通道设置为较大的  $2^R$  值，则它会阻挡所有其他通道执行 DMA 传输，直到低优先级 DMA 传输完成。因此，在设置 R\_power 时必须小心，不要显著增加高优先级通道的延迟。



### 32.4-5.1 无效

这意味着该通道未使能任何 DMA 传输。控制器完成一个 DMA 周期后，它将周期类型设置为无效，以防止它重复相同的 DMA 周期。

### 32.4-5.2 基本

在此模式下，控制器可配置为使用主要数据结构或备用数据结构。

在信道 C(信道号)使能之后 当控制器接收该通道的请求，则须按照 DMA 周期的操作如下：

Step-1. 控制器执行  $2^R$  个传输。如果剩余的传输次数变为 0，则流程到 Step-3 继续。

Step-2. 控制器仲裁：

- 如果更高优先级的通道正在请求服务，则控制器为该通道提供服务
- 如果外围设备或软件向控制器发出请求，则它继续执行 Step-1.

Step-3. 控制器将 `dma_done[C]` 设置为高一个 HCLK 周期，表示传输结束，控制器为对应的 DMA 通道产生中断。

Performs  $2^R$  transfers per request until N transfers have been done. Arbitrates after  $2^R$

例： $2^R=2, N=6$ :

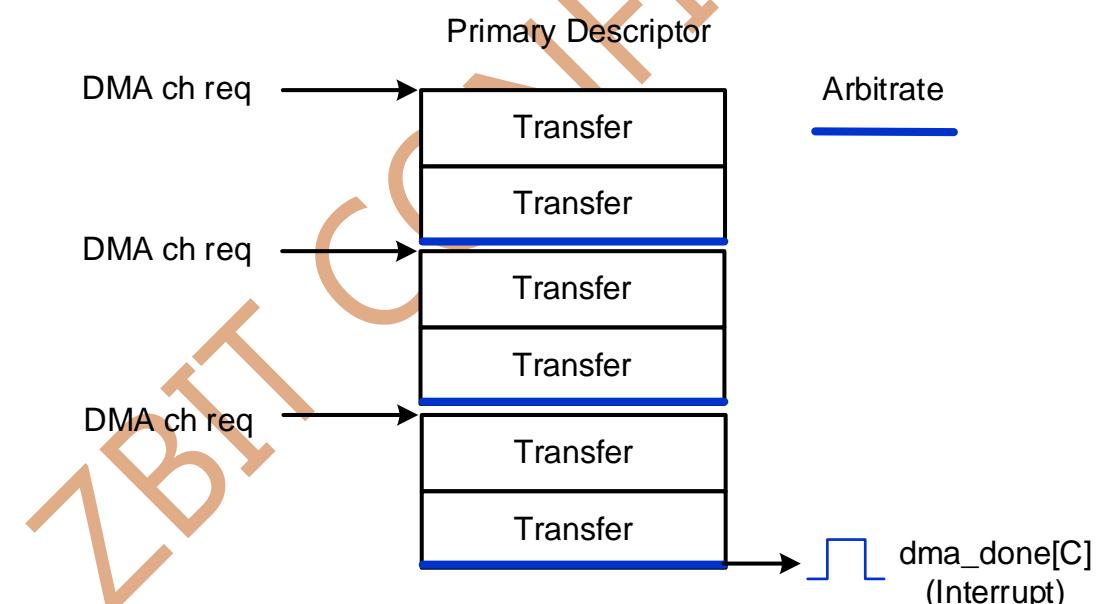


Figure 32-6 基本模式图



### 32.4-5.3 自动请求模式

当控制器在此模式下工作时，它只需要接收一个请求以便能它便可完成整个 DMA 周期。这样可以进行大量数据传输而不 会显著增加服务更高优先级请求的延迟时间，或者要求处理器或外设提供多个请求。这种模式对存储器到存储器复制应用非 常有用(自动请求不适合外设使用)。

您可以将控制器配置为主要数据结构或备用数据结构。在使能通道 C 且控制器收到此通道的请求后，此 DMA 周期的操作如下：

Step-1. 控制器执行  $2^R$  个传输。如果剩余的传输次数变为零，则转到 Step-3 继续。

Step-2. 自动产生该通道 C 的请求.控制器仲裁：当通道 C 为最高优先级时，则 DMA 周期转到第 1 步.

Step-3. 控制器将 `dma_done[C]` 设置为高一个 HCLK 周期, 表示传输结束，控制器为对应的 DMA 通道产生中断。

Performs N transfers per request. Arbitrates for every  $2^R$  transfers

例： $2^R=2$ ,  $N=6$ :

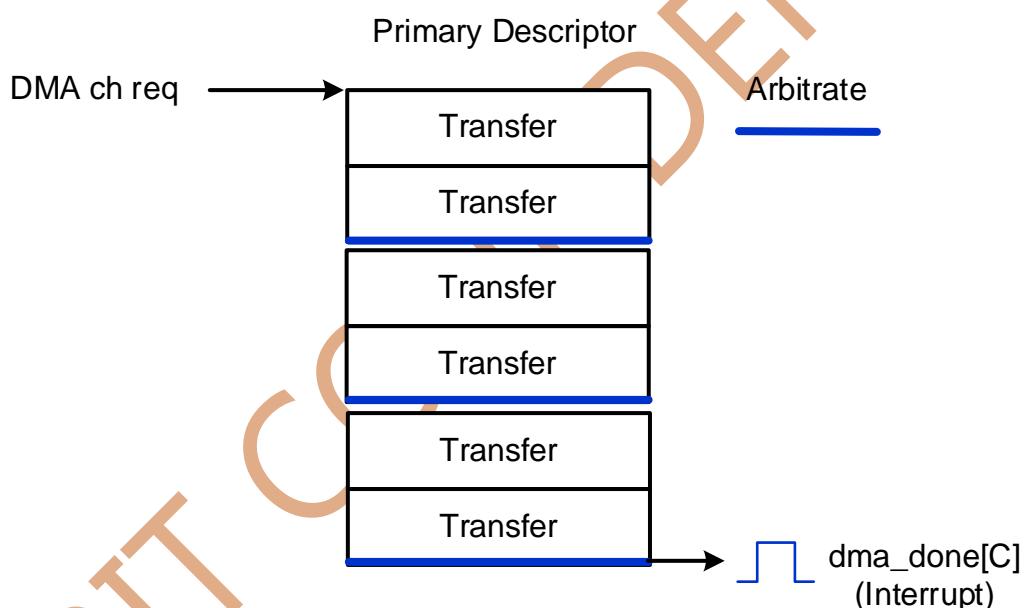


Figure 32-7 自动请求模式范例图



#### 32.4-5.4 Ping-pong 模式

在乒乓(Ping-pong)模式下，控制器使用其中一个(主要/备用)数据结构执行一个 DMA 周期，然后利用另一个(备用 /主要)数据结构执行一个 DMA 周期。控制器不停地交替使用主要数据结构和备用数据结构，直至读取到无效数据结构或者主机处理器禁用该通道。

这种模式适合于利用存储器中的不同缓冲器来将数据从外设传输到存储器。在典型应用中，启动传输之前，主机必须同时配置主要数据结构和备用数据结构。传输进行时，主机随后可以在对应的传输结束时，在中断服务例程中配置主要或备用控制数据结构。

完成与各控制数据结构相关的传输之后，DMA 控制器中断处理器。利用主要或备用控制数据结构的各次传输的工作方式与基本 DMA 传输完全相同。

- 举例说明：

N requests on Primary, then N requests on Alternate, then Primary..(Arbitration for every  $2^R$  transfers)

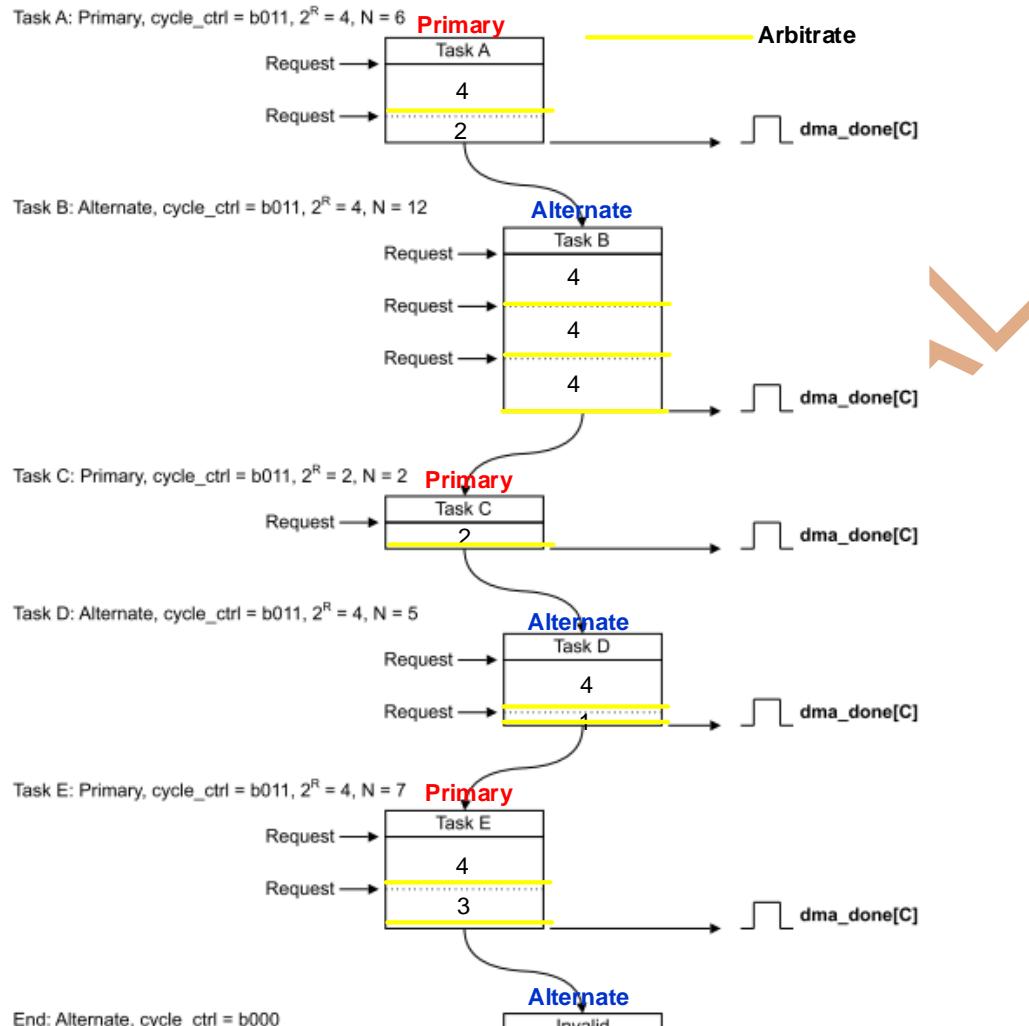


Figure 32-8 Ping-pong 模式范例图

说明:

Task A:

1. CPU 为任务 A 配置主要数据结构。
2. CPU 为任务 B 配置备用数据结, 并将其使能 DMA 控制器可以在任务 A 完成后立即切换到任务 B, 前提是更高优先级的通道没有请求服务。
3. DMA 控制器接收请求并执行四次 DMA 传输。
4. DMA 控制器仲裁后, DMA 控制器收到对该通道的请求后如果具有最高优先级就继续往下执行。
5. DMA 控制器接收请求并 DMA 控制器执行剩余的两次 DMA 传输。
6. DMA 控制器设置 dma\_done[C] 一个周期(HCLK )的 high, 并进入仲裁过程



任务 A 传输完成后,CPU 就能对任务 C 配置主要(primary)数据结构,并将任务 C 使能会在任务 B 执行完成后,DMA 控制器立即切换到任务 C 执行,前提是更高优先级的通道没有请求服务.

在 DMA 控制器收到收到从通道发出的新请求并且它具有最高优先级之后, 任务 B 开始:

Task B :

1. DMA 控制器接收请求并执行四次 DMA 传输。
2. DMA 控制器仲裁后, DMA 控制器收到对该通道的请求后如果具有最高优先级就继续往下执行。
3. DMA 控制器接收请求 DMA 控制器执行剩余的两次 DMA 传输。
4. DMA 控制器仲裁后, DMA 控制器收到对该通道的请求后如果具有最高优先级就继续往下执行。
5. DMA 控制器接收请求并 DMA 控制器执行剩余的两次 DMA 传输。
6. DMA 控制器设置 `dma_done[C]` 一个周期(HCLK )的 high,并进入仲裁过程

任务 B 传输完成后,CPU 就能对 Task D 配置备用(alternate)数据结构

在 DMA 控制器收到收到从通道发出的新请求并且它具有最高优先级之后, 任务 C 开始:

Task C :

1. DMA 控制器接收请求并执行二次 DMA 传输。
2. DMA 控制器设置 `dma_done[C]` 一个周期(HCLK )的 high,并进入仲裁过程

任务 C 传输完成后,CPU 就能对 Task E 配置主要(primary)数据结构.

在控制器收到对通道的新请求并且它具有最高优先级之后, 任务 D 开始:

Task D :

1. DMA 控制器接收请求并执行四次 DMA 传输。
2. DMA 控制器仲裁后, DMA 控制器收到对该通道的请求后如果具有最高优先级就继续往下执行。
3. DMA 控制器接收请求并 DMA 控制器执行剩余的一次 DMA 传输。
4. DMA 控制器设置 `dma_done[C]` 一个周期(HCLK )的 high,并进入仲裁过程

在控制器收到对通道的新请求并且它具有最高优先级之后, 任务 E 开始

Task E :

1. DMA 控制器接收请求并执行四次 DMA 传输。
2. MA 控制器仲裁后, DMA 控制器收到对该通道的请求后如果具有最高优先级就继续往下执行。
3. DMA 控制器接收请求并 DMA 控制器执行剩余的三次 DMA 传输。
4. DMA 控制器设置 `dma_done[C]` 一个周期(HCLK )的 high,并进入仲裁过程

在控制器收到对通道的新请求并且它具有最高优先级之后,会试着启动下一个任务.



但是，由于 CPU 尚未 配置备用数据结构，并且在任务 E 完成时，DMA 控制器将设置 cycle\_ctrl 位为 b000，然后乒乓(Ping-Pong) DMA 事务完成

ZBT CONFIDENTIAL

---

---



### 32.4-5.5 存储器分散/聚集式

DMA 来撷取分段内存并对该内存执行移动操作。

这种模式用于执行多个存储器到存储器复制任务的场合。处理器可以同时配置所有 任务，无需在任务之间进行干预。

在存储器分散/聚集模式下，控制器必须配置为同时使用主要数据结构和备用数据结构。控制器利用主要数据结构设置备用数据结构的控制配置。备用数据结构用于实际数据传输，其与自动请求 DMA 传输类似。每次主要传输完成后，控制器执行仲裁。控制器只需一个请求就能完成全部传输。当利用基本周期完成全部分散/聚集处理时，控制器在 NVIC 中产生对应的 DMA 通道中断。

操作：

这种模式下，控制器接收到初始请求后，利用主要数据结构执行 4 次 DMA 传输，依设置备用数据结构的控制结构。完成此传输之后，控制器利用备用数据结构启动 DMA 周期。完成该周期之后，控制器再次利用主要数据结构执行 4 个 DMA 周期。控制器 不停地交替使用主要数据结构和备用数据结构，直至处理器将备用数据结构配置为基本周期或者 DMA 读取到无效数据结构。

**Note:** 控制器完成 N 次主要传输后，通过将 cycle\_ctrl 设置为 b000 时主要(primary)数据结构为无效。当内存分散-聚集模式在使用 auto-request cycle 传输完成,DMA 控制器会设置 dma\_done[C] 为 high. 在分散-聚集模式下，DMA 控制器使用主要数据结构来编程备用数据结构。表 33-3 列出了主数据的 channel\_cfg 内存位置的字段结构，您必须使用常量值(Constant-value fields)和那些可以由用户定义的值(User defined values)对其进行编程。

Table 存储器分散/聚集模式下主要数据结构的 CHNL\_CFG, CHNL\_CFG[2:0] = 100

定量值设定

位	名称	值	描述
31:30	DST_INC	10	10: 配置控制器对地址使用字增量。
29:28	DST_SIZE	10	10: 配置控制器使用字转换。
27:26	SRC_INC	10	10: 配置控制器对地址使用字增量。
25:24	SRC_SIZE	10	10: 配置控制器使用字转换。
23:18	保留	--	未定义。写入 0。
17:14	R_POWER	0010	0010: 表示 DMA 控制器要执行 4 次传输。
13:4	N_MINUS_1	-	配置控制器执行N次DMA传输，其中N为4的倍数。 The number of alternative task to be prepared ×4 is specified.
3	next_useburst	0	写入 0。"0" is specified in memory scatter / gather mode.
2:0	CYCLE_CTRL	100	100: 配置控制器执行存储器分散/聚集 DMA 周期。

Table 32-6 存储器分散/聚集式-信道配置表



范例说明：

- 存储器分散/聚集式：使用主要数据结构预先定义 A,B,C Task 的描述元到指定位置的储存器

主要数据结构 来源端指针内容 指向 TaskA,B,C	来源端指针	目标端指针	控制数据配置			保留
			Cycle_ctrl	2 <sup>R</sup>	N	
Data for Task A	0x0A000000	0x0AE00000	2' b101	2	2	--
Data for Task C	0x0B000000	0x0BE00000	2' b101	2	4	--
Data for Task A	0x0C000000	0x0CE00000	2' b001	2	4	--

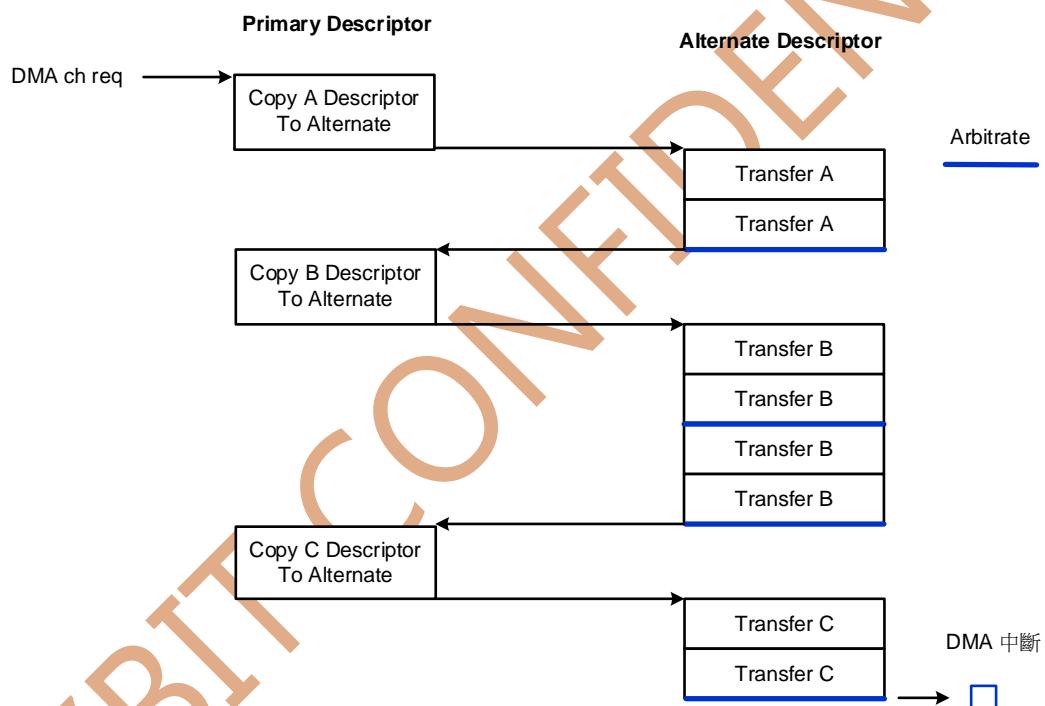


Figure 32-9 存储器分散/聚集式 图例



- 执行步骤:

- 1) 相对的通道产生 DMA 请求
- 2) 复制 Task A 描述元的配置数据(从主要数据结构定义的位置)到备用数据结构(相对的信道位置)
- 3) 执行相对通道的备用数据结构 Task A 配置的数据传输.
- 4) Task A 传输完成 go to step 5
- 5) 复制 Task B 描述元的配置数据到备用数据结构(相对的信道位置)
- 6) 执行相对的通道的备用数据结构 Task B 配置的数据传输.
- 7) Task B 传输完成 go to step 8
- 8) 复制 Task C 描述元的配置数据到备用数据结构(相对的信道位置)
- 9) 执行相对的通道的备用数据结构 Task A 配置的数据传输.
- 10) Task C 传输完成, DMA 产生中断到 CPU

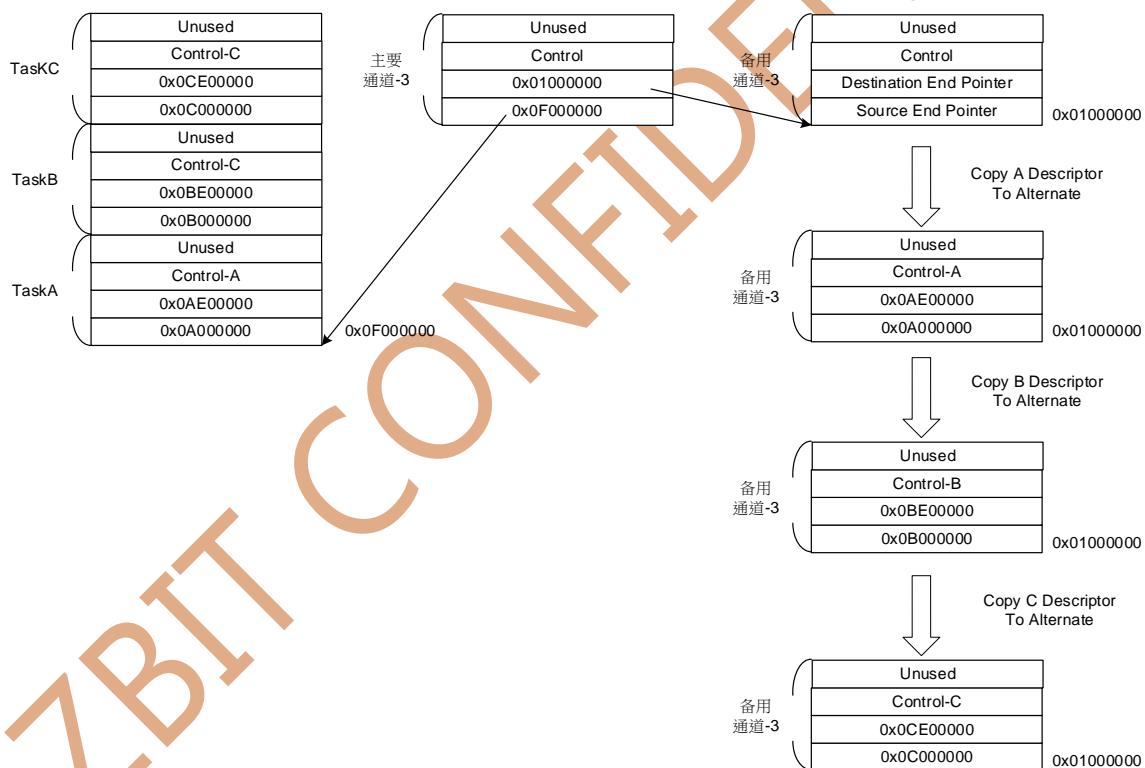


Figure 32-10 复制 Task A,B,C 描述元的配置数据图



### 32.4-5.6 外设分散/聚集式

这种模式用于要执行多个外设到存储器 **DMA** 任务的场合。除了仲裁和请求要求之外，这与存储器分散/聚集模式非常相似。

在外设分散/聚集模式下，控制器必须配置为同时使用主要数据结构和备用数据结构。控制器利用主要数据结构设置备用数据结构的控制结构。备用数据结构用于实际数据传输，每次传输采用备用数据结构和基本 DMA 传输进行。每次主要传输完成 后，**控制器不执行仲裁**。当利用基本周期完成全部分散/聚集处理时，控制器在 NVIC 中产生对应的 DMA 通道中断。

在外设分散/聚集模式下，控制器从外设接收到初始请求后，利用主要数据结构执行 4 次 DMA 传输，以设置备用控制数据结构。然后，控制器利用备用数据结构立即启动 DMA 周期，而不重新仲裁。

完成此周期之后，控制器重新仲裁；如果其从具有最高优先级的外设收到请求，则利用主要数据结构再次执行 4 次 DMA 传输。然后，控制器利用备用数据结构立即启动 DMA 周期，而不重新仲裁。控制器不停地交替使用主要数据结构和备用数据结构，直至处理器将备用数据结构配置为基本周期或者 DMA 读取到无效数据结构。

定量值设定

Table 32-7 外设分散/聚集式 -信道配置表

位	名称	值	描述
31:30	DST_INC	10	配置控制器对地址使用字增量。
29:28	DST_SIZE	10	配置控制器使用字转换。
27:26	SRC_INC	10	配置控制器对地址使用字增量。
25:24	SRC_SIZE	10	配置控制器使用字转换。
23:18	保留	--	未定义。写入 0。
17:14	R_POWER	0010	表示 DMA 控制器执行了 4 次传输而不重新仲裁。
13:4	N_MINUS_1	-	配置控制器执行N次DMA传输，其中N为4的倍数。 The number of alternative task to be prepared ×4 is specified.
3	next_useburst	-	When set to 1, the controller sets the chnl_useburst_set [C] bit to 1 after the alternate transfer completes
2:0	CYCLE_CTRL	110	配置控制器执行存外设器分散/聚集 DMA 周期。



范例说明：

- 存储器分散/聚集式：使用主要数据结构预先定义 A,B,C Task 的描述元到到指定位置的储存器

	来源端指针	目标端指针	控制数据配置			保留
			Cycle_ctrl	2 <sup>R</sup>	N	
Data for Task A	0x0A000000	0x0AE00000	2' b101	2	2	--
	0x0B000000	0x0BE00000	2' b101	2	4	--
	0x0C000000	0x0CE00000	2' b001	2	4	--

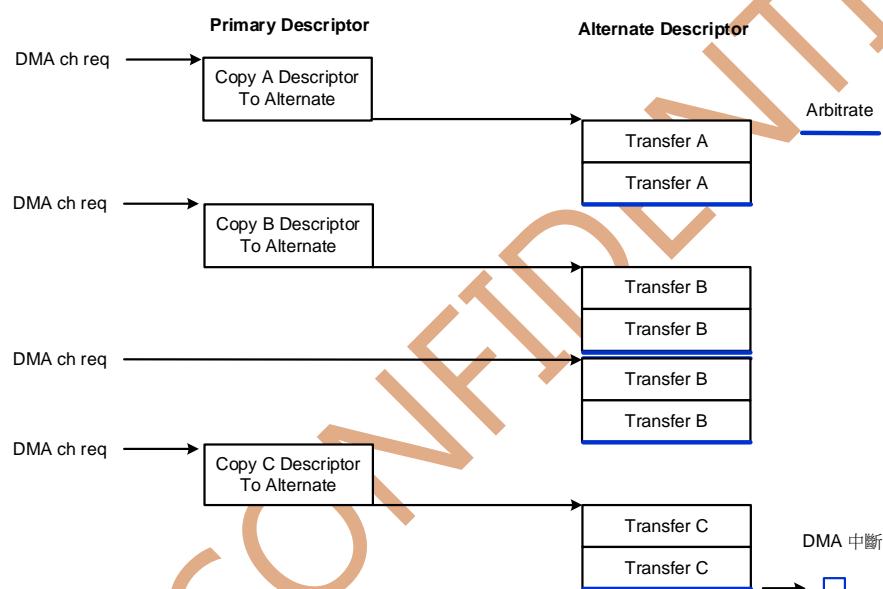


Figure 32-11 外设分散/聚集式范例图

● 执行步骤：

- 1) 相对的通道产生 DMA 请求
- 2) 复制 Task A 描述元的配置数据(从主要数据结构定义的位置)到备用数据结构(相对的信道位置)
- 3) 执行相对通道的备用数据结构 Task A 配置的数据传输.
- 4) Task A 传输完成, 如相对的通道产生 DMA 请求 goto step 5
- 5) 复制 Task B 描述元的配置数据到备用数据结构(相对的信道位置)
- 6) 执行相对的通道的备用数据结构 Task B 配置的数据传输.
- 7) Task B 传输完成, 如相对的通道产生 DMA 请求 goto step 8
- 8) 复制 Task C 描述元的配置数据到备用数据结构(相对的信道位置)
- 9) 执行相对的通道的备用数据结构 Task A 配置的数据传输.
- 10) Task C 传输完成, DMA 产生中断到 CPU.



### 32.4.6 错误管理

读取或写入保留的地址空间，会产生 DMA 传输错误。在 DMA 读或写访问期间发生 DMA 传输错误时，会自动禁用故障通道。若 NVIC 中使能了该 DMA 错误中断，则还会产生一个中断。  
主机处理器检测到 `dma_err` 为高电平后，必须检查哪个通道发生 ERROR 时处于活动状态。它可以通过以下方式为记录哪个信道发生 ERROR 的步骤：

11. 读取 DMA\_CHENSET 寄存器以创建禁用通道列表。

→当通道完成传输，控制器将禁用该通道。在主机处理器上运行程序必须始终记录哪些通道已传输完成需读取标志寄存器 DMA\_CHTCIF。

12. 它必须将步骤 1 中禁用的频道列表与的记录进行比较 最近设置其 DMA\_CHENSET 输出频道。

→没有设置禁用该通道 (发生 ERROR)记录此通道是会再发生 ERROR 的通道。

## 32.5 中断

各 DMA 通道传输完成或 ERROR 发生时，可以产生一个中断。针对各 DMA 通道，NVIC 提供了独立的中断使能位。

DMA 控制器取出位于 SRAM 存储器中的通道控制数据结构以执行数据传输。当允许使用 DMA 操作时，支持 DMA 的外设请求 DMA 控制器进行传输。在一个通道的设定 DMA 传输次数结束时，DMA 控制器产生一个对应该通道的中断。此中断表示 DMA 传输已完成。

中断事件	事件旗标志位	使能中断控制位
传输完成	DMA_CHTCIF[C] C: 0~15	DMA_CHINTEN[C] C: 0~15
传输错误	DMA_CHTCIF[31]	DMA_CHINTEN[31]



### 32.6 DMAX 寄存器列表(X=1,2)

DMA1 基地址: 0x4001 4000 DMA2 基地址: 0x40014400

偏移地址	名称	描述	访问类型	默认值
0x000	DMAX_STA	状态寄存器	RO	0x0000 0000
0x004	DMAX_CFG	配置寄存器	WO	0x0000 0000
0x008	DMAX_PDBPTR	主要控制数据基址指针寄存器	RW	0x0000 0000
0x00C	DMAX_ADBPTR	备选控制数据基址指针寄存器	RO	0x0000 00nn <sup>(2*)</sup>
0x010	DMAX_WTREQ_STA	DMAX 等待请求状态	RO	0x0000 0000
0x014	DMAX_SWREQ	软件请求寄存器	WO	0x0000 0000
0x020	DMAX_RMSKSET	请求屏蔽设置寄存器	RW	0x0000 0000
0x024	DMAX_RMSKCLR	请求屏蔽清零寄存器	WO	0x0000 0001
0x028	DMAX_ENSET	使能设置寄存器	RW	0x0000 0001
0x02C	DMAX_ENCLR	使能清零寄存器	WO	0x0000 0000
0x030	DMAX_ALTSET	主要-备选设置寄存器	RW	0x0000 0000
0x034	DMAX_ALTCLR	主要-备选清除寄存器	WO	0x0000 0000
0x038	DMAX_PRISET	优先级设置寄存器	RW	0x0000 0000
0x03C	DMAX_PRICLR	优先级清零寄存器	WO	0x0000 0000
0x04C	DMAX_ERRCLR	总线错误清零寄存器	RW	0x0000 0000
0X0060	DMAX_CHINTSET	DMAX 通道传输完成中断设置寄存器	RW	0x0000 0000
0X0064	DMAX_CHINTCLR	DMAX 通道传输完成中断清除寄存器	RW	0x0000 0000
0X0068	DMAX_CHTCIF	DMAX 通道传输完成中断旗帜寄存器	RW	0x0000 0000
0X006C	DMAX_CHWTIONREG	DMAX 通道等待请求使能寄存器	RW	0x0000 0000
0X0070	DMAX_CHSELREQ	DMAX 通道选择请求寄存器	RW	0x0000 0000
0xE00(1*)	DMAX_INTCFG	DAM 集成配置寄存器	RW	0x0000 0000
0xE08(1*)	DMAX_STALL	停滞状态寄存器	RO	0x0000 0001
0xE10(1*)	DMAX_CHREQ	DMAX 请求状态寄存器	RO	0x0000 0001
0xE18(1*)	DMAX_CHSREQ	DMAX 通道单笔请求状态	RO	0x0000 0000
0xE20(1*)	DMAX_CHDONESET	DMAX 通道传输完成的状态设置寄存器	RW	0x0000 0000
0xE24(1*)	DMAX_CHDONECLR	DMAX 通道传输完成状态清 0 寄存器	WO	--
0xE28(1*)	DMAX_CHACTSET	DMAX 通道正在传输中的状态设置寄存器	RW	0x0000 0000
0xE2C(1*)	DMAX_CHACTCLR	通道正在传输中的状态清 0 寄存器	WO	--
0xE48(1*)	DMAX_ERRSET	DMAX 总线错误清除寄存器寄存器	WO	--
0xFD0	DMAX_PERID4	DMAX 外设 ID 4 寄存器	RO	0x0000 0004
0xFE0	DMAX_PERID0	DMAX 外设 ID 0 寄存器	RO	0x0000 0030
0xFE4	DMAX_PERID1	DMAX 外设 ID 1 寄存器	RO	0x0000 00B2
0xFE8	DMAX_PERID2	DMAX 外设 ID 2 寄存器	RO	0x0000 00-B
0xFEC	DMAX_PERID3	DMAX 外设 ID 3 寄存器	RO	0x0000 0000



偏移地址	名称	描述	访问类型	默认值
0xFF0	DMAX_PCELLID0	DMAX PrimeCell ID 0 寄存器	RO	0x0000 000D
0xFF4	DMAX_PCELLID1	DMAX PrimeCell ID 1 寄存器	RO	0x0000 00F0
0xFF8	DMAX_PCELLID2	DMAX PrimeCell ID 2 寄存器	RO	0x0000 0005
0XFFC	DMAX_PCELLID3	DMAX PrimeCell ID3 寄存器	RO	0x0000 00B1

Table 32-8 DMA 寄存器列表和复位值

Note:1\* : DMA 测试寄存器 2\* : 重置的值根据 DMA 信道配置



### 32.7 寄存器说明

#### 32.7-1 DMAX 状态寄存器(DMAX\_STA)

地址偏移: 0x00

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								STATE				保留			ENABLE
RO															

位	标记	功能描述	复位值	读写
31:21	-	保留	0x0	-
20:16	CHNLSMINUS1	可用DMAX通道数减1。 例如，若有 12 个通道可用(12-1=11)，寄存器的这些位将读出 0xB。	0x0	RO
15:8	--	保留。	0x0000	RO
7:4	STATE	DMAX控制器状态机的当前状态。读取此寄存器可了解DMAX执行的操作。 0000: IDLE: 空闲。 0001: RDCHNLDATA: 读取通道控制器数据。 0010: RDSRCENDPTR: 读取来源数据端指针。 0011: RDDSTENDPTR: 读取目标端指针。 0100: RDSRCDATA: 读取来源数据。 0101: WRDSTDATA: 写入目标数据。 0110: WAITDMAXREQCLR: 等待DMAX请求清零。 0111: WRCHNLDATA: 写入通道控制器数据。 1000: STALLED: 停止。 1001: DONE: 完成。 1010: SCATRGATHR: 外设分散/聚集过渡。 1011 到 1111: 保留。	0x0	RO
3:1	--	保留	0x0	RO
0	MasterEN	DMAX控制器使能状态。 0: 控制器已禁用。 1: 控制器已使能。	0	RO



### 32.7.2 DMAX 配置寄存器(DMAX\_CFG)

地址偏移: 0x04

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:1	--	保留	0x0000	--
0	MasterEN	控制器使能。 0: DMAX控制器已禁用。 1: DMAX 控制器已使能	0	R/W



### 32.7.3 DMAX 主要控制数据基址指针寄存器(DMAX\_PDBPTR)

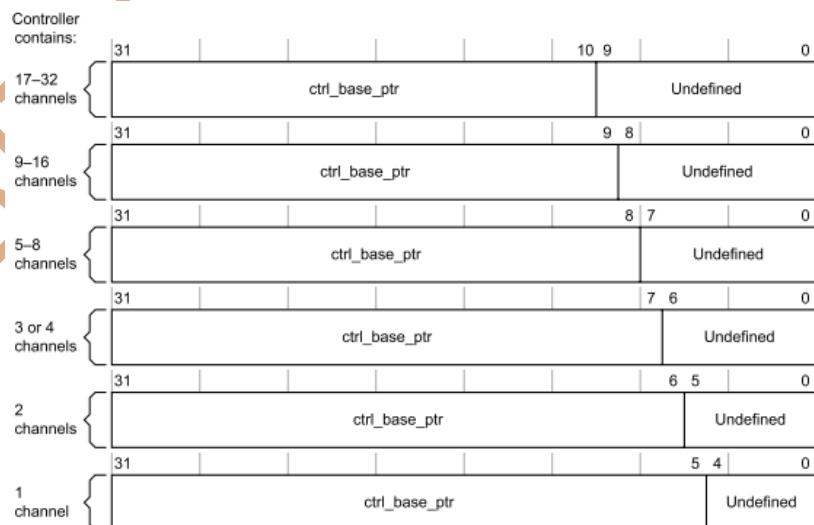
地址偏移: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAX_PDBPTR[31:16]															
R/W															

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

位		标记	功能描述	复位值	读写														
31:CH_BITS+5		ctrl_base_ptr	指向主要数据结构基址的指针	0x0000	R/W														
CH_BITS+4:0		--	保留 <table border="1"><tr><td>CH_BITS</td><td>说明</td></tr><tr><td>0</td><td>使用 1 DMAX 通道</td></tr><tr><td>1</td><td>使用 2 DMAX 通道</td></tr><tr><td>2</td><td>使用 3,4 DMAX 通道</td></tr><tr><td>3</td><td>使用 5~8 DMAX 通道</td></tr><tr><td>4</td><td>使用 9~16DMAX 通道</td></tr><tr><td>5</td><td>使用 17~32 DMAX 通道</td></tr></table>	CH_BITS	说明	0	使用 1 DMAX 通道	1	使用 2 DMAX 通道	2	使用 3,4 DMAX 通道	3	使用 5~8 DMAX 通道	4	使用 9~16DMAX 通道	5	使用 17~32 DMAX 通道	0x00	R/W
CH_BITS	说明																		
0	使用 1 DMAX 通道																		
1	使用 2 DMAX 通道																		
2	使用 3,4 DMAX 通道																		
3	使用 5~8 DMAX 通道																		
4	使用 9~16DMAX 通道																		
5	使用 17~32 DMAX 通道																		





### 32.7-4 DMAX 备用控制数据基地址指针寄存器(DMAX\_ADBPTR)

地址偏移: 0x0

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
alt_ctrl_base_ptr [31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	alt_ctrl_base_ptr	指向备用数据结构基址的指针	0x0000	R/W



### 32.7.5 DMAX 通道等待请求状态寄存器(DMAX\_CHWTREQ)

地址偏移: 0x10

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHWTREQ[15:0]															
RO															

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R
15	CHWTREQ[C*]	1: 通道(15)在等待请求状态	1	RO
14	CHWTREQ[C*]	1: 通道(14)在等待请求状态	1	RO
13	CHWTREQ[C*]	1: 通道(13)在等待请求状态	1	RO
12	CHWTREQ[C*]	1: 通道(12)在等待请求状态	1	RO
11	CHWTREQ[C*]	1: 通道(11)在等待请求状态	1	RO
10	CHWTREQ[C*]	1: 通道(10)在等待请求状态	1	RO
9	CHWTREQ[C*]	1: 通道(9)在等待请求状态	1	RO
8	CHWTREQ[C*]	1: 通道(8)在等待请求状态	1	RO
7	CHWTREQ[C*]	1: 通道(7)在等待请求状态	1	RO
6	CHWTREQ[C*]	1: 通道(6)在等待请求状态	1	RO
5	CHWTREQ[C*]	1: 通道(5)在等待请求状态	1	RO
4	CHWTREQ[C*]	1: 通道(4)在等待请求状态	1	RO
3	CHWTREQ[C*]	1: 通道(3)在等待请求状态	1	RO
2	CHWTREQ[C*]	1: 通道(2)在等待请求状态	1	RO
1	CHWTREQ[C*]	1: 通道(1)在等待请求状态	1	RO
0	CHWTREQ[C*]	1: 通道(0)在等待请求状态	1	RO

Note : C\* : 15~0 通道



### 32.7-6 DMAX 通道软件请求寄存器(DMAX\_CHSWREQ)

地址偏移: 0x14

复位值: 0x0000 0000

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

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

Note : 完成对应的软件请求之后，这些位自动由硬件清零

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	WO
15:0	CHSWREQ[C*]	写入时： 0: 不产生通道(C)的DMAX 软件请求。 1: 产生通道(C)的DMAX软件请求。	0	WO

Note : C\* : 15~0 通道



### 32.7.7 DMAX 通道使用多笔传输设定寄存器 (DMAX\_CHTSET)

地址偏移: 0x18

复位值: 0x0000 0000

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

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

Note : 完成对应的软件请求之后，这些位自动由硬件清零

位	标记	功能描述	复位值	读写
31:12	Revered	Revered	0x0000	R/W
15:0	CHBTSET[C*]	读取时： 0: 通道(C)为 single or burst requests 传输。 1: 通道(C)为 burst requests( $2^R$ transfers) 传输。 写入时 0: 不起作用。 1: 设置通道(C*)为 burst requests( $2^R$ transfers)。	0	R/W

Note : C\* : 15~0 通道



### 32.7-8 DMAX 通道使用多笔传输清除寄存器(DMAX\_CHTCLR)

地址偏移: 0x1C

复位值: 0x0000 0000

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

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

Note : 完成对应的软件请求之后，这些位自动由硬件清零

位	标记	功能描述	复位值	读写
31:12	Revered	Revered	0x0000	WO
15:0	CHBTCLR[C*]	写入时 0: 不起作用 1: 清除通道(C)为 burst requests 的设定	0	WO

Note : C\* : 15~0 通道



### 32.7-9 DMAX 通道请求屏蔽设置寄存器(DMAX\_CHRMKSET)

地址偏移: 0x20

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:12	--	Revered	0x0000	R/W
15:0	CHRMKSET[C*]	读取时: 0: 使能通道(C) 的请求。 1: 禁用通道(C) 的请求。 写入时 0: 不起作用(使用 DMAXRMKCLR 寄存器使能 DMAX 请求) 1: 禁用与通道(C)为相关的外设产生 DMAX 请求。:	0	R/W

Note : C\* : 15~0 通道



### 32.7-10 DMAX 通道请求屏蔽清除寄存器(DMAX\_CHRMKCLR)

地址偏移: 0x24

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:12	--	Revered	0x0000	R/W
15:0	CHRMKCLR[C*]	写入时 0: 不起作用(使用 DMAXRMKCLR 寄存器使能 DMAX 请求) 1: 启用与通道(C)为相关的外设产生 DMAX 请求。:	0	R/W

Note : C\* : 15~0 通道



### 32.7.11 DMAX 通道使能设置寄存器(DMAX\_CHENSET)

地址偏移: 0x28

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHENSET[C*]	0: 不起作用(使用 DMAXENCLR 寄存器禁用通道(C)) 1: 使能通道(C)	0	R/W

Note : C\* : 15~0 通道



### 32.7.12 DMAX 通道使能清除寄存器 (DMAX\_CHENCLR)

地址偏移: 0x2C

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15~1	CHENCLR[15]	0: 不起作用(使用 CHENSET 寄存器使能通道(15)) 1: 禁用通道(15)	0	R/W
.	CHENCLR[C*]	.		
.	CHENCLR[C*]	.		
.	CHENCLR[C*]	.		
0	CHENCLR[0]	0: 不起作用(使用 CHENSET 寄存器使能通道(0)) 1: 禁用通道(0)	0	R/W

Note : C\* : 15~0 通道



### 32.7.13 DMAX 通道备用设置寄存器(DMAX\_CHALTSET)

地址偏移: 0x30

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:12	--	Revered	0x0000	R/W
15:0	CHALTSET[C*]	读取时: 0: 通道(15)使用主要数据结构 1: 通道(15)使用备用数据结构 写入时: 0: 不起作用 1: 设置通道(15)为备用数据结构	0	R/W
.	CHENCLR[C*]	.	.	
0	CHALTSET[0]	读取时: 0: 通道(0)使用主要数据结构 1: 通道(0)使用备用数据结构 写入时: 0: 不起作用 1: 设置通道(0)为备用数据结构	0	R/W

Note : C\* : 15~0 通道

Note: 对于乒乓式、存储器分散/聚集式和外设分散/聚集式传输, DMAX 控制器会视需要自动将这些位置 1/清 0。



### 32.7.14 DMAX 通道备用清除设置寄存器(DMAX\_CHALTCLR)

地址偏移: 0x34

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHALTCLR[C*]	0: 不起作用 1: 设置通道(C)为主要数据结构	0	R/W

Note : C\* : 15~0 通道

Note: 对于乒乓式、存储器分散/聚集式和外设分散/聚集式传输, DMAX 控制器会视需要自动将这些位置 1/清 0。



### 32.7-15 DMAX 通道优先级设置寄存器(DMAX\_CHPRISET)

地址偏移: 0x38

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHPRISET[C*]	读取时: 0: 通道(C)正在使用默认优先级 1: 通道(C) 正在使用高优先级 写入时: 0: 不起作用 1:设置通道(C)为高优先级	0	R/W

Note : C\* : 15~0 通道



### 32.7.16 DMAX 通道优先级清除寄存器(DMAX\_CHPRICLR)

地址偏移: 0x3C

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHPRICLR[C*]	0: 不起作用 1: 设置通道(C)为默认优先级	0	R/W

Note : C\* : 15~0 通道



### 32.7.17 DMAX 总线错误清除寄存器(DMAX\_CERRCLR)

地址偏移: 0x40

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:1	--	Revered	0x0000	R/W
0	ERRCLR	<p>总线错误状态。 此寄存器用于读取和清除 DMAX 总线错误状态。 如果控制器在执行传输时遇到总线错误，就会设置错误状态。 如果一个通道发生总线错误，控制器会自动禁用该通道。 其它通道不受影响。写入 1 可将此位清 0。</p> <p>读取时： 0: 未发生总线错误。 1: 有一个总线错误待处理。</p> <p>写入时： 0: 不起作用。 1: 位清 0。</p>	0	R/W



### 32.7.18 DMAX 通道传输完成中断设置寄存器(DMAX\_CHINTSET)

地址偏移: 0x0060

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHINTEN[C*]															
RW															

位	标记	功能描述	复位值	读写
31	ERRINTEN	0 : 不起作用 1 : 设置 DMAX 传输发生错误时会产生中断	0	RW
30:16	Revered	Revered	0x0000	RW
15:0	CHINTEN[C*]	读取时: 0: 通道(C) 传输完成 1 不会产生中断 1: 通道(C) 正在使用传输完成产生中断 写入 0 : 不起作用 1 : 设置通道(C*)传输完成产生中断	0x0000	RW

Note : C\* : 15~0 通道



### 32.7.19 DMAX 通道中断使能清除寄存器(DMAX\_CHINTCLR)

地址偏移: 0X0064

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31	ERRINTCLR	写入时: 0: 不起作用 1: 清除 DMAX ERROR 中断的使能	0x0000	R/W
30~16	Revered	Revered		
15:0	CHINTCLR[C*]	写入时: 0: 不起作用 1: 清除通道中断的使能	0	WO

Note : C\* : 15~0 通道



### 32.7-20 DMAX 通道传输完成中断旗标寄存器(DMAX\_CHTCIF)

地址偏移: 0X0068

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW															

位	标记	功能描述	复位值	读写
31	ERRIF	读取时: 1: DMAX 传输中发生错误中断旗标 写入时 1: 清除旗标为 0	0	RW
30:16	Revered	Revered	0x0000	RW
15:0	CHTCIF[C]	读取时: 1: 通道(C) 传输完成产生的中断旗标 写入时 1: 清除旗标为 0	0	RW

Note : C\* : 15~0



### 32.7-21 DMAX 通道等待请求使能寄存器(DMAX\_CHWTONREG)

地址偏移: **0X006C**

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CHWTONREQ[C*]																
RW																

位	标记	功能描述	复位值	读写
31:16	Revered	Revered	0x0000	RW
15:0	CHWTONREQ[C*] (DMAx_waitonreg[C])	当 Burst 传输时:[1*] 0 : 通道(C*)每读写一笔数据都须设定信道配置 1 : 通道(C*),DMAX 连续传输.	0	RW

Note : C\* : 15~0

[1\*]

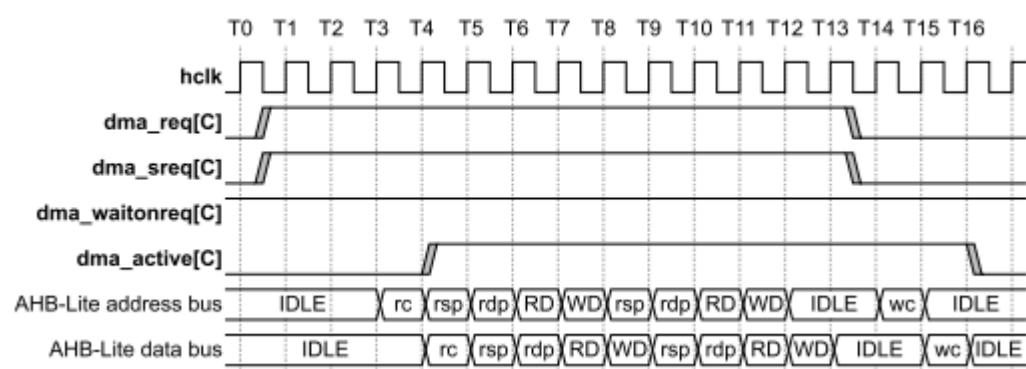


Figure 32-12 CHWTONREQ[C] ( DMAx\_waitonreg[C]=1 ) 的时序图

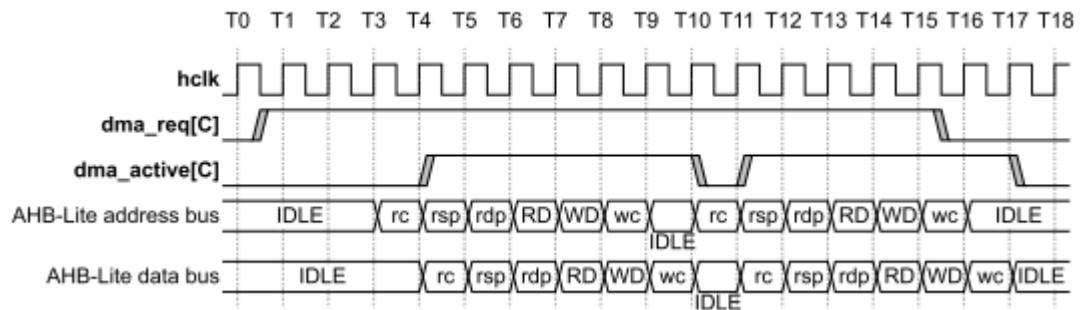


Figure 32-13 CHx\_CHWTREQ=0 的时序图

ZBT CONFIDENTIAL



### 32.7-22 DMAX 通道选择请求寄存器(DMAX\_CHSELREQ)

地址偏移: **0X0070**

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSELREQ[C*]															
RW															

位	标记	功能描述	复位值	读写
31:16	Revered	Revered	0x0000	RW
15:0	CHSELREQ[C*]	DMAX 传输时: 0 : 通道(C) 为原始的请求(Default) 1 : 信道(C)为选项的请求	0x0000	RW

Note : C\* : 15~0 通道



### 32.7-23 DMAX 集成配置寄存器(DMAX\_INTCFG)

地址偏移: 0xE00

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Revered															
RO															

位	标记	功能描述	复位值	读写
31:1	--	Revered	0x0000	--
0	DMAX_INTCFGEN	使能测试讯号 0:禁用。 1: 使能测试控制讯号状态 (1) DMAx_active[] (2) DMAx_done[] (3) DMAx_err.	0	RO



### 32.7-24 DMAX 停滞状态寄存器(DMAX\_STALL)

地址偏移: 0xE08

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Revered															
RO															

位	标记	功能描述	复位值	读写
31:1	--	Revered	0x0000	--
0	DMAX_STALL	使能测试讯号 0:正常工作 1: DMAX 停滞状态	0	RO



### 32.7-25 DMAX 请求状态 (DMAX\_CHREQ)

地址偏移: 0xE10

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15	CHREQ[C*]	0: 通道(C)为没有任何请求 1: 通道(C)为请求状态	0	RO
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
0	CHREQ[0]	0: 通道(0)为没有任何请求 1: 通道(0)为请求状态	0	RO

Note : C\* : 15~0 通道



### 32.7-26 DMAX 通道单笔请求状态 (DMAX\_CHSREQ)

地址偏移: 0xE18

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHSREQ[C*]	0: 通道(C)没有单笔请求 1: 通道(C)为单笔请求状态	0	RO

Note : C\* : 15~0 通道



### 32.7-27 DMAX 通道传输完成的状态设置寄存器(DMAX\_CHDONESET)

地址偏移: 0xE20

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	DONESET[C*]	读取时: 0: 通道(C*) 无任何状态 1: 通道(C*) 传输完成 写入时: 0: 不起作用 1: 使用此位为通道(C*) 传输完成状态 (需 DMAX_INTCFGEN = 1)	0	R/W

Note : C\* : 15~0 通道



### 32.7-28 DMAX 通道传输完成状态清 0 寄存器(DMAX\_CHDONECLR)

地址偏移: 0xE24

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DONECLR[C*]															
WO															

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	DONECLR[C*]	写入时: 0: 不起作用 1: 清除通道(11) 的传输完成讯号 (DMAX_Done[11])为 0 (需 DMAX_INTCFGEN = 1)	0	WO

Note : C\* : 15~0 通道



### 32.7-29 DMAX 通道正在传输中的状态设置寄存器(DMAX\_CHACTSET)

地址偏移: 0xE28

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHACTSET[C*]	读取时: 0: 通道(C*) 无任何状态 1: 通道(C*) 传输中 写入时: 0: 不起作用 1: 使用此位为通道(C*) 传输中状态 (需 DMAX_INTCFGEN = 1)	0	R/W

Note : C\* : 15~0 通道



### 32.7-30 DMAX 通道正在传输中的状态清 0 寄存器(DMAX\_CHACTCLR)

地址偏移: 0xE2C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHACTCLR[C*]															
WO															

位	标记	功能描述	复位值	读写
31:16	--	Revered	0x0000	R/W
15:0	CHACTCLR[C*]	写入时: 0: 不起作用 1: 清除通道(C*) 的传输中讯号(DMAX_ACT[C*]) 为 0 (需 DMAX_INTCFGEN = 1)	0	WO

Note : C\* : 15~0 通道



### 32.7.31 DMAX 总线错误清除寄存器寄存器(DMAX\_ERRSET)

地址偏移: 0xE48

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:1	--	Revered	0x0000	R/W
0	ERRSET	设置总线错误状态。  写入时： 0: 不起作用。 1: 使能 ERROR	0	WO



## 【33】 QUAD-SPI 接口(QSPI)

### 33.1 概述

QUADSPI 是一种专门针对单、双或四 SPI 闪存的通信接口。它可以工作在以下三种模式中的任何一种：

- (7) 间接模式：所有操作都使用 QUADSPI 寄存器执行
- (8) 状态轮询模式：定期读取外部闪存状态寄存器，并在标志设置的情况下产生中断
- (9) 内存映像模式：将外部闪存映像到设备地址空间，系统将其视为内部存储器，使用双闪存模式，吞吐量和容量都可以增加两倍，其中两个 Quad-SPI 闪存可以同时访问。

### 33.2 QSPI 功能框图

QSPI 功能框图，使用 BANK1 见图 (Dual Flash Disable set QUADSPI\_CR.DFM=0):

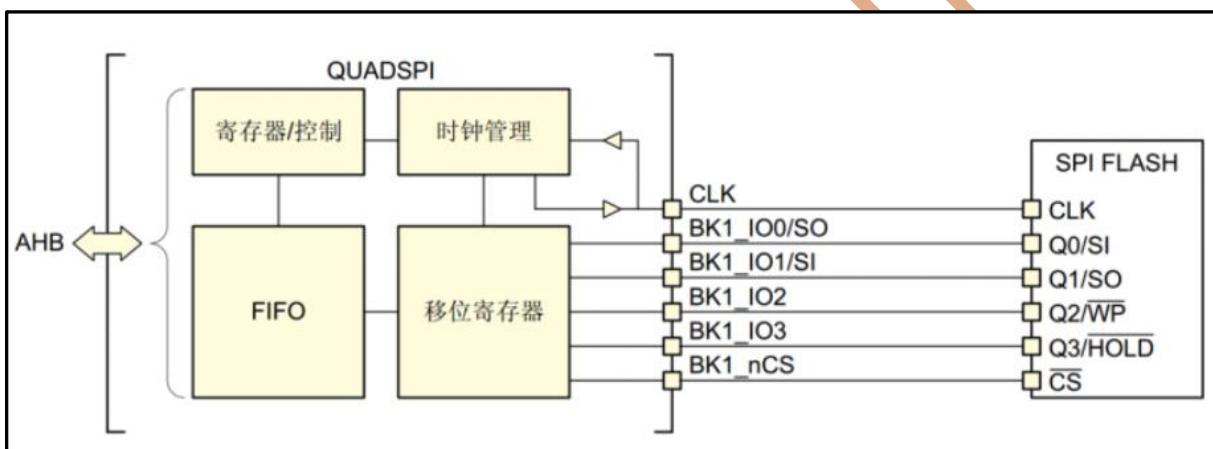


Figure 33-1 QSPI 功能框图(BANK1)

QSPI 使用 6 个信号连接 Flash，分别是四个数据线 BK1\_IO0~BK1\_IO3，一个时钟输出 CLK，一个晶片选输出（低电平有效）BK1\_nCS，它们的作用介绍如下：

- (1) **BK1\_nCS**: 晶片选输出（低电平有效），适用于 FLASH 1。如果 QSPI 始终在双闪存模式下工作，则其也可用于 FLASH 2 从设备选择信号线。QSPI 通讯以 BK1\_nCS 线置低电平为开始信号，以 BK1\_nCS 线被拉高作为结束信号。
- (2) **CLK**: 时钟输出，适用于两个存储器，用于通讯数据同步。它由通讯主机产生，决定了通讯的速率，不同的设备支持的最高时钟频率不一样，如 STM32 的 QSPI 时钟频率最大为  $f_{pclk}/2$ ，两个设备之间通讯时，通讯速率受限于低速设备。
- (3) **BK1\_IO0**: 在双线 / 四线模式中为双向 IO，单线模式中为串行输出，适用于 FLASH 1。
- (4) **BK1\_IO1**: 在双线 / 四线模式中为双向 IO，单线模式中为串行输入，适用于 FLASH 1。
- (5) **BK1\_IO2**: 在四线模式中为双向 IO，适用于 FLASH 1。
- (6) **BK1\_IO3**: 在四线模式中为双向 IO，适用于 FLASH 1。



QSPI 功能框图, 使用 BANK1,BANK2 见图 ((Dual Flash Enable set QUADSPI\_CR.DFM=1)):

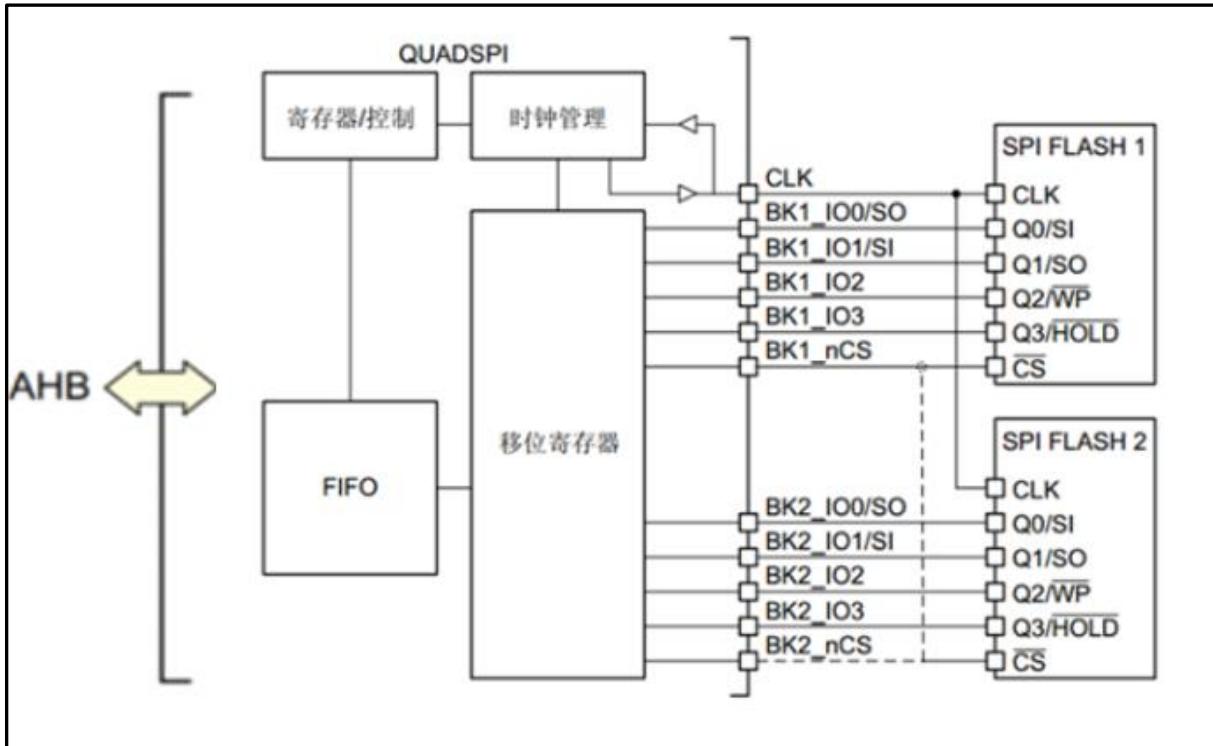


Figure 33-2 QSPI 功能框图(BANK1,BANK2)

- (7) BK2\_IO0/SO : 在 2 线或者 4 线模式中作为双向 IO, 1 线模式作为单向输出, 供 Flash2 使用。
- (8) BK2\_IO1/SI : 在 2 线或者 4 线模式中作为双向 IO, 1 线模式作为单向输入, 供 Flash2 使用。
- (9) BK2\_IO2: 在 4 线模式中作为双向 IO, 供 Flash2 使用。
- (10) BK2\_IO3 : 在 4 线模式中作为双向 IO, 供 Flash2 使用。
- (11) BK2\_nCS: 片选信号, 低电平有效, 供 Flash2 使用。如果工作在双 bank 模式下, 也可用于 Flash1。

ZBT



### 33.3 QSPI 主要特性

- (1) 三种功能模式：间接、状态轮询和内存映射
- (2) 双闪存模式，通过并行访问两个闪存，可以同时发送/接收 8 位数据
- (3) 间接和内存映像模式的完全可编程操作码.
- (4) 间接和内存映像模式的完全可编程帧格式
- (5) 用于接收和发送的集成 FIFO
- (6) 允许位数据访问 8,16,32
- (7) 用于间接模式操作的 DMAXX 信道
- (8) 在 FIFO 阈值、超时、操作完成和访问错误时产生中断

### 33.4 QSPI 命令序列

QUADSPI 通过命令与 Flash 通信 每条命令包括指令、地址、交替字节、空指令和数据这五个阶段 任一阶段均可跳过，但至少要包含指令、地址、交替字节或数据阶段之一。nCS 在每条指令开始前下降，在每条指令完成后再次上升。先看看 QSPI 四线模式下的读命令时序，见图 34-2

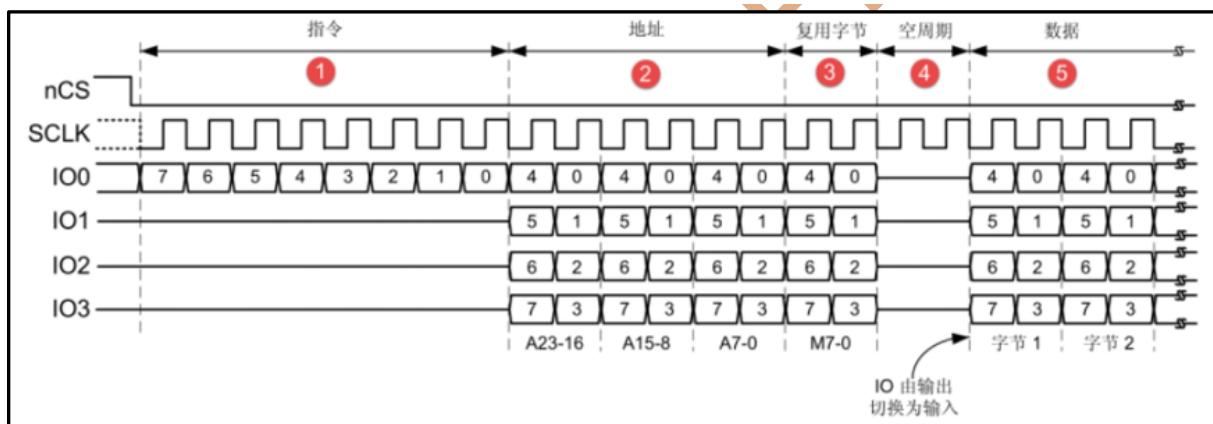


Figure 33-3 四线模式下的读命令时序

#### 33.4-1 地址阶段

在地址阶段，将 1-4 字节发送到 Flash，指示操作地址。待发送的地址字节数在 QUADSPI\_CCR[13:12] 寄存器的 ADSIZE[1:0] 字段中进行配置。在间接模式和自动轮询模式下，待发送的地址字节在 QUADSPI\_AR 寄存器的 ADDRESS[31:0] 中指定。在内存映像模式下，则通过 AHB（来自于 Cortex® 或 DMAXX）直接给出地址。地址阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI\_CCR[11:10] 寄存器中的 ADMODE[1:0] 字段进行配置。

若 ADMODE = 00，则跳过地址阶段，命令序列直接进入下一阶段（如果存在）。

#### 33.4-2 四线交替字节阶段

在交替字节阶段，将 1-4 字节发送到 Flash，一般用于控制操作模式。待发送的交替字节数在 QUADSPI\_CCR[17:16] 寄存器的 ABSIZE[1:0] 字段中进行配置。待发送的字节在 QUADSPI\_ABR 寄存器中指定。



交替字节阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI\_CCR[15:14] 寄存器中的 ABMODE[1:0] 字段进行配置。

若 ABMODE = 00，则跳过交替字节阶段，命令序列直接进入下一阶段（如果存在）。交替字节阶段存在仅需发送单个半字节而不是一个全字节的情况下，比如采用双线模式并且仅使用两个周期发送交替字节时。在这种情况下，固件可采用四线模式（ABMODE = 11）并发送一个字节，方法是 ALTERNATE 的位 7 和 3 置“1”（IO3 保持高电平）且位 6 和 2 置“0”（IO2 线保持低电平）。此时，半字节的高 2 位存放在 ALTERNATE 的位 4:3，低 2 位存放在位 1 和 0 中。例如，如果半字节 2 (0010) 通过 IO0/IO1 发送，则 ALTERNATE 应设置为 0x8A (1000\_1010)。

### 33.4-3 空指令周期阶段

在空指令周期阶段，给定的 1-31 个周期内不发送或接收任何数据，目的是当采用更高的时钟频率时，给 Flash 留出准备数据阶段的时间。这一阶段中给定的周期数在 QUADSPI\_CCR[22:18] 寄存器的 DCYC[4:0] 字段中指定。在 SDR 和 DDR 模式下，持续时间被指定为一定个数的全时钟周期。

若 DCYC 为零，则跳过空指令周期阶段，命令序列直接进入数据阶段（如果存在）。空指令周期阶段的操作模式由 DMODE 确定。为确保数据信号从输出模式转变为输入模式有足够的“周转”时间，使用双线和四线模式从 Flash 接收数据时，至少需要指定一个空指令周期。

### 33.4-4 数据阶段

在数据阶段，可从 Flash 接收或向其发送任意数量的字节。

在间接模式和自动轮询模式下，待发送/接收的字节数在 QUADSPI\_DLR 寄存器中指定。在间接写入模式下，发送到 Flash 的数据必须写入 QUADSPI\_DR 寄存器。在间接读取模式下，通过读取 QUADSPI\_DR 寄存器获得从 Flash 接收的数据。在内存映像模式下，读取的数据通过 AHB 直接发送回 Cortex 或 DMAXX。数据阶段可一次发送/接收 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI\_CCR[15:14] 寄存器中的 ABMODE[1:0] 字段进行配置。若 DMODE = 00，则跳过数据阶段，命令序列在拉高 nCS 时立即完成。这一配置仅可用于仅间接写入模式。



### 33.5 QUADSPI 信号接口协议模式

#### 33.5-1 单线 SPI 模式(Single SPI Mode)

传统 SPI 模式允许串行发送/接收单独的 1 位。在此模式下，数据通过 SO 信号（其 I/O 与 IO0 共享）发送到 Flash。从 Flash 接收到的数据通过 SI（其 I/O 与 IO1 共享）送达。通过将（QUADSPI\_CCR 中的）IMODE/ADMODE/ABMODE/DMODE 字段设置为 01，可对不同的命令阶段分别进行配置，以使用此单个位模式。在每个已配置为单线模式的阶段中：

- IO0 (SO) 处于输出模式
- IO1 (SI) 处于输入模式（高阻抗）
- IO2 处于输出模式并强制置 “0”（以禁止“写保护”功能）
- IO3 处于输出模式并强制置 “1”（以禁止“保持”功能）

若 DMODE = 01，这对于空指令阶段也同样如此。

#### 33.5-2 双线 SPI 模式(Dual SPI Mode)



在双线模式下，通过 IO0/IO1 信号同时发送/接收两位。通过将 QUADSPI\_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 字段设置为 10，可对不同的命令阶段分别进行配置，以使用双线 SPI 模式。在每个已配置为单线模式的阶段中：

- IO0/IO1 在数据阶段进行读取操作时处于高阻态（输入），在其他情况下为输出
- IO2 处于输出模式并强制置 “0”
- IO3 处于输出模式并强制置 “1”

在空指令阶段，若 DMODE = 01，则 IO0/IO1 始终保持高阻态。

#### 33.5-3 四线 SPI 模式(Quad SPI Mode)



在四线模式下，通过 IO0/IO1/IO2/IO3 信号同时发送/接收四位。通过将 QUADSPI\_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 字段设置为 11，可对不同的命令阶段分别进行配置，以使用四线 SPI 模式。在每个已配置为四线模式的阶段中，IO0/IO1/IO2/IO3 在数据阶段进行读取操作时均处于高阻态（输入），在其他情况下为输出。在空指令阶段中，若 DMODE = 11，则 IO0/IO1/IO2/IO3 均为高阻态。IO2 和 IO3 仅用于 Quad SPI 模式 如果未配置任何阶段使用四线 SPI 模式，即使 UADSPI 激活，对应 IO2 和 IO3 的引脚也可用于其他功能。





### 33.5-4 双闪存模式

若 DFM 位 (QUADSPI\_CR[6]) 为 1, QUADSPI 处于双闪存模式。QUADSPI 使用两个外部四线 SPI Flash (FLASH 1 和 FLASH 2), 在每个周期中发送/接收 8 位 (在 DDR 模式下为 16 位), 能够有效地将吞吐量和容量扩大一倍。每个 Flash 使用同一个 CLK 并可选择使用同一个 nCS 信号, 但其 IO0、IO1、IO2 和 IO3 信号是各自独立的。双闪存模式可与单比特模式、双比特模式以及四比特模式结合使用, 也可与 SDR 或 DDR 模式相结合。Flash 的大小在 FSIZE[4:0] (QUADSPI\_DCR[20:16]) 中指定, 指定的值应能够反映 Flash 的总容量, 即单个组件容量的 2 倍。如果地址 X 为偶数, QUADSPI 赋给地址 X 的字节是存放于 FLASH 1 的地址 X/2 中的字节, QUADSPI 赋给地址 X+1 的字节是存放于 FLASH 2 的地址 X/2 中的字节。也就是说, 偶地址中的字节存储于 FLASH 1, 奇地址中的字节存储于 FLASH 2。

在双闪存模式下读取 Flash 状态寄存器时, 需要读取的字节数是单闪存模式下的 2 倍。这意味着在状态寄存器获取指令到达后, 如果每个 Flash 给出 8 个有效位, 则 QUADSPI 必须配置为 2 个字节 (16 位) 的数据长度, 它将从每个 Flash 接收 1 个字节。如果每个 Flash 给出一个 16 位的状态, 则 QUADSPI 必须配置为读取 4 字节, 以在双闪存模式下可获取两个 Flash 的所有状态位。结果 (在数据寄存器中) 的最低有效字节是 FLASH 1 状态寄存器的最低有效字节, 而下一个字节是 FLASH 2 状态寄存器的最低有效字节。数据寄存器的第三个字节是 FLASH 1 的第二个字节, 第四个字节是 FLASH 2 的第二个字节 (Flash 具有 16 位状态寄存器时)。

偶数个字节必须始终在双闪存模式下访问。因此, 若 DRM = 1, 则数据长度字段 (QUADSPI\_DLR[0]) 的位 0 始终保持为 1。

在双闪存模式下, FLASH 1 接口信号的行为基本上与正常模式下相同。在指令、地址、交替字节以及空指令周期阶段, FLASH 2 接口信号具有与 FLASH 1 接口信号完全相同的波形。也就是说, 每个 Flash 总是接收相同的指令与地址。然后, 在数据阶段, BK1\_IOx 和 BK2\_IOx 总线并行传输数据, 但发送到 FLASH 1 (或从其接收) 的数据与 FLASH 2 中的不同。

### 33.6 QUADSPI 间接模式

在间接模式下, 通过写入 QUADSPI 寄存器来触发命令; 并通过读写数据寄存器来传输数据, 就如同对待其他通信外设那样。

若 FMODE = 00 (QUADSPI\_CCR[27:26]), 则 QUADSPI 处于间接写入模式, 字节在数据阶段中发送到 Flash。通过写入数据寄存器 (QUADSPI\_DR) 的方式提供数据。

若 FMODE = 01, 则 QUADSPI 处于间接读取模式, 在数据阶段中从 Flash 接收字节。通过读取 QUADSPI\_DR 来获取数据。

读取/写入的字节数在数据长度寄存器 QUADSPI\_DLR 中指定。

如果 QUADSPI\_DLR = 0xFFFF\_FFFF (全为 "1"), 则数据长度视为未定义, QUADSPI 将继续传输数据, 直到到达 (由 FSIZE 定义的) Flash 的结尾。如果不传输任何字节, DMODE (QUADSPI\_CCR[25:24]) 应设置为 00。如果 QUADSPI\_DLR = 0xFFFF\_FFFF 并且 FSIZE = 0x1F (最大值指示一个 4GB 的 Flash), 在此特殊情况下, 传输将无限继续下去, 仅在出现终止请求或 QUADSPI 被禁止后停止。在读取最后一个存储器地址后 (地址为 0xFFFF\_FFFF), 将从地址 = 0x0000\_0000 开始继续读取。



当发送或接收的字节数达到编程设定值时，如果 **TCIE = 1**，则 **TCF** 置 1 并产生中断。在数据数量不确定的情况下，将根据 **QUADSPI\_CR** 中定义的 Flash 大小，在达到外部 SPI 的限制时，**TCF** 置 1。

### 33.6-1 触发命令启动

从本质上讲，在固件给出命令所需的最后一点信息时，命令即会启动。根据 **QUADSPI** 的配置，在间接模式下有三种触发命令启动的方式。在出现以下情形时，命令立即启动：

- 1、对 **INSTRUCTION[7:0]** (**QUADSPI\_CCR**) 执行写入操作，如果没有地址是必需的（当 **ADMODE = 00**）并且不需要固件提供数据（当 **FMODE = 01** 或 **D MODE = 00**）；
- 2、对 **ADDRESS[31:0]** (**QUADSPI\_AR**) 执行写入操作，如果地址是必需的（当 **ADMODE = 00**）并且不需要固件提供数据（当 **FMODE = 01** 或 **D MODE = 00**）；
- 3、对 **DATA[31:0]** (**QUADSPI\_DR**) 执行写入操作，如果地址是必需的（当 **ADMODE != 00**）并且需要固件提供数据（当 **FMODE = 00** 并且 **D MODE != 00**）。

写入交替字节寄存器 (**QUADSPI\_ABR**) 始终不会触发命令启动。如果需要交替字节，必须预先进行编程。如果命令启动，**BUSY** 位 (**QUADSPI\_SR** 的位 5) 将自动置 1。

### 33.6-2 FIFO 和数据管理



在间接模式中，数据将通过 **QUADSPI** 内部的一个 32 字节 FIFO。

**FLEVEL[5:0](QUADSPI\_SR[13:8])** 指示 FIFO 目前保存了多少字节。

在间接写入模式下 (**FMODE = 00**)，固件写入 **QUADSPI\_DR** 时，将在 FIFO 中加入数据。字写入将在 FIFO 中增加 4 个字节，半字写入增加 2 个字节，而字节写入仅增加 1 个字节。如果固件在 FIFO 中加入的数据过多（超过 **DL[31:0]** 指示的值），将在写入操作结束（**TCF** 置 1）时从 FIFO 中清除超出的字节。

对 **QUADSPI\_DR** 的字节/半字访问必须仅针对该 32 位寄存器的最低有效字节/半字。**FTHRES[3:0]** 用于定义 FIFO 的阈值。如果达到阈值，**FTF** (FIFO 阈值标志) 置 1。在间接读取模式下，从 FIFO 中读取的有效字节数超过阈值时，**FTF** 置 1。从 Flash 中读取最后一个字节后，如果 FIFO 中依然有数据，则无论 **FTHRES** 的设置为何，**FTF** 也都会置 1。在间接写入模式下，当 FIFO 中的空字节数超过阈值时，**FTF** 置 1。

如果 **FTIE = 1**，则 **FTF** 置 1 时产生中断。如果 **DMAXXEN = 1**，则 **FTF** 置 1 时启动数据传送。如果阈值条件不再为“真”（CPU 或 **DMAXX** 传输了足够的数据后），则 **FTF** 由 HW 清零。在间接模式下，当 FIFO 已满，**QUADSPI** 将暂时停止从 Flash 读取字节以避免上溢。请注意，只有在 FIFO 中的 4 个字节为空 (**FLEVEL ≤ 11**) 时才会重新开始读取 Flash。因此，若 **FTHRES ≥ 13**，应用程序必须读取足够的字节以确保 **QUADSPI** 再次从 Flash 检索数据。否则，只要  $11 < \text{FLEVEL} < \text{FTHRES}$ ，**FTF** 标志将保持为“0”。



### 33.7 QUADSPI 状态轮询模式

在自动轮询模式下，QUADSPI 会定期启动命令以读取定义数量的状态字节（最多 4 个）。可以屏蔽接收到的字节以隔离一些状态位，并且当所选位具有定义的值时可以生成中断。对闪存的访问以与间接读取模式相同的方式开始：如果不需要地址 (AMODE=00)，则在写入 QUADSPI\_CCR 后立即开始访问。否则，如果需要地址，则在写入 QUADSPI\_AR 时开始第一次访问。BUSY 在这一点变高，即使在周期性访问之间也保持高电平。

MASK[31:0] (QUADSPI\_PSMAR) 的内容用于在自动轮询模式下屏蔽来自闪存的数据。如果 MASK[n] = 0，则结果的第 n 位被屏蔽而不考虑。如果 MASK[n] = 1，并且 bit[n] 的内容与 MATCH[n] (QUADSPI\_PSMAR) 相同，则表示第 n 位匹配。

如果轮询匹配模式位 (PMM, QUADSPI\_CR 的第 23 位) 为 0，则激活“与”匹配模式。这意味着仅当所有未屏蔽位都匹配时才设置状态匹配标志 (SMF)。

如果 PMM=1，则激活“OR”匹配模式。这意味着如果任何未屏蔽位匹配，则设置 SMF。

如果 SMIE=1，当 SMF 置位时调用中断。

如果设置了自动轮询模式停止 (APMS) 位，一旦检测到匹配，操作就会停止并且 BUSY 变为 0。否则，BUSY 保持为“1”并且周期性访问将继续，直到出现中止或 QUADSPI 被禁用 (EN=0)。

数据寄存器 (QUADSPI\_DR) 包含最新接收的状态字节 (FIFO 已停用)。数据寄存器的内容不受匹配逻辑中使用的屏蔽的影响。新的状态读取完成后，FTF 状态位立即置位，读取数据后立即清除 FTF。

### 33.8 QUADSPI 内存映像模式

在内存映像模式下配置时，外部 SPI 设备被视为内部存储器。在以下情况下使用内存映像模式

(1) 用于阅读操作

(2) 像使用内部 Flash 一样使用外部 QSPI Flash，任何 AHB 总线主控都可以自主读取数据

(3) 用于从外部 QSPI Flash 执行代码

在正确配置和启用 QUADSPI 外设之前，禁止访问 QUADSPI Flash bank 区域。

即使闪存容量更大，也不能超过 256MB。(在内存映射模式下地址范围是 0x9000 0000~0x9FFF FFFF)

如果访问的地址超出 FSIZE 定义的范围但仍在 256MB 范围内，则会出现总线错误。

此错误的影响取决于尝试访问的总线主控器：

(1) 如果是 Cortex® CPU，则启用时产生总线故障异常（或禁用总线故障时产生硬故障异常）

(2) 如果是 DMAXX，则会产生 DMAXX 传输错误，并自动禁用相应的 DMAXX 通道。

内存映像模式字节、半字和字访问类型均受支持。

实现了对就地执行 (XIP) 操作的支持，其中 QUADSPI 预计下一次访问并提前加载以下地址处的字节。

如果后续访问确实是在一个连续的地址上进行的，那么访问将更快地完成，因为该值已经被预取了默认情况下，QUADSPI 从不停止其预取操作，在 nCS 保持低电平的情况下保持先前的读取操作处于活动状态，即使长时间未访问闪存也是如此。由于当 nCS 保持低电平时闪存往往消耗更多，因此应用程序可能需要激活超时计数器 (TCEN=1, QUADSPI\_CR 的第 3 位)，以便在 TIMEOUT[15:0]



(QUADSPI\_LPTR) 一段时间后释放 nCS 自从 FIFO 被预取数据填满以来，已经过去了几个周期而没有任何访问

第一次内存映射访问发生后，BUSY 变为高电平。由于预取操作，BUSY 不会下降，直到超时、中止或外设被禁用。

### 33.9 QUADSPI Flash 配置

外部 SPI Flash 的参数可以通过配置寄存器 (QUADSPI\_DCR) 实现。这里配置 Flash 的容量是设置 FSIZE[4:0] 字段，使用下面的公式定义外部存储器的大小：

FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，Flash 容量最高可达 4GB（使用 32 位进行寻址），但在内存映像模式下的可寻址空间限制为 256MB。如果 DFM = 1，FSIZE 表示两个 Flash 容量的总和。QUADSPI 连续执行两条命令时，它在两条命令之间将片选信号 (nCS) 为高电平默认仅一个 CLK 周期时长。如果 Flash 需要命令之间的时间更长，

可使用片选高电平时间 (CSHT) 字指定 nCS 必须保持高电平的最少 CLK 周期数（最大为 8）。

时钟模式 (CKMODE) 位指示命令之间的 CLK 信号逻辑电平 (nCS = 1 时)。

### 33.10 QUADSPI 延迟数据采样数据进行采样

默认情况下，QUADSPI 在 Flash 驱动信号后半个 CLK 周期对 Flash 驱动的，在外部信号延迟的情况下，稍后对数据进行采样可能是有益的。使用 SSHIFT 位 (QUADSPI\_CR 的第 4 位)，可以将数据采样移位半个 CLK 周期。

DDR 模式不支持时钟移位：当 DDRM 位被设置时，SSHIFT 位必须清零

### 33.11 QUADSPI 配置

QUADSPI 配置分两个阶段完成：

- (1) QUADSPI IP 配置。
- (2) QUADSPI 闪存配置

配置并启用后，QUADSPI 可用于三种操作模式之一：间接模式、状态轮询模式或存储器映像模式



### 33.12 QUADSPI 用法

使用 FMODE[1:0] (QUADSPI\_CCR[27:26]) 选择工作模式.

#### 间接模式程序

当 FMODE 被编程为 00 时，间接写入模式被选择并且数据可以被发送到闪存。当 FMODE=01 时，选择间接读取模式，可以从 Flash 存储器读取数据. 在间接模式下使用 QUADSPI 时，帧的构造方式如下:

- (1) 在 QUADSPI\_DLR 中指定要读取或写入的数据字节数.
- (2) 在 QUADSPI\_CCR 中指定帧格式、模式和指令代码.
- (3) 指定在 QUADSPI\_ABR 中的地址阶段之后立即发送的可选备用字节.
- (4) 在 QUADSPI\_CR 中指定操作模式。如果 FMODE=00 (间接写入模式) 且 DMAXXEN=1，则 QUADSPI\_AR 应在 QUADSPI\_CR 之前指定，否则 QUADSPI\_DR 可能在 QUADSPI\_AR 更新之前由 DMAXX 写入 (如果 DMAXX 控制器已启用)
- (5) 指定目标 QUADSPI\_AR 中的地址.
- (6) 通过 QUADSPI\_DR 从/向 FIFO 读取/写入数据.

写入控制寄存器 (QUADSPI\_CR) 时，用户指定以下设置:

- SE 使能位 (EN) 设置为 "1"
- 从 RAM 到/传送数据的 DMAXX 使能位 (DMAXXEN)
- 超时计数器使能位 (TCEN)
- 样本移位设置 (SSSHIFT)
- 样本移位设置 (SSSHIFT)
- 中断使能

在写入通信配置寄存器 (QUADSPI\_CCR) 时，用户指定以下参数:

- 指令字节通过 INSTRUCTION 位
- 指令必须通过 IMODE 位 (1/2/4 行) 发送的方式
- 地址必须通过 ADMODE 位发送的方式 (无/1/2/4 行)
- 地址大小 (8/16/24/32 位) 通过 ADSIZE 位
- 交替字节数 (1/2/3/4) 通过 ABSIZE 位
- 通过 DBMODE 位的虚拟字节的存在与否
- 通过 DCYC 位的哑字节数
- 必须通过 DMODE 位发送/接收数据的方式 (无/1/2/4 行)

如果地址寄存器 (QUADSPI\_AR) 和数据寄存器 (QUADSPI\_DR) 都不需要为特定命令更新，则命令序列在写入 QUADSPI\_CCR 后立即开始。当 ADMODE 和 DMODE 均为 00 时，或者在间接读取模式 (FMODE=01) 下仅 ADMODE=00 时，就会出现这种情况。

当需要地址 (ADMODE 不是 00) 并且不需要写入数据寄存器时 (当 FMODE=01 或 DMODE=00 时)，一旦地址通过写入 QUADSPI\_AR 进行更新，命令序列就会开始。



在数据传输的情况下 (FMODE=00 和 DMODE!=00), 通信开始由通过 QUADSPI\_DR 写入 FIFO 触发.

#### 状态标志轮询模式

状态标志轮询模式被启用, 将 FMODE 字段 (QUADSPI\_CCR[27:26]) 设置为 10。在此模式下, 将发送编程帧并定期检索数据. 每帧读取的最大数据量为 4 个字节。如果在 QUADSPI\_DLR 中请求更多数据, 它将被忽略并且只会读取 4 个字节.

周期在 QUADSPI\_PISR 寄存器中指定, 一旦检索到状态数据, 就可以在内部对其进行处理, 以便:

- 设置状态匹配标志并在启用时产生中断
- 自动停止状态字节的周期性检索

接收到的值可以用存储在 QUADSPI\_PSMKR 中的值进行屏蔽, 并与存储在 QUADSPI\_PSMAR 中的值进行 ORed 或 AND 运算. 在匹配的情况下, 状态匹配标志被置位, 如果使能则产生中断, 如果 AMPS 位被置位, QUADSPI 可以自动停止. 无论如何, 最新检索到的值在 QUADSPI\_DR 中可用.

#### 内存映像模式

在内存映像模式下, 外部闪存被视为内部存储器, 但在访问期间有一些延迟。在这种模式下, 只允许对外部 Flash 进行读操作.

通过将 QUADSPI\_CCR 寄存器中的 FMODE 设置为 11 来进入存储器映像模式.

当主设备访问存储器映射空间时发送编程指令和帧, FIFO 用作预取缓冲区以预测线性读取, 对任何访问都在此模式下 QUADSPI\_DR 的返回零.

数据长度寄存器 (QUADSPI\_DLR) 在内存映像模式下没有意义.



### 33.13 只发送一次指令

一些闪存（例如 Winbound）可能提供一种模式，在这种模式下，一条指令必须只与第一个命令序列一起发送，而后续命令则直接从地址开始。可以使用 SIOO 位 (QUADSPI\_CCR[28]) 来利用这一特性。SIOO 适用于所有功能模式（间接、自动轮询和内存映射）。如果设置了 SIOO 位，则仅针对写入 QUADSPI\_CCR 之后的第一个命令发送指令。随后的命令序列跳过指令阶段，直到写入 QUADSPI\_CCR。

当 IMODE=00（无指令）时，SIOO 无效

### 33.14 QUADSPI 错误管理

在以下情况下会产生错误：

- 在间接模式或状态标志轮询模式下，当在 QUADSPI\_AR 中编程了错误的地址时（根据 QUADSPI\_DCR.FSIZE[4:0] 中定义的闪存大小）这将设置 TEF 并在产生中断启用时。
- 同样在间接模式下，如果地址加数据长度超过 Flash 存储器大小，则一旦触发访问，TEF 将被设置。
- 在内存映像模式下，当主机执行超出范围的访问或禁用 QUADSPI 时：这将生成总线错误作为对故障总线主机请求的响应。
- 当一个主设备在内存映像模式被禁用时访问内存映射空间：这将产生一个总线错误作为对错误总线主设备请求的响应。

### 33.15 QUADSPI 忙位和中止功能

一旦 QUADSPI 开始对闪存进行操作，QUADSPI\_SR 中的 BUSY 位将自动设置，在间接模式下，一旦 QUADSPI 完成请求的命令序列并且 FIFO 为空，BUSY 位就会复位。在自动轮询模式下，由于 APMS=1 时的匹配或由于中止，BUSY 仅在最后一次定期访问完成后变低。

在内存映像模式下第一次访问后，BUSY 仅在超时事件或中止时变低。

可以通过设置 QUADSPI\_CR 中的 ABORT 位来中止任何操作。一旦中止完成，BUSY 位和 ABORT 位自动复位，FIFO 被刷新。

注意：如果对状态寄存器的写操作被中止，某些闪存可能会出现错误行为。



### 33.16 nCTS 行为

默认情况下，nCS 为高电平，取消选择外部闪存。nCS 在操作开始前下降，并在操作完成后立即上升，当 CKMODE=0 (“mode0”，此时没有操作时 CLK 保持低电平) nCS 在操作第一个 CLK 上升沿前下降一个 CLK 周期，nCS 在操作最后一个 CLK 上升沿后上升一个 CLK 周期，如图所示图 34-6

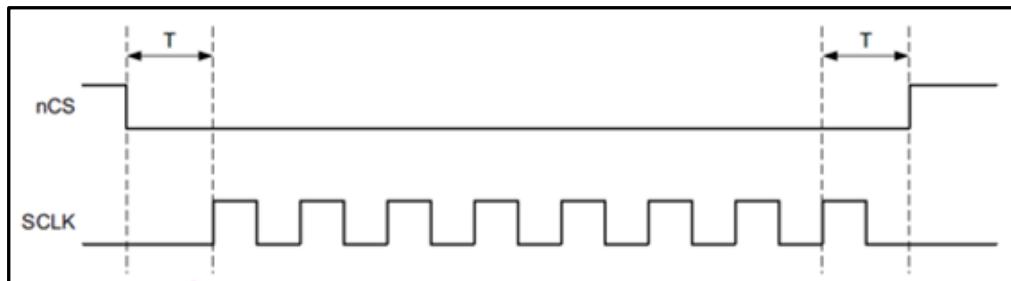


Figure 33-4 nCS when CKMODE=0 ( $T = \text{CLK period}$ )

当 CKMODE=1 (“mode3”，此时 CLK 在没有操作时变高)，nCS 在操作的第一个 CLK 上升沿前仍下降一个 CLK 周期，nCS 在操作最后一个 CLK 上升沿后上升一个 CLK 周期，如图如图 34-7 所示

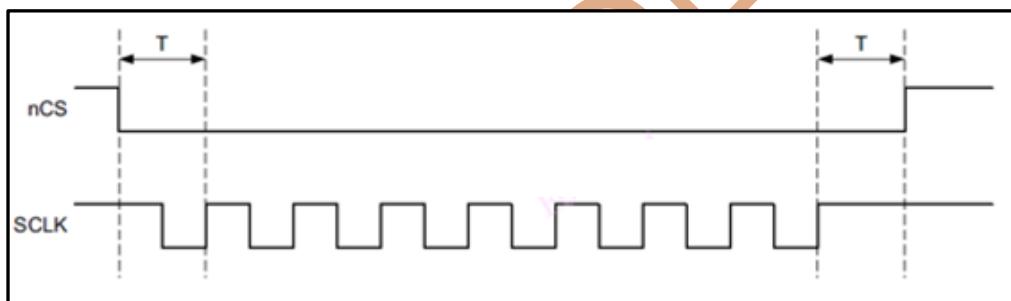


Figure 33-5 nCS when CKMODE=1 in SDR mode ( $T=\text{CLK period}$ )

当 FIFO 在读操作中保持满状态或如果 FIFO 在写操作中保持空状态，操作将停止并且 CLK 保持低电平，直到固件为 FIFO 提供服务。如果在操作停止时发生中止，nCS 会在请求中止后立即上升，然后 CLK 会在半个 CLK 周期后上升，如图 34-8 所示

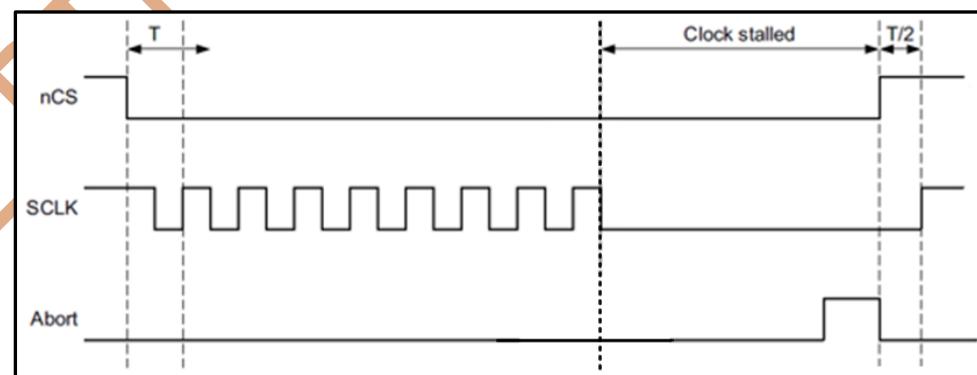


Figure 33-6 nCS when CKMODE=1 with an abort ( $T=\text{CLK period}$ )



当不处于双闪存模式（DFM=0）时，仅访问 FLASH 1，因此 BK2\_nCS 保持高电平。在双闪存模式下，BK2\_nCS 的行为与 BK1\_nCS 完全相同。因此，如果有 FLASH 2 且应用程序始终处于双闪存模式，则 FLASH 2 可以使用 BK1\_nCS，而输出 BK2\_nCS 的引脚可以用于其他功能。

### 33.17 QUADSPI 中断

以下事件可产生中断(单独的中断使能位以提高灵活性)：

Interrupt event	Event flag	Enable control bit
Timeout	TOF	TOIE
Status match	SMF	SMIE
FIFO threshold	FTF	FTIE
Transfer complete	TCF	TCIE
Transfer error	TEF	TEIE

Table 33-1 QSPI 中断表



### 33.18 QSPI 寄存器列表

基址: 0x4001 4000

偏移地址	名称	描述	访问类型	默认值
0x000	QUADSPI_CR	状态寄存器	RW	0x0000 0000
0x004	DMAXX_DCR	装置配置寄存器	RW	0x0000 0000
0x008	QUADSPI_SR	状态寄存器	RW	0x0000 0000
0x00C	QUADSPI_FCR	标志清除寄存器	RW	0x0000 0000
0x010	QUADSPI_DLR	数据长度寄存器	RW	0x0000 0000
0x014	QUADSPI_CCR	通信配置寄存器	RW	0x0000 0000
0x018	QUADSPI_AR	地址寄存器	RW	0x0000 0000
0x01C	QUADSPI_ABR	选择字节寄存器	RW	0x0000 0001
0x020	QUADSPI_DR	数据寄存器	RW	0x0000 0001
0x024	QUADSPI_PSMKR	轮询屏蔽寄存器	RW	0x0000 0000
0x028	QUADSPI_PSMAR	轮询状态匹配寄存器	RW	0x0000 0000
0x02C	QUADSPI_PIR	轮询间隔寄存器	RW	0x0000 0000
0x030	QUADSPI_LPTR	低功耗超时寄存器	RW	0x0000 0000

Table 33-2 QSPI 寄存器列表和复位值



### 33.19 QSPI 寄存器列表

#### 33.19-1 QSPI 控制寄存器 1(QUADSPI\_CR)

偏移地址: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	ro	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			FTHRES					FSEL	DFM	Res.	SSHIFT	TCEN	N	DMEA	ABORT
ro	ro	ro	rw	rw	rw	rw	rw	rw	rw	ro	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:24	PRFTBS	QSPI 时钟预除器(基于AHB时钟): 0: QCLK = FAHB (AHB 的时钟) 1:QCLK = FAHB/2 2:QCLK = FAHB/3 ... 255: QCLK = FAHB/256  Note : 设置需在BUSY=1.	0x00	R/W
23	PMM	轮询匹配模式 该位指示在期间应使用哪种方法来确定“匹配, 自动轮询模式” 0: AND 匹配模式。如果从闪存接收到的所有未屏蔽位则 SMF 被设置 与匹配寄存器中的相应位匹配 1: OR 匹配模式。如果从闪存接收到的任何未屏蔽位, 则 SMF 被设置与匹配寄存器中的相应位匹配。 仅当 BUSY = 0 时才能修改该位	0	R/W
22	APMS	自动轮询模式停止 该位确定在匹配后是否停止自动轮询 0: 仅通过中止或禁用 QUADSPI 来停止自动轮询模式; 1: 一匹配就停止自动轮询模式。	0	R/W



位	标记	功能描述	复位值	读写
		该位只能在BUSY = 0 时修改。		
21	Revred	--	0	R/W
20	TOIE	超时中断使能 0: 中断禁止 1: 中断使能		R/W
19	SMIE	状态匹配中断使能 该位使能状态匹配中断 0: 中断禁止 1: 中断使能	0	R/W
18	FTIE	FIFO 阈值中断使能 该位使能 FIFO 阈值中断 0: 禁止中断; 1: 允许中断。	0	R/W
17	TCIE	传输完成中断使能 该位使能传输完成中断。 0: 中断禁止 1: 中断使能	0	R/W
16	TEIE	传输错误中断使能 该位使能传输错误中断。 0: 中断禁止 1: 中断使能	0	R/W
15:13	Revered	--	0x0	R/W
12:8	FTHRES[4:0]	FTHRES[4:0] FIFO 阈值水平 在间接模式下, 定义 FIFO 中将导致的阈值字节数 FIFO 阈值标志 (FTF、QUADSPI_SR[2]) 置位。 在间接写入模式下 (FMODE = 00): 0: 如果 FIFO 中有 1 个或更多可用字节可写入, 则设置 FTF 1: 如果有 2 个或更多可用字节可写入, 则设置 FTF FIFO FIFO 中 ...	0x0	R/W



位	标记	功能描述	复位值	读写
		<p>31: 如果有 32 个可用字节可供写入，则设置 FTF</p> <p><b>在间接读取模式下 (FMODE = 01) :</b></p> <p>0: 如果有 1 个或多个可写入的有效字节，则设置 FTF 从 FIFO 读取</p> <p>1: 如果可以从 FIFO 中读取 2 个或更多有效字节， ... 则设置 FTF31: 如果可以从 FIFO 中读取 32 个有效字节，则设置 FTF</p> <p>如果 DMAXXEN = 1, 那么必须禁用相应通道的 DMAXX 控制器在更改 FTHRES 值之前。</p>		
7	FSEL	<p>Flash 存储器选择</p> <p>该位选择要在单 Flash 模式下(当 DFM = 寻址的 Flash 存储器0)</p> <p>0: 选择 FLASH 1选择</p> <p>1: FLASH 2</p> <p>该位只有在 BUSY = 0 时才能修改。</p> <p>当 DFM = 1 时, 该位被忽略。</p>	0x0	R/W
6	DFM	<p>双闪存模式</p> <p>该位激活双闪存模式，其中两个外部 Flash 使用内存同时使吞吐量和容量翻倍。</p> <p>0: 禁用双闪模式</p> <p>1: 启用双闪模式</p> <p>该位只能在 BUSY = 0 时修改。</p>	0x0	R/W
5	Revered	--	0x0	R/W
4	SSSHIFT	<p>采样移位</p> <p>默认情况下, QUADSPI 在数据被驱动后采样 1/2 个 CLK 周期的数据</p> <p>闪存。该位允许稍后对数据进行采样以解决外部信号延迟。</p> <p>0: 无移位</p> <p>1: 1/2 周期移位</p> <p>固件必须确保在 DDR 模式下 (当 DDRM = 1 时) SSHIFT = 0。</p> <p>只有当 BUSY = 0 时才能修改该字段</p>	0x0	R/W



位	标记	功能描述	复位值	读写
3	TCEN	<p>超时计数器使能 该位仅在选择内存映像模式 (FMODE = 11) 时有效。激活 如果则该位会导致片选 (nCS) 被释放 (从而减少消耗) 在一定时间后没有访问，， 该时间 由 TIMEOUT[15:0] (QUADSPI_LPTR) 定义。 启用超时计数器。 默认情况下， QUADSPI 从不停止其预取操作，在情况下保持先前的读取 nCS 保持低电平的操作处于活动状态，即使没有访问闪存 长时间也是如此。由于当 nCS 保持时闪存往往消耗更多 低电平，因此应用程序可能需要激活超时计数器 (TCEN = 1, 3 位 QUADSPI_CR 的第 )，以便在 TIMEOUT[15:0] 一段时间后释放 nCS (QUADSPI_LPTR)自从 FIFO以来，已经过去了没有访问的周期 被预取数据填满。 0: 超时计数器禁用，因此，片选 (nCS) 将保持活动状态 在存储器映像模式下访问后无限期。 1: 超时计数器使能，因此在，芯片选择在存储器映射被释放 闪存不活动的 TIMEOUT[15:0] 周期后模式下。 仅当 BUSY = 0 时才能修改该位</p>	0x0	R/W
2	DMAXXEN	<p>DMAXX 使能 在间接模式下， DMAXX 可用于通过 QUADSPI_DR 输入或输出数 据 寄存器。 DMAXX 传输在 FIFO 阈值标志 FTF 置位时启动。 0: DMAXX 在间接模式下禁用 1: DMAXX 在间接模式下启用</p>	0x0	R/W
1	ABORT	<p>中止请求 该位中止正在进行的命令序列。一旦中止它会自动重置 完成，。 该位停止当前传输。 在轮询模式或内存映像模式下，该位还会复位 APM 位或 DM 位。 0: 没有中止请求 1: 中止请求</p>	0x0	R/W



位	标记	功能描述	复位值	读写
0	EN	使能 QUADSPI。 0: 禁止QUADSPI 1: 使能 QUADSPI	0x0	R/W

ZBT CONFIDENTIAL



### 33.19-2 QSPI 控制寄存器 1(QUADSPI\_DCR)

偏移地址: 0x04 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:21	Reserved		0x0	R/W
20:16	FSIZE [4:0]	闪存大小  该字段使用以下公式定义外部存储器的大小：  闪存中的字节数 = $2[\text{FSIZE}+1]$ FSIZE+1 实际上是地址位数需要寻址闪存 Flash 内存容量在最高可达 4GB (使用 32 位)  间接模式下寻址，但内存映像模式下的可寻址空间限制为 256MB。  如果 DFM = 1, FSIZE 表示两个 Flash 存储器的总容量。  该字段只能在 BUSY = 0 时修改。	0	R/W
15:11	Reserved	--	--	--
10:8	CSHT [2:0]	芯片选高电平时间  CSHT+1 定义了片选 (nCS) 必须的最小 CLK 周期数 在向闪存发出命令之间保持高电平。  0: nCS 在闪存命令之间保持高电平至少 1 个周期在闪存命令之间 1: nCS 保持高电平至少 2 个周期 ... 7: nCS 在闪存命令之间保持高电平至少 8 个周期  此字段可以修改仅当 BUSY = 0 时。		R/W



位	标记	功能描述	复位值	读写
7:1	Reserved	--		--
0	CKMODE	<p>模式 0 / 模式 3 该位指示 CLK 在命令之间（当 nCS = 1 时）的电平。 0: 当 nCS 为高电平时 CLK 必须保持低电平（片选释放）。这被称为 模式 0。 1: 当 nCS 为高电平时，CLK 必须保持高电平（片选释放 ）。这称为 模式 3。 仅当 BUSY = 0 时才能修改此字段</p>		R/W



### 33.19-3 QSPI 状态寄存器(QUADSPI\_DCR)

偏移地址: 0x08 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FLEVEL				Reserved		BUSY	TOF	SMF	FTF	TCF	TEF			
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro

位	标记	功能描述	复位值	读写
31:14	Reserved		0x0	RO
13:8	FLEVEL[5 :0]	<p>FIFO 级别 该字段给出了 FIFO 中保存的有效字节数。FLEVEL = 0，当 FIFO 为空时当 FIFO 已满时为 32。在内存映像模式和自动状态轮询模式下，FLEVEL 为零。</p>	0	RO
7:6	Reserved	--	--	RO
5	BUSY	<p>Busy 当操作正在进行时设置该位。当时，该位自动清零</p>		RO
4	TOF	<p>Flash 存储器操作完成且 FIFO 为空 超时标志 该位在超时发生时置位。它通过向 CTOF 写入 1 来清除</p>		RO
3	SMF	<p>状态匹配标志 当未屏蔽的接收数据与中的匹配时，该位在自动轮询模式下设置 匹配寄存器 (QUADSPI_PSMAR)相应位。它通过写入 1 向 CSMF 来清除</p>		RO



位	标记	功能描述	复位值	读写
2	FTF	<p>FIFO 阈值标志</p> <p>在间接模式下，当达到 FIFO 阈值或还有时，该位被设置</p> <p>从 Flash 存储器读取完成后 FIFO 中任何数据。它就会清除</p> <p>一旦阈值条件不再为真，自动。</p> <p>在自动轮询模式下，每次读取状态寄存器时都会设置该位，</p> <p>读取该位</p> <p>数据寄存器时会清除</p>		RO
1	TCF	<p>传输完成标志</p> <p>该位在间接模式下在编程数量的数据时</p> <p>传输或在传输中止时在任何模式下设置通过写入CTCF位 1</p> <p>清除</p>		RO
0	TEF	<p>传输错误标志</p> <p>当在间接访问无效地址时，该位在间接模式下设置</p> <p>模式下。它通过向 CTEF 写入 1 来清除。</p>		RO



### 33.19-4 QSPI 标志清除寄存器(QUADSPI\_FCR)

偏移地址: 0x0C 复位值: 0x0000 0000

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

	记	功能描述	位值	写
31:5	Reserved	--	0x0	R
4	CTOF	清除超时标志位 写入1清除 QUADSPI_SR 寄存器中的 TOF 标志	0	R/W
3	CSMF	清除状态匹配标志位 写入1清除 QUADSPI_SR 寄存器中的 SMF 标志	--	--
2	Reserved	--		R/W
1	CTCF	清除传输完成标志 写1清除 QUADSPI_SR 寄存器中的 TCF 标志	--	--
0	CTEF	清除传输错误标志 写1清除 QUADSPI_SR 寄存器中的 TEF 标志		R/W



### 33.19-5 QSPI 数据长度寄存器(QUADSPI\_DLR)

偏移地址: 0x010 复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:5	DL[31:0]	<p>在间接和状态轮询模式下要检索的数据数（值+1）。其值状态轮询模式应使用不大于 3（表示 4 个字节）。</p> <p>间接模式中的全 1 表示未定义的长度，其中 QUADSPI 将继续直到内存结束，如 FSIZE 所定义</p> <p>0x0000_0000: 要传输 1 个字节</p> <p>0x0000_0001: 要传输 2 个字节</p> <p>0x0000_0002: 要传输 3 个字节</p> <p>0x0000_0003: 要传输 4 个字节</p> <p>0xFFFF_FFFD: 要传输 1 个字节 (4,294,294 FFFD) 将传输 4,294,967,295 (4G-1) 个字节将传输</p> <p>0xFFFF_FFFF: 未定义长度直到闪存结束 (FSIZE 定义所有字节)</p> <p>如果 FSIZE = 0x1F，则无限期地继续阅读</p> <p>DL[0] 在双闪存模式 (DFM = 1) 中一直保持在 “1”，即使向该位写入 “0”，从而确保每次访问传输偶数个字节</p> <p>当处于内存映像模式 (FMODE = 10) 时，该字段无效</p> <p>仅当 BUSY = 0 时才能写入此字段</p>	0x0	R/W



### 33.19-6 QSPI 通信配置寄存器(QUADSPI\_CCR)

偏移地址: 0x014 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			SIO0	FMODE		DMODE		Res.	DCYC				ABSIZE		
ro	ro	ro	rw	rw	rw	rw	rw	ro	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE		ADSIZE		ADMODE		IMODE		INSTRUCTION							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:29	Reserved	Reserved, must be kept at reset value.	0x0	ro
28	SIO0	<p>S 仅发送指令一次模式 参见第 421 页的第 14.3.12 节：仅发送指令一次。该位 当 IMODE = 00 时，无效。</p> <p>0: 每次事务 发送指令 1：仅发送第一条指令的指令 仅当 BUSY = 0 时才能写入该字段。</p>	0x0	rw
27:26	FMODE	<p>功能模式 该字段定义 QUADSPI 功能操作模式。</p> <p>00: 间接写入模式 01: 间接读取模式 10: 自动轮询模式 11: 存储器映像模式 如果 DMXXEN = 1，则必须相应通道的 DMXX 控制器 在更改 FMODE 值之前禁用。 该字段只能在 BUSY = 0 时写入</p>	0x0	rw



位	标记	功能描述	复位值	读写
25:24	DMODE	<p>数据模式 该字段定义数据阶段的操作模式:</p> <p>00: 无数据 01: 单行行 数据 10: 两数据第行四行 11: 数据 该字段还决定了虚拟相操作模式。 只有当 <b>BUSY = 0</b> 时才能写入该字段</p>	0x0	rw
23	Revered	--	0	r/ w
22:18	DCYC[4:0]	<p>虚拟周期数 该字段定义虚拟阶段的持续时间。在 SDR 和 DDR 模式下, 它指定了 CLK 周期数 (0-31)。</p> <p>该字段只能在 <b>BUSY = 0</b> 时写入</p>	0x00	rw
17:16	ABSIZE[1:0]	<p>交替字节大(Alternate bytes size) 该位定义交替字节大小:</p> <p>00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes 该字段只能在 <b>BUSY = 0</b> 时写入</p>	0x0	rw
15:14	ABMODE	<p>交替字节模式(Alternate bytes mode) 该字段定义交替字节相位操作模式:</p> <p>00: No alternate bytes 01: Alternate bytes on a single line</p>	0x0	rw



位	标记	功能描述	复位值	读写
		10: Alternate bytes on two lines 11: Alternate bytes on four lines  该字段只能在 BUSY = 0 时写入		
13:12	ADSIZE	地址大小(Address size)  This bit defines address size: 00: 8-bit address 01: 16-bit address 10: 24-bit address 11: 32-bit address  This field can be written only when BUSY = 0.	0x0	rw
11:10	ADMODE	地址模式(Address mode)  该字段定义地址阶段的操作模式: 00: No address 01: Address on a single line 10: Address on two lines 11: Address on four lines  该字段只能在 BUSY = 0 时写入	0x0	rw
9:8	IMODE	指令模式( Instruction mode)  该字段定义了指令阶段的操作模式: 00: No instruction 01: Instruction on a single line 10: Instruction on two lines 11: Instruction on four lines  该字段只能在 BUSY 时写入 = 0	0x0	rw
7:0	INSTRUCTION	Instruction  要发送到外部 SPI 设备的指令  仅当 BUSY = 0 时才能写入此字段	0x00	rw



### 33.19-7 QSPI 地址寄存器(QUADSPI\_AR)

偏移地址: 0x018 复位值: 0x0000 0000

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

记	能描述	位值	写
31:0	ADDRESS  地址 要发送到外部闪存的地址 当 BUSY = 0 或 FMODE = 11 (存储器映射时, 忽略对该字段的写入模式)。 在双闪存模式下, ADDRESS[0] 自动固定为 “0”, 因为地址应始终为偶数	0x00000000	rw



### 33.19-8 QSPI 可选字节寄存器(QUADSPI\_ABR)

偏移地址: 0x01C 复位值: 0x0000 0000

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

	记	功能描述	位值	写
31:0	ALTERNATE	在地址之后立即发送到外部 SPI 设备的可选据。  仅当 BUSY = 0 时才能写入此字段。	0x00000000	rw



### 33.19-9 QSPI 数据寄存器(QUADSPI\_DR)

偏移地址: 0x020 复位值: 0x0000 0000

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

记	功能描述	位值	写
31:0  Data [31:0]	<p>要向/从外部 SPI 设备发送/接收的数据。</p> <p>在间接写入模式下, 写入该寄存器的数据在数据阶段发送到闪存之前先存储在 FIFO 中。如果 FIFO 太满, 则写入操作将暂停, 直到 FIFO 有足够的空间来接受正在写入的数据量。</p> <p>在间接读取模式下, 读取该寄存器提供 (通过 FIFO) 从闪存接收到的数据。如果 FIFO 没有读取操作所请求的字节数并且如果 BUSY=1, 则读取操作将暂停, 直到存在足够的数据或传输完成 (以先发生者为准)。</p> <p>在自动轮询模式下, 该寄存器包含从闪存中读取的最后数据 (无屏蔽)。</p> <p>支持对该寄存器的字、半字和字节访问。在间接写入模式下, 字节写入向 FIFO 添加 1 个字节, 半字写入 2, 字写入 4。类似地, 在间接读取模式下, 字节读取从 FIFO 中删除 1 个字节, 半字读取 2, 然后字读取 4.</p> <p>间接模式下的访问必须与该寄存器的底部对齐: 字节读取必须读取 DATA[7:0], 半字读取必须读取 DATA[15:0]。</p>	0x00000000	rw



### 33.19-10 QSPI 轮询屏蔽寄存器(QUADSPI\_PSMKR)

偏移地址: 0x024 复位值: 0x0000 0000

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

	记	描述	功能	位值	写
31:0	Mask [31:0]	<p><b>状态屏蔽</b> 应用于在轮询模式下接收的状态字节的屏蔽。 对于第 n 位： 0: 自动轮询模式接收的数据的第 n 位被屏蔽，匹配逻辑中不考虑其值 1: 自动轮询模式接收的数据的第 n 位未屏蔽，匹配时考虑其值逻辑 该字段只能在 BUSY = 0 时写入</p>		0x00000000	rw



### 33.19-11 QSPI 轮询状态匹配寄存器(QUADSPI\_PSMAR)

偏移地址: 0x028 复位值: 0x0000 0000

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

	记	功能描述	位值	写
31:0	MATCH [31:0]	状态匹配 要与屏蔽状态寄存器进行比较以获得匹配的值。 该字段只能在 BUSY = 0 时写	0x00000000	rw



### 33.19-12 QSPI 轮询间隔寄存器(QUADSPI\_PIR)

偏移地址: 0x02C 复位值: 0x0000 0000

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

	记	功能描述	位值	写
31:16	Reserved	--	0x0000	ro
15:0	INTERVAL	轮询间隔 在自动轮询阶段读取之间的 CLK 周期数。 仅当 BUSY = 0 时才能写入此字段。	0x0000	rw



### 33.19-13 QSPI 超时寄存器(QUADSPI\_LPTR)

偏移地址: 0x030 复位值: 0x0000 0000

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

	记 述	功能描	位值	写
31:16	Reserved	--	0x0000	ro
15:0	TIMEOUT	超时周期 在内存映像模式下每次访问后, QUADSPI 预取后续 字节并将这些字节保存在 FIFO 中。该字段指 示多少个 CLK 周期, 在 FIFO 变满之前 QUADSPI 等待直到它提 高 nCS, 从而使闪存 处于较低消耗状态。 仅当 BUSY = 0 时才能写入此字段。	0x0000	rw



## 【34】真随机数发生器 (TRNG)

### 34.1 概述

真随机数模块产生 32 位真随机数。

### 34.2 功能框图

TRNG 模块的数据流如下图示：

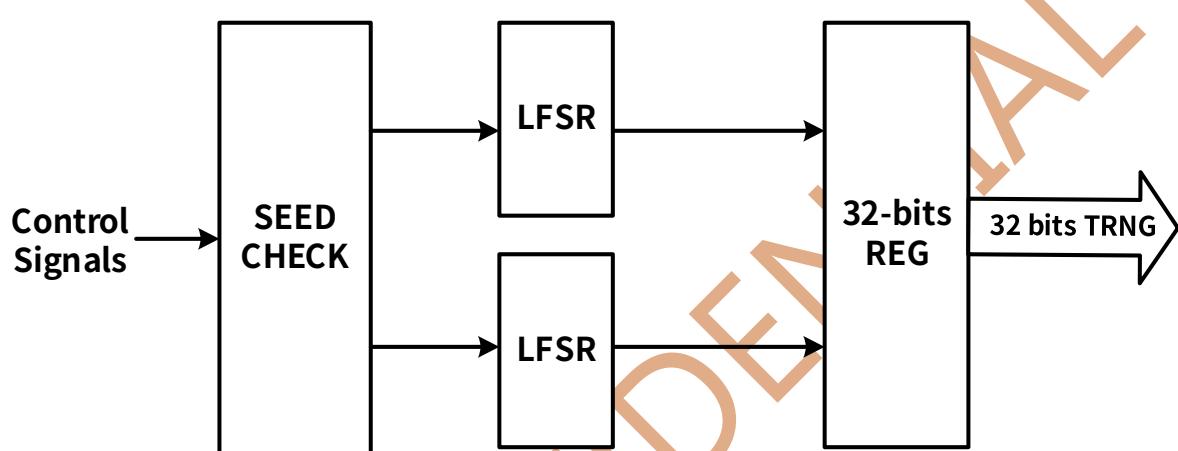


Figure 34-1 真随机数框图

### 34.3 功能描述

本模块采用内部的模拟随机源，每次启动都可以产生 32bits 真随机数,可与 DMAXX 请求完成真随机数.



#### 34.4 TRNG 寄存器列表

TRNG 基地址: 0x4002 2400

偏移地址	名称	描述	默认值
0x000	TRNG_CR	TRNG 控制寄存器	0x00000000
0x004	TRNG_SR	TRNG 状态寄存器	0x00000000
0x008	TRNG_DR	TRNG 数据寄存器	0x00000000

Table 34-1 寄存器列表和复位值



### 34.5 寄存器说明

#### 34.5-1 TRNG 控制寄存器(TRNG\_CR)

地址偏移: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
-															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
-										CED	TRNGD2EN	TRNGIE	TRNGEN	Reserved	
-										RW	-	RW	RW	-	

位	标记	功能描述	复位值	读写
31:6	Reserved	-	-	-
5	CED	时钟错误侦测 0: 打开时钟错误侦测 1: 关闭时钟错误侦测	0	RW
4	TRNGD2EN	0: 不使用 DMA2 1: 使用 DMA2 Note: 當 DRDY='1' & TRNGIE='1' & DRDY=1 会发 DMA2 请求时或使用軟件請求	-	-
3	TRNGIE	TRNG 中断始能 0: 无中断 1:TRNG 中断始能	0	RW
2	TRNGEN	TRNG 始能 0: 关闭随机源 1:打开随机源	0	RW
1:0	TRNG_CLK_DIV	TRAN 时钟除频 (HCLK/8,4,2,1) 0: 除 8 1: 除 4 2: 除 2 3: 除 1 当 TRNGEN=1, 无法实时变更此选项 Note: TRANG 时钟 <= 8MHz	0x0	R/W



### 34.5.2 TRNG 状态寄存器(TRNG\_SR)

地址偏移: 0x00

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:7	Reserved	-	-	-
6	SEIS	Seed 产生时错误中断始能 0: 无中断. 1: 始能中断,至少有一次随机数数产生时错误发生,.写 0 清除.	0	W
5	CEIS	时钟错误中断始能 0: 无中断 1: 始能中断时钟错误发生..写 0 清除	0	W
4:3	Reserved	-	-	-
2	SECS	目前 Seed 状态: 0: 随机顺序产生正常. 1: 随机顺序产生有误.	0	R
1	CECS	目前 TRNG 时钟的状态 0: TRNG 时钟正常 1: TRNG 时钟错误	0	R
0	DRDY	随机数数据转换完成 0: 尚未完成 1: 随机数数据有效,读取后此为清为 0. DRDY=' 1' & TRNGIE=' 1' & DRDY=1 会发生中断.	0	R



### 34.5.3 TRNG 数据寄存器(TRNG\_DR)

地址偏移: 0x08

复位值: 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:0	RNDATA	随机数数据 34its, 当 DRDY=' 1' RNDATA 有效, DRDY=' 0' 数据全为 RNDATA =0	0	R



## 【35】高级加密标准模块 (AES)

### 35.1 概述

AES 硬件加速器可用于使用 AES 算法加密和解密数据。它是以下标准的完全兼容实现：

由联邦信息处理标准出版物 (FIPS PUB 197, 2001 年 11 月 26 日) 定义的高级加密标准 (AES)

--加速器使用 128 -位密钥长度。

它还可以执行密钥派生。加密或解密密钥存储在内部寄存器中，以便在使用相同密钥处理多个数据块时最大限度地

减少 CPU 或 DMAXX 的写操作。

默认情况下，选择电子代码簿模式 (ECB)。硬件还支持密码块链接 (CBC) 或计数器 (CTR) 模式链接算法。

AES 支持传入和传出数据的 DMAXX 传输 (需要 2 个 DMAXX 通道)。

AES 硬件加速器可用于使用 AES 算法加密和解密数据。

### 35.2 特性

- (1) 使用 AES Rijndael Block Cipher 算法的加密/解密
- (2) 符合 NIST FIPS 197 的 AES 加密/解密算法的实现
- (3) 用于存储加密或派生密钥的内部 128 位寄存器 (4x 32 位寄存器)
- (4) 一个 128 位寄存器用于在 CBC 模式下配置 AES 时的初始化向量或在选择 CTR 模式时用于 32 位计数器初始化。
- (5) 1x32 位输入缓冲区和 1x32 位输出缓冲区。
- (6) 支持使用直接内存访问 (DMAXX) 的自动数据流控制，一个用于输入数据，另一个用于输出数据。

### 35.3 AES 中断

Interrupt event	Event flag	Enable control bit	Exit from wait
AES computation completed flag	CCF	CCFIE	yes
AES read error flag	RDERR	ERRIE	yes
AES write error flag	WRERR	ERRIE	yes

Table 35-1 AES 中断表



### 35.4 AES 功能描述

当选择 CBC 或 CTR 链接模式时，AES 加速器使用长度为 128 位的密钥和初始化向量处理 128 位（4 个字）的数据块。

提供 4 种操作模式：

#### 模式 1(Encryption):

使用存储在 AES\_KEYRx 寄存器中的加密密钥进行加密。

#### 模式 2(Key derivation):

密钥派生内部存储在 AES\_KEYRx 寄存器中，在从启用 AES 之前存储在此寄存器中的加密密钥处理的密派生结束时。此模式独立于 AES 链接模式选择。

#### 模式 3(Decryption):

使用存储在 AES\_KEYRx 寄存器中的给定（预先计算的）解密密钥进行解密。

#### 模式 4(Key derivation + decryption):

使用存储在 AES\_KEYRx 寄存器中的加密密钥的密钥派生 + 解密（当 AES 配置为计数器模式以执行链接算法时不使用）。

通过将 MODE[1:0] 位编程到 AES\_CR 寄存器中来选择操作模式。仅当禁用 AES (AES\_CR 寄存器中的 EN 位 = 0) 时，才必须更改模式。在启用 AES 之前必须存储 KEY 寄存器 (AES\_KEYRx)。

要选择 ECB、CBC 或 CTR 模式中的哪一种用于加密解决方案，必须写入 AES\_CR 寄存器和 AES\_IVR 寄存器的 CHMOD[1:0] 位（仅用于 CBC 和 CTR 链接模式）禁用 AES (AES\_CR 寄存器中的 EN 位 = 0)。一旦启用 (AESEN 位 = 1)，AES 处于输入阶段，等待软件将输入数据字写入模式 1、3 或 4 的 AES\_DINR (4 个字)。数据对应于明文消息或密码消息。在两次连续写入 AES\_DINR 寄存器之间自动插入一个等待周期，以便将密钥与数据交错发送到 AES 处理器。

对于模式 2，密钥派生处理在 AES\_CR 寄存器中的 EN 位被设置后立即开始。它要求在启用 AES 之前使用加密的 KEY 加载 AES\_KEYRx 寄存器。在密钥派生处理结束时（设置 CCF 标志），派生密钥在 AES\_KEYRx 寄存器中可用，并且 AES 被硬件禁用。在此模式下，AES\_KEYRx 寄存器在启用 AES 时不能被读取，直到 CCF 标志被硬件设置为 1。

一旦计算阶段完成，AES\_SR 寄存器中的状态标志 CCF（计算完成标志）就被设置。如果 AES\_CR 寄存器中的 CCFIE 位 = 1，则可以产生中断。然后，软件可以从 AES\_DOUTR 寄存器（对于模式 1、3、4）或从 AES\_KEYRx 寄存器（如果选择模式 2）读回数据。

当 AES\_CR 寄存器中的 DMAXXOUTEN = 1 时，标志 CCF 没有意义，因为读取 AES\_DOUTR 寄存器由 DMAXX 自动管理，在计算阶段结束时无需任何软件操作。

操作以输出阶段结束，在此期间，软件在模式 1、3 或 4 中依次从 AES\_DOUTR 寄存器中读取 4 个输出数据字。在模式 2（密钥导出模式）中，数据自动存储在 AES\_KEYRx 寄存器中并 AES 被硬件禁用。然后，软件可以在启用 AES 使用此派生密钥开始解密之前选择模式 3（解密模式）。



在输入和输出阶段，软件必须连续读取或写入数据字节（模式 2 除外），但 AES 可以容忍每次读取或写入操作之间发生的任何延迟（例如：如果此时服务另一个中断）。

当检测到意外的读或写操作时，AES\_SR 寄存器中的 RDERR 和 WRERR 标志被设置。如果 AES\_CR 寄存器中的 ERRIE 位被设置，则可以产生中断。AES 在检测到错误后不会被禁用并继续正常处理。

也可以使用通用 DMAXX 写入输入字和读取输出字。

通过重置 AES\_CR 寄存器中的 AESEN 位，可以随时重新初始化 AES。然后可以通过设置 EN=1 从头重新启动 AES，等待写入第一个输入数据字节（模式 2 除外，其中一旦设置了 EN 位，密钥推导处理就开始，从值开始存储在 AES\_KEYRx 寄存器中）

#### 35.4-1 加密和派生密钥(Encryption and derivation keys)

AES\_KEYRx 寄存器用于存储加密或解密密钥。这四个寄存器以 little-endian 配置组织：寄存器 AES\_KEYR0 必须加载密钥的 32 位 LSB。因此，AES\_KEYR3 必须加载 128 位密钥的 32 位 MSB。当禁用 AES (AES\_CR 寄存器中的 EN = 0) 时，加密或解密的密钥必须存储在这些寄存器中。他们的字节序是固定的。

在模式 2 (密钥派生) 中，AES\_KEYRx 需要加载加密密钥。然后，必须启用 AES。在计算阶段结束时，派生密钥自动存储在 AES\_KEYRx 寄存器中，覆盖之前的加密密钥。当派生密钥可用时，AES 被硬件禁用。如果软件需要将 AES 切换到模式 3 (解密模式)，如果 AES\_KEYRx 寄存器的内容对应于派生密钥 (之前由模式 2 计算)，则无需写入 AES\_KEYRx 寄存器。

在模式 4 (密钥推导 + 解密) 中，AES\_KEYRx 寄存器仅包含加密密钥。派生密钥在内部计算，无需对这些寄存器进行任何写入。

#### 35.4-2 加密和 AES 链接算法(AES chaining algorithms)

AES 硬件支持三种算法，当 AES 禁用 (位 EN = 0) 时，可以通过 AES\_CR 寄存器中的 CHMOD[1:0] 位选择：

- (10) 电子密码本 (ECB)
- (11) 密码块链接 (CBC)
- (12) 计数器模式 (CTR)



### 35.4-2.1 电子密码本 (ECB)

这是默认模式。此模式不使用 AES\_IVR 寄存器。没有链式操作。消息被分成块，每个块单独加密。

图 Figue36-1,2 分别描述了电子密码本算法的加解密原理：

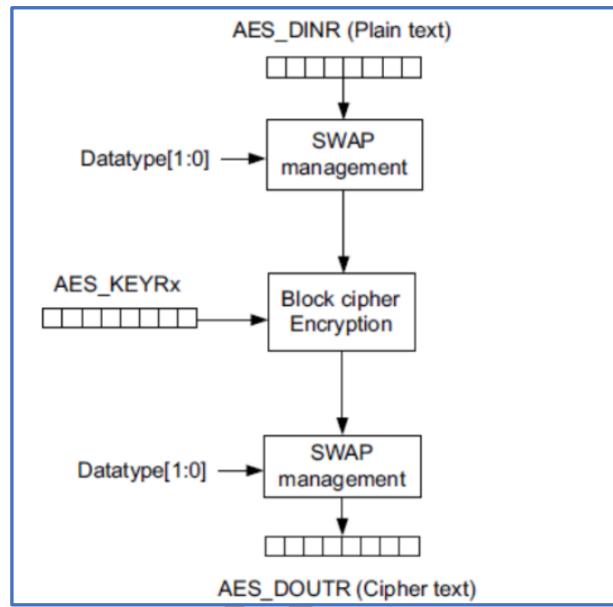


Figure 35-1 ECB 加密模式.

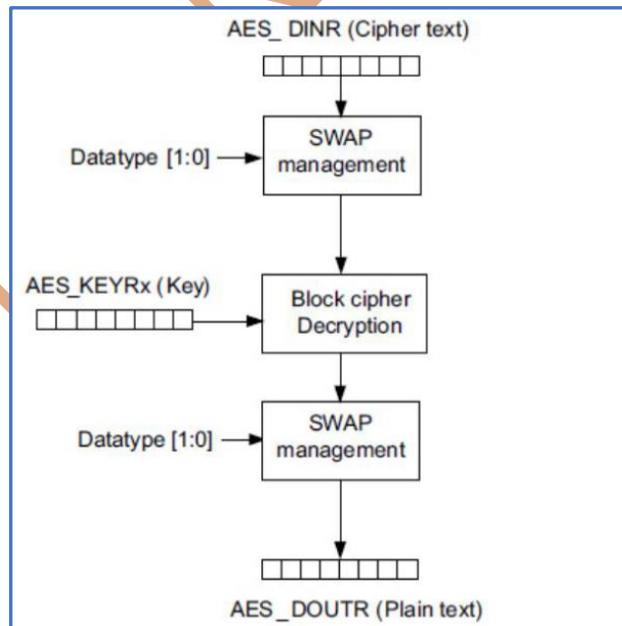


Figure 35-2 ECB 解密模式



### 35.4-2.2 密码块链接 (CBC)

在密码块链接 (CBC) 模式下，每个明文块在加密之前与前一个密文块进行 XOR。为了使每条消息都是唯一的，在第一个块处理期间使用了一个初始化向量 (AES\_IVRx)。

初始化向量在加密模式期间交换管理块之后和解密模式之前进行异或（参见图 Fig.36-3,4）

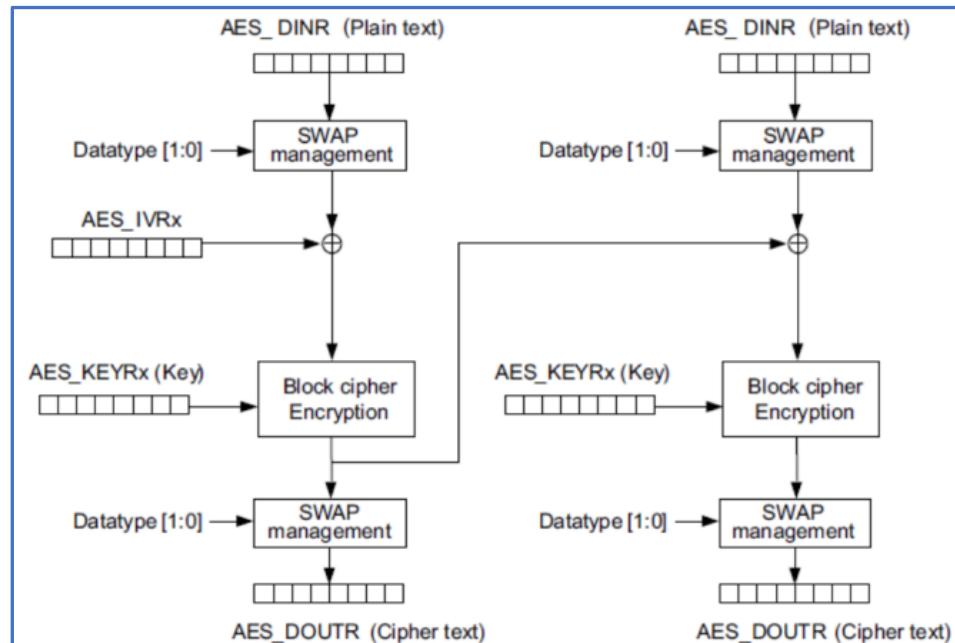


Figure 35-3 CBC 加密模式.

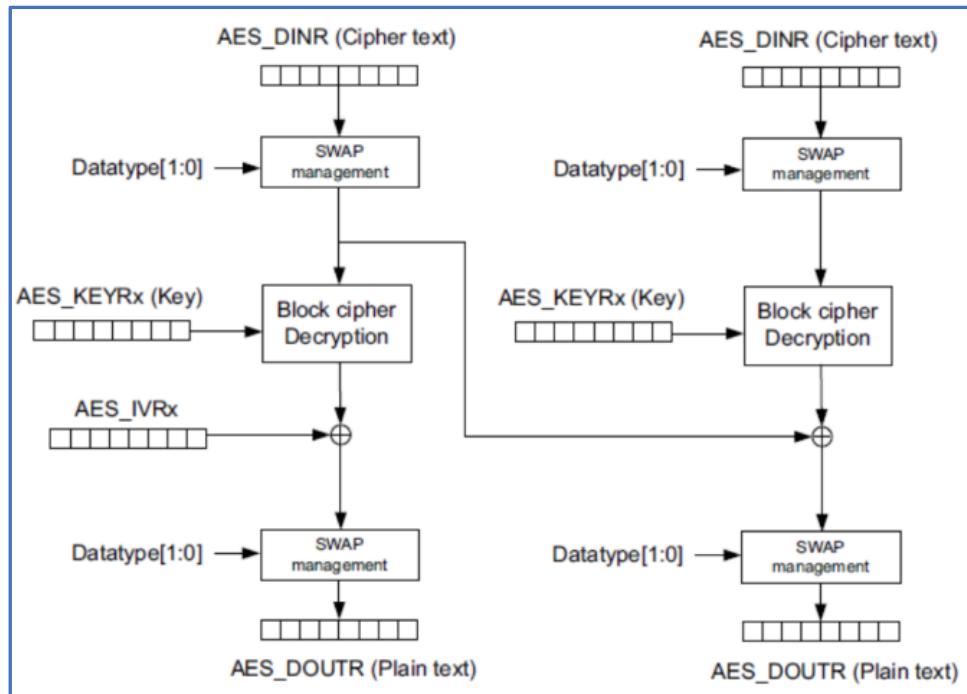


Figure 35-4 CBC 解密模式

### 35.4-2.3 计数器模式(CTR)

在计数器模式下，除了随机数值之外，还使用 32 位计数器与进行 XOR 操作（参见图 Fig.36-4,5）

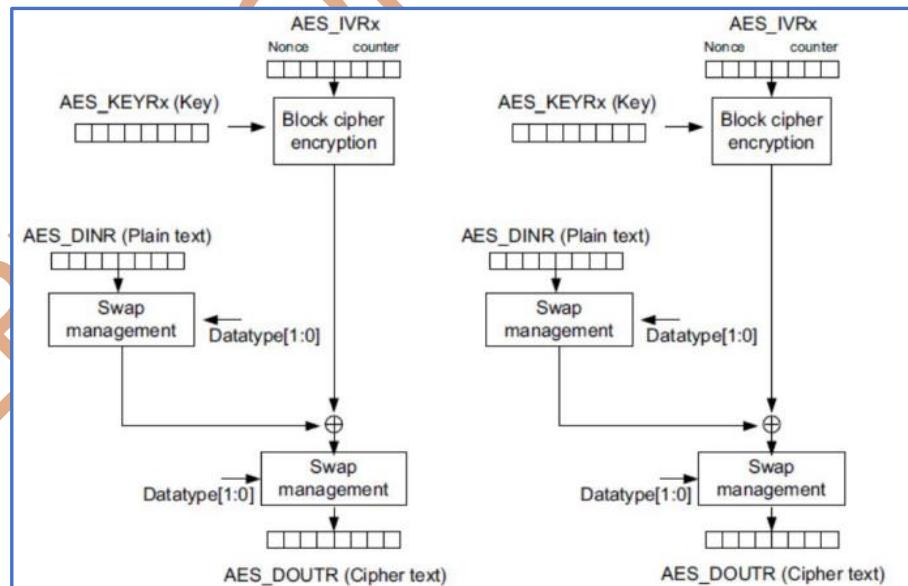


Figure 35-5 CTR 加密模式.

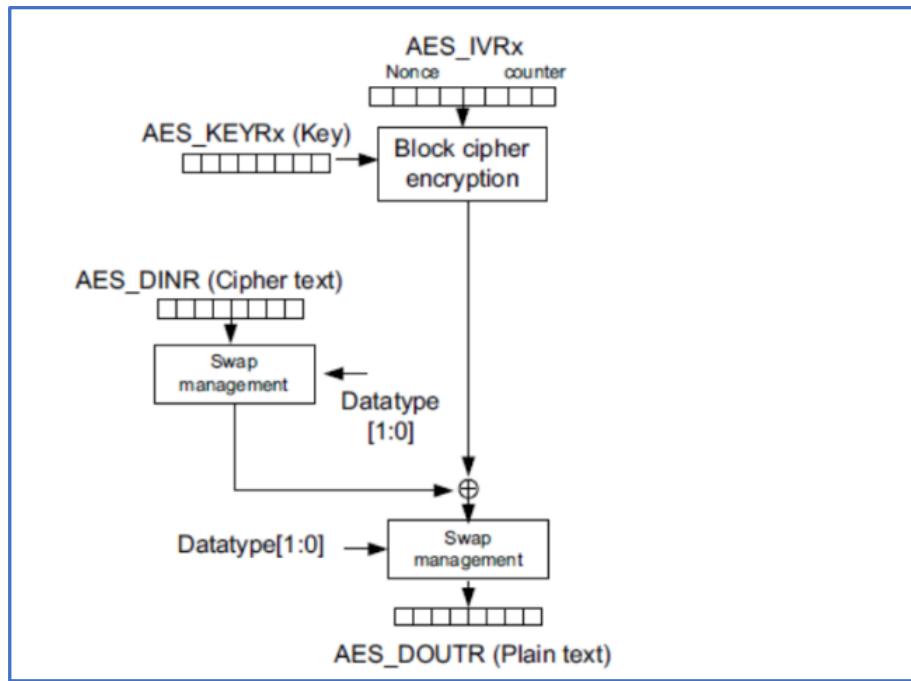


Figure 35-6 CTR 解密模式



随机数(nonce) 值和 32 位计数器可通过 AES\_IVRx 寄存器访问，其组织方式如下图

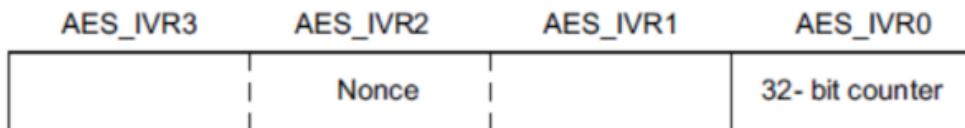


Figure 35-7 32 位计数器+随机数组织

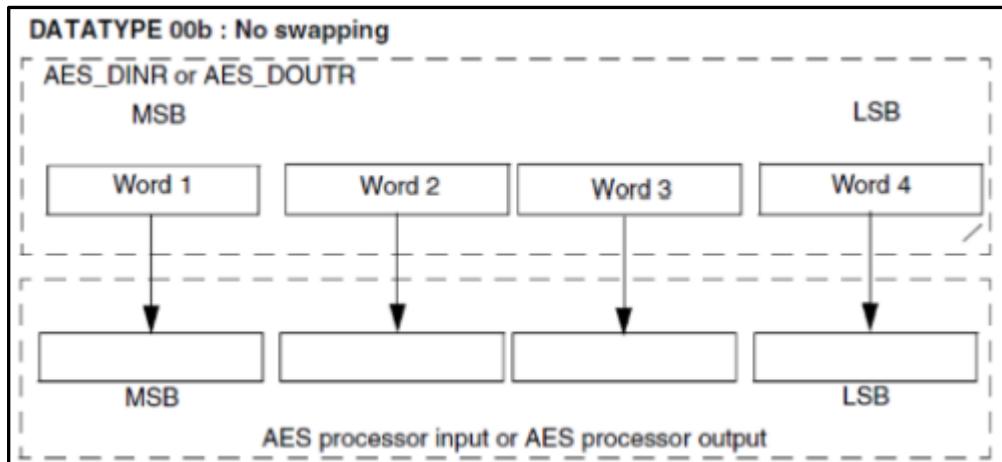
在计数器模式下，计数器从每个要块的初始化值开始递增，处理的以保证一个唯一的序列，不会长时间重复。它是一个 32 位计数器，这意味着 nonce 消息保持时存储的初始化值为禁用 AES。只有 128 位初始化向量寄存器的 32 位 LSB 代表计数器。与 CBC 模式（时仅使用一次 AES\_IVRx 寄存器）相反 处理第一个数据块，在计数器模式下，AES\_IVRx 寄存器用于处理每个数据块。

#### 35.4-3 数据型态(Data Type)

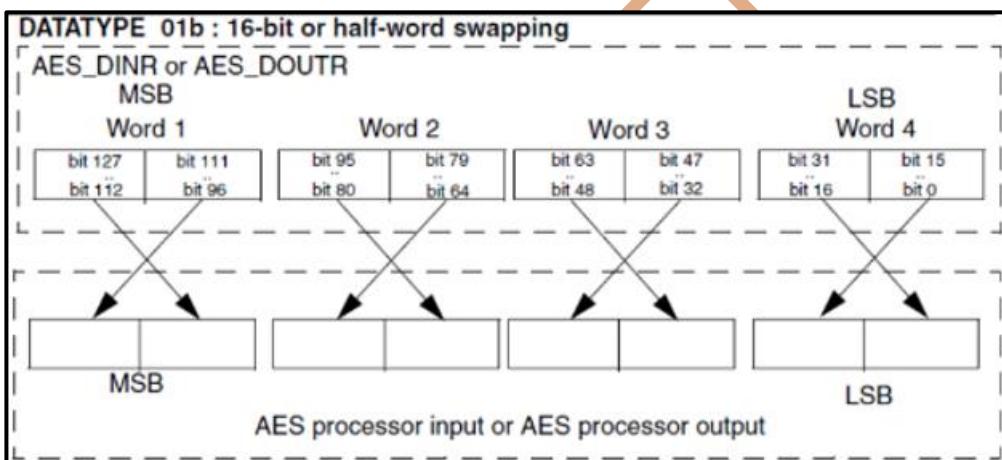
通过将数据写入 AES\_DINR 寄存器，将数据一次 32 位（字）输入 AES 处理器。AES 处理 128 位数据块。必须读取或写入 AES\_DINR 或 AES\_DOUTR 寄存器四次才能处理一个 128 位数据块，MSB 在前。系统内存组织是小端的：无论使用什么数据类型（位、字节、16 位半字、32 位字），低位数据占据最低地址位置。因此，在进入 AES 处理器之前，必须对系统内存中要写入 AES\_DINR 的数据进行位、字节或半字交换操作，并且必须对要从中读取的 AES 数据进行相同的交换操作。AES\_DOUTR 寄存器到系统内存，取决于要加密或解密的数据类型。

AES\_CR 寄存器中的 DATATYPE 位提供了不同的交换模式，在将其发送到 AES 处理器之前应用于 AES\_DINR 寄存器，并应用于 AES\_DOUTR 寄存器上的来自处理器的数据（参见图 1.10）。

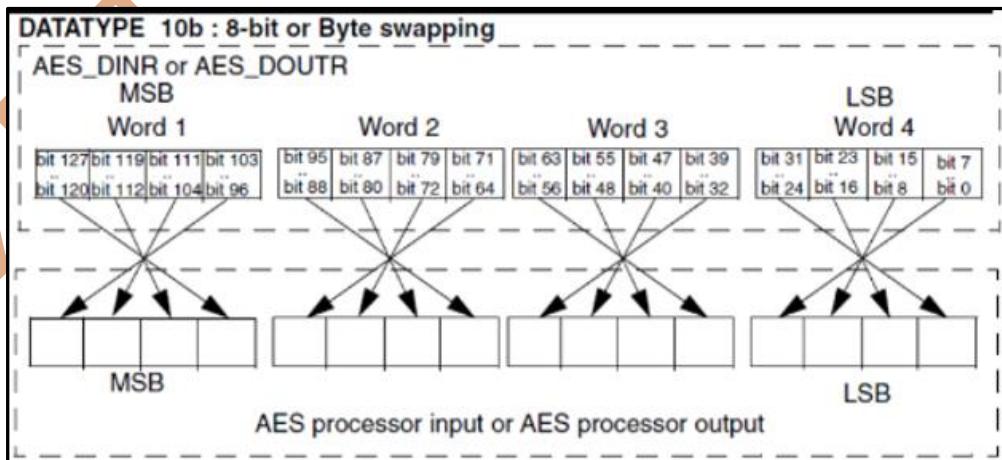
注意：交换操作仅涉及 AES\_DOUTR 和 AES\_DINR 寄存器。AES\_KEYRx 和 AES\_IVRx 寄存器对选择的交换模式不敏感。它们具有固定的小端配置



**Figure 35-8** Data Type= 2' b00



**Figure 35-9** Data Type= 2' b01



**Figure 35-10** Data Type= 2' b10

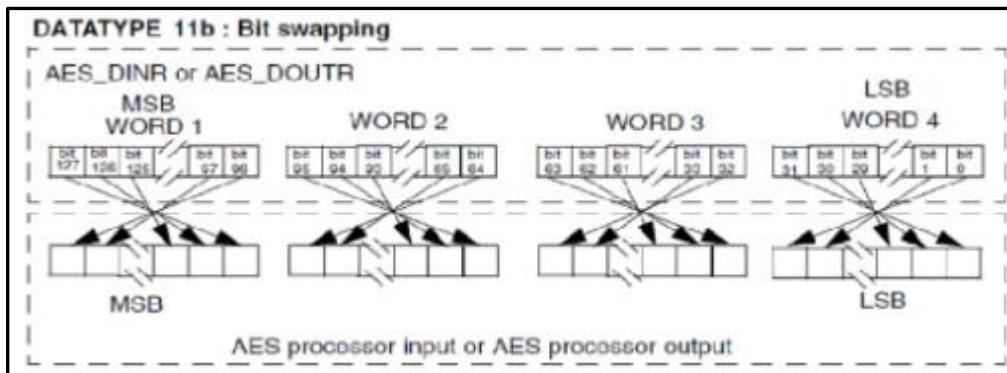


Figure 35-11 Data Type= 2' b11



### 35.5 AES 寄存器列表

AES 基地址: 0x4002 2800

偏移地址	名称	描述	默认值
0x000	AES_CR	AES 控制寄存器	0x00000000
0x004	AES_SR	AES 状态寄存器	0x00000000
0x008	AES_DINR	AES 输入数据寄存器	0x00000000
0x00C	AES_OUTR	AES 输出数据寄存器	0x00000000
0x010	AES_KEYR0	AES Key0寄存器(LSB: key[31:0])	0x00000000
0x014	AES_KEYR1	AES Key1寄存器(LSB: key[63:32])	0x00000000
0x018	AES_KEYR2	AES Key2寄存器(MSB: key[95:64])	0x00000000
0x01C	AES_KEYR3	AES Key3寄存器(MSB: key[127:96])	0x00000000
0x020	AES_IVR0	AES IVR0寄存器 (LSB: IVR [31:0])	0x00000000
0x024	AES_IVR1	AES IVR1寄存器 (LSB: IVR [63:32])	0x00000000
0x028	AES_IVR2	AES IVR2寄存器 (MSB: IVR[95:64])	0x00000000
0x01C	AES_IVR3	AES IVR3寄存器 (MSB: IVR [127:96])	0x00000000



### 35.6 寄存器说明

#### 35.6-1 AES 控制寄存器(AES\_CR)

地址偏移: 0x000

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMA2OUTEN	DMA2INEN	ERRIE	CCFIE	ERRC	CCFC	CHMOD[1:0]	MODE[1:0]	DATATYPE[1:0]		AESSEN		
R	R	R	R/W	R/W	R/W	R/W	R/WC	R/WC	RW	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:13	Reserved		0x0	--
12	DMA2OUTEN	AES 数据输出阶段, 使用 DMA2 功能 0: 不使用 DMA2 1: 使用 DMA2 数据输出, 当 Mode1/2/4 产生 DMA2 请求时.	0x0	R/W
11	DMA2INEN	AES 数据输入阶段, 使用 DMA2 功能 0: 不使用 DMA2 1: 使用 DMA2 数据输入当, Mode1/2/4 产生 DMA2 请求时.	0x0	R/W
10	ERRIE	Error interrupt enable 错误中断使能 RDERR /WRERR 须被设置. 0: 禁用中断 1: 致能中断	0x0	R/W
9	CCFIE	计算完成旗标(CCF)中断致能 0: 禁用 CCF 中断	0x0	R/W



位	标记	功能描述	复位值	读写
		1: 致能 CCF 中断		
8	ERRC	错误清除 写入 0 清除 RDERR , WRERR 旗标.	0x0	R/WC
7	CCFC	CCF 旗标 Clear 清除 写入 0 清除 CCF 旗标.	0x0	R/WC
6:5	CHMOD[1:0]	AES 的加密模式 00: 电子密码本模式(EBC) 01: 密码块链接模式(CBC) 10: 计数器模式 (CTR) 11: Reserved.  只有在禁用 AES 时才能更改 AES 链接模式。写入这些位, 禁止在启用 AES 时以避免不可预测的 AES 行为	0x0	RW
4:3	MODE[1:0]	AES 操作模式 00: 加密 01: 密钥派生 10: 解密 11: 密钥派生 + 解密  只有在禁用 AES 时才能更改操作模式。AES 时写入这些位, 禁止在启用以避免不可预测的 AES 行为。如果选择 CTR 模式, 则禁止模式 4。尽管如此, 如果软件尝试为此 CTR 模式配置设置模式 4, 它将被强制为模式 3。	0x0	R/W
2:1	DATATYPE[1:0]	数据类型选择 (用于加密块的数据输入和输出) 00: 32 位数据。没有交换。  01: 16 位数据或半字。在这个字中, 每个半字都被交换。例如, 如果写入 AES_DINR 寄存器的四个 32 位数据之一是 0x764356AB, 则赋予加密块的值为 0x56AB7643	0x0	R/W



位	标记	功能描述	复位值	读写
		<p>10: 8 位数据或字节。在字中，所有字节都被交换。例如，如果写入 AES_DINR 寄存器的四个 32 位数据之一是 0x764356AB，则赋予加密块的值为 0xAB564376</p> <p>11: 位数据。在字中所有位都被交换。例如，如果写入 AES_DINR 寄存器的四个 32 位数据之一是 0x764356AB，则赋予加密块的值为 0xD56AC26E</p> <p>如果禁用 AES，则必须更改数据类型选择。禁止在启用 AES 时写入这些位，以避免不可预测的 AES 行为</p>		
0	AESEN	<p>AES 启用</p> <p>0: AES 禁用</p> <p>1: AES 启用</p> <p>可以通过重置该位随时重新初始化 AES: 当 EN 置位时，AES 准备好开始处理新块。</p> <p>当 AES 计算在模式 2 (密钥推导) 下完成时，该位由硬件清零</p>	0x0	R/W



### 35.6-2 AES 状态寄存器(AES\_SR)

地址偏移: 0x004

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved[15:3]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位	标记	功能描述	复位值	读写
31:13	Reserved		0x0	--
2	WRERR	<p>写入错误旗标(Write error flag) 当检测到对 AES_DINR 寄存器的意外写操作时（在计算或数据输出阶段），该位由硬件设置。当启用 AES 时尝试更改 CHMOD、MODE、DATATYPE、AES_KEY* 或 AES_IVR* 时，也会设置该位。如果先前已在 AES_CR 寄存器中设置了 ERRIE 位，则会产生中断。此标志对 AES 没有影响，即使设置了 WERR，它也会继续运行。它由软件通过设置 AES_CR 寄存器中的 ERRC 位来清除 0: 未检测到写入错误 1: 检测到写入错误</p>	0x0	R/W
1	RDERR	<p>读取错误旗标(Read error flag) 当检测到来自 AES_DOUTR 寄存器的意外读取操作时（在计算或数据输入阶段），该位由硬件设置。如果先前已在 AES_CR 寄存器中设置了 ERRIE 位，则会产生中断。此标志对 AES 没有影响，即使设置了 RDERR 它也会继续运行。 它由软件通过设置 AES_CR 寄存器中的 ERRC 位 i 来清除 0: 未检测到读取错误检测 1: 读取错误</p>	0x0	R/W
0	CCF	<p>计算完成旗标 该位在计算完成时由硬件置位。如果了 CCFIE 则会产生中断，先前已在 AES_CR 寄存器中设置位，它由软件</p>		



位	标记	功能描述	复位值	读写
		<p>通过设置 AES_CR 寄存器中的 CCFC 位来清除。 0: 计算完成 1: 计算未完成 注: 该位仅在 DMAAXXOUTEN = 0 时有效。当 DMAAXX_EN = 1 时, 它可能保持高电平</p>		

ZBT CONFIDENTIAL



### 35.6-3 AES 输入数据寄存器(AES\_DINR)

地址偏移: 0x008

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DINR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	DINR[31:0]	数据输入寄存器。 该寄存器在输入阶段必须写入 4 次。 --在模式 1 (加密) 中, 必须写入 4 个字, 代表从 MSB 到 LSB 的纯文本。 --在模式 2 (密钥派生) 中, 不使用该寄存器, 因为该模式仅涉及从 AES_KEYRx 寄存器开始的派生密钥计算。 --在模式 3 (解密) 和模式 4 (密钥导出+解密) 中, 必须写入 4 个字, 代表密文 MSB 到 LSB. 注意: 该寄存器必须以 32 位数据宽度访问	0x0	R/W



### 35.6-4 AES 输出数据寄存器(AES\_OUTR)

地址偏移: 0x00C

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTR[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTR[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	标记	功能描述	复位值	读写
31:0	OUTR[31:0]	数据输出寄存器. 该寄存器是只读的 一旦设置了 CCF 标志 (计算完成标志), 读取该数据寄存器 4 次即可访问 128 位输出结果. --在模式 1 (加密) 中, 读取的 4 个字代表从 MSB 到 LSB 的密文. --在模式 2 (密钥派生) 中, 不需要读取该寄存器, 因为派生密钥位于 AES_KEYRx 寄存器中. --在模式 3 (解密) 和模式 4 (密钥派生+解密) 中, 读取的 4 个字代表从 MSB 到 LSB 的纯文本. 注意: 该寄存器必须以 32 位数据宽度访问	0x0	R



### 35.6-5 AES KEY0 寄存器(AES\_KEY0R) (LSB: key[31:0])

地址偏移: 0x010

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	KEY0[31:0]	<p>数据输出寄存器 (LSB 密钥 [31:0]) 该寄存器必须在 AES_CR 寄存器中的 EN 位设置之前写入。 在模式 1 (加密)、模式 2 (密钥派生) 和模式 4 (密钥派生 + 解密) 中, 要写入的值代表来自 LSB 的加密密钥, 即 Key [31:0]。 在模式 3 (解密) 中, 要写入的值代表来自 LSB 的解密密钥, 即 Key [31:0]。当在这种解密模式下使用加密密钥写入寄存器时, 在启用 AES 之前读取它会返回加密值。在设置 CCF 标志后读取它会返回派生密钥。 在启用 AES 时读取该寄存器会返回一个不可预测的值。 <b>注意:</b> 该寄存器不包含方式 4 (推导+解密) 中的推导密钥。它始终包含加密密钥值</p>	0x0	R/W



### 35.6-6 AES KEY1 寄存器(AES\_KEY1R) (LSB: key[63:32])

地址偏移: 0x014

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	KEY1[31:0]	AES key register (key [63:32]) Refer to the description of AES_KEYR0.	0x0	R/W



### 35.6-7 AES KEY2 寄存器(AES\_KEY1R) (MSB: key[95:64])

地址偏移: 0x018

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	KEY2[31:0]	AES key register (key [95:64]) Refer to the description of AES_KEYR0.	0x0	R/W

### 35.6-8 AES KEY3 寄存器(AES\_KEY1R) (MSB: key[127:96])

地址偏移: 0x01C

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	KEY3[31:0]	AES key register (key [127:96]) Refer to the description of AES_KEYR0.	0x0	R/W



### 35.6-9 AES IVR0 寄存器(AES\_IVR0)(IVR[31:0])

地址偏移: 0x020

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVR0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	IVR0[31:0]	初始化向量寄存器 (IVR [31:0]) 该寄存器必须在 AES_CR 寄存器中的 EN 位被设置之前写入. 寄存器值在以下情况下没有意义: - 选择了 EBC 模式 (电子码本) - 除了密钥派生外, 还选择了 CTR 或 CBC 模式. 在 CTR 模式 (Counter mode) 下, 该寄存器包含 32 位计数器值. 在启用 AES 时读取该寄存器将返回值 0x00000000	0x0	R/W

### 35.6-10 AES IVR1 寄存器(AES\_IVR1)(IVR[63:32])

地址偏移: 0x024

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	IVR1[31:0]	初始化向量寄存器(IVR [95:64]) 请参考 IVR0 的说明.	0x0	R/W



### 35.6-11 AES IVR2 寄存器(AES\_IVR2)( IVR[95:64])

地址偏移: 0x028

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	IVR2[31:0]	初始化向量寄存器 (IVR [95:64]) 该寄存器必须在 AES_CR 寄存器中的 EN 位被设置之前写入. 寄存器值在以下情况下没有意义: - 选择了 EBC 模式 (电子码本) - 除了密钥派生外, 还选择了 CTR 或 CBC 模式. 在 CTR 模式 (Counter mode) 下, 该寄存器包含 32 位计数器值. 在启用 AES 时读取该寄存器将返回值 0x00000000	0x0	R/W

### 35.6-12 AES IVR3 寄存器(AES\_IVR3)( IVR[127:96])

地址偏移: 0x02C

复位值: 0x0000 0000

Bit31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位	标记	功能描述	复位值	读写
31:0	IVR3[31:0]	初始化向量寄存器( [127:96]) 请参考 IVR0 的说明.	0x0	R/W



## 【36】USB 全速设备接口(USB)

### 36.1 功能與特徵

USB 外设实现了 USB2.0 全速总线和 AHB 总线间的接口, USB 外设支持 USB 挂起/恢复操作, 可以停止设备时钟实现低功耗。

主要特性:

- (1) 符合 USB2.0 全速设备的技术规范
- (2) 1 個控制端点(INOUT)和可配置可配置多达 15 个额外的,
- (3) CRC(循环冗余校验)生成/校验, 反向不归零 (NRZI) 编码/解码和位填充
- (4) 支持同步传输
- (5) 支持批量/同步端点的双缓冲区机制
- (6) 支持 USB 挂起/恢复操作
- (7) 帧锁定时钟脉冲生成
- (8) Soft connect/disconnect.

### 36.2 USB 方框图

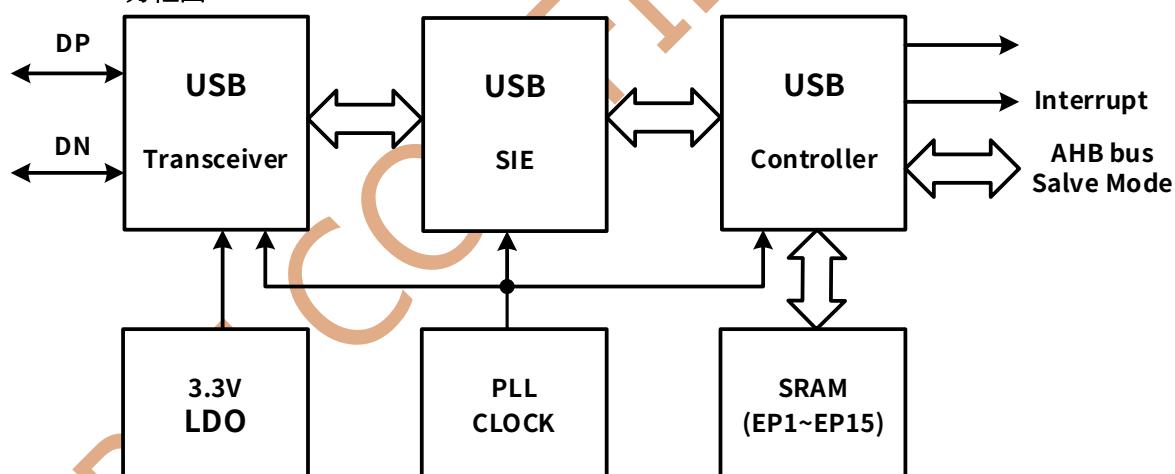


Figure 36-1 USB 设备框图



### 36.3 功能描述

USB 模块为 PC 主机和微控制器所实现的功能之间提供了符合 USB 规范的通信连接。PC 主机和微控制器之间的数据传输是通过共享一专用的数据缓冲区来完成的，该数据缓冲区能被 USB 外设直接访问。这块专用数据缓冲区的大小由所使用的端点数目和每个端点最大的数据分组大小所决定，每个端点最大可使用 512 字节缓冲区，最多可用于 16 个单向或 8 个双向端点。USB 模块同 PC 主机通信，根据 USB 规范实现令牌分组的检测，数据发送/接收的处理，和握手分组的处理。整个传输的格式由硬件完成，其中包括 CRC 的生成和校验。每个端点都有一个缓冲区描述块，描述该端点使用的缓冲区的地址，大小和必须被传输的字节数。

**36.4**

**36.5**

**36.6 Aa**

ZBT CONFIDENTIAL



## 【37】 USB 全速设备接口(USB)

### 37.1 功能與特徵

USB 外设实现了 USB2.0 全速总线和 AHB 总线间的接口, USB 外设支持 USB 挂起/恢复操作, 可以停止设备时钟实现低功耗。

主要特性:

- (1) 符合 USB2.0 全速设备的技术规范
- (2) 1 個控制端点(INOUT)和可配置可配置多达 15 个额外的,
- (3) CRC(循环冗余校验)生成/校验, 反向不归零 (NRZI) 编码/解码和位填充
- (4) 支持同步传输
- (5) 支持批量/同步端点的双缓冲区机制
- (6) 支持 USB 挂起/恢复操作
- (7) 帧锁定时钟脉冲生成
- (8) Soft connect/disconnect.

### 37.2 USB 方框图

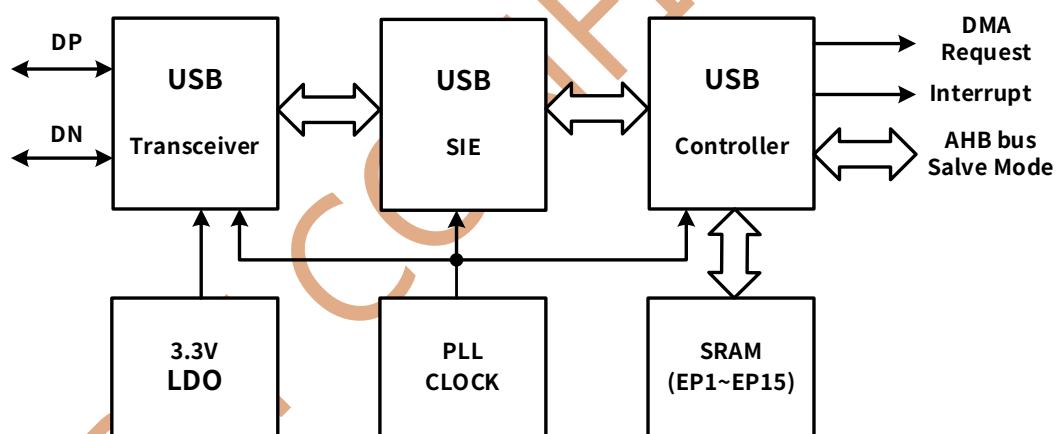


Figure 37-1 USB 设备框图



### 37.3 功能描述

USB 模块为 PC 主机和微控制器所实现的功能之间提供了符合 USB 规范的通信连接。PC 主机和微控制器之间的数据传输是通过共享一专用的数据缓冲区来完成的，该数据缓冲区能被 USB 外设直接访问。这块专用数据缓冲区的大小由所使用的端点数目和每个端点最大的数据分组大小所决定，每个端点最大可使用 512 字节缓冲区，最多可用于 16 个单向或 8 个双向端点。USB 模块同 PC 主机通信，根据 USB 规范实现令牌分组的检测，数据发送/接收的处理，和握手分组的处理。整个传输的格式由硬件完成，其中包括 CRC 的生成和校验。每个端点都有一个缓冲区描述块，描述该端点使用的缓冲区的地址，大小和必须被传输的字节数。

### 37.4

### 37.5

### 37.6 Aa



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBT CONFIDENTIAL



## 【38】 SDIO 接口(SDIO)

### 38.1 SDIO 主要功能

SD/SDIO MMC 卡主机模块(SDIO)在 AHB 外设总线和多媒体卡(MMC)、SD 存储卡、SDIO 卡和 CE-ATA 设备间提供了操作接口。

多媒体卡系统规格书由 MMCA 技术委员会发布，可以在多媒体卡协会的网站上([www.mmca.org](http://www.mmca.org))获得。

CE-ATA 系统规格书可以在 CE-ATA 工作组的网站上([www.ce-ata.org](http://www.ce-ata.org))获得。

SDIO 的主要功能如下：

- 多媒体卡 (MMC - 4.41 版、eMMC - 4.5 版、4.51 版和 5.0 版)。MMC 4.4 标准已被 MMC 4.41 (JESD84-A441, 2010 年 3 月) 取代。DWC 移动存储主机支持从 MMC 4.4 到 MMC 4.41 所做的更改，包括 eMMC 支持
- 与较早的多媒体卡系统规格版本全兼容(向前兼容)。
- 与 SD 存储卡规格版本 3.0/3.01 全兼容.( Includes eSD support)
- 与 SD I/O 卡规格版本 3.0 全兼容.( includes eSD support)
- 完全支持 CE-ATA 功能(与 CE-ATA 数字协议版本 1.1 全兼容)。
- 8 位总线模式下数据传输速率可达 48MHz。
- 数据和命令输出使能信号，用于控制外部双向驱动器。

### 38.2 SDIO 总线拓扑

SD\_MMC 存储卡通常是用于 FLASH 大容量存储的设备。SDIO 卡通常 I/O 应用程序中的功能，也可以有可选的 FLASH 存储器。CE-ATA 卡移动设备应用程序中的功能。SD\_MMC\_CEATA 总线模式包括以下信号：

- (1) **CLK:** Host 给 Device 的时钟信号
- (2) **CMD:** 双向的命令、响应信号线，Command 由 Host 发送给 Device，Response 由 Device 返回给 Host
- (3) **DATA[7:0]:** 双向的数据传输线，数据位宽允许动态配置为 1-bit (默认)、4-bit、8-bit 模式 (数据线与命令线通常要保持上拉状态，以保护数据免受总线浮动影响)

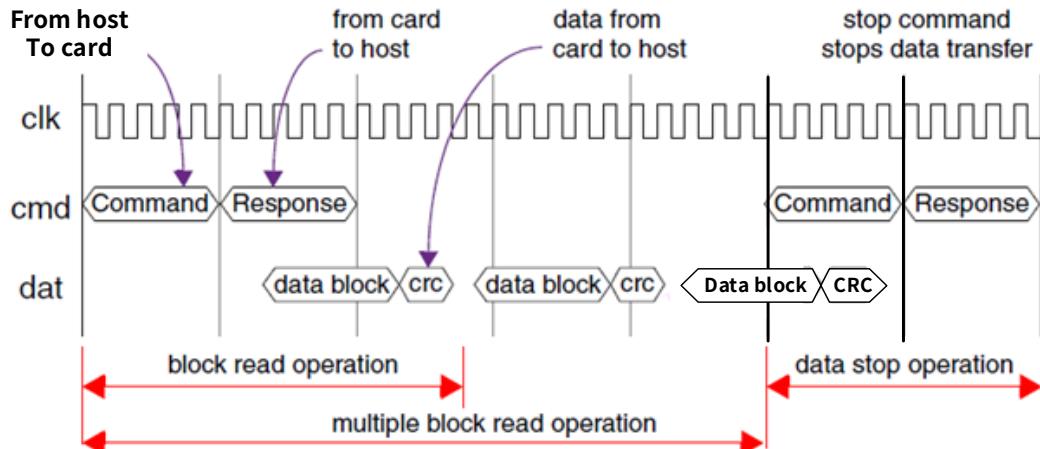


Figure 38-1 SDIO(多)数据块读操作图

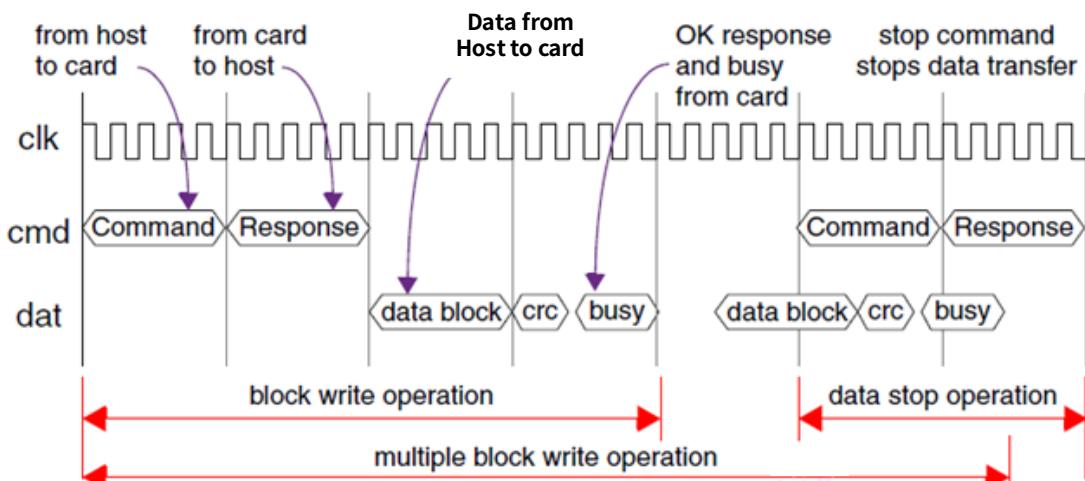


Figure 38-2 SDIO(多)数据块写操作图

- 主机发送的示例命令:
- 来自卡片的简短响应示例



### 38.3 要求和特性

- (1) 支持安全数字内存與多媒体卡协议命令.
- (2) 支持 Secure Digital I/O 协议指令.
- (3) 支持多媒体卡协议命令.
- (4) 支持命令完成信号和对主处理器的中断與完成信号禁用功能.
- (5) 仅主接口支持 64 位寻址與从接口只能是 32 位寻址.
- (6) 支持内部 DMA 功能, CRC 生成和错误检测與写保护.
- (7) 可配置为 MMC-Ver3.3-only 控制器或 SD\_MMC 控制器.
- (8) 可编程波特率。支持多达 4 个时钟分频器, 以支持同时具有不同时钟速度要求的多张卡的操作.
- (9) 提供单独的时钟控制以选择性地打开或关闭卡的时钟.
- (10) 支持主机上拉控制,卡片检测和初始化
- (11) 支持 SDIO 读等待,挂起和恢复操作;SDIO3.0 支持 8 位模式.
- (12) 块最大字节 65,535 字节.
- (13) 支持 Busy Clear Interrupt 在写数据传输到卡時.
- (14) Two FIFOs, TX and RX FIFO (FIFO depth = 32 and FIFO data width = 32 bits).

### 38.4 基本配置

SD/MMC 接口配置如下:

■时钟(**clock**):

- (1) 启用将要使用的时钟源，如果它尚未运行（大多数振荡器可以在不需要时关闭以节省电力）。
- (2) 选择将在 SDIOCLKSEL 寄存器中使用的时钟源.(参考)
- (3) 当使用相位延迟时，在 SDIOCLKSEL 寄存器中选择的输入时钟,必须是 SDIO 时钟的 2 倍.(请参见第 XXXX 节，SDIOCLKCTRL 寄存器)
- (4) 设置时钟源选择复用器之后的时钟分频器 (SDIOCLKDIV)以获得所需的时钟速率。(请参见第 XXX 节)
- (5) 在 AHBCLOCKCTRL2 寄存器中启用外设时钟。请参阅第 XXX 节

■复位(**Reset**): 可以使用 PRESETCTRL1 寄存器专门复位外设，但在继续之前必须从重置状态中删除。  
(请参阅第 XXXX 节)

■复用引脚: 在 AFIO 寄存器块中配置将用于该外设的引脚。有关 AFIO 的详细信息，请参阅第 11 章.

■中断: 如果该外设将使用中断，请在 NVIC 中启用. 请参阅第 7 章

■外围内部设置: 根据应用需要配置 SDIO/SDMMC.

■采样和驱动输入和输出的延迟值可以使用 SYSCON 模块中的 SDIOCLKCTRL 寄存器



### 38.5 功能描述

SD/MMC 控制器接口由以下主要功能塊組成：

- 總線接口單元 (BIU) - 為寄存器和數據提供 AHB 和 DMA 接口讀/寫。
- 通过提供主机接口到寄存器和数据 FIFO 主机接口单元 (HIU)。此外，它通过 DMA 提供独立的数据 FIFO 访问界面。您可以将主机接口配置为 AMBA APB 或 AMBA AHB 从机
- 卡接口單元 (CIU) - 處理卡協議並提供時鐘管理。
- 內部 MCI DMA 控制器(IDMAC): AHB 總線主控 DMA 控制器

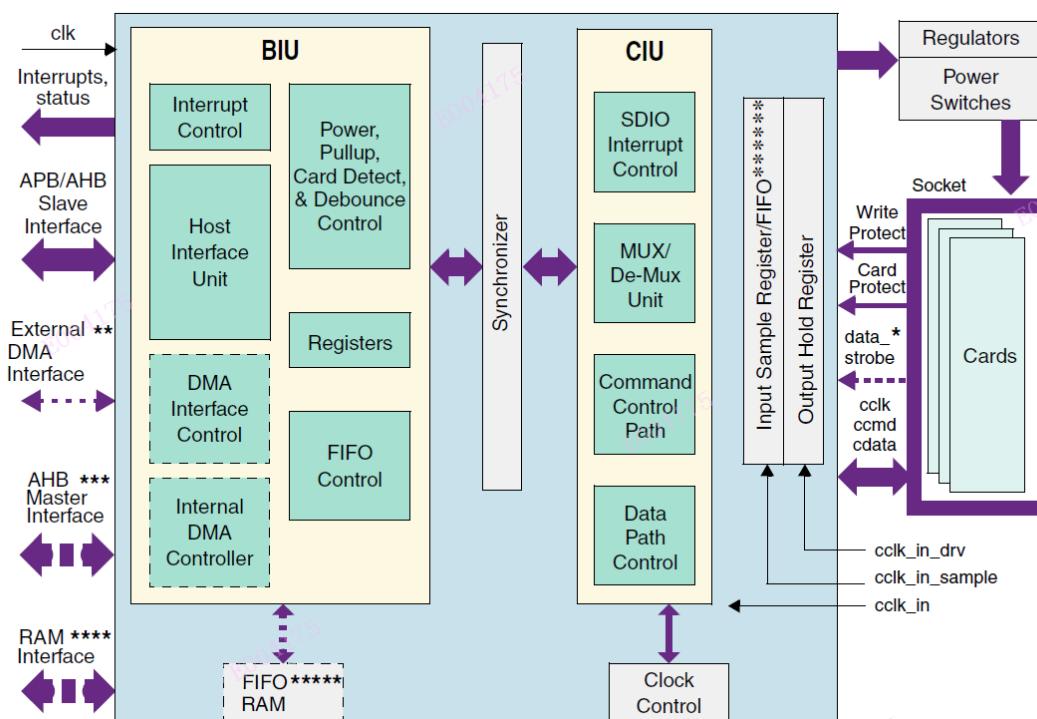


Figure 38-3 SDIO Block Diagram



### 38.5-1 Bus Interface Unit (BIU)

BIU 提供以下功能：

- 主机界面(HIU)
- DMA 接口
- 中断控制
- Register access
- FIFO access
- Power/pullup control and card detection(电源/上拉控制和卡检测)

#### 38.5-1.1 主机界面(HIU)

主机接口单元 (HIU) 是一个 AHB/APB 从接口，它提供接口在 SD 单元块中 和主机总线之间可以在中配置 主机接口 coreConsultant 作为 AMBA 高级高性能总线 (AHB) 或 AMBA 高级外围总线 (APB)；它为 AMBA AHB 或 AMBA 提供无缝接口 APB 总线。在 AHB 访问中，寄存器访问有一个等待状态。访问 寄存器区分为单等待状态单次访问。

在对数据 FIFO 区域的突发访问中，第一次数据访问会产生一个等待状态；连续的访问没有等待状态。 在对数据 FIFO 区域进行部分访问期间，实际的 FIFO 数据弹出或推送发生在 FIFO 数据宽度边界访问。 例如，如果 FIFO 数据宽度为 32，则 FIFO 当您执行 32 位访问时会发生 pop 或 push，地址 [1] 等于 1'b1 的 16 位访问，或地址 [1:0] 等于 2'b11 的 8 位访问。部分写入存储在临时 寄存器直到 FIFO 数据宽度边界访问。建议你做 full bus-size 访问数据 FIFO。如果您需要进行部分总 线大小的访问，您必须首先访问较低的字节。通过突发访问访问 FIFO 时，突发类型为“无关”。只要 htrans=SEQ 或背靠背 SINGLE 访问是对 FIFO，DWC\_mobile\_storage 是 1 时钟、0 等待状态访问。

#### 38.5-1.2 中断控制

中断控制器单元产生一个依赖于控制器原始中断的中断状态、中断屏蔽寄存器和全局中断使能寄存器位。 一旦中断检测到条件，它设置原始中断状态寄存器中的相应中断位。原始中断状态位保持打开状态，直 到软件通过向中断写入 1 清除该位少量; 0 保留该位不变。中断端口 int 是一个高电平有效、电平敏感 的中断。中断端口仅处于活动状态当原始中断状态寄存器中的任何位有效时，对应的中断屏蔽位为 1， 并且全局中断使能位为 1。中断端口被注册，以避免任何组合故障。



### 38.5-1.3 FIFO Controller Unit

FIFO 控制器将内部 FIFO 连接到主机/DMA 接口和卡控制器单元。 FIFO 深度可以配置在 8 到 4096 Byte 之间。

### 38.5-2 Card Interface Unit (CIU)

#### 38.5-2.1 Command Path

命令路径执行以下功能：

- 加载时钟参数
- 加载卡片命令参数
- 向卡总线发送命令 (`ccmd_out` 行)
- 从卡总线接收响应 (`ccmd_in` 行)
- 向 BIU 发送响应
- 在命令行上驱动 P 位

#### 38.5-2.2 Data Path

数据路径模块在写数据传输期间弹出数据 FIFO 并在 `cdata_out` 上传输数据，或者它在读取数据传输期间接收 `cdata_in` 上的数据并将其推入 FIFO。数据路径加载新数据参数——即预期数据、读/写数据传输、流/块传输、块大小、字节计数、卡类型、超时寄存器——只要数据传输命令不在进行中。

如果在命令寄存器中设置了 `data_expected` 位，则新命令是数据传输命令，并且数据路径开始以下之一：

- 如果读/写位 = 1，则发送数据
- 如果读/写位 = 0，则接收数据

##### 38.5-2.2.1 Data Transmit

数据传输状态机，如图 38-4 所示，在一个时钟后两个时钟开始数据传输。接收到对数据写入命令的响应；即使命令路径检测到响应也会发生这种情况错误或响应 CRC 错误。如果由于响应超时而未从卡收到响应，则数据不传输。根据命令寄存器中的 `transfer_mode` 位的值，数据发送状态机将数据以流或块的形式放在卡数据总线上。

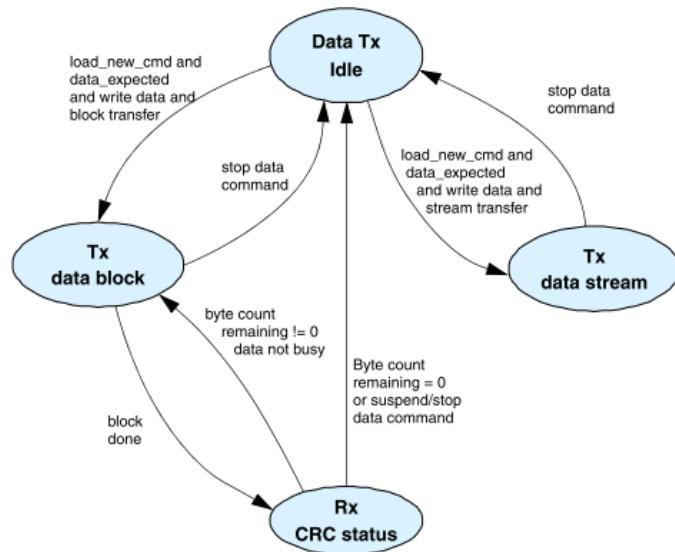


Figure 38-4 S Data Transmit State Machine

#### 38.5-2.2.2 Data Receive

数据接收状态机，如图 38-5 所示，在一个结束位后两个时钟周期接收数据数据读取命令，即使命令路径检测到响应错误或响应 CRC 错误。如果有回应由于发生响应超时，没有从卡接收到，BIU 没有接收到信号数据传输完成；如果 SD 方块发送的命令是非法的，就会发生这种情况卡的操作，使卡无法开始读取数据传输。如果在数据超时之前没有接收到数据，则数据路径向 BIU 发出数据超时信号并结束数据传输完成。根据命令寄存器中的 transfer\_mode 位的值，数据接收状态机以流或块的形式从卡数据总线获取数据。

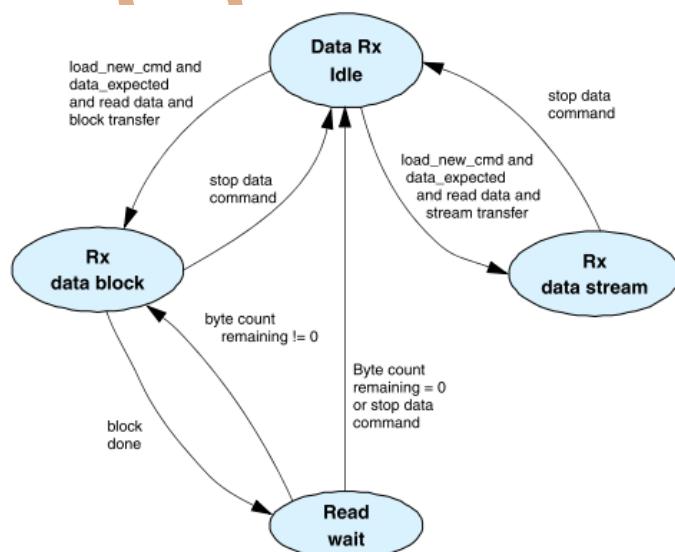


Figure 38-5 S Data-Receive State Machine



### 38.5-2.2.3 Auto-Stop

SD 方块 内部生成停止命令，并在命令路径中加载时 `send_auto_stop` 位在命令寄存器中设置。自动停止命令有助于发送准确的使用 MMC 的流读取或写入以及 SD 的多块读取或写入的数据字节数 SD 卡的内存传输。

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
MMC	Stream read	0	No	Open-ended stream
MMC	Stream read	> 0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stream
MMC	Stream write	> 0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	> 0	No	Byte count = 0 is illegal
MMC	Single-block write	> 0	No	Byte count = 0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	> 0	Yes **	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	> 0	Yes **	Pre-defined multiple block
SDMEM	Single-block read	> 0	No	Byte count = 0 is illegal
SDMEM	Single-block write	> 0	No	Byte count = 0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	> 0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	> 0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	> 0	No	Byte count = 0 is illegal
SDIO	Single-block write	> 0	No	Byte count = 0 illegal
SDIO	Multiple-block read	0	No	Open-ended multiple block
SDIO	Multiple-block read	> 0	No	Pre-defined multiple block
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block write	> 0	No	Per-defined multiple block

Note : 将传输模式设置为块传输且 `byte_count` 等于块大小的条件被视为 MMC 和 SD 卡的单块数据传输命令。如果 `byte_count = n*block_size` ( $n = 2, 3, \dots$ )，则条件被视为预定义的多块数据传输命令。在 MMC 卡的情况下，主机软件可以通过两种方式执行预定义的数据传输：1) 在发出之前发出 CMD23 命令 CMD18/CMD25 命令到卡 - 在这种情况下，发出 CMD18/CMD25 命令而不设置



send\_auto\_stop 位。2) 发出 CMD18/CMD25 命令而不向卡发出 CMD23 命令，使用 send\_auto\_stop 位设置。在这种情况下，多块数据传输由内部生成的自动停止终止.编程字节计数后的命令。

以下列出了自动停止命令的条件。

- 字节数大于 0 的 MMC 卡的流读取 - SD 方块 生成一个内部停止命令并将其加载到命令路径中，以便停止命令的结束位当从卡中读取最后一个数据字节并且没有接收到额外的数据字节时发送。如果字节数小于 6 (48 位)，在结束位之前从卡接收到几个额外的数据字节停止命令的发送。
- 字节数大于 0 的 MMC 卡的流写入 - SD 方块 生成一个内部停止命令并将其加载到命令路径中，以便停止命令的结束位当在卡总线上传输最后一个数据字节并且没有传输额外的数据字节时发送。如果字节数小于 6 (48 位)，则数据路径最后传输数据以满足上述要求健康) 状况。
- 字节数大于 0 的 SD 卡的多块读取内存 - 如果块大小小于 4 (单位数据总线)、16 (4 位数据总线) 或 32 (8 位数据总线)，自动停止命令加载到读取所有字节后的命令路径。否则，顶部命令被加载到命令路径，以便在接收到最后一个数据块后发送停止命令的结束位。
- 字节数大于 0 的 SD 卡的多块写入内存 - 如果块大小小于 3 (单位数据总线)、12 (4 位数据总线) 或 24 (8 位数据总线)，自动停止命令加载到传输完所有数据块后的命令路径。否则，将停止命令加载到命令路径，以便在 CRC 状态的结束位之后发送停止命令的结束位收到。
- 自动停止期间主机软件的注意事项 - 每当发出自动停止命令时，主机软件不应向 SD 方块 发出新命令，直到自动停止由 SD 方块和数据传输完成。如果主机在正在进行自动停止的数据传输，可能会在新的数据传输后发送自动停止命令发送命令并接收其响应；这可能会延迟发送停止命令，从而传输额外的数据字节。对于流写入，额外的数据字节是错误数据，可能会破坏卡数据。如果主机想要在数据传输完成之前终止数据传输，它可以发出停止或中止命令，在这种情况下 SD 方块 不会生成自动停止命令。

### 38.5-2.3 Power/Pullup Control and Card Detection Unit

寄存器单元具有控制电源和 MMC 开漏上拉的寄存器。每张 v 的电源可以有选择地打开或关闭。此外，还有两个 4 位调整 Card 电压可以控制两个稳压器电压的信号。控制寄存器有一个 enable\_OD\_pullup 位，用于在 MMC 期间开启开漏模式上拉初始化。此外，SD 提供了一个 8 位通用输入端口，gp\_in 和一个 16 位通用输出端口 gp\_out。Card 检测单元查找卡片插入或卡片的卡片检测信号的任何变化移动。它过滤掉与机械插入或移除相关的去抖动，并且向主机产生一个中断。您可以对去抖动过滤器值进行编程。上电时，控制器应读取 card\_detect 端口并将值存储在记忆。收到卡检测中断后，它应该再次读取 card\_detect 端口并与之前的卡检测状态进行异或，以找出哪个卡已中断。如果不只一个卡同时拔出或插入，只有一个卡检测中断：XOR value 表示哪些卡受到了干扰。内存应该更新为新的卡检测值。卡检测信号根据指定的块数去抖动去抖计数寄存器 (DEBNCE)。当此信号连接到卡检测引脚



卡槽，则 CDETECT 寄存器的位 0 状态将被过滤的数量 DEBNCE 中指定的去抖动周期。这保证了与卡相关的中断检测信号在发生之前被去抖动。

#### 38.5.2.4 Non-Data Transfer Commands that Use Data Path

一些非数据传输命令（非读/写命令）也使用数据路径。

#### 38.5.2.5 SDIO Interrupt Control

SD 卡的中断發生需要兩個時鐘的時間通知给 BIU. SDIO 卡通过在中断期间将 cdata\_in 置为低电平来发出中断信号；一个所选卡的中断周期由中断控制状态机确定。一个中断周期对于非活动或非选定卡始终有效，并且为 1 位数据模式选择的卡。宽总线活动或选定卡的中断周期对以下情况有效条件：

- 卡闲置
- 非数据传输命令正在进行中
- 两个数据块之间的数据块结束位后的第三个时钟
- 从最后一个数据结束位后的两个时钟到下一个数据传输命令的结束位

#### 38.5.3 Auto-Stop

自动停止命令有助于使用流读取发送准确数量的数据字节或为 MMC 写入，为 SD 内存传输进行多块读取或写入牌。模块内部生成停止命令，并在命令中加载在命令寄存器 (CMD) 中设置 SEND\_AUTO\_STOP 位时的路。

软件应根据列表中列出的详细信息设置 SEND\_AUTO\_STOP 位

下表：

#### 38.6

#### 38.7 SDIO PIN 描述

Pin function	Type	Description
SDIO_CLK	O	SD/SDIO/MMC clock.
SDIO_CMD	I/O	Command input/output.
SDIO_D[7:0]	I/O	Data input/output for data lines DAT[7:0].



### 38.8 功能描述

### 38.9

### 38.10

## 【39】 选项字节区(Option Bytes)

选项字节区位于地址为 0x0800 0000- 0x0800 01FF。

选项字节区的各配置字段用于用户实现部分系统功能的配置。

地址	名称	说明
0x0800 0000	USERCFG0	ISP_CON 配置字
0x0800 0004	USERCFG1	SWD 保护位配置字
0x0800 0008	USERCFG2	IWDGCNT[19:0]、IWDGMODE、IWDGINTMASK、IWDGON 配置字

### 39.1 用户配置寄存器 0(USERCFG0)

地址: 0x0800 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保 留	保留			保 留	保 留	ISP_CON									
保 留	保留			保 留	保 留	R/ W									

位	标记	功能描述	默认值	读写
31:1	-	保留。	1	-
0	ISP_CON	ISP 功能使能控制位。 0: 使能 ISP 功能 1: 不使能 ISP 功能  注: 用户更改该位后, 必须产生 CPURST 以外的复位, 该位的功能才会有效。	1	R/W



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBIT CONFIDENTIAL



### 39.2 用户配置寄存器 1(USERCFG1)

地址: 0x0800 0000

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

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

位	标记	功能描述	默认值	读写
31:1	-	保留。	0xFFFFFFFF	-
0	SWDP	SWDP: SWD 保护位。 0: SWD 接口配置为受保护； 1: SWD 接口配置为不受保护。 注: 用户更改该位后, 必须产生 CPURST 以外的复位, 该位的功能 才会有效。	1	R/W



### 39.3 用户配置寄存器 2(USERCFG2)

地址: 0x0800 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IWDGON	保留	IWDGINTMASK	IWDGMODE									IWDTCNT[19:16]			
R/W		R/W	R/W									R/W			

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

位	标记	功能描述	默认值	读写
31	IWDGON	IWDG 配置位。 0: 硬件不自动启动IWDG 1: 硬件自动启动IWDG	1	R/W
30	-	保留。	1	-
29	IWDGINTMASK	IWDG 中断屏蔽位。 0: IWDG 不屏蔽中断 1: IWDG 屏蔽中断	1	R/W
28	IWDGMODE	IWDG 模式配置位。 0: 复位方式 1: 中断方式	1	R/W
27:20	-	保留。	0xFF	-
19:0	IWDGCNT[19:0]	IWDG 计数值寄存器	0xFFFF	R/W

### 39.4 注意

由于对选项字节区的改写需要符合 Flash 的擦写时序，用户改写选项字节区参数时需要把整个选项字节区的数据读出备份，擦除整个选项字节区后改写需要配置的字段，再全部写回选项字节区。



## 【40】 Debug 支持(DBG)

### 40.1 简介

通用 MCU 的 JTAG/SWD 是非加密的传统型 JTAG/SWD 调试接口，这在客户程序的保密性以及系统的安全性方面具有非常大的隐患。ZB32F103 芯片为了保护用户程序及提高系统安全性，在 SWD 端口上集成设计了安全授权电路。本芯片出厂时，端口默认配置 SWD 调试接口，配合本产品的上位机软件或用户程序，在上位机软件 SWDP 位置处写入 0 值时，SWD 调试端口在复位或下次上电后自动断开。

调试接口 SWD 不可以通过 SWDP 写 0xFF 打开，如果需要打开 SWD 接口只有对芯片进行全片擦除。

注：SWDP 在选项字节区(Option Bytes)的用户配置寄存器 1 (USERCFG1)中，请参阅 [38.2 用户配置寄存器 1\(USERCFG1\)](#)。

注意：

ARM Cortex™-M3 内核提供集成的片上调试功能，可参考：

- Cortex™-M3 技术参考手册(TRM)
- ARM 调试接口 V5
- ARM CoreSight 开发工具集(r1p0 版)技术参考手册

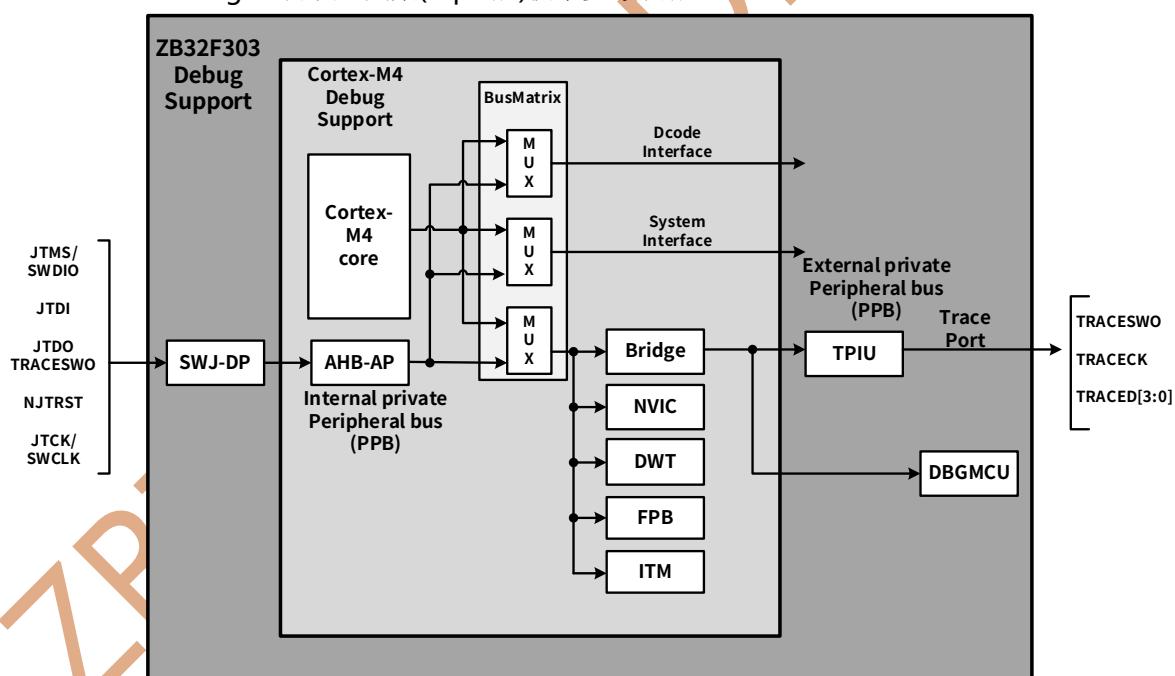


Figure 40-1 ZB32F103 级别和 Cortex™-M3 级别的调试框图

MCUDBG 模块可帮助调试器调试低功耗模式，定时器，I2C，bxCAN，WWWDG 与 IWDG。当相应位置位，在低功耗模式下提供时钟或保持计数器定时器，WWWDG，IWDG，CAN 或 I2C 的当前状态。

MCU 调试模块协助调试器提供以下功能：

- 低功耗模式的调试支持
- 在断点时提供定时器、看门狗、I2C 和 bxCAN 的时钟控制



-- ID 代码

-- 对跟踪脚分配的控制(64-Pin not Support)

ZB32F103 使用 Cortex™-M3 内核，该内核内含硬件调试模块，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都是可以查询的。完成查询后，内核和外设可以被复原，程序将继续执行。

当 ZB32F103 微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。支持两种调试接口：

- SWD 串行接口
- JTAG 调试接口

SWJ 调试端口脚请参考[调试接口信号引脚配置表](#)

## 40.2 SWD 调试接口说明

### 40.2-1 SWD 调试接口的引脚分配

该芯片的 2 个 GPIO 可以用作 SWD 接口引脚。这些引脚在所有的封装上都存在。

SWD 接口引脚名称	SWD 接口类型	SWD 接口功能	引脚分配	
SWDIO	输入/输出	串行数据输入/输出	PA13	PA11
SWCLK	输入	串行时钟	PA14	PA12

### 40.2-2 SWD 引脚的内部上拉和下拉

保证 SWD 的引脚不是悬空的是非常必要的，因为他们直接连接到 D 触发器控制着调试模式。必须特别注意 SWCLK 引脚，因为它直接连接到一些 D 触发器的时钟端。

为了避免任何未受控制的 I/O 电平，本芯片在 SWD 引脚上嵌入了内部上拉和下拉电阻。

- SWDIO: 内部上拉
- SWCLK: 带下拉的输入

软件也可以把这些 I/O 口配置为 GPIO 使用。



#### 40.3 SWD 保护位工作原理

- 客户拿到的芯片是空白片，SWDP 的值是 1，所以 SWD 是默认打开的。
- 客户使用 Keil/IAR 进行软件开发，开发完成后，可以通过 Keil/IAR 直接下载，也可以通过烧录器下载。
- 客户下载完程序之后，通过上位机配置 SWDP 的值为 0，开启保护位。
- MCU 重新复位之后，SWD 保护位即刻生效。
- 客户再需要进行 SWD 调试的话，需要将芯片全片擦除，才能重新打通 SWD 通道。

#### 40.4 支持定时器、看门狗、bxCAN 和 I2C 的调试

在产生断点时，有必要根据定时器和看门狗的不同用途选择计数器的工作模式：

- 在产生断点时，计数器继续计数。这在输出 PWM 控制电机时常常要用到。
- 在产生断点时，计数器停止计数。这对于看门狗的计数器是必需的。

对于 bxCAN，用户可以选择在断点期间阻止接收寄存器的更新。

对于 I2C，用户可以选择在断点期间阻止 SMBUS 超时。

#### 40.5 ID 代码

在 ZB32F103 微控制器内部有多个 ID 编码，强烈建议工具设计者使用映射到外部 PPB 存储器上地为 0xE0042000 的 MCUDeviceID 来锁定调试器。这个 ID 定义了 MCU 的部件号和硅片版本。它是 MCUDBG 的一个组成部分，并且映射到外部 PPB 总线上。

使用 JTAG 调试口（4~5 个引脚）或 SW 调试口（2 个引脚）或通过用户代码都可以访问此编码。即使当 MCU 处于系统复位状态下这个编码也可以被访问。

##### 40.5-1 微控制器设备 ID 编码

微控制器 STM32F10xxx 内含一个 MCU ID 编码。这个 ID 定义了 ST MCU 的部件号和硅片版本。它是 DBG\_MCU 的一个组成部分，并且映射到外部 PPB 总线上(见 29.16 节)。使用 JTAG 调试口(4~5 个引脚)或 SW 调试口(2 个引脚)或通过用户代码都可以访问此编码。即使当 MCU 处于系统复位状态下这个编码也可以被访问。

地址：0xE004 2000

只支持 32 位访问

只读=0xXXXXX410，其中 X 为内容不确定的位

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



#### 40.6 在低功耗模式下使用 SWD

使用 WFI 和 WFE 可以进入低功耗模式。MCU 支持多种低功耗模式，分别可以关闭 CPU 时钟，或降低 CPU 的能耗。内核不允许在调试期间关闭 FCLK 或 HCLK。这些时钟对于调试操作是必要的，因此在调试期间，它们必须工作。MCU 使用一种特殊的方式，允许用户在低功耗模式下调试代码。为实现这一功能，调试器必须先设置一些配置寄存器来改变低功耗模式的特性。

- 在睡眠模式下，调试器必须先置位 `DBG_CR` 寄存器的 `DBG_SLEEP` 位。这将为 `HCLK` 提供与 `FCLK`（由代码配置的系统时钟）相同的时钟。
- 在停止模式下，调试器必须先置位 `DBG_STOP` 位。这将激活内部 RC 振荡器，在停止模式下为 `FCLK` 和 `HCLK` 提供时钟。

Note : (1) FCLK 是 Cortex™-M3 的自由运行时钟, 详情见 ARM 的 Cortex™-M3 技术参考手册

(2) HCLK to MCU AHB Bus and Memory.



#### 40.7 DBG 寄存器列表

DBG 基址: 0xE004 2004

偏移地址	名称	描述	默认值
0x000	DBG_CR	Debug 模式控制寄存器	0x00000000

Table 40-1 Debug 寄存器映象和复位值

#### 40.8 DBG\_CR 寄存器列表 DBG 寄存器说明

##### 40.8-1 Debug 模式控制寄存器(DBG\_CR)

地址偏移: 0x00 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	DBG_TIM2A_STOP	DBG_TIM2B_STOP	DBG_TIM2_STOP	DBG_TIM11_STOP	DBG_TIM10_STOP	DBG_LPTIM_STOP	DBG_RTC_STOP	Reserved	DBG_STANDBY	DBG_SLEEP	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG_I2C1_SMBUS_TIMEOUT	DBG_CAN1_STOP	DBG_PCA_STOP	DBG_TIM1B_STOP	DBG_TIM1A_STOP	DBG_WWDG_STOP	DBG_IWWDGSTOP	TRACE_MODE[1:0]	TRACE_IOEN	R/W						



位	标记	功能描述	复位值	读写
31:28	Revered	--	0x0	r
27	DBG_TIM2C_STOP	DBG_TIM2C_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出	0x0	rw
26	DBG_TIM2B_STOP	DBG_TIM2B_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出	0x0	rw
25	DBG_TIM2A_STOP	DBG_TMR2A_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出	0x0	rw
24	DBG_TIM2_STOP	DBG_TIM2_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出 (就好象对某一暂停事件的紧急响应, 停止定时器)。	0x0	rw
23	DBG_TIM11_STOP	DBG_TIM11_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出 (就好象对某一暂停事件的紧急响应, 停止定时器)。	0x0	rw
22	DBG_TIM10_STOP	DBG_TIM10_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出 (就好象对某一暂停事件的紧急响应, 停止定时器)。	0x0	rw
21	DBG_LPTIM_STOP	Low Power Timer 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
20	DBG_RTC_STOP	RTC 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W



位	标记	功能描述	复位值	读写
19:17	Revered	--	0x0	r
16	DBG_I2C2_SMBUS_T IMEOUT	DBG_I2C2_SMBUS_TIMEOUT: 当核心停止时停止 SMBUS 超时模式。 0: 与正常模式操作相同; 1: 冻结 SMBUS 的超时控制。	0x0	rw
15	DBG_I2C1_SMBUS_T IMEOUT	DBG_I2C1_SMBUS_TIMEOUT: 当核心停止时停止 SMBUS 超时模式。 0: 与正常模式操作相同; 1: 冻结 SMBUS 的超时控制。	0x0	rw
14	DBG_CAN1_STOP	当内核进入调试状态时, CAN1 停止运行。 0: CAN1 仍然正常运行; 1: CAN1 的接收寄存器不继续接收数据。	0x0	rw
13	DBG_PCA_STOP	DBG_PCA_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出。	0x0	rw
12	DBG_TIM1B_STOP	DBG_TIM1B_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出。	0x0	rw
11	DBG_TIM1A_STOP	DBG_TIM1A_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出。	0x0	rw
10	DBG_TIM1_STOP	DBG_TMR1_STOP: 当核心停止时停止定时器计数器 0: 当核心停止时, 仍然向相关定时器的计数器提供时钟, 定时器输出工作正常; 1: 当核心停止时, 切断相关定时器的计数器的时钟, 同时关闭定时器的输出 (就好象对某一暂停事件的紧急响应, 停止定时器)。	0x0	rw
9	DBG_WWDG_STOP	DBG_WWDG_STOP: 当内核进入调试状态时调试窗口看门狗停止工作。 0: 窗口看门狗计数器仍然正常工作; 1: 窗口看门狗计数器停止工作。	0x0	rw
8	DBG_IWDG_STOP	DBG_IWDG_STOP: 当内核进入调试状态时看门狗停止工作	0x0	rw



位	标记	功能描述	复位值	读写
		0: 看门狗计数器仍然正常工作; 1: 看门狗计数器停止工作。		
7:6	TRACE_MODE	TRACE_MODE[1: 0]和 TRACE_IOEN: 跟踪引脚分配控制 -当 TRACE_IOEN=0 时: TRACE_MODE=xx: 不分配跟踪引脚 (默认状态)。 -当 TRACE_IOEN=1 时: TRACE_MODE=00: 跟踪引脚使用异步模式; TRACE_MODE=01: 跟踪引脚使用同步模式, 并且数据长度为 1; TRACE_MODE=10: 跟踪引脚使用同步模式, 并且数据长度为 2; TRACE_MODE=11: 跟踪引脚使用同步模式, 并且数据长度为 4。 Synchronous Mode with a TRACEDATA size of 2 TRACE_MODE=11: TRACE pin assignment for Synchronous Mode with a TRACEDATA size of 4	0x0	rw
5	TRACE_IOEN	Trace pin assignment enable	0x0	rw
4:3	Reserved	--	0x0	rw
2	DBG_STANDBY	DBG_STANDBY: 调试待机模式。 0: (FCLK 关, HCLK 关) 整个数字电路部分都断电。从软件的观点看, 退出 STANDBY 模式与复位是一样的 (除了一些状态位指示了微控制器刚从 STANDBY 状态退出)。 1: (FCLK 开, HCLK 开) 数字电路部分不下电, FCLK 和 HCLK 时钟由内部 RLD 振荡器 提供时钟。另外, 微控制器通过产生系统复位来退出 STANDBY 模式和复位是一样的。	0x0	rw
1	DBG_STOP	DBG_STOP: 调试停止模式。 0: (FCLK 关, HCLK 关) 在停止模式时, 时钟控制器禁止一切时钟 (包括 HCLK 和 FCLK)。当从 STOP 模式退出时, 时钟的配置和复位之后的配置一样 (微控制器由 8MHz 的内部 RC 振荡器 (HSI) 提供时钟)。因此, 软件必需重新配置时钟控制系统启动 PLL, 晶振等。	0x0	rw



位	标记	功能描述	复位值	读写
		1: (FCLK 开, HCLK 开) 在停止模式时, FCLK 和 HCLK 时钟由内部 RC 振荡器提供。当退出停止模式时, 软件必需重新配置时钟系统启动 PLL, 晶振等 (与配置此比特位为 0 时的操作一样)。		
0	DBG_SLEEP	DBG_SLEEP: 调试睡眠模式 0: (FCLK 开, HCLK 关) 在睡眠模式时, FCLK 由原先已配置好的系统时钟提供, HCLK 则关闭。由于睡眠模式不会复位已配置好的时钟系统, 因此从睡眠模式退出时, 软件不需要重新配置时钟系统。 1: (FCLK 开, HCLK 开) 在睡眠模式时, FCLK 和 HCLK 时钟都由原先配置好的系统时钟提供。		



## 【41】 在线电路编程(ICP)

通过在线电路编程 (ICP) 编程 Flash。如果产品在开发中，或在终端客户的产品需要固件升级，采用硬件编程模式非常困难且不方便。采用 ICP 方式将很简单，且不需要将微控制器从板上拆下来。ICP 方式同样允许客户在量产电路板上编程设备，在设备装配完成后再编程，这样允许设备编程最新的固件或定制化固件。

执行 ICP 功能，仅需要 5 个引脚 ICP\_CLK、ICP\_DIN、ICP\_CS、ICP\_DOUT、RSTN 用于进入或退出 ICP 模式，ICP\_DIN、ICP\_DOUT 为数据输入输出脚，ICP\_CLK 为编程时钟输入脚，ICP\_CS

和 RSTN 是编程时的控制功能引脚。用户需要在系统板上预留 VDD、GND 以及这五个脚。

ICP 编程器是根据 MCU 的电气特性专门设计的，是很高效稳定的编程方式。具体内容请参考 Application Notes。

ICP 接口连接方式示意图：

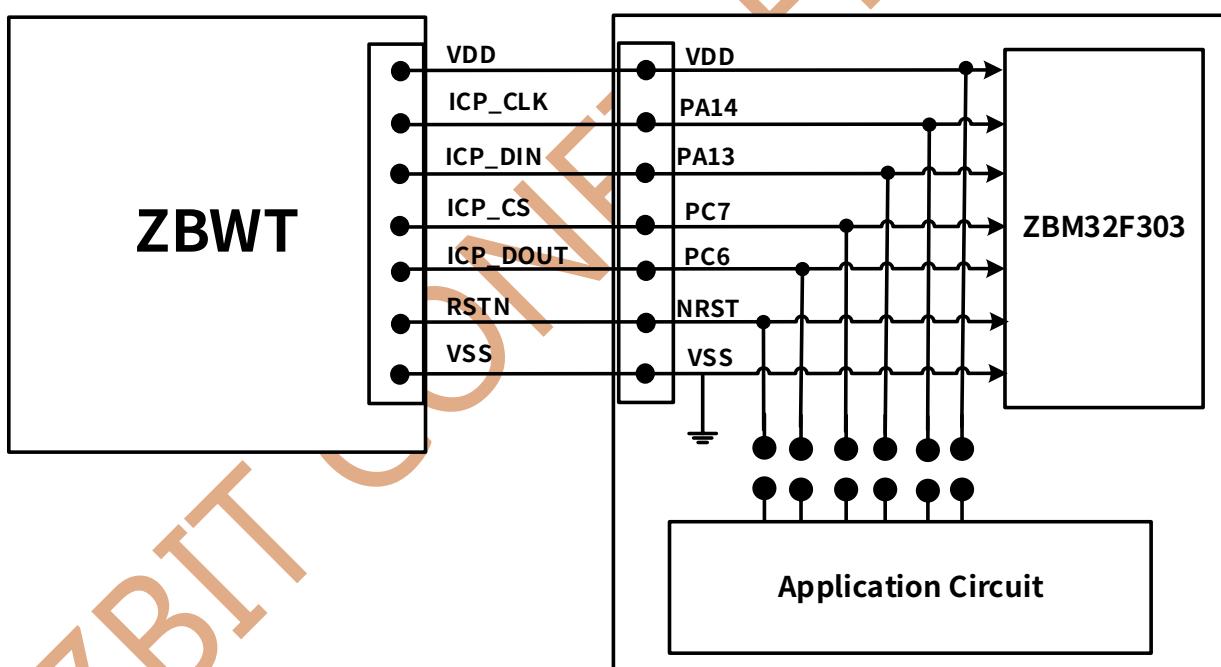


Figure 41-1 ICP 接口连接方式示意图

Note:

在使用 ICP 更新代码时，NRST、PA14、PA13、PC7、PC6 必须断开与系统版负载的连接。在 ICP 编程结束后，建议关闭系统电源移去 ZBWT，然后再接上电源。



## 【42】 附录 1: SysTick 定时器(SYST)

### 42.1 SysTick 定时器简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0+ 处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 中断（中断号为 15），该异常事件会引起 SysTick 中断处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 中断的产生是可控的，如果中断被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询溢出标志。

### 42.2 设置 SysTick 定时器

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

配置 SYST\_CSR.ENABLE 为 0，禁止 SysTick。

配置 SYST\_CSR.CLKSOURCE，选择 SysTick 的时钟源。

配置 SYST\_RVR，选择 SysTick 的溢出周期。

向 SYST\_CVR 写入任意值，清零 SYST\_CVR 及 SYST\_CSR.COUNTFLAG。

配置 SYST\_CSR.TICKINT 为 1，使能 SysTick 中断。

配置 SYST\_CSR.ENABLE 为 1，使能 SysTick，计数开始。

在中断服务程序中读取 SYST\_CSR 以清除溢出标志。

注：SysTick 定时器溢出时间为 SYST\_RVR+1 个 SysTick 时钟周期，配置举例：

SysTick 时钟源	SysTick 时钟周期	SYST_RVR	SysTick 定时器溢出时间
内核时钟 HCLK (被设为 4MHz)	0.25us	3999	$1\text{ms} = 0.25\text{us} * (3999+1)$



### 42.3 SysTick 定时器寄存器列表

地址	名称	描述	复位值	CMSIS 符号
0xE000 E010	SYST_CSR	SysTick 定时器控制和状态寄存器	0x0000 0000	SysTick->CTRL
0xE000 E014	SYST_RVR	SysTick 定时器重载值寄存器	未定义	SysTick->LOAD
0xE000 E018	SYST_CVR	SysTick 定时器当前值寄存器	未定义	SysTick->VAL
0xE000 E01C	SYST_CALIB	SysTick 定时器校准值寄存器	0x4000 9C3F	SysTick->CALIB

Table 42-1 SysTick 定时器寄存器列表和复位值

注：SYST\_CALIB 寄存器与 RCC\_STICKCR 有对应关系，请参考 7.4.15 节 SysTick Timer 控制寄存器(RCC\_STICKCR)。

### 42.4 SysTick 定时器寄存器说明

#### 42.4-1 SysTick 定时器控制和状态寄存器(SYST\_CSR)

位	符号	功能描述	类型	复位值
31:17	保留	-	-	-
16	COUNTFLAG	SysTick 定时器溢出标志 1: SysTick 定时器发生下溢出。 0: SysTick 定时器未发生溢出。 读该寄存器，可清除COUNTFLAG 标志	RO	0
15:3	保留	-	-	-
2	CLKSOURCE	SysTick 时钟源选择 1: 使用内核时钟 (HCLK) 0: HCLK/4	RW	0
1	TICKINT	SysTick 中断使能 1: 使能中断 0: 禁止中断	RW	0



0	ENABLE	SysTick 定时器使能1: 使能 SysTick 0: 禁止 SysTick	RW	0
---	--------	--	----	---

ZBT CONFIDENTIAL



#### 42.4-2 SysTick 定时器重载值寄存器(SYST\_RVR)

位	符号	功能描述	类型	复位值
31:24	保留	-	-	-
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

#### 42.4-3 SysTick 定时器当前值寄存器(SYST\_CVR)

位	符号	功能描述	类型	复位值
31:24	保留	-	-	-
23:0	CURRENT	读该寄存器，获取SysTick 定时器的当前计数值；写任意值到该寄存器，清零该寄存器及COUNTFLAG。	RW	未定义

#### 42.4-4 SysTick 定时器校准值寄存器(SYST\_CALIB)

位	符号	功能描述	类型	复位值
31	NOREF	SysTick 定时器是否使用外部参考时钟 0: HCLK/4 1: 使用内核时钟(HCLK)	RO	0
30	SKEW	10ms TENMS 值是否准确 0: 准确 1: 不准确	RO	1
29:24	保留		-	-
23:0	TENMS[23:0]	SysTick 10ms 校准值，此值为使用外部参考时钟 HCLK/4(4MHZ)的10ms 校准值。	RO	0x009C3F



## 【43】 Flash 控制器(Flash)

### 43.1 Flash 控制器概述

本芯片包含 1 颗 64K / 32K Byte 容量的嵌入式 Flash 存储器，包括一个 128 / 64 sector 的 Main Array

区域，一个 8 sector 的 NVR 区域。每个 sector 的容量为 512 Byte。Flash 的 Main Array 区域是给用

户使用的，可以存放用户开发的程序和数据。NVR 区域中，一个 sector 用于存放系统配置，一个 sector

用于存放选项字节，其余的 6 个 sector 用于存放系统的 ISP 程序。本模块支持对 Flash 存储器的擦除、编程以及读取操作。此外，本模块支持对 Flash 存储器擦写的保护，以及控制寄存器的写保护。

### 43.2 Flash 结构框图

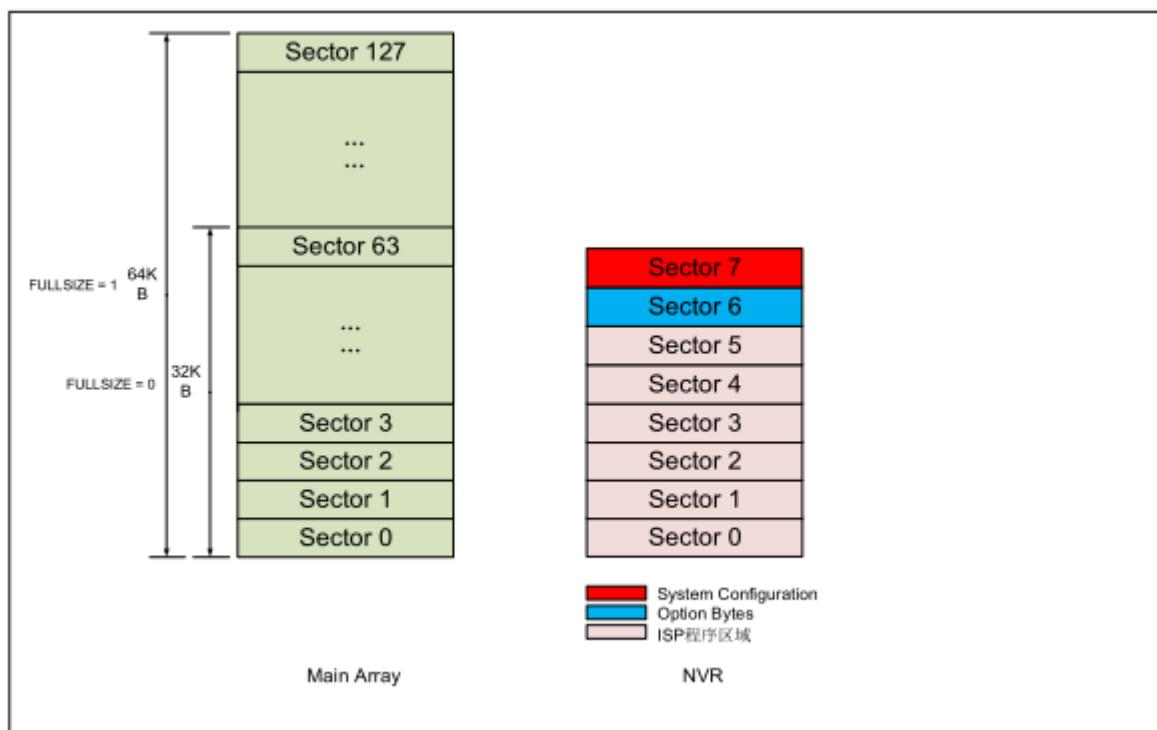


Figure 43-1 Flash 结构框图

Main Array 区域用于存放用户的代码。

NVR 区域的 sector 0~5 这 6 个 sector 用于存放厂家提供的系统 ISP(在系统可编程)代码，用户可以通过 ISP 代码下载开发的应用程序，用户程序不能对 ISP 代码进行读取或者擦除操作。NVR 区域的



Sector 6 是选项字节区域，用于用户实现部分系统功能的配置。NVR 区域的 Sector 7 是系统配置区，用于存放系统配置值。

### 43.3 功能描述

本控制器支持对 Flash 的 byte(8bits)、half-word(16bits)、word(34bits)三种位宽读写操作。注意，byte

操作的地址必须按 byte 对齐，half-word 操作的目标地址必须按 half-word 对齐(地址最低位为 0)，word 操作的目标地址必须按 word 对齐(地址最低两位为 0)。如果读写操作的地址没有按照位宽规定对齐，该操作无效，并且系统会进入 hard fault 出错中断。

#### 43.3-1 擦除操作

##### 43.3-1.1 Sector 擦除

Sector 擦除操作步骤如下图所示：

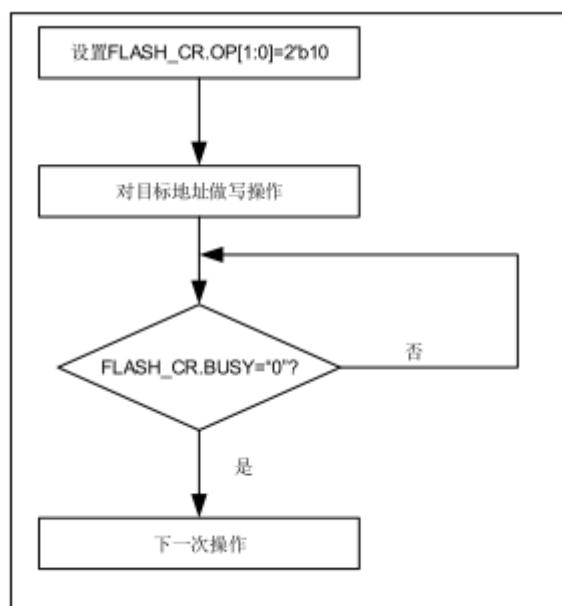


Figure 43-2 擦除操作步骤

注意：

1. 控制器忽略目标地址的低 9 位，只要目标地址落在该页地址范围之内即可。
2. 该写操作用于触发 sector 擦除操作，所写数据可以是任意数据。
3. 如果当前 sector 擦除指令是在 Flash 内执行，则 CPU 会暂停取指令和执行指令的动作，自动等待 Flash 的 BUSY 状态结束后继续取指令并执行。



4. 如果当前 sector 擦除指令是在 RAM 内执行，则 CPU 不会停止取指和执行，在对 Flash 进行

任何操作之前，软件必须判断 Flash 的 BUSY 状态是否结束。

ZBT CONFIDENTIAL

---

---



### 43.3-1.2 Chip 擦除

Chip 擦除操作步骤如下图所示：

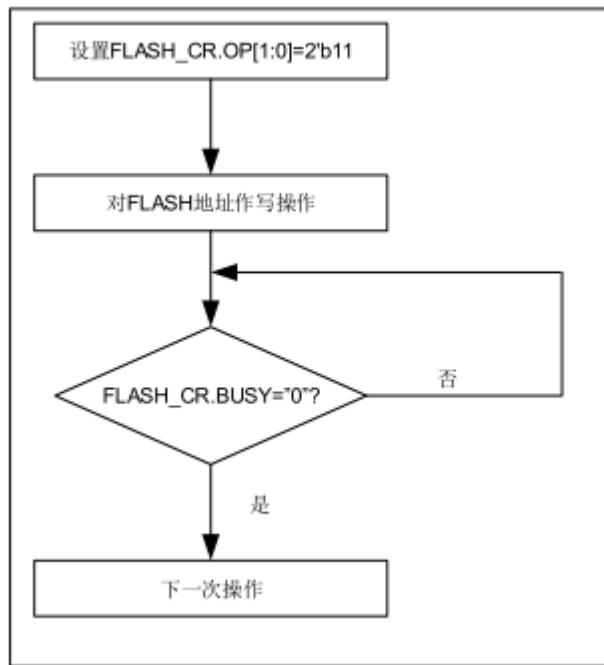


Figure 43-3 Chip 擦除操作步骤

注意：

1. Chip 擦除只能针对 Flash 的 Main Array 操作，NVR 区域只能进行 sector 擦除；
2. 控制器忽略目标地址的低 15 位，只要目标地址落在 Flash 地址范围之内即可；
3. 该写操作用于触发 chip 擦除操作，所写数据可以是任意数据；
4. 如果当前 chip 擦除指令是在 Flash 内执行，则 CPU 会暂停取指令和执行指令的动作，自动等待 Flash 的 BUSY 状态结束后继续取指令并执行。
5. 如果当前 chip 擦除指令是在 RAM 内执行，则 CPU 不会停止取指和执行，在对 Flash 进行任何操作之前，软件必须判断 Flash 的 BUSY 状态是否结束。



### 43.3-2 写操作

写操作步骤如下图所示：

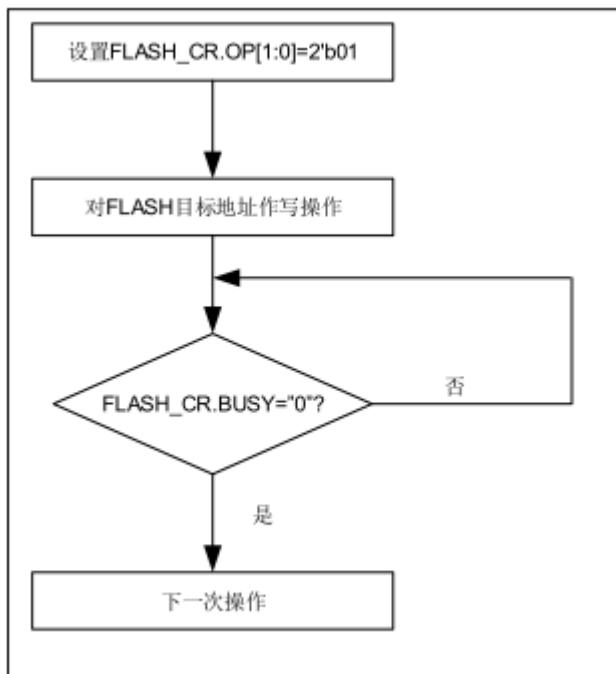


Figure 43-4 写操作步骤

注意：

- (1) 如果当前写操作指令是在 Flash 内执行，则 CPU 会暂停取指令和执行指令的动作，自动等待 Flash 的 BUSY 状态结束后继续取指令并执行。
- (2) 如果当前写操作指令是在 RAM 内执行，则 CPU 不会停止取指和执行，在对 Flash 进行任何操作之前，软件必须判断 Flash 的 BUSY 状态是否结束。

### 43.3-3 读操作

读 Flash 操作和 CPU 读 SRAM 存储器的操作一样，无特殊要求。本芯片片内 Flash 的最大读出速度为 20ns，如要优化 CPU 的执行须欲取等待一个周期针对 Flash 读取，如下所示：

- (SYSCLK)频率与闪存访问时间的关系：
  - 0 等待周期，当  $0 < \text{SYSCLK} < 32\text{MHz}$
  - 1 等待周期，当  $32\text{MHz} < \text{SYSCLK} \leq 64\text{MHz}$



#### 43.3-4 擦写时间

Flash 存储器对擦除和编程操作的控制信号都有严格的时间要求，控制信号的有效时间超出设计要求会带来擦除和编程操作失败，本芯片设置了 Flash sector 擦除时间寄存器(Flash\_TSERASE)、Flash chip 擦除时间寄存器(Flash\_TMERASE)、Flash 编程时间寄存器(Flash\_TPROG)三个寄存器可以分别对 sector、chip 擦除以及编程(写)时间进行调整。

注：芯片在 CP 阶段会对 Flash 的擦除和编程的时间参数进行测试并写入系统参数配置区域，在芯片上电后电路会自动把这些参数加载到 sector、chip 擦除时间寄存器和编程时间寄存器中，因此一般情况下不建议改动这些已设置好的擦写时间参数。

#### 43.3-5 擦写保护

##### 43.3-5.1 擦写保护位

整个 64K Byte Flash 存储器被划分为 128 个 sector，为了防止应用中的意外擦写操作改变 Flash 中的内容，一共设置了 64 个擦写保护位，每个擦写保护位负责保护 2 个 sector 的区域。保护位寄存器 Flash\_SLOCK.SLOCK0/1[31:0]默认值为 “0000\_0000”，即不允许擦写。只有把相应保护位修改为“1”，该 sector 才能允许被擦写。当 Flash 存储器中的任意 sector 被擦写保护时，对该 Flash 的 chip

擦写也会被控制器自动屏蔽，并且会发生报警标志位和中断信号。如需进行 chip 擦写，必须修改保护位寄存器 Flash\_SLOCK.SLOCK0/1[31:0]的值为 “0xFFFFFFFF”。

##### 43.3-5.2 PC 地址擦写保护

CPU 在 Flash 中运行程序时会遇到一种情况：当前运行的 PC 指针正好落在软件擦写 Flash 的 sector 地址范围之内，那么该擦写操作也会被控制器自动屏蔽，并且会产生报警标志位和中断信号。

##### 43.3-5.3 寄存器写保护

为了防止应用时意外的 Flash 擦写操作改变 Flash 的内容，对 Flash 控制器寄存器的写操作和擦写操作时对 Flash 的写操作必须用写序列方式才能修改。具体操作步骤如下图所示：

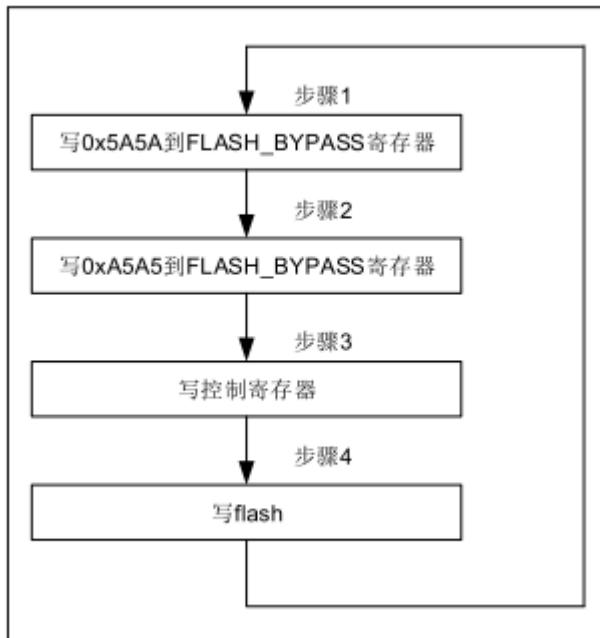


Figure 43-5 写寄存器 **BYPASS** 序列

注意：

步骤 1~4 操作之间不可插入任何其他写操作，否则该 BYPASS 序列操作失效，需要重新写入 0x5A5A, 0xA5A5 序列。



### 43.3-6 系统 BOOT 地址映像

#### 43.3-6.1 APP 程序 boot

下图是 APP 程序区 boot 的地址映像图。

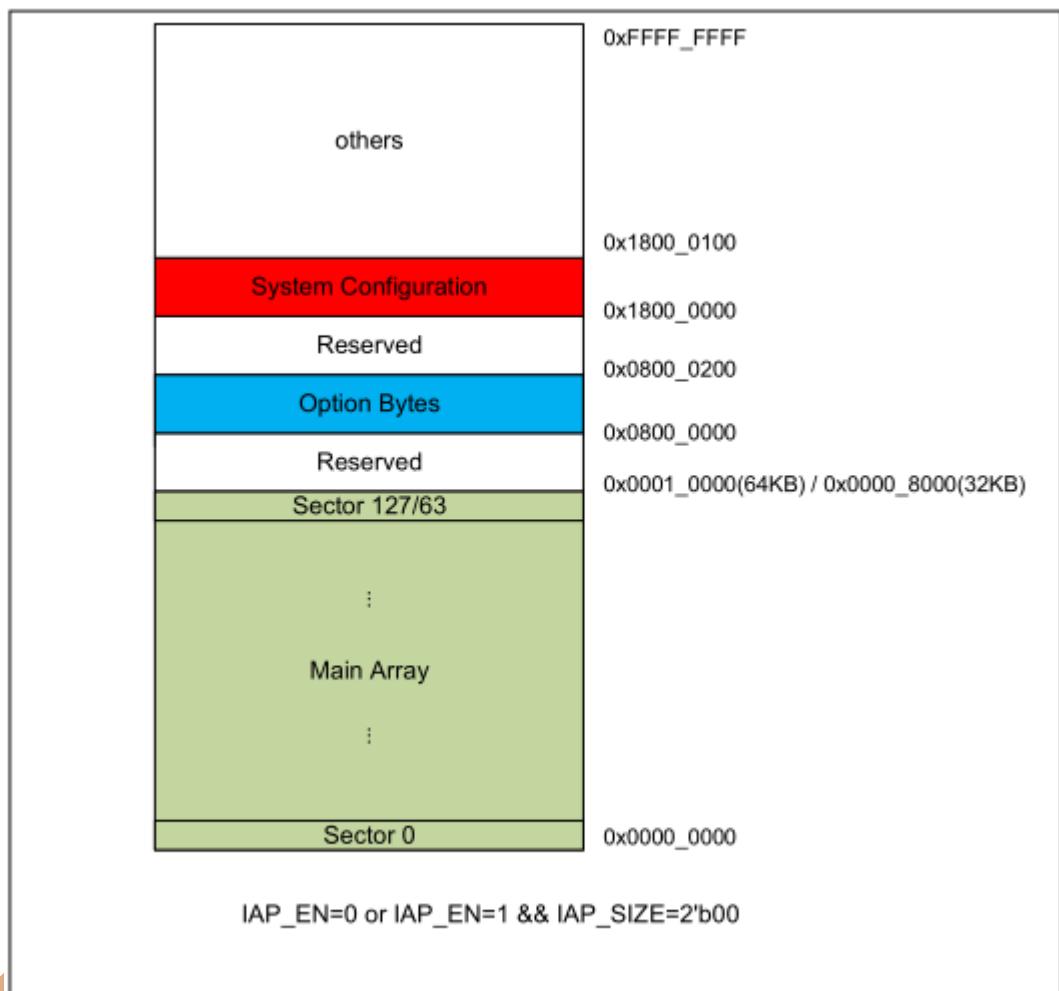


Figure 43-6APP 程序区 boot 地址映像图

APP 程序区 boot 状态也是芯片提供给客户的初始状态。此时 Main Array 被映像到逻辑地址 0x0000\_0000 开始的 64K/32K Byte 区域，选项字节区域被映像到逻辑地址 0x0800\_0000 开始的 512 Byte 区域，系统配置区域的低 256 Byte 被映射到 0x1800\_0000 开始的 256 Byte，程序对这 256 Byte 只能读，不能擦写。



#### 43.3-7 系选项字节区(Option Bytes)

具体内容请参考第 33 章选项字节区。

ZBT CONFIDENTIAL



### 43.3-8 在系统编程(ISP)

本芯片具有在系统编程(ISP)功能,ISP 程序由芯片生产厂家提供,烧录在 Flash 的 NVR 区域.用户通过执行 ISP 程序把应用程序编程到 Flash 的应用程序区,有两种软件方法可以启动 ISP 程序的执行。

#### 43.3-8.1 内部配置与软件进入 ISP

第一种方法是把选项字节区 USERCFG0 的 ISP\_CON 写为 0, 芯片上电或者 CPURST 以外复位后, 首先硬件自动把选项字节区 USERCFG0 的 ISP\_CON 加载到 FMC 寄存器 Flash\_ISPCON 的 ISP\_CON, CPU 启动后 Bootloader 程序识别到 FMC 寄存器 Flash\_ISPCON 的 ISP\_CON 为 0 就会调用 ISP 程序, 配合厂家提供的 ISP 上位机软件把应用程序下载到 Flash 的应用程序区域里。第二种方法客户不必修改选项字节区 USERCFG0 的 ISP\_CON, 也不需要重新上电, 只要直接修改 FMC 寄存器 Flash\_ISPCON 的 ISP\_CON 位为 0, 然后产生 CPURST 就能使设置生效。在系统编程的流程如下图所示。( Set BOOT0 =0 )

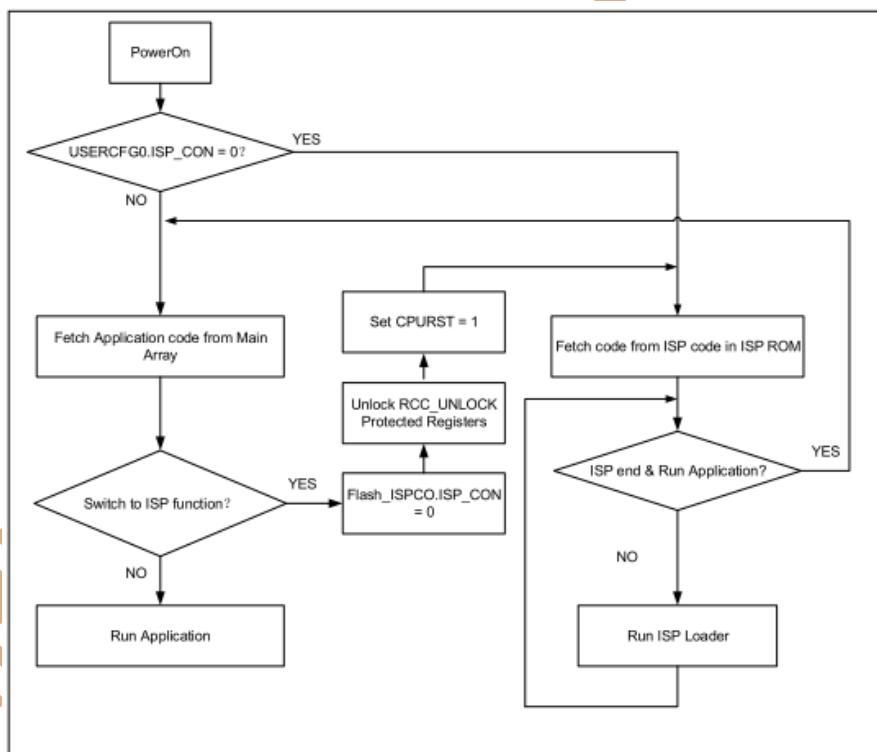


Figure 43-7 内部配置与软件进入 ISP 流程图

除了软件方法启动 ISP 之外, 芯片还有一种硬件启动 ISP 的途径, 通过外接厂方提供的特定 ISP 工具注并配合厂方提供的上位机软件也可以启动 ISP。需要注意的是如果使用硬件启动 ISP, 需要使用芯片的 NRST、PD1 和 PC7 这三个引脚, 推荐在制作 PCB 板时将这三个引脚引出至与特定 ISP 工具的接口适配的接插件上。



### 43.3-8.2 外部 BOOT PIN 选择

BOOT PIN 程序請參考 Fig11-8. 說明如下：

(1) 当复位时 **BOOT0 (PD03)** 管脚为高电平

- (a) **BOOT1 (PA09)** 管脚为低电平，
  - 芯片工作于 Test mode on SRAM(3KB: 0x0000\_0000~0x0000\_0BFF)，可通过 SWD 协议对 SRAM 进行编程。
- (b) **BOOT1 (PA09)** 管脚为高电平，
  - 芯片工作于用户模式 QSPI 内存映像模式(128MB: 0x0000\_3000~0x07FF\_FFFF)，可通过 SWD 协议对 SRAM 进行编程。

(2) 当复位时 **BOOT0 (PD03)** 管脚为低电平，

- 芯片工作于用户模式 eflash (64KB: 0x0000\_0000~0x0000\_FFFF)

**Step1** : ISP 编程模式，→ 可通过 ISP 协议对 FLASH 进行编程。

**Step2** : 芯片执行 main FLASH 内的程序代码，可通过 SWD 协议对 FLASH 进行编程。

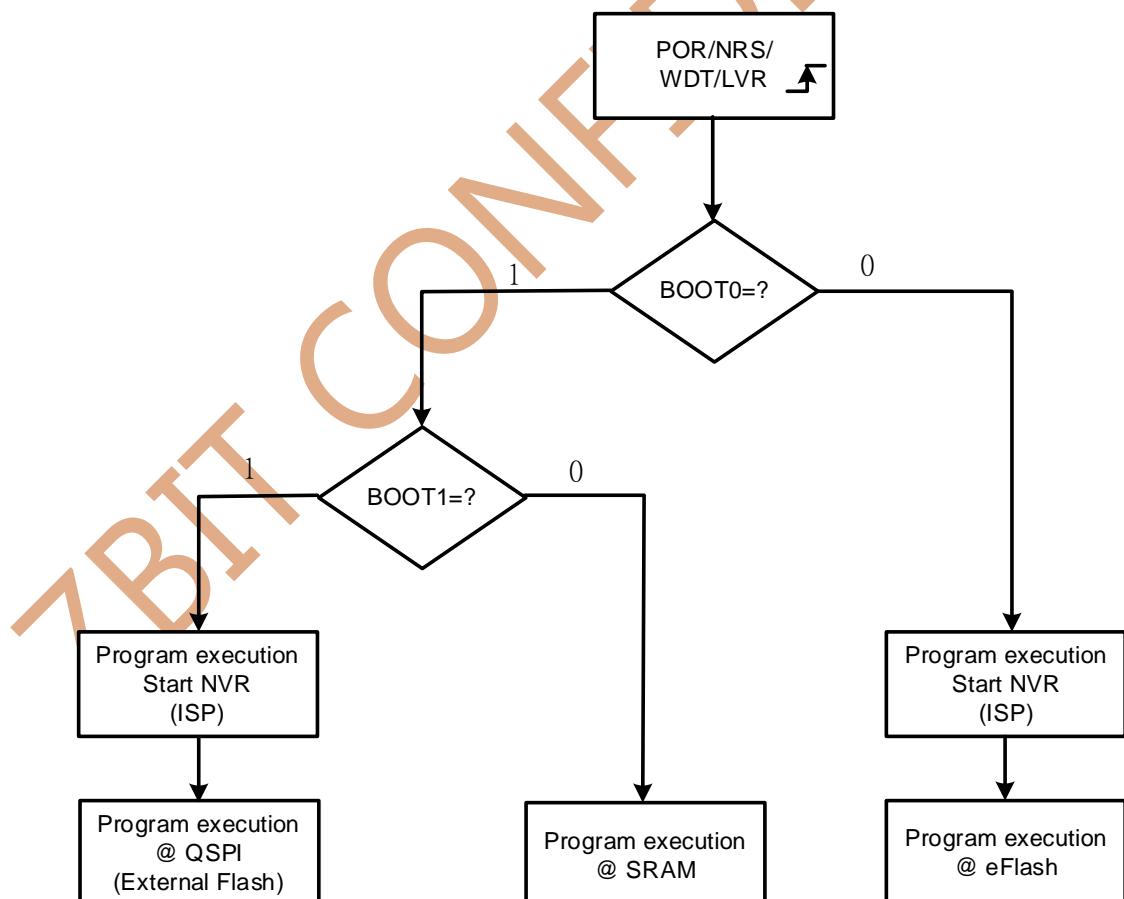


Figure 43-8 外部 BOOT0,BOOT1 PIN 选择流程图

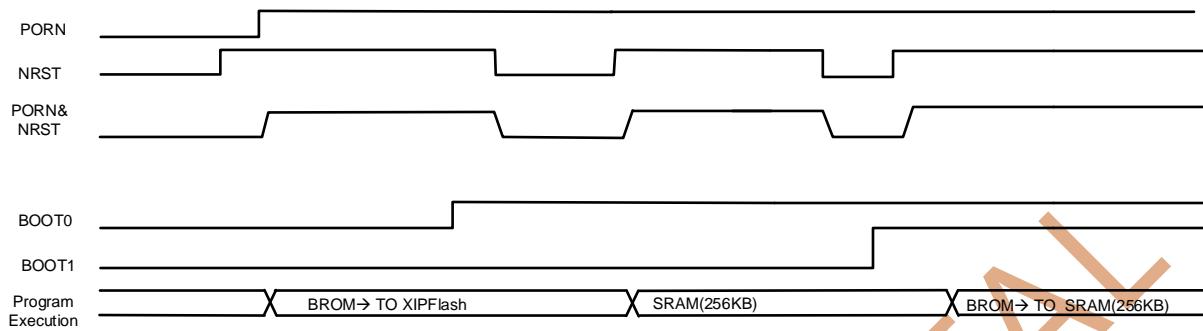


Figure 43-9BOOT0,BOOT1 PIN 时序图



#### 43.4 Flash 寄存器列表

基址址: 0x4002 0400

偏移地址	名称	描述	默认值
0x00	Flash_CR	控制寄存器	0x0000 0000
0x04	Flash_IFR	中断标志寄存器	0x0000 0000
0x08	Flash_ICLR	中断标志清除寄存器	0x0000 0000
0x0C	Flash_BYPASS	0x5A5A-0xA5A5 序列寄存器	0x0000 0000
0x10	Flash_SLOCK0	sector 擦写保护寄存器 0	0x0000 0000
0x14	Flash_SLOCK1	sector 擦写保护寄存器 1	0x0000 0000
0x18	Flash_ISPCON	Flash ISP 控制寄存器	0x0000 0001
0x28	Flash_RCR	Flash 读取周期寄存器	

Table 43-1 寄存器列表和复位值

注: 所有 Flash 寄存器只能按 word 方式读写。



### 43.5 寄存器说明

#### 43.5-1 Flash\_CR(Flash 控制寄存器)

地址偏移: 0x00 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
保留															

位	标记	功能描述	复位值	读写
31:5	-	保留, 始终读为 0。	0	-
4:3	IE[1:0]	IE[1]: Flash 擦写被保护地址中断使能; 0: 不使能; 1: 使能。 IE[0]: Flash 擦写 PC 值中断使能; 0: 不使能; 1: 使能。	2' b00	R/W
2	BUSY	空闲/忙标志位; 0: 空闲状态; 1: 忙状态。	1' b0	RO
1:0	OP[1:0]	Flash 操作模式寄存器 2' b00: Read 2' b01: Program 2' b10: Sector erase 2' b11: Chip erase	2' b00	R/W



#### 43.5.2 Flash\_IFR(Flash 中断标记寄存器)

地址偏移: 0x04 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
保留															

位	标记	功能描述	复位值	读写
31:2	-	保留, 始终读为 0。	0	-
1	IF1	擦写保护报警中断标志位	1' b0	RO
0	IF0	擦写 PC 地址报警中断标志位	1' b0	RO



#### 43.5.3 Flash\_ICLR(Flash 中断标记清除寄存器)

地址偏移: 0x08 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
保留															
保留															

位	标记	功能描述	复位值	读写
31:2	-	保留, 始终读为 0。	0	-
1	ICLR1	清除保护报警中断标志位: 写 0 清除; 写 1 无效	-	WO
0	ICLR0	清除 PC 地址报警中断标志位: 写 0 清除; 写 1 无效	-	WO



#### 43.5-4 Flash\_BYPASS(BYPASS 序列寄存器)

地址偏移: 0x0C 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYPASSSEQ[15:0]															
WO															

位	标记	功能描述	复位值	读写
31:16	-	保留, 始终读为 0。	0	-
15:0	BYPASSSEQ[15:0]	修改本模块寄存器之前必须对 BYPASSSEQ[15:0] 寄存器写入 0x5A5A, 0xA5A5 序列。一次正确写入序列后只允许写一次寄存器, 如需再次修改寄存器必须再次输入正确的 BYPASS 序列	16' h0000	WO



#### 43.5.5 Flash\_SLOCK0(sector 擦写保护寄存器 0)

地址偏移: 0x10 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK0[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK0[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	SLOCK0[31:0]	<p>Sector 擦写保护位</p> <p>0: 不允许擦写;</p> <p>1: 允许擦写。</p> <p>SLOCK0 对应: sector 62-63</p> <p>SLOCK0 对应: sector 60-61</p> <p>SLOCK0 对应: sector 58-59</p> <p>.</p> <p>.</p> <p>.</p> <p>SLOCK0 对应: sector 2-3</p> <p>SLOCK0 对应: sector 0-1</p>	32' h00 00_0000	R/W



#### 43.5-6 Flash\_SLOCK1(sector 擦写保护寄存器 1)

地址偏移: 0x14 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK1[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	SLOCK1[31:0]	<p>Sector 擦写保护位</p> <p>0: 不允许擦写;</p> <p>1: 允许擦写。</p> <p>SLOCK1 对应: sector 126-127</p> <p>SLOCK1 对应: sector 124-125</p> <p>SLOCK1 对应: sector 122-123</p> <p>.</p> <p>.</p> <p>.</p> <p>SLOCK1 对应: sector 66-67</p> <p>SLOCK1 对应: sector 64-65</p>	32' h00 00_0000	R/W



#### 43.5.7 Flash\_ISPCON(Flash ISPCON 配置寄存器)

地址偏移: 0x18 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
I															
保留															

位	标记	功能描述	复位值	读写
31:1		保留, 始终读为 0。	0	-
0	ISP_CON	ISP 功能配置位; 0: BOOTLOAD 流程中使能 ISP 功能。 1: BOOTLOAD 流程中跳过 ISP 功能直接切换至应用程序; ISP_CON 只能被 MCURST 复位, CPURST 不能复位 ISP_CON。 用户修改 ISP_CON 后通过 CPURST 使设置生效。	1' b1	R/W

注意: 写 Flash\_ISPCON 寄存器的 ISP\_CON 时, 高 16 位的数据必须为 16' h5A69。



#### 43.5.8 Flash 读取周期寄存器(Flash\_RCR)

地址偏移: 0x28

复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:1 6	Key	只有高位写 0x96A5 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
1	fdo_mask_en	0: flash dout always output 1: flash dout wait read signal(if wcycle = 0)	1'b0	R/W
0	WCYCLE	闪存读取 1: 等待0周期, 当 $0 < \text{SYSCLK} < 32\text{MHz}$ 0: 等待1周期, 当 $32\text{MHz} < \text{SYSCLK} \leq 64\text{MHz}$	0	R/W

Note : 写入 16' h96A5,解除保护.( 写 Flash\_RCR 寄存器的 wait\_cycle 时, 高 16 位的数据必须为 16' h96A5)



【44】 版本修订纪录

Version	Date.	Description
V0.00	2022-0622	初稿发布

ZBT CONFIDENTIAL



恒烁半导体  
Zbit Semi, Inc

ZB32F103 用户手册

ZBIT CONFIDENTIAL