

# ZB32L032

## ARM® Cortex®-M0+ 32-bit Micro-Controller User Manual

Version V1.09(For A-Version)

3/27, 2023

## 目录

目录 .....	2
图索引 .....	28
表索引 .....	37
<b>Note 索引 .....</b>	40
【1】简介 .....	41
【2】产品特性 .....	42
【3】产品功能描述 .....	43
3.1 设备概述 .....	43
3.2 内部功能框图结构 .....	44
3.3 32 位 Cortex®-M0+内核 .....	45
3.3-1 Flash .....	45
3.3-2 16K Bytes SRAM .....	45
3.4 时钟系统 .....	45
3.5 工作模式 .....	46
3.6 中断控制器(NVIC) .....	46
3.7 复位控制器 .....	47
3.8 通用 IO 端口(GPIO) .....	48
3.9 定时器和看门狗 .....	49
3.9-1 高级控制定时器(TIM1,1A,1B) .....	50
3.9-2 通用定时器(TIM2,2A,2B,2C) .....	50
3.9-3 可编程计数器阵列(PCA) .....	51
3.9-4 低功耗定时器(LPTIM) .....	51
3.9-5 基础定时器(TIM10/TIM11) .....	51
3.9-6 独立看门狗(IWDG) .....	51
3.9-7 系统窗口看门狗(WWDG) .....	51
3.9-8 SysTick 定时器(SYST) .....	52
3.10 实时时钟(RTC) .....	52
3.11 通用异步收发器(UART0/UART1/USART2/USART3) .....	52
3.12 通用异步收发器(LPUART) .....	52
3.13 串行外设接口(SPI) / 内部集成音频接口(I2S) .....	52
3.14 I2C 接口(I2C) .....	52

---

3.15 One-Wire 接口(OWIRE) .....	52
3.16 蜂鸣器(BEEP).....	53
3.17 自唤醒定时器(AWK).....	53
3.18 时钟校准/监测模块(CLKTRIM) .....	53
3.19 唯一 ID 号(UID) .....	53
3.20 循环冗余校验计算单元(CRC16/32) .....	53
3.21 模拟/数字转换器(ADC).....	53
3.22 低电压检测器(LVD) .....	54
3.23 电压比较器(VC).....	54
3.24 运算放大器(OPA) .....	54
3.25 嵌入式调试系统.....	54
3.26 加密型嵌入式 Debug 支持(DBG) .....	54
3.27 高级加密标准模块 (AES).....	54
3.28 真随机数发生器 (TRNG).....	54
3.29 编程模式.....	55
3.30 内嵌 SRAM .....	55
3.31 直接存储器存取(PDMA) .....	55
3.32 Flash 存储器.....	55
3.33 内嵌的自举程序(Bootloader) .....	55
<b>【4】 引脚配置及功能 .....</b>	<b>56</b>
4.1 引脚定义 .....	56
4.2 引脚功能配置 .....	58
4.3 选择芯片引脚复用功能说明 .....	61
4.4 模块与引脚信号说明 .....	64
4.5 串行引脚接口说明 .....	66
<b>【5】 系统和存储器概要 .....</b>	<b>67</b>
5.1 系统架构图 .....	67
5.2 存储器映射 .....	68
5.3 存储空间和模块地址 .....	69
5.4 BOOT PIN 地址映射 .....	70
<b>【6】 工作模式和电源管理 .....</b>	<b>72</b>
6.1 运行模式(Active Mode) .....	73
6.2 休眠模式(Sleep Mode).....	74
6.3 深度休眠模式(Deep Sleep Mode) .....	75

---

---

6.4 Cortex®-M0+内核系统控制寄存器(SCR).....	77
<b>【7】 系统复位与时钟(RCC) .....</b>	<b>78</b>
7.1 复位 .....	78
7.1-1 复位控制器介绍 .....	78
7.1-2 复位源 .....	79
7.2 系统时钟 .....	81
7.2-1 系统时钟模块图 .....	82
7.2-2 内部高速 RC 时钟 (HIRC) .....	83
7.2-3 内部低速 RC 时钟 (SIRC) .....	83
7.2-4 外部高速晶振时钟 (HXT) .....	84
7.2-5 外部低速晶振时钟 (LXT) .....	84
7.2-6 PLL 时钟 .....	85
7.2-7 系统时钟启动过程 .....	86
7.2-8 系统时钟切换 .....	86
7.2-9 系统时钟输出 .....	88
7.2-10 系统时钟安全控制 .....	88
7.2-11 IWDG 时钟 .....	89
7.2-12 RTC 时钟 .....	89
7.2-13 AWK 时钟 .....	89
7.2-14 低功耗模式 .....	89
7.3 寄存器列表 .....	90
7.4 寄存器说明 .....	91
7.4-1 AHB 时钟分频寄存器(RCC_HCLKDIV) .....	91
7.4-2 APB 时钟分频寄存器(RCC_PCLKDIV) .....	92
7.4-3 AHB 周边模块时钟使能寄存器(RCC_HCLKEN) .....	93
7.4-4 APB 周边模块时钟使能寄存器(RCC_PCLKEN) .....	95
7.4-5 时钟输出控制寄存器(RCC_MCOCR) .....	98
7.4-6 系统复位控制寄存器(RCC_RSTCR) .....	99
7.4-7 系统复位状态寄存器(RCC_RSTSR) .....	100
7.4-8 系统时钟源配置寄存器(RCC_SYSCLKCR) .....	102
7.4-9 系统时钟源选择寄存器(RCC_SYSCLKSEL) .....	104
7.4-10 内部高速 RC 振荡器控制寄存器(RCC_HIRCCR) .....	105
7.4-11 外部高速晶体振荡器控制寄存器(RCC_HXTCR) .....	106
7.4-12 内部低速 RC 振荡器控制寄存器(RCC_SIRCCR) .....	107

---

---

7.4-13 外部低速晶体振荡器控制寄存器(RCC_LXTCR) .....	108
7.4-14 Cortex M0+ IRQ 延时控制寄存器(RCC_IRQLATENCY) .....	110
7.4-15 SysTick Timer 控制寄存器(RCC_STICKCR) .....	111
7.4-16 SWDIO 端口控制寄存器(RCC_SWDIOCR) .....	112
7.4-17 周边模块复位控制寄存器(RCC_PERIRST) .....	113
7.4-18 RTC 复位控制寄存器(RCC_RTCRST) .....	116
7.4-19 MASK 中断时间控制寄存器(RCC_MASK_INT_T) .....	117
7.4-20 LPM 控制寄存器(RCC_LPM) .....	118
7.4-21 周边模块复位控制 1 寄存器(RCC_PERIRST1) .....	119
7.4-22 PLL 控制寄存器 1(RCC_PLLCR1) .....	120
7.4-23 PLL 控制寄存器 2(RCC_PLLCR2) .....	121
7.4-24 寄存器写保护控制寄存器(RCC_UNLOCK) .....	122
<b>【8】系统控制(SYSCON) .....</b>	<b>123</b>
8.1 寄存器列表 .....	123
8.2 寄存器说明 .....	124
8.2-1 系统配置寄存器 0(SYSCON_CFGR0) .....	124
8.2-2 端子 Deep Sleep 中断模式控制寄存器(SYSCON_PORTINTCR) .....	125
8.2-3 端子控制寄存器(SYSCON_PORTCR) .....	126
8.2-4 PCA 捕获通道控制寄存器(SYSCON_PCACR) .....	128
8.2-5 TIM1/1A/1B 通道输入源选择(SYSCON_TIM1xCR) .....	129
8.2-6 TIM2/2A/2B/2C 通道输入源选择(SYSCON_TIM2XCR) .....	131
8.2-7 SYSCON 寄存器写保护(SYSCON_UNLOCK) .....	133
8.2-8 SYSCON CIDL 寄存器(SYSCON_CIDL) .....	134
8.2-9 SYSCON CIDL 寄存器(SYSCON_CIDH) .....	134
8.2-10 SYSCON TST0 寄存器(SYSCON_TST_REG0) .....	135
8.2-11 SYSCON TST1 寄存器(SYSCON_TST_REG1) .....	135
8.2-12 SYSCON TST2 寄存器(SYSCON_TST_REG2) .....	136
8.2-13 SYSCON BOOT 状态寄存器(SYSCON_BOOT) .....	137
<b>【9】中断控制器(NVIC) .....</b>	<b>138</b>
9.1 概述 .....	138
9.2 特征 .....	138
9.3 中断优先级 .....	138
9.4 中断向量表 .....	139
9.5 中断唤醒控制 WIC .....	141

---

---

9.5-1 NVIC 从深度休眠模式唤醒设置进入中断 ISR 设置 .....	141
9.5-2 NVIC 从深度休眠模式唤醒设置不执行中断 ISR 设置 .....	142
9.5-3 使用退出休眠特性 .....	142
<b>9.6 软件基本操作 .....</b>	<b>143</b>
9.6-1 外部中断使能 .....	143
9.6-2 NVIC 中断使能和清除使能 .....	143
9.6-3 NVIC 中断挂起和清除挂起 .....	143
9.6-4 NVIC 中断优先级 .....	143
9.6-5 NVIC 中断屏蔽 .....	144
<b>9.7 寄存器列表 .....</b>	<b>145</b>
<b>9.8 寄存器说明 .....</b>	<b>146</b>
9.8-1 中断使能设置寄存器(NVIC_ISER) .....	146
9.8-2 中断使能清除寄存器(NVIC_ICER) .....	147
9.8-3 中断挂起设置寄存器(NVIC_ISPR) .....	148
9.8-4 中断挂起清除寄存器(NVIC_ICPR) .....	149
9.8-5 中断优先级控制寄存器 0(NVIC_IPR0) .....	150
9.8-6 中断优先级控制寄存器 1(NVIC_IPR1) .....	150
9.8-7 中断优先级控制寄存器 2(NVIC_IPR2) .....	151
9.8-8 中断优先级控制寄存器 3(NVIC_IPR3) .....	151
9.8-9 中断优先级控制寄存器 4(NVIC_IPR4) .....	152
9.8-10 中断优先级控制寄存器 5(NVIC_IPR5) .....	152
9.8-11 中断优先级控制寄存器 6(NVIC_IPR6) .....	153
9.8-12 中断优先级控制寄存器 7(NVIC_IPR7) .....	153
<b>【10】通用输入输出口(GPIO) .....</b>	<b>154</b>
10.1 GPIO 简介 .....	154
10.2 GPIO 主要特性 .....	154
10.3 GPIO 功能描述 .....	155
10.3-1 通用 I/O(GPIO) .....	157
10.3-2 I/O 端口控制寄存器 .....	157
10.3-3 I/O 端口控制寄存器 .....	157
10.3-4 I/O 数据位处理 .....	158
10.3-5 输入配置 .....	158
10.3-6 输出配置 .....	159
10.3-7 外部中断/唤醒线 .....	159

---

---

10.3-8 I/O 引脚的复用功能和重映射.....	160
10.3-9 模拟配置.....	162
10.3-10 HXT 或 LXT 引脚用作 GPIO .....	162
10.4 GPIOX 寄存器列表 .....	163
10.5 GPIO 寄存器说明.....	164
10.5-1 GPIO 端口方向寄存器(GPIOx_DIRCR)(x = A,B,C,D).....	164
10.5-2 GPIO 端口输出类型寄存器(GPIOx_OTYPER)(x = A,B,C,D) .....	166
10.5-3 GPIO 端口输出数据寄存器(GPIOx_ODR)(x = A,B,C,D) .....	168
10.5-4 GPIO 端口输入数据寄存器(GPIOx_IDR)(x = A,B,C,D) .....	170
10.5-5 GPIO 端口中断使能寄存器(GPIOx_INTEN)(x = A,B,C,D).....	172
10.5-6 GPIO 端口中断原始状态寄存器(GPIOx_RAWINTSR)(x = A,B,C,D) .....	174
10.5-7 GPIO 端口中断状态寄存器(GPIOx_MSKINTSR)(x = A,B,C,D) .....	176
10.5-8 GPIO 端口中断清除寄存器(GPIOx_INTCLR)(x = A,B,C,D) .....	178
10.5-9 GPIO 端口中断类型寄存器(GPIOx_INTTYPSCR)(x = A,B,C,D).....	180
10.5-10 GPIO 端口中断类型寄存器(GPIOx_INTPOLCR)(x = A,B,C,D) .....	182
10.5-11 GPIO 端口任意边沿触发中断寄存器(GPIOx_INTANY)(x = A,B,C,D).....	184
10.5-12 GPIO 端口输出置位寄存器(GPIOx_ODSET)(x = A,B,C,D) .....	186
10.5-13 GPIO 端口输出清除寄存器(GPIOx_ODCLR)(x = A,B,C,D) .....	188
10.5-14 GPIO 端口输入去抖动寄存器(GPIOx_INDBEN)(x = A,B,C,D) .....	190
10.5-15 GPIO 端口输入去抖动时钟配置寄存器(GPIOx_DBCLKCR)(x = A,B,C,D) .....	191
10.5-16 GPIO 端口上拉/下拉寄存器(GPIOx_INDBEN)(x = A,B,C,D) .....	192
10.5-17 GPIO 端口电压转换速率配置(GPIOx_SLEWCR)(x = A,B,C,D).....	195
10.5-18 GPIO 端口驱动强度配置寄存器(GPIOx_DRVCR)(x = A,B,C,D) .....	197
10.5-19 GPIOA 端口复用功能寄存器(GPIOA_AFR) .....	199
10.5-20 GPIOB 端口复用功能寄存器(GPIOB_AFR).....	200
10.5-21 GPIOC 端口复用功能寄存器(GPIOC_AFR).....	201
10.5-22 GPIOD 端口复用功能寄存器(GPIOD_AFR).....	202
10.5-23 GPIO 端口输入触发器始能(GPIOx_DRVCR)(x = A,B,C,D) .....	203
<b>【11】 Flash 控制器(Flash) .....</b>	<b>205</b>
11.1 Flash 控制器概述 .....	205
11.2 Flash 结构框图 .....	205
11.3 功能描述 .....	206
11.3-1 擦除操作 .....	206
11.3-2 写操作 .....	208

---

---

11.3-3 读操作.....	208
11.3-4 擦写时间.....	209
11.3-5 擦写保护.....	209
11.3-6 系统 BOOT 地址映像 .....	211
11.3-7 系选项字节区(Option Bytes) .....	211
11.3-8 在系统编程(ISP) .....	212
11.4 Flash 寄存器列表 .....	215
11.5 寄存器说明.....	216
11.5-1 Flash_CR(Flash 控制寄存器).....	216
11.5-2 Flash_IFR(Flash 中断标记寄存器).....	217
11.5-3 Flash_ICLR(Flash 中断标记清除寄存器) .....	217
11.5-4 Flash_BYPASS(BYPASS 序列寄存器) .....	218
11.5-5 Flash_SLOCK0(sector 擦写保护寄存器 0).....	219
11.5-6 Flash_SLOCK1(sector 擦写保护寄存器 1).....	220
11.5-7 Flash_ISPCON(Flash ISPCON 配置寄存器) .....	221
11.5-8 Flash 读取周期寄存器(Flash_RCR) .....	222
<b>【12】循环冗余校验计算单元(CRC).....</b>	<b>223</b>
12.1 概述.....	223
12.2 功能描述.....	223
12.2-1 CRC 编码模式.....	224
12.2-2 CRC 检验模式.....	224
12.3 CRC 寄存器列表 .....	225
12.4 寄存器说明.....	226
12.4-1 CRC 结果寄存器(CRC_RESULT) .....	226
12.4-2 CRC 控制寄存器寄存器(CRC_CR).....	227
12.4-3 CRC 数据寄存器(CRC_DATA) .....	228
<b>【13】高级控制定时器(TIM1/1A/1B) .....</b>	<b>229</b>
13.1 Advanced Timer 简介 .....	229
13.2 Advanced Timer 主要特性 .....	229
13.3 Advanced Timer 功能描述 .....	231
13.3-1 时基单元.....	231
13.3-2 计数器模式.....	232
13.3-3 重复计数器 .....	242
13.3-4 时钟选择 .....	243

---

---

13.3-5 捕获/比较通道 .....	247
13.3-6 输入捕获模式.....	249
13.3-7 PWM 输入模式.....	250
13.3-8 强置输出模式.....	251
13.3-9 输出比较模式.....	251
13.3-10 PWM 模式.....	252
13.3-11 互补输出和死区插入 .....	256
13.3-12 使用刹车功能 .....	257
13.3-13 在外部事件时清除 OCxREF 信号 .....	260
13.3-14 产生六步 PWM 输出.....	261
13.3-15 单脉冲模式 .....	262
13.3-16 编码器接口模式 .....	263
13.3-17 定时器输入异或功能 .....	265
13.3-18 与霍尔传感器的接口 .....	266
13.3-19 TIM1X* 定时器和外部触发的同步 .....	268
13.3-20 定时器同步 .....	271
13.3-21 调试模式 .....	271
13.4 高级控制定时器(TIM1/1A/1B)寄存器列表 .....	272
13.5 TIM1X*寄存器说明 .....	273
13.5-1 TIM1X* 控制寄存器 1(TIMX*_CR1) .....	273
13.5-2 TIM1X* 控制寄存器 2(TIM1X*_CR2) .....	275
13.5-3 TIM1X* 从模式控制寄存器(TIM1X*_SMCR).....	277
13.5-4 TIM1X* 中断与 DMA 请求使能寄存器(TIM1X*_DIER).....	280
13.5-5 TIM1X* 状态寄存器(TIM1X*_SR) .....	282
13.5-6 TIM1X* 事件产生寄存器(TIM1X*_EGR) .....	284
13.5-7 TIM1X* 捕获/比较模式寄存器 1 (TIM1X*_CCMR1).....	286
13.5-8 TIM1X* 捕获/比较模式寄存器 2(TIM1X*_CCMR2) .....	291
13.5-9 TIM1X* 捕获/比较使能寄存器(TIM1X*_CCER).....	293
13.5-10 TIM1X* 计数器(TIM1X*_CNT).....	296
13.5-11 TIM1X* 预分频器(TIM1X*_PSC) .....	296
13.5-12 TIM1X* 自动重装载寄存器(TIM1X*_ARR) .....	297
13.5-13 TIM1X* 重复计数寄存器(TIM1X*_RCR) .....	298
13.5-14 TIM1X* 捕获/比较寄存器 1(TIM1X*_CCR1) .....	299
13.5-15 TIM1X* 捕获/比较寄存器 2(TIM1X*_CCR2) .....	300

---

---

13.5-16 TIM1X* 捕获/比较寄存器 3(TIM1X*_CCR3) .....	301
13.5-17 TIM1X* 捕获/比较寄存器 4(TIM1X*_CCR4) .....	302
13.5-18 TIM1X* 刹车和死区寄存器(TIM1X*_BDTR).....	303
13.5-19 TM1X*DMA 控制寄存器(TIM1X*_DCR) .....	306
13.5-20 TM1X*连续模式的 DMA 地址(TIM1X*_DMAR) .....	307
<b>【14】通用定时器 TIM2X*(TIM2/2A/2B/2C) .....</b>	<b>308</b>
14.1 通用定时器简介 .....	308
14.2 通用定时器主要功能 .....	308
14.3 TIM2X* 功能描述 .....	310
14.3-1 时基单元 .....	310
14.3-2 计数器模式 .....	311
14.3-3 时钟选择 .....	319
14.3-4 捕获/比较通道 .....	322
14.3-5 输入捕获模式 .....	324
14.3-6 PWM 输入模式 .....	325
14.3-7 强置输出模式 .....	326
14.3-8 输出比较模式 .....	326
14.3-9 PWM 模式 .....	328
14.3-10 单脉冲模式 .....	331
14.3-11 在外部事件时清除 OCxREF 信号 .....	333
14.3-12 编码器接口模式 .....	334
14.3-13 定时器输入异或功能 .....	336
14.3-14 定时器和外部触发的同步 .....	336
14.3-15 定时器同步 .....	339
14.3-16 调试模式 .....	344
14.4 TIM2X* 寄存器列表 .....	345
14.5 TIM2X* 寄存器说明 .....	346
14.5-1 TIM2X* 控制寄存器 1(TIM2X*_CR1) .....	346
14.5-2 TIM2X* 控制寄存器 2(TIM2X*_CR2) .....	348
14.5-3 TIM2X* 从模式控制寄存器(TIM2X*_SMCR) .....	350
14.5-4 TIM2X* 中断与 DMA 请求使能寄存器(TIM2X*_DIER) .....	353
14.5-5 TIM2X* 状态寄存器(TIM2X*_SR) .....	355
14.5-6 TIM2X* 事件产生寄存器(TIM2X*_EGR) .....	357
14.5-7 TIM2X* 捕获/比较模式寄存器 1(TIM2X*_CCMR1) .....	359

---

---

14.5-8 TIM2X* 捕获/比较模式寄存器 2(TIM2X*_CCMR2) .....	363
14.5-9 TIM2X* 捕获/比较使能寄存器(TIM2X*_CCER) .....	365
14.5-10 TIM2X* 计数器(TIM2X*_CNT) .....	367
14.5-11 TIM2X* 预分频器(TIM2X*_PSC) .....	367
14.5-12 TIM2X* 自动重装载寄存器(TIM2X*_ARR) .....	368
14.5-13 TIM2X* 捕获/比较寄存器 1(TIM2X*_CCR1) .....	369
14.5-14 TIM2X* 捕获/比较寄存器 2(TIM2X*_CCR2) .....	370
14.5-15 TIM2X* 捕获/比较寄存器 3(TIM2X*_CCR3) .....	371
14.5-16 TIM2X* 捕获/比较寄存器 4(TIM2X*_CCR4) .....	372
14.5-17 TM2X*DMA 控制寄存器(TIM2X*_DCR) .....	373
14.5-18 TM2X*连续模式的 DMA 地址(TIM2X*_DMAR) .....	374
<b>【15】可编程计数阵列(PCA) .....</b>	<b>375</b>
15.1 PCA 简介 .....	375
15.2 PCA 功能描述 .....	376
15.2-1 PCA 定时/计数器 .....	376
15.2-2 捕获功能 .....	377
15.2-3 PCA 比较功能 .....	378
15.3 PCA 模块与其他模块互连及控制 .....	383
15.3-1 ECI 互连 .....	383
15.3-2 PCACAP0 .....	383
15.3-3 PCACAP1/2/3/4 .....	383
15.4 PCA 寄存器列表 .....	384
15.5 寄存器说明 .....	385
15.5-1 控制寄存器(PCA_CR) .....	385
15.5-2 模式寄存器(PCA_MOD) .....	387
15.5-3 计数寄存器(PCA_CNT) .....	388
15.5-4 中断清除寄存器(PCA_INTCLR) .....	389
15.5-5 比较捕获模式寄存器(PCA_CCAPM0~4) .....	390
15.5-6 比较捕获数据寄存器低 8 位(PCA_CCAP0~4L) .....	392
15.5-7 比较捕获数据寄存器高 8 位(PCA_CCAP0~4H) .....	393
15.5-8 比较高速输出标志寄存器(PCA_CCAPO) .....	394
15.5-9 端子输出控制寄存器(PCA_POCR) .....	395
15.5-10 比较捕获 16 位寄存器(PCA_CCAP0~4) .....	397
<b>【16】基础定时器 Base Timer(TIM10/TIM11) .....</b>	<b>398</b>

---

---

16.1 Base Timer 简介 .....	398
16.2 Base Timer 功能描述 .....	399
16.2-1 计数功能 .....	401
16.2-2 定时功能 .....	401
16.2-3 Buzzer 功能 .....	402
16.3 Base Timer 互连 .....	402
16.3-1 GATE 互联 .....	402
16.3-2 Toggle 输出互联 .....	402
16.4 Base Timer 寄存器列表 .....	403
16.5 Base Timer 寄存器说明 .....	404
16.5-1 控制寄存器(TIMx_CR) .....	404
16.5-2 立即重载寄存器(TIMx_LOAD) .....	406
16.5-3 计数器寄存器(TIMx_CNT) .....	406
16.5-4 原始中断状态寄存器(TIMx_RAWINTSR) .....	407
16.5-5 中断标志寄存器(TIMx_MSKINTSR) .....	407
16.5-6 中断清除寄存器(TIMx_INTCLR) .....	408
16.5-7 周期重载寄存器(TIMx_BGLOAD) .....	408
<b>【17】低功耗定时器(LPTIM) .....</b>	<b>409</b>
17.1 LPTIM 功能描述 .....	409
17.1-1 计数功能 .....	411
17.1-2 定时功能 .....	411
17.2 LPTIM 互连 .....	412
17.2-1 GATE 互联 .....	412
17.2-2 EXT 互联 .....	412
17.2-3 TOGGLE 输出互联 .....	412
17.3 LPTIM 寄存器列表 .....	413
17.4 LPTIM 寄存器说明 .....	414
17.4-1 LPTIM 计数值只读寄存器(LPTIM_CNTVAL) .....	414
17.4-2 LPTIM 控制寄存器(LPTIM_CR) .....	415
17.4-3 LPTIM 立即重载寄存器(LPTIM_LOAD) .....	417
17.4-4 LPTIM 中断寄存器(LPTIM_INTSR) .....	417
17.4-5 LPTIM 中断寄存器(LPTIM_INTCLR) .....	418
17.4-6 LPTIM 周期重载寄存器(LPTIM_BGLOAD) .....	418
<b>【18】自唤醒定时器(AWK) .....</b>	<b>419</b>

---

---

18.1 寄存器列表.....	420
18.2 寄存器说明.....	421
18.2-1 自唤醒定时器控制寄存器(AWK_CR) .....	421
18.2-2 自唤醒定时器重装载数据寄存器(AWK_RLOAD).....	422
18.2-3 自唤醒定时器状态寄存器(AWK_SR) .....	422
18.2-4 自唤醒中断清除寄存器(AWK_INTCLR) .....	423
<b>【19】蜂鸣器(BEEP).....</b>	<b>424</b>
19.1 简介.....	424
19.2 功能描述.....	424
19.2-1 蜂鸣器操作.....	424
19.2-2 蜂鸣器校准.....	425
19.3 寄存器列表.....	426
19.4 寄存器说明.....	427
19.4-1 蜂鸣器控制/状态寄存器(BEEP_CSR) .....	427
<b>【20】独立看门狗(IWDG) .....</b>	<b>428</b>
20.1 IWDG 概述 .....	428
20.2 IWDG 的功能 .....	428
20.2-1 超时周期.....	429
20.2-2 IWDG 溢出后产生中断 .....	429
20.2-3 IWDG 溢出后产生复位 .....	430
20.3 寄存器列表.....	431
20.4 寄存器说明.....	432
20.4-1 IWDG 控制命令寄存器(IWDG_CMDCR) .....	432
20.4-2 IWDG 配置寄存器(IWDG_CFGR) .....	433
20.4-3 IWDG 计数器重装载寄存器(IWDG_RLOAD) .....	434
20.4-4 IWDG 计数器值寄存器(IWDG_CNTVAL) .....	434
20.4-5 IWDG 中断状态寄存器(IWDG_SR) .....	435
20.4-6 IWDG 中断清除寄存器(IWDG_INTCLR) .....	436
20.4-7 IWDG 保护寄存器(IWDG_UNLOCK) .....	437
20.5 注意.....	437
<b>【21】系统窗口看门狗(WWDG).....</b>	<b>438</b>
21.1 概述.....	438
21.2 特征.....	438
21.3 结构框图.....	438

---

---

21.4 基本配置.....	438
21.5 功能描述.....	439
21.5-1 窗口看门狗定时器的计数.....	439
21.5-2 窗口看门狗定时器比较中断.....	440
21.5-3 窗口看门狗定时器复位系统.....	440
21.5-4 窗口看门狗定时器的窗口设置限制.....	440
21.6 与独立看门狗定时器(IWDG)比较.....	440
21.6-1 复位条件和复位延时.....	440
21.6-2 唤醒功能.....	440
21.7 寄存器列表.....	441
21.8 寄存器说明.....	442
21.8-1 窗口看门狗定时器重载计数寄存器(WWDG_RLOAD).....	442
21.8-2 窗口看门狗定时器控制寄存器(WWDG_CR).....	443
21.8-3 窗口看门狗定时器中断使能寄存器(WWDG_INTEN).....	444
21.8-4 窗口看门狗定时器状态寄存器(WWDG_SR).....	444
21.8-5 窗口看门狗定时器中断清除寄存器(WWDG_INTCLR).....	445
21.8-6 窗口看门狗定时器计数器值寄存器(WWDG_CNTVAL).....	445
<b>【22】通用异步/同步收发器(UART0/1,USART2/3).....</b>	<b>446</b>
22.1 概述.....	446
22.2 结构框图.....	448
22.3 USART 工作模式.....	450
22.3-1 Mode 0(同步模式, 半双工) .....	450
22.3-2 Mode 1(异步模式, 全双工) .....	451
22.3-3 Mode 2(异步模式, 全双工) .....	452
22.3-4 Mode 3(异步模式, 全双工) .....	453
22.3-5 波特率编程.....	453
22.3-6 帧错误检测.....	454
22.3-7 多机通讯.....	454
22.3-8 自动地址识别.....	454
22.3-9 给定地址.....	454
22.3-10 广播地址 .....	455
22.3-11 收发端缓存 .....	455
22.3-12 IrDA 红外功能.....	456
22.4 USART 工作模式.....	459

---

---

22.4-1 USART 特征描述.....	459
22.4-2 发送器.....	460
22.4-3 接收器.....	463
22.4-4 多处理器通信.....	465
22.4-5 校验控制.....	466
22.4-6 LIN (局域互联网) 模式 .....	467
22.4-7 USART 同步模式.....	469
22.4-8 单线半双工通信模式.....	471
22.4-9 红外线数据传输(IrDA).....	472
22.4-10 使用 DMA 通信 .....	474
22.4-11 硬件流控制 .....	474
22.4-12 中断请求 .....	476
22.5 不同波特率的分频设置.....	477
22.6 USART/USART 寄存器列表.....	481
22.7 USART0,1 寄存器说明 .....	482
22.7-1 USART 控制寄存器(USARTx_SCON) .....	482
22.7-2 USART 数据寄存器(USARTx_SBUF) .....	484
22.7-3 USART 地址寄存器(USARTx_SADDR) .....	484
22.7-4 USART 地址掩码寄存器(USARTx_SADEN) .....	485
22.7-5 USART 标志位寄存器(USARTx_INTSR) .....	485
22.7-6 USART 标志位清除寄存器(USARTx_INTCLR) .....	486
22.7-7 USART 波特率控制寄存器(USARTx_BAUDCR) .....	487
22.7-8 IrDA 控制寄存器(USARTx_IRDACR) .....	488
22.8 USART2,3 寄存器说明 .....	489
22.8-1 状态寄存器(USARTx_SR) .....	489
22.8-2 数据寄存器(USARTx_DR) .....	491
22.8-3 波特比率寄存器(USARTx_BRR) .....	492
22.8-4 控制寄存器 1 (USARTx_CR1).....	493
22.8-5 控制寄存器 2(USART_CR2) .....	495
22.8-6 控制寄存器 3(USART_CR3) .....	497
22.8-7 保护时间和预分频寄存器(USART_GTPR) .....	499
22.8-8 低功耗波特比率寄存器(USARTx_LP_CR) .....	500
22.8-9 低功耗波特比率寄存器(USARTx_LP_BRR) .....	501
【23】低功耗通用异步收发器(LPUART) .....	502

---

---

23.1 概述.....	502
23.2 结构框图.....	502
23.3 工作模式.....	503
23.3-1 Mode 0(同步模式, 半双工) .....	503
23.3-2 Mode 1(异步模式, 全双工) .....	504
23.3-3 Mode 2(异步模式, 全双工) .....	505
23.3-4 Mode 3(异步模式, 全双工) .....	506
23.4 波特率编程.....	506
23.4-1 Mode 0.....	506
23.4-2 Mode 1/3.....	507
23.4-3 Mode 2.....	507
23.5 帧错误检测.....	508
23.6 多机通讯.....	508
23.7 自动地址识别.....	508
23.8 给定地址.....	508
23.9 广播地址.....	509
23.10 收发端缓存 .....	509
23.10-1 接收缓存 .....	509
23.10-2 发送缓存 .....	509
23.11 寄存器列表 .....	510
23.12 寄存器说明 .....	511
23.12-1 LPUART 数据寄存器(LPUART_SBUF) .....	511
23.12-2 LPUART 控制寄存器(LPUART_SCON).....	512
23.12-3 LPUART 地址寄存器(LPUART_SADDR) .....	514
23.12-4 LPUART 地址掩码寄存器(LPUART_SADEN) .....	514
23.12-5 LPUART 标志位寄存器(LPUART_INTSR) .....	515
23.12-6 LPUART 标志位清除寄存器(LPUART_INTCLR) .....	516
23.12-7 LPUART 波特率控制寄存器(LPUART_BAUDCR).....	517
<b>【24】 I2C 接口(I2C0/1)</b> .....	<b>518</b>
24.1 I2C 简介 .....	518
24.2 I2C 主要特性 .....	518
24.3 I2C 协议描述 .....	518
24.3-1 I2C 总线上数据传输 .....	519
24.3-2 起始位或重复起始信号 .....	519

---

---

24.3-3 从机地址传输.....	520
24.3-4 数据传输.....	520
24.4 I2C 功能描述 .....	521
24.5 I2C 工作模式 .....	522
24.5-1 仲裁与同步逻辑.....	522
24.5-2 串行时钟发生器.....	523
24.5-3 输入滤波器.....	523
24.5-4 地址比较器.....	523
24.5-5 中断产生器.....	523
24.5-6 I2C 主机发送模式 .....	524
24.5-7 I2C 主机接收模式 .....	527
24.5-8 I2C 从机接收模式 .....	529
24.5-9 I2C 从机发送模式 .....	532
24.5-10 I2C 其他杂项状态 .....	534
24.6 I2C 操作模式 .....	534
24.6-1 初始化程序.....	534
24.6-2 端口配置程序.....	534
24.6-3 启动主机发送功能.....	535
24.6-4 启动主机接收功能.....	536
24.6-5 I2C 中断程序 .....	537
24.6-6 无指定模式状态.....	537
24.6-7 主发送状态.....	538
24.6-8 主接收状态.....	539
24.6-9 从接收状态 .....	540
24.6-10 从发送状态 .....	542
24.7 I2CX* 寄存器列表 .....	543
24.8 I2C 寄存器说明 .....	544
24.8-1 I2C 配置寄存器(I2CX*_CR) .....	544
24.8-2 I2C 数据寄存器(I2CX*_DATA) .....	545
24.8-3 I2C 地址寄存器(I2CX*_ADDR) .....	545
24.8-4 I2C 状态寄存器(I2C_SR) .....	546
24.8-5 I2C 计数器 DMA 使能寄存器(I2CX*_TIMDMA) .....	547
24.8-6 I2C 波特率计数器配置寄存器(I2CX*_BAUDCR) .....	548
【25】串行外设接口(SPI/I2S) .....	549

---

---

25.1 SPI 简介 .....	549
25.2 SPI/I2S 主要特性 .....	549
25.3 功能描述 .....	551
25.3-1 概述 .....	551
25.3-2 配置 SPI 为从模式 .....	555
25.3-3 配置 SPI 为主模式 .....	556
25.3-4 配置 SPI 为单工通信 .....	557
25.3-5 数据发送与接收过程 .....	558
25.3-6 CRC 计算 .....	565
25.3-7 状态标志 .....	566
25.3-8 关闭 SPI .....	567
25.3-9 使用 DMA 的 SPI 通信 .....	568
25.3-10 错误标志 .....	570
25.3-11 SPI 中断 .....	570
25.4 I2S 功能描述 .....	571
25.4-1 I2S 功能描述 .....	571
25.4-2 支持的音频协议 .....	572
25.4-3 时钟发生器 .....	579
25.4-4 I2S 主模式 .....	581
25.4-5 I2S 从模式 .....	583
25.4-6 状态标志位 .....	585
25.4-7 错误标志位 .....	586
25.4-8 I2S 中断 .....	586
25.4-9 DMA 功能 .....	586
25.5 SPI/I2S 寄存器列表 .....	587
25.6 SPI/I2S 寄存器说明 .....	588
25.6-1 SPI 控制寄存器 1(SPI_CR1)(I2S 模式下不使用) .....	588
25.6-2 SPI 控制寄存器 2(SPI_CR2) .....	591
25.6-3 SPI 状态寄存器(SPI_SR) .....	592
25.6-4 SPI 数据寄存器(SPI_DR) .....	594
25.6-5 SPI CRC 多项式寄存器(SPI_CRCPR)(I2S 模式下不使用) .....	595
25.6-6 SPI Rx CRC 寄存器(SPI_RXCRCR)(I2S 模式下不使用) .....	596
25.6-7 SPI Tx CRC 寄存器(SPI_TXCRCR) .....	597
25.6-8 SPI Tx CRC 寄存器(SPI_TXCRCR) .....	598

---

---

25.6-9 SPI_I2S 预分频寄存器(SPI_I2SPR) .....	600
<b>【26】 One-Wire 接口(OWIRE) .....</b>	<b>601</b>
26.1 单总线协议(One-Wire).....	601
26.1-1 特点.....	601
26.1-2 优点.....	601
26.2 单总线通信过程.....	601
26.2-1 初始化.....	601
26.2-2 写时间间隙.....	601
26.2-3 读时间隙.....	602
26.3 配置说明.....	603
26.3-1 初始化配置说明.....	603
26.3-2 读数据配置说明.....	603
26.3-3 写数据配置说明.....	603
26.4 One-Wire 接口寄存器列表 .....	604
26.5 寄存器说明.....	605
26.5-1 1-Wire 模块控制寄存器(OWIRE_CR) .....	605
26.5-2 1-Wire 输入端子滤波控制寄存器(OWIRE_NFCR).....	606
26.5-3 1-Wire RESET 宽度控制寄存器(OWIRE_RSTCNT) .....	607
26.5-4 1-Wire Presence Pulse 宽度计数寄存器(OWIRE_PRESCNT) .....	607
26.5-5 1-Wire Bit rate 设计计数器(OWIRE_BITRATECNT).....	608
26.5-6 1-Wire 主器件读/写 PULL0 驱动时间(OWIRE_DRVCNT) .....	608
26.5-7 1-Wire 主器件读采样时间设定(OWIRE_RDSMPCNT).....	609
26.5-8 1-Wire Recover Time 计数区间值(OWIRE_RECCNT) .....	609
26.5-9 1-Wire 数据寄存器(OWIRE_DATA) .....	610
26.5-10 1-Wire 总线操作命令寄存器(OWIRE_CMD) .....	610
26.5-11 1-wire 中断使能寄存器(OWIRE_INTEN).....	611
26.5-12 1-wire 状态寄存器(OWIRE_SR).....	612
26.5-13 1-wire 状态清除寄存器(OWIRE_INTCLR) .....	613
<b>【27】 时钟校准/监测模块(CLKTRIM) .....</b>	<b>614</b>
27.1 简介.....	614
27.2 主要特性.....	614
27.3 CLKTRIM 功能描述 .....	615
27.3-1 CLKTRIM 校准模式 .....	615
27.3-2 CLKTRIM 监测模式 .....	616

---

---

27.4 CLKTRIM 寄存器列表 .....	617
27.5 CLKTRIM 寄存器说明 .....	618
27.5-1 配置寄存器(CLKTRIM_CR) .....	618
27.5-2 参考计数器初值配置寄存器(CLKTRIM_REFCON) .....	619
27.5-3 参考计数器值寄存器(CLKTRIM_REFCNT) .....	619
27.5-4 校准计数器值寄存器(CLKTRIM_CALCNT) .....	620
27.5-5 中断标志位寄存器(CLKTRIM_IFR) .....	621
27.5-6 中断标志位清除寄存器(CLKTRIM_ICLR) .....	622
27.5-7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON) .....	622
<b>【28】 实时时钟(RTC) .....</b>	<b>623</b>
28.1 简介 .....	623
28.2 主要特性 .....	623
28.3 RTC 功能描述 .....	624
28.3-1 RTC 结构框图 .....	624
28.3-2 RTC 时钟 .....	625
28.3-3 复位过程 .....	625
28.3-4 寄存器的写保护 .....	625
28.3-5 日历初始化及配置 .....	626
28.3-6 读出计数寄存器 .....	627
28.3-7 写入计数寄存器 .....	627
28.3-8 闹钟设定 .....	628
28.3-9 校准 1Hz 输出 .....	628
28.3-10 RTC 时钟校准 .....	628
28.4 RTC 中断 .....	629
28.4-1 RTC 闹钟中断 .....	629
28.4-2 RTC 周期中断 .....	629
28.5 RTC 寄存器列表 .....	630
28.6 RTC 寄存器说明 .....	631
28.6-1 RTC 控制寄存器(RTC_CR) .....	631
28.6-2 RTC 时钟控制寄存器(RTC_CLKCR) .....	633
28.6-3 RTC 时间寄存器(RTC_TIME) .....	634
28.6-4 RTC 日期寄存器(RTC_DATE) .....	635
28.6-5 RTC 时间闹钟寄存器(RTC_ALM1TIME) .....	636
28.6-6 RTC 日期闹钟寄存器(RTC_ALM1DATE) .....	637

---

---

28.6-7 RTC 周期闹钟寄存器(RTC_ALM2PRD).....	638
28.6-8 RTC 时钟调校寄存器(RTC_CLKTRIM).....	639
28.6-9 RTC 初始化和状态寄存器(RTC_ISR).....	640
28.6-10 RTC 状态清除寄存器(RTC_INTCLR).....	641
28.6-11 RTC 写保护寄存器(RTC_WPR) .....	642
<b>【29】 模拟/数字转换器(ADC).....</b>	<b>643</b>
29.1 模块简介 .....	643
29.2 ADC 框图 .....	643
29.2-1 ADC 输入阻抗 .....	644
29.3 转换时序及速度 .....	645
29.4 单次转换模式 .....	645
29.5 连续转换模式 .....	646
29.6 连续转换累加模式 .....	648
29.7 ADC 转换结果比较 .....	650
29.8 ADC 中断 .....	651
29.9 ADC 寄存器列表 .....	651
29.10 寄存器说明 .....	652
29.10-1 ADC 配置寄存器 0(ADC_CR0).....	652
29.10-2 ADC 配置寄存器 1(ADC_CR1).....	655
29.10-3 ADC 配置寄存器 2(ADC_CR2).....	658
29.10-4 ADC 通道 0 转换结果(ADC_RESULT0).....	659
29.10-5 ADC 通道 1 转换结果(ADC_RESULT1).....	659
29.10-6 ADC 通道 2 转换结果(ADC_RESULT2).....	660
29.10-7 ADC 通道 3 转换结果(ADC_RESULT3).....	660
29.10-8 ADC 通道 4 转换结果(ADC_RESULT4).....	661
29.10-9 ADC 通道 5 转换结果(ADC_RESULT5).....	661
29.10-10 ADC 通道 6 转换结果(ADC_RESULT6) .....	662
29.10-11 ADC 通道 7 转换结果(ADC_RESULT7) .....	662
29.10-12 ADC 转换结果(ADC_RESULT) .....	663
29.10-13 ADC 转换结果累加值(ADC_RESULT_ACC).....	663
29.10-14 ADC 比较上阈值(ADC_LT) .....	664
29.10-15 ADC 比较下阈值(ADC_LT) .....	664
29.10-16 ADC 中断使能寄存器(ADC_INTEN) .....	665
29.10-17 ADC 中断清除寄存器(ADC_INTCLR) .....	666

---

---

29.10-18 ADC 掩码前中断状态寄存器(ADC_RAWINTSR) .....	667
29.10-19 ADC 掩码后中断状态寄存器(ADC_MSKINTSR) .....	668
29.10-20 ADC/VTSE TRIM 寄存器 2(ADCVTSETTRIM) .....	669
<b>【30】低电压检测器(LVD).....</b>	<b>670</b>
30.1 LVD 简介 .....	670
30.2 LVD 框图 .....	670
30.3 数字滤波 .....	671
30.4 配置示例 .....	672
30.4-1 LVD 配置为低电压复位 .....	672
30.4-2 LVD 配置为电压变化中断 .....	672
30.5 LVD 寄存器列表 .....	673
30.6 寄存器说明 .....	674
30.6-1 LVD 控制寄存器(LVD_CR) .....	674
30.6-2 LVD 状态寄存器(LVD_SR) .....	676
<b>【31】电压比较器(VC) .....</b>	<b>677</b>
31.1 VC 简介 .....	677
31.2 VC 框图 .....	677
31.3 数字滤波 .....	678
31.4 配置示例 .....	678
31.5 VC 寄存器列表 .....	679
31.6 VC 寄存器说明 .....	680
31.6-1 VC0 电压控制寄存器(VC0_CR0) .....	680
31.6-2 VC0 控制寄存器(VC0_CR1) .....	682
31.6-3 VC0 输出配置寄存器(VC0_OUTCFG) .....	684
31.6-4 VC0 状态寄存器(VC0_SR) .....	686
31.6-5 VC1 电压控制寄存器(VC1_CR0) .....	687
31.6-6 VC1 控制寄存器(VC1_CR1) .....	689
31.6-7 VC1 输出配置寄存器(VC1_OUTCFG) .....	691
31.6-8 VC1 状态寄存器(VC1_SR) .....	693
<b>【32】运算放大器 (OPA) .....</b>	<b>694</b>
32.1 OPA 简介 .....	694
32.2 OPA 特性 .....	694
32.3 OPA 框图 .....	695
32.4 PGA 增益 .....	696

---

---

32.4-1 单位增益 PGA .....	696
32.4-2 正向输入 PGA .....	697
32.4-3 互补输入 PGA .....	697
32.4-4 串级 PGA .....	699
32.5 OPA 寄存器列表 .....	700
32.6 寄存器说明 .....	701
32.6-1 OP1 控制寄存器 0 (OP1_CR0) .....	701
32.6-2 OP2 控制寄存器 0 (OP2_CR0) .....	702
32.6-3 OPDAC 控制寄存器 (OPDAC_CR) .....	703
<b>【33】 PDMA 控制器 .....</b>	<b>704</b>
33.1 PDMA 概述 .....	704
33.2 PDMA 主要特性 .....	704
33.3 功能框图 .....	705
33.4 功能描述 .....	706
33.4-1 通道设置 .....	707
33.4-2 PDMA 仲裁率 .....	708
33.4-3 通道优先级 .....	709
33.4-4 DMA 通道控制数据结构 .....	711
33.4-5 DMA 传输周期的类型 .....	715
33.4-6 错误管理 .....	726
33.5 中断 .....	726
33.6 DMA 寄存器列表 .....	727
33.7 寄存器说明 .....	728
33.7-1 DMA 状态寄存器(DMA_STA) .....	728
33.7-2 DMA 配置寄存器(DMA_CFG) .....	729
33.7-3 DMA 主要控制数据基地址指针寄存器(DMA_PDBPTR) .....	730
33.7-4 DMA 备用控制数据基地址指针寄存器(DMA_ADBPTR) .....	731
33.7-5 DMA 通道等待请求状态寄存器(DMA_CHWTREQ) .....	732
33.7-6 DMA 通道软件请求寄存器(DMA_CHSWREQ) .....	733
33.7-7 DMA 通道使用多笔传输设定寄存器 (DMA_CHTSET) .....	734
33.7-8 DMA 通道使用多笔传输清除寄存器(DMA_CHTCLR) .....	735
33.7-9 DMA 通道请求屏蔽设置寄存器(DMA_CHRMKSET) .....	736
33.7-10 DMA 通道请求屏蔽清除寄存器(DMA_CHRMKCLR) .....	737
33.7-11 DMA 通道使能设置寄存器(DMA_CHENSET) .....	738

---

---

33.7-12 DMA 通道使能清除寄存器 (DMA_CHENCLR) .....	739
33.7-13 DMA 通道备用设置寄存器(DMA_CHALTSET) .....	740
33.7-14 DMA 通道备用清除设置寄存器(DMA_CHALTCLR) .....	741
33.7-15 DMA 通道优先级设置寄存器(DMA_CHPRISET) .....	742
33.7-16 DMA 通道优先级清除寄存器(DMA_CHPRICLR) .....	743
33.7-17 DMA 总线错误清除寄存器寄存器(DMA_CHERRCLR) .....	744
33.7-18 DMA 通道传输完成中断设置寄存器(DMA_CHINTSET) .....	745
33.7-19 DMA 通道中断使能清除寄存器(DMA_CHINTCLR) .....	746
33.7-20 DMA 通道传输完成中断旗标寄存器(DMA_CHTCIF) .....	747
33.7-21 DMA 通道等待请求使能寄存器(DMA_CHWTIONREG) .....	748
33.7-22 DMA 通道选择请求寄存器(DMA_CHSELREQ) .....	749
33.7-23 DMA 停滞状态寄存器(DMA_STALL) .....	750
33.7-24 DMA 请求状态 (DMA_CHREQ) .....	751
33.7-25 DMA 通道单笔请求状态 (DMA_CHSREQ) .....	752
33.7-26 DMA 通道传输完成的状态设置寄存器(DMA_CHDONESET) .....	753
33.7-27 DMA 通道传输完成状态清 0 寄存器(DMA_CHDONECLR) .....	754
33.7-28 DMA 通道正在传输中的状态设置寄存器(DMA_CHACTSET) .....	755
33.7-29 DMA 通道正在传输中的状态清 0 寄存器(DMA_CHACTCLR) .....	756
33.7-30 DMA 总线错误清除寄存器寄存器(DMA_ERRSET) .....	757
<b>【34】 QUAD-SPI 接口(QSPI) .....</b>	<b>758</b>
34.1 概述 .....	758
34.2 QSPI 功能框图 .....	758
34.3 QSPI 主要特性 .....	760
34.4 QSPI 命令序列 .....	760
34.4-1 地址阶段 .....	760
34.4-2 四线交替字节阶段 .....	761
34.4-3 空指令周期阶段 .....	761
34.4-4 数据阶段 .....	761
34.5 QUADSPI 信号接口协议模式 .....	762
34.5-1 单线 SPI 模式(Single SPI Mode) .....	762
34.5-2 双线 SPI 模式(Dual SPI Mode) .....	762
34.5-3 四线 SPI 模式(Quad SPI Mode) .....	762
34.5-4 双闪存模式 .....	763
34.6 QUADSPI 间接模式 .....	763

---

---

34.6-1 触发命令启动.....	764
34.6-2 FIFO 和数据管理.....	765
34.7 QUADSPI 状态轮询模式.....	765
34.8 QUADSPI 内存映像模式.....	766
34.9 QUADSPI Flash 配置.....	766
34.10 QUADSPI 延迟数据采样数据进行采样.....	767
34.11 QUADSPI 配置 .....	767
34.12 QUADSPI 用法 .....	768
34.13 只发送一次指令 .....	770
34.14 QUADSPI 错误管理 .....	770
34.15 QUADSPI 忙位和中止功能 .....	770
34.16 nCTS 行为 .....	771
34.17 QUADSPI 中断 .....	772
34.18 QSPI 寄存器列表 .....	773
34.19 QSPI 寄存器说明 .....	774
34.19-1 QSPI 控制寄存器 1(QUADSPI_CR) .....	774
34.19-2 QSPI 控制寄存器 1(QUADSPI_DCR) .....	778
34.19-3 QSPI 状态寄存器(QUADSPI_SR) .....	779
34.19-4 QSPI 标志清除寄存器(QUADSPI_FCR).....	780
34.19-5 QSPI 数据长度寄存器(QUADSPI_DLR) .....	781
34.19-6 QSPI 通信配置寄存器(QUADSPI_CCR) .....	782
34.19-7 QSPI 地址寄存器(QUADSPI_AR).....	785
34.19-8 QSPI 备用字节寄存器(QUADSPI_ABR) .....	786
34.19-9 QSPI 数据寄存器(QUADSPI_DR) .....	787
34.19-10 QSPI 轮询屏蔽寄存器(QUADSPI_PSMKR).....	788
34.19-11 QSPI 轮询状态匹配寄存器(QUADSPI_PSMAR).....	789
34.19-12 QSPI 轮询间隔寄存器(QUADSPI_PIR) .....	790
34.19-13 QSPI 超时寄存器(QUADSPI_LPTR) .....	791
【35】 真随机数发生器 (TRNG) .....	792
35.1 概述.....	792
35.2 功能框图.....	792
35.3 功能描述.....	792
35.4 TRNG 寄存器列表 .....	793
35.5 寄存器说明.....	794

---

---

35.5-1 TRNG 控制寄存器(TRNG_CR).....	794
35.5-2 TRNG 状态寄存器(TRNG_SR) .....	795
35.5-3 TRNG 数据寄存器(TRNG_DR) .....	796
<b>【36】 高级加密标准模块 (AES) .....</b>	<b>797</b>
36.1 概述.....	797
36.2 特性.....	797
36.3 AES 中断 .....	797
36.4 AES 功能描述 .....	798
36.4-1 加密和派生密钥(Encryption and derivation keys).....	799
36.4-2 加密和 AES 链接算法(AES chaining algorithms).....	799
36.4-3 数据型态(Data Type) .....	803
36.5 AES 寄存器列表 .....	805
36.6 寄存器说明.....	806
36.6-1 AES 控制寄存器(AES_CR) .....	806
36.6-2 AES 状态寄存器(AES_SR) .....	808
36.6-3 AES 输入数据寄存器(AES_DINR).....	809
36.6-4 AES 输出数据寄存器(AES_OUTR).....	810
36.6-5 AES KEY0 寄存器(AES_KEYR0) (LSB: key[31:0]) .....	811
36.6-6 AES KEY1 寄存器(AES_KEYR1) (LSB: key[63:32]) .....	812
36.6-7 AES KEY2 寄存器(AES_KEYR2) (MSB: key[95:64]) .....	813
36.6-8 AES KEY3 寄存器(AES_KEYR1) (MSB: key[127:96]).....	813
36.6-9 AES IVR0 寄存器(AES_IVR0)(IVR[31:0]) .....	814
36.6-10 AES IVR1 寄存器(AES_IVR1)(IVR[63:32]) .....	814
36.6-11 AES IVR2 寄存器(AES_IVR2)(IVR[95:64]) .....	815
36.6-12 AES IVR3 寄存器(AES_IVR3)(IVR[127:96]) .....	815
<b>【37】 选项字节区(Option Bytes).....</b>	<b>816</b>
37.1 用户配置寄存器 0(USERCFG0).....	816
37.2 用户配置寄存器 1(USERCFG1).....	817
37.3 用户配置寄存器 2(USERCFG2).....	818
37.4 注意.....	818
<b>【38】 Debug 支持(DBG).....</b>	<b>819</b>
38.1 SWD 调试接口说明 .....	819
38.1-1 SWD 调试接口的引脚分配 .....	819
38.1-2 SWD 引脚的内部上拉和下拉 .....	819

---

---

38.2 SWD 保护位工作原理 .....	820
38.3 在低功耗模式下使用 SWD .....	820
38.4 DBG 寄存器列表 .....	821
38.5 DBG 寄存器说明 .....	821
38.5-1 Debug 模式控制寄存器(DBG_APBFZ).....	821
<b>【39】附录 1: SysTick 定时器(SYST).....</b>	<b>823</b>
39.1 SysTick 定时器简介 .....	823
39.2 设置 SysTick 定时器.....	823
39.3 SysTick 定时器寄存器列表 .....	824
39.4 SysTick 定时器寄存器说明 .....	824
39.4-1 SysTick 定时器控制和状态寄存器(SYST_CSR) .....	824
39.4-2 SysTick 定时器重载值寄存器(SYST_RVR).....	825
39.4-3 SysTick 定时器当前值寄存器(SYST_CVR) .....	825
39.4-4 SysTick 定时器校准值寄存器(SYST_CALIB) .....	825
<b>【40】版本修订纪录.....</b>	<b>826</b>

---

## 图索引

Figure 3 - 1 内部功能框图结构 .....	44
Figure 4 - 1 ZB32L032 LQFP48 .....	56
Figure 4 - 2 ZB32L032 LQFP32 .....	57
Figure 4 - 3 ZB32L032 TSSOP28 .....	57
Figure 5 - 1 系统的模块框图 .....	67
Figure 5 - 2 储存器映像图 .....	68
Figure 5 - 3 BOOT PIN 地址映射图 .....	71
Figure 6 - 1 控制模式框图 .....	72
Figure 7 - 1 复位来源示意图 .....	78
Figure 7 - 2 系统时钟控制模块图 .....	82
Figure 7 - 3 HXT/LXT 时钟源 .....	84
Figure 7 - 4 PLL 时钟源 .....	85
Figure 7 - 5 内部高速时钟启动示意图 .....	86
Figure 7 - 6 时钟切换示意图 .....	87
Figure 10 - 1 标准 I/O 端口的位基本结构 .....	155
Figure 10 - 2 浮空输入/上拉/下拉配置 .....	158
Figure 10 - 3 输出配置 .....	159
Figure 10 - 4 复用功能配置 .....	161
Figure 10 - 5 高阻抗模拟配置 .....	162
Figure 11- 1 Flash 结构框图 .....	205
Figure 11- 2 擦除操作步骤 .....	206
Figure 11- 3 Chip 擦除操作步骤 .....	207
Figure 11- 4 写操作步骤 .....	208
Figure 11- 5 写寄存器 BYPASS 序列 .....	210
Figure 11- 6 APP 程序区 boot 地址映像图 .....	211
Figure 11- 7 内部配置与软件进入 ISP 流程图 .....	212
Figure 11- 8 外部 BOOT0,BOOT1 PIN 选择流程图 .....	213
Figure 11- 9 BOOT0,BOOT1 PIN 时序图 .....	214

---

Figure 12- 1CRC 应用示意图 .....	223
Figure 13- 1 高级控制定时器框图 .....	230
Figure 13- 2 当预分频器的参数从 1 变到 2 时, 计数器的时序图 .....	231
Figure 13- 3 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	232
Figure 13- 4 计数器时序图: 内部时钟分频因子为 1 .....	233
Figure 13- 5 计数器时序图: 内部时钟分频因子为 2 .....	233
Figure 13- 6 计数器时序图: 内部时钟分频因子为 4 .....	233
Figure 13- 7 计数器时序图: 内部时钟分频因子为 N .....	234
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).....	234
Figure 13- 9 计数器时序图: 当 ARPE=1 时的更新事件(预装入了 TIM1X*_ARR).....	235
Figure 13- 10 计数器时序图: 内部时钟分频因子为 1 .....	236
Figure 13- 11 计数器时序图: 内部时钟分频因子为 2 .....	237
Figure 13- 12 计数器时序图: 内部时钟分频因子为 4 .....	237
Figure 13- 13 计数器时序图: 内部时钟分频因子为 N .....	237
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 .....	238
Figure 13- 15 计数器时序图: 内部时钟分频因子为 1(TIM1X*_ARR=0x6) .....	239
Figure 13- 16 计数器时序图: 内部时钟分频因子为 2 .....	240
Figure 13- 17 计数器时序图: 内部时钟分频因子为 4(TIM1X*_ARR=0x36) .....	240
Figure 13- 18 计数器时序图: 内部时钟分频因子为 N .....	240
Figure 13- 19 计数器时序图: ARPE=1 时的更新事件(计数器下溢).....	241
Figure 13- 20 计数器时序图: ARPE=1 时的更新事件(计数器溢出) .....	241
Figure 13- 21 不同模式下更新速率的例子, 及 TIM1X*_RCR 的寄存器设置 .....	242
Figure 13- 22 一般模式下的控制电路, 内部时钟分频因子为 1 .....	243
Figure 13- 23 TI2 外部时钟连接例子 .....	244
Figure 13- 24 外部时钟模式 1 下的控制电路.....	245
Figure 13- 25 外部触发输入框图 .....	245
Figure 13- 26 外部时钟模式 2 下的控制电路 .....	246
Figure 13- 27 捕获/比较通道(如: 通道 1 输入部分).....	247
Figure 13- 28 捕获/比较通道 1 的主电路 .....	247
Figure 13- 29 捕获/比较通道的输出部分(通道 1 至 3) .....	248
Figure 13- 30 捕获/比较通道的输出部分(通道 4).....	248
Figure 13- 31 PWM 输入模式时序.....	250
Figure 13- 32 输出比较模式, 翻转 OC1.....	252

---

Figure 13- 33 边沿对齐的 PWM 波形(ARR=8) .....	253
Figure 13- 34 中央对齐的 PWM 波形(APR=8).....	254
Figure 13- 35 带死区插入的互补输出 .....	256
Figure 13- 36 死区波形延迟大于负脉冲.....	256
Figure 13- 37 死区波形延迟大于正脉冲.....	257
Figure 13- 38 响应刹车的输出 .....	259
Figure 13- 39 清除 TIM1X* 的 OCxREF .....	260
Figure 13- 40 产生六步 PWM, 使用 COM 的例子(OSSR=1) .....	261
Figure 13- 41 单脉冲模式的例子 .....	262
Figure 13- 42 编码器模式下的计数器操作实例 .....	264
Figure 13- 43 IC1FP1 反相的编码器接口模式实例.....	265
Figure 13- 44 霍尔传感器接口的实例 .....	267
Figure 13- 45 复位模式下的控制电路 .....	268
Figure 13- 46 门控模式下的控制电路 .....	269
Figure 13- 47 触发器模式下的控制电路 .....	270
Figure 13- 48 外部时钟模式 2 + 触发模式下的控制电路 .....	271
 Figure 14- 1 通用定时器框图 .....	309
Figure 14- 2 当预分频器的参数从 1 变到 2 时, 计数器的时序图 .....	310
Figure 14- 3 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	311
Figure 14- 4 计数器时序图: 内部时钟分频因子为 1 .....	312
Figure 14- 5 计数器时序图: 内部时钟分频因子为 2 .....	312
Figure 14- 6 计数器时序图: 内部时钟分频因子为 4 .....	312
Figure 14- 7 计数器时序图: 内部时钟分频因子为 N .....	313
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).....	313
Figure 14- 9 计数器时序图: 当 ARPE=1 时的更新事件(预装入了 TIM2X*_ARR).....	313
Figure 14- 10 计数器时序图: 内部时钟分频因子为 1 .....	314
Figure 14- 11 计数器时序图: 内部时钟分频因子为 2 .....	314
Figure 14- 12 计数器时序图: 内部时钟分频因子为 4 .....	315
Figure 14- 13 计数器时序图: 内部时钟分频因子为 N .....	315
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.....	315
Figure 14- 15 计数器时序图: 内部时钟分频因子为 1, TIM2X*_ARR=0x6 .....	316
Figure 14- 16 计数器时序图: 内部时钟分频因子为 2 .....	317
Figure 14- 17 计数器时序图: 内部时钟分频因子为 4, TIM2X*_ARR=0x36.....	317

---

Figure 14- 18 计数器时序图：内部时钟分频因子为 N .....	317
Figure 14- 19 计数器时序图：ARPE=1 时的更新事件(计数器下溢).....	318
Figure 14- 20 计数器时序图：ARPE=1 时的更新事件(计数器溢出).....	318
Figure 14- 21 一般模式下的控制电路，内部时钟分频因子为 1 .....	319
Figure 14- 22 TI2 外部时钟连接例子 .....	320
Figure 14- 23 外部时钟模式 1 下的控制电路.....	320
Figure 14- 24 外部触发输入框图 .....	321
Figure 14- 25 外部时钟模式 2 下的控制时序 .....	321
Figure 14- 26 捕获/比较通道(如：通道 1 输入部分).....	322
Figure 14- 27 捕获/比较通道 1 的主电路 .....	322
Figure 14- 28 捕获/比较通道的输出部分(通道 1).....	323
Figure 14- 29 PWM 输入模式时序 .....	325
Figure 14- 30 输出比较模式，翻转 OC1 .....	327
Figure 14- 31 边沿对齐的 PWM 波形(ARR=8) .....	329
Figure 14- 32 中央对齐的 PWM 波形(APR=8).....	330
Figure 14- 33 单脉冲模式的例子 .....	331
Figure 14- 34 清除 TIM2X* 的 OCxREF .....	333
Figure 14- 35 编码器模式下的计数器操作实例 .....	335
Figure 14- 36 IC1FP1 反相的编码器接口模式实例.....	335
Figure 14- 37 复位模式下的控制电路 .....	336
Figure 14- 38 门控模式下的控制电路.....	337
Figure 14- 39 触发器模式下的控制电路.....	338
Figure 14- 40 外部时钟模式 2 + 触发模式下的控制电路 .....	339
Figure 14- 41 主/从定时器的例子 .....	339
Figure 14- 42 的 OC1REF 控制定时器 2,2A,2B,2C .....	340
Figure 14- 43 通过使能定时器 1 可以控制定时器 2.....	341
Figure 14- 44 使用定时器 1 的更新触发定时器 2 .....	342
Figure 14- 45 利用定时器 1 的使能触发定时器 2 .....	342
Figure 14- 46 利用定时器 1 的使能触发定时器 2.....	343
Figure 15- 1 PCA 整体框图 .....	375
Figure 15- 2 PCA 计数器框图 .....	377
Figure 15- 3 PCA 捕获功能框图 .....	378
Figure 15- 4 PCA 比较功能框图.....	380

---

---

Figure 15- 5 PCA PWM 功能框图 .....	381
Figure 15- 6 PCA PWM 输出波形 .....	382
Figure 16- 1 Base Timer 框图 .....	398
Figure 16- 2 Timer 模式 1 框图 .....	399
Figure 16- 3 Timer 模式 2 框图 .....	400
Figure 16- 4 32 位模式 1 时序图(max=0xFFFF FFFF) .....	401
Figure 16- 5 32 位模式 2 时序图(PCLK 二分频, max=0xFFFF FFFF) .....	401
Figure 17- 1 LPTIMER 结构框图 .....	409
Figure 17- 2 LPTIMER 模式 1 .....	410
Figure 17- 3 LPTIMER 模式 2 .....	411
Figure 18- 1 自唤醒定时器结构图 .....	419
Figure 19- 1 蜂鸣器功能图 .....	424
Figure 20- 1 IWDG 整体框图 .....	428
Figure 21- 1 WWDG 结构框图 .....	438
Figure 22- 1 UART 结构图 .....	448
Figure 22- 2 USART 结构图 .....	448
Figure 22- 3 Mode0 发送数据 .....	450
Figure 22- 4 Mode0 接收数据 .....	450
Figure 22- 5 Mode1 发送数据 .....	451
Figure 22- 6 Mode1 接收数据 .....	451
Figure 22- 7 Mode2 发送数据 .....	452
Figure 22- 8 Mode2 接收数据 .....	452
Figure 22- 9 接收缓存 .....	455
Figure 22- 10 IrDA 结构框图 .....	457
Figure 22- 11 IrDA 收发脉冲 .....	458
Figure 22- 12 USART 工作模式选择表 .....	459
Figure 22- 13 USART 工作模式控制表 .....	459
Figure 22- 14 字长设置 .....	460

---

---

Figure 22- 15 配置停止位 .....	461
Figure 22- 16 配置停止位 .....	462
Figure 22- 17 利用空闲总线检测的静默模式 .....	465
Figure 22- 18 利用地址标记检测的静默模式 .....	466
Figure 22- 19 LIN 模式下的断开检测(11 位断开长度 – 设置了 LBDL 位) .....	468
Figure 22- 20 LIN 模式下的断开检测与帧错误的检测 .....	468
Figure 22- 21 USART 同步传输的例子 .....	469
Figure 22- 22 USART 数据时钟时序示例(M=0) .....	470
Figure 22- 23 USART 数据时钟时序示例(M=1) .....	470
Figure 22- 24 RX 数据采样/保持时间 .....	471
Figure 22- 25 IrDA 框图 .....	473
Figure 22- 26 IrDA 数据调制 .....	473
Figure 22- 27 两个 USART 间的硬件流控制 .....	474
Figure 22- 28 RTS 流控制 .....	475
Figure 22- 29 CTS 流控制 .....	475
Figure 22- 30 USART 中断映像图 .....	476
Figure 23- 1 LPUART 结构框图 .....	502
Figure 23- 2 Mode 0 发送数据 .....	503
Figure 23- 3 Mode 0 接收数据 .....	504
Figure 23- 4 Mode 1 发送数据 .....	504
Figure 23- 5 Mode 1 接收数据 .....	505
Figure 23- 6 Mode 2 发送数据 .....	505
Figure 23- 7 Mode 2 接收数据 .....	506
Figure 24- 1 I2C 传输协议 .....	518
Figure 24- 2 主机向从机传输数据 .....	519
Figure 24- 3 主机向从机传输数据 .....	519
Figure 24- 4 START 和 STOP 条件 .....	519
Figure 24- 5 I2C 总线上位传输 .....	520
Figure 24- 6 I2C 总线上应答信号 .....	520
Figure 24- 7 I2C 功能模块图 .....	521
Figure 24- 8 总线上的仲裁 .....	522
Figure 24- 9 I2C 主机发送状态图 .....	526
Figure 24- 10 I2C 主机接收状态图 .....	528

---

---

Figure 24- 11 I2C 从机接收状态图.....	531
Figure 24- 12 I2C 从机发送状态图.....	533
Figure 24- 13 I2C0 Master DMA TX .....	535
Figure 24- 14 I2C0 Master DMA RX.....	536
Figure 25- 1 SPI 的方框图 .....	551
Figure 25- 2 单主和单从应用 .....	552
Figure 25- 3 硬件/软件的从选择管理.....	552
Figure 25- 4 数据时钟时序图 .....	554
Figure 25- 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.....	560
Figure 25- 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.....	561
Figure 25- 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 .....	562
Figure 25- 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.....	562
Figure 25- 9 只接收模式(BIDIMODE=0 并且 RXONLY=1)下连续传输时， RXNE 变化示意图 .....	563
Figure 25- 10 非连续传输发送(BIDIMODE=0 并且 RXONLY=0)时， TXE/BSY 变化示意图 .....	564
Figure 25- 11 使用 DMA 发送.....	568
Figure 25- 12 使用 DMA 接收.....	569
Figure 25- 13 I2S 框图 .....	571
Figure 25- 14 I2S 飞利浦协议波形(16/32 位全精度， CPOL = 0) .....	573
Figure 25- 15 I2S 飞利浦协议标准波形(24 位帧， CPOL = 0).....	573
Figure 25- 16 发送 0x8EAA33 .....	573
Figure 25- 17 接收 0x8EAA33 .....	574
Figure 25- 18 I2S 飞利浦协议标准波形(16 位扩展至 32 位包帧， CPOL = 0) .....	574
Figure 25- 19 MSB 对齐 16 位或 32 位全精度， CPOL = 0 .....	575
Figure 25- 20 MSB 对齐 24 位数据， CPOL = 0.....	575
Figure 25- 21 MSB 对齐 16 位数据扩展到 32 位包帧， CPOL = 0.....	575
Figure 25- 22 LSB 对齐 16 位或 32 位全精度， CPOL = 0 .....	576
Figure 25- 23 LSB 对齐 24 位数据， CPOL = 0.....	576

---

---

Figure 25- 24 要求发送 0x3478AE 的操作 .....	576
Figure 25- 25 要求接收 0x3478AE 的操作 .....	577
Figure 25- 26 LSB 对齐 16 位数据扩展到 32 位包帧, CPOL =0 .....	577
Figure 25- 27 PCM 标准波形(16 位).....	578
Figure 25- 28 PCM 标准波形(16 位扩展到 32 位包帧).....	578
Figure 25- 29 音频采样频率定义 .....	579
Figure 25- 30 I2S 时钟发生器结构 .....	579
Figure 26- 1 初始化过程中的复位与应答脉冲 .....	601
Figure 26- 2 单总线通信协议中写时间隙时序图.....	602
Figure 26- 3 单总线通信协议中读时间隙时序图.....	602
Figure 28- 1 RTC 框图.....	624
Figure 29- 1 ADC 架构方块图.....	643
Figure 29- 2 ADC 转换时序图 .....	645
Figure 29- 3 ADC 连续转换过程示例 .....	646
Figure 29- 4 连续转换累加过程示例 .....	648
Figure 30 - 1 LVD 结构框图 .....	670
Figure 30 - 2 LVD 滤波输出 .....	671
Figure 31 - 1 VC0,VC1 结构框图 .....	677
Figure 31 - 2 VC 滤波输出 .....	678
Figure 32 - 1 OPA1,OPA2 结构框图 .....	695
Figure 33- 1 PDMA 框图 .....	705
Figure 33- 2 通道请求框图 .....	707
Figure 33- 3 Polling flowchart.....	710
Figure 33- 4 基本模式图 .....	716
Figure 33- 5 自动请求模式范例图.....	717
Figure 33- 6 Ping-pong 模式范例图 .....	719
Figure 33- 7 存储器分散/聚集式 图例.....	722
Figure 33- 8 复制 Task A,B,C 描述元的配置数据图 .....	723

---

Figure 33- 9 外设分散/聚集式范例图.....	725
Figure 33- 10 CHWTIONREQ[C*]( dma_waitonreg[C*] )=1 的时序图 .....	748
Figure 33- 11 CHx_CHWTIONREQ=0 的时序图.....	748
Figure 34- 1 QSPI 功能框图(BANK1) .....	758
Figure 34-2 QSPI 功能框图(BANK1,BANK2) .....	759
Figure 34-3 四线模式下的读命令时序.....	760
Figure 34- 4 nCS when CKMODE=0 (T = CLK period) .....	771
Figure 34- 5 nCS when CKMODE=1 in SDR mode (T=CLK period).....	771
Figure 34- 6 nCS when CKMODE=1 with an abort (T=CLK period) .....	771
Figure 35- 1 真随机数框图 .....	792
Figure 36- 1 ECB 加密模式.....	800
Figure 36- 2 ECB 解密模式.....	800
Figure 36- 3 CBC 加密模式.....	801
Figure 36- 4 CBC 解密模式 .....	801
Figure 36- 5 CTR 加密模式.....	802
Figure 36- 6 CTR 解密模式.....	802
Figure 36- 7 32 位计数器+随机数组织 .....	803
Figure 36- 8 Data Type= 2' b00.....	803
Figure 36- 9 Data Type= 2' b01 .....	804
Figure 36- 10 Data Type= 2' b10 .....	804
Figure 36- 11 Data Type= 2' b11 .....	804

## 表索引

Table 3 - 1 ZB32L032 芯片特性与周边配备 .....	43
Table 3 - 2 中断源 .....	47
Table 3 - 3 定时器菜单 .....	49
Table 4 - 1 引脚功能说明表 .....	58
Table 4 - 2 选择芯片引脚表 .....	61
Table 4 - 3 模块与引脚信号说明表 .....	65
Table 4 - 4 串行接口说明表 .....	66
Table 5 - 1 地址对应表 .....	70
Table 6 - 1 MCU 唤醒菜单 .....	72
Table 6 - 2 运行模式下的菜单 .....	73
Table 6 - 3 休眠模式下的模块工作表 .....	74
Table 6 - 4 深度休眠模式下的模块工作表 .....	76
Table 7 - 1 模块时钟对应表 .....	83
Table 7 - 2 PLL 参数表 .....	85
Table 7 - 3 RCC 寄存器列表和复位值 .....	90
Table 8- 1 SYSCON 寄存器列表和复位值 .....	123
Table 9 - 1 中断向量表 .....	139
Table 9 - 2 NVIC 寄存器列表和复位值 .....	145
Table 10 - 1 端口位配置表 .....	156
Table 10 - 2 GPIO 口和外设引脚的复用功能映射,请参考 4.3 .....	161
Table 10 - 3 GPIOx 寄存器列表和复位值 .....	163
Table 11- 1 Flash 寄存器列表和复位值 .....	215
Table 12- 1 CRC 寄存器列表和复位值 .....	225
Table 13- 1 计数方向与编码器信号的关系 .....	264
Table 13- 2 TIM1,TIM1A,TIM1B 寄存器列表和复位值 .....	272

---

Table 13- 3 TIM1X* 内部触发连接 .....	279
Table 13- 4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位.....	295
Table 14- 1 计数方向与编码器信号的关系 .....	334
Table 14- 2 TIM2X* 寄存器列表和复位值.....	345
Table 14- 3 TIM2X* 内部触发连接.....	352
Table 15- 1 PCA 比较/捕获功能模块设置 .....	382
Table 15- 2 PCA 寄存器列表和复位值 .....	384
Table 16- 1 Base Timer 寄存器列表和复位值.....	403
Table 17- 1 LPTIM 寄存器列表和复位值 .....	413
Table 18- 1 AWK 寄存器列表和复位值.....	420
Table 19- 1 BEEP 寄存器列表和复位值 .....	426
Table 20- 1 IWDG 寄存器列表和复位值 .....	431
Table 21- 1 WWDG 寄存器列表和复位值.....	441
Table 22- 1 USART 帧格式 .....	466
Table 22- 2 USART 中断请求 .....	476
Table 22- 3 UART 寄存器列表 .....	481
Table 22- 4 USART 寄存器列表 .....	481
Table 23- 1 LPUART 寄存器列表 .....	510
Table 24- 1 I2CX* 寄存器列表 .....	543
Table 25- 1 SPI 中断请求 .....	570
Table 25- 2 使用标准的 8MHz HSE 时钟得到精确的音频频率.....	580
Table 25- 3 I2S 中断请求.....	586
Table 25- 4 SPI/I2S 寄存器列表和复位值 .....	587
Table 26- 1 One-Wire 接口寄存器列表和复位值 .....	604
Table 27- 1 CLKTRIM 寄存器列表和复位值.....	617

---

Table 28- 1 RTC 寄存器列表和复位值 .....	630
Table 29- 1 ADC 寄存器列表和复位值.....	651
Table 30- 1 LVD 寄存器列表.....	673
Table 31- 1 VC 寄存器列表和复位值 .....	679
Table 32- 1 OPA 寄存器列表和复位值 .....	700
Table 33- 1 DMA 使用的通道与请求列表 .....	707
Table 33- 2 AHB bus transfer arbitration interval .....	708
Table 33- 3 通道控制数据结构表.....	711
Table 33- 4 使用 16 通道的储存器映像含盖主要结构与备用结构.....	712
Table 33- 5 控制数据配置表 .....	713
Table 33- 6 存储器分散/聚集式-信道配置表.....	721
Table 33- 7 外设分散/聚集式 -信道配置表 .....	724
Table 33- 8 DMA 寄存器列表和复位值.....	727
Table 34- 1 QSPI 中断表 .....	772
Table 34- 2 QSPI 寄存器列表和复位值 .....	773
Table 35- 1 寄存器列表和复位值.....	793
Table 36- 1 AES 中断表 .....	797
Table 38- 1 Debug 寄存器映象和复位值 .....	821
Table 40- 1 SysTick 定时器寄存器列表和复位值 .....	824

## Note 索引

Note_25. 1 : 目前 SPI 从模式的非连续传送仅支持硬件 DMA 更新发送缓冲器, 不支持软件更新发送缓冲器 .....	550
Note_25. 2 : 目前 SPI 从模式非连续接收仅支持字节与字节的空隙需小于 4*SPI clock 的时间 .....	550
Note_25. 3 : 目 RXNE 在发送时, RXNE flag 是无用的, 请忽略此标志位 .....	564
Note_25. 4 : 使用 SPI_EN 开关方式重置 CRC 计算资料 .....	566
Note_22.1 1 : 禁用 CPOL=1 and CPHA=0 .....	496

## 【1】简介

**ZB32L032** 是一款内嵌 32 位 ARM® Cortex®-M0+ 内核的超低功耗和宽电压工作范围 (2.5V~5.5V) 的微控制器，最高可运行在 64MHz，内置 32K/64K 字节的嵌入式 Flash，16K 字节的SRAM，集成了12位1Msps高精度SAR型ADC(16 channel)、DAC(6-bits)、OPAx2、RTC、比较器x2、UARTx2、USARTx2、LPUART、SPI/I2S、QSPI、I2Cx2 和 PWMx (多路独立输出或互补式输出)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。

**ZB32L032** 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，广泛适用于下列应用：  
小家电、充电器、遥控器、电子烟、燃气报警器、数显表、温控器、记录仪、电机驱动、智能门锁、智能传感器、智能家居等。

## 【2】产品特性

- **ARM® Cortex®-M0+ 内核**
  - 32K/64K字节嵌入式Flash, 具有擦写保护功能
  - 16K字节SRAM
  - 最高运行到64MHz
  - 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
  - 4.3us 唤醒时间@Fcpu=32Mhz
- **工作条件**
  - 宽电压工作范围2.5V至5.5V
  - 宽工作频率12M高至64MHz
  - 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
  - 两种低功耗工作模式: Sleep、Deep Sleep 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: 所有GPIO PIN 与 RTC(低速时钟工作)
- **中断**
  - 嵌套向量中断控制器(NVIC)用于控制32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48-Pin封装下39个I/O
- **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
- **蜂鸣器频率发生器**
  - 可产生5个1KHz, 2KHz, 4KHz的蜂鸣信号
- **通信接口**
  - UART0,1; USART3,4标准通讯接口
  - 支持低速时钟的超低功耗LPUART
  - QSPI,SPI/I2S 标准通讯接口, 最高达20Mbps.(有复用的 I2S 接口)
  - I2CX2标准通讯接口, 主模式最高支持1Mbps, 从模式最高支持800Kbps
  - One-Wire通讯接口
- **ADC**
  - 12位1Msps采样速率, 12位SAR型ADC
  - 24通道: 16路外部引脚, 1路内部温度传感器电压, 2路OPA输出, 1路1/3\*VDD, 1路BGR 1.2V, 1路DAC
  - 外部参考电压 : VDD,GPIO(PB01) PIN
  - 内置参考电压 : 2.5V, 2.048V
- **PDMA 控制器(Primcell μDMA PL230)**
  - 支持 8 chs : SPI,I2C,USART,Timer,ADC
- **PWM**
  - 支持最多3个互补式输出
- **DAC(6-bits)x2**
- **运算放大器(OPA)x2**
- **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,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: TSSOP28, QFN32, LQFP32/48**

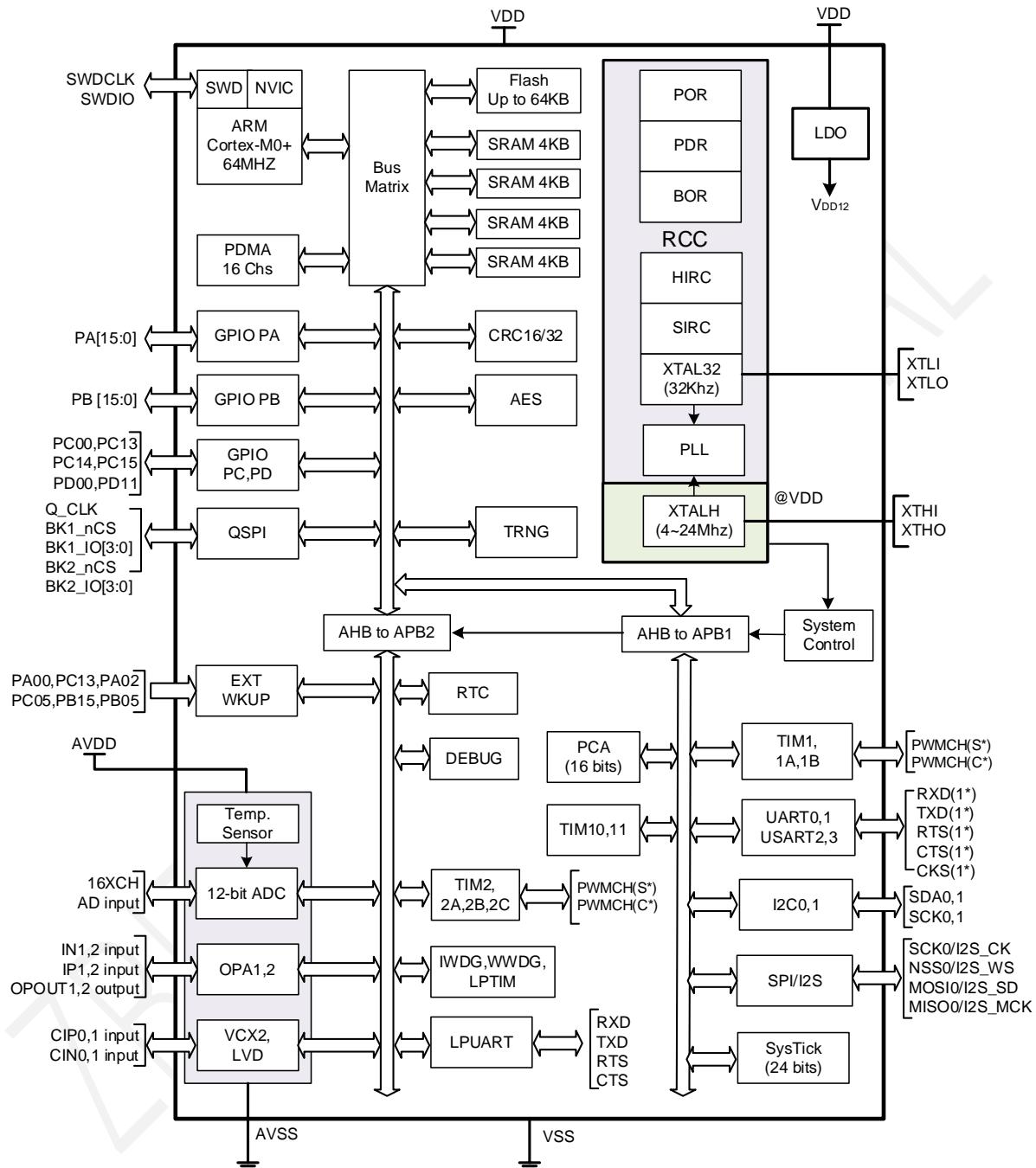
## 【3】产品功能描述

### 3.1 设备概述

	ZB32L032XXX	ZB32L032XXX	ZB32L032XXX
引脚数	48	32	28
GPIO	39	25	22
MCU	内核 CPU 频率	Cortex M0+ Up to 64 MHz	
Flash		64K Bytes (50 MHz Access)	
SRAM		16K Bytes	
PDMA		8 ch (SPI, I2C, USART, Timer, ADC)	
Timer	基础(16-bits)	2x16 bits / 1x32 bits	
	通用(16-bits)	4(T2, T2A, T2B, T2C)	
	高级(16 bits)	3(T1, T1A, T1B)	
	编程	2	
	SysTick(24 bits)	1	
	省电(LPTIMER)	1	
RTC/SYSTICK/IWDG/WWDG		1/1/1/1	
AWK		1	
工作电压范围		2.5~5.5V	
工作温度		-40~85 °C	
调试功能		SWD	
唯一标识符		16 Bytes	
通信界面	UART, USART	2,2	1,1
	LPUART	1	1
	SPI	2(16/20Mhz)	1(16/20Mhz)
	I2C	2	2
比较器	2	2	2
AES		Yes(128 bytes)	
TRNG		1	
CRC16/32		1	
内部温度传感器		1	
时钟	内部高速晶振	HIRC : 12/11.059 MHz	
	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~64Mhz	
12 Bits A/D	16 CH	10 CH	10 CH
OPAx2	2	2	0
蜂鸣器		1	
封装	LQP48	LQFP32/QFN32	TSSOP28

Table 3 - 1 ZB32L032 芯片特性与周边配备

### 3.2 内部功能框图结构



**Figure 3 - 1 内部功能框图结构**

Note : S\*: Maximum channels for Single output ; C\*: Maximum channels for complement output .

1\* : is UART0, UART1, USART2, USART3

### 3.332 位 Cortex®-M0+内核

ARM® Cortex®-M0+处理器是最新一代的嵌入式 32 位 RISC 处理器，该处理器引脚数少、功耗低，能够提供满足 MCU 实现需要的低成本平台，同时提供卓越的计算性能和先进的中断系统响应。

Cortex®-M0+处理器全面支持 Keil、IAR 等调试器，包含了一个硬件调试电路，支持 2 线式的 SWD 调试接口。

Cortex®-M0+特性：

指令集	Thumb / Thumb-2
流水线	2 级流水线
CoreMark/MHz	2.46
DMIPS/MHz	0.95
中断	32 个中断源
中断优先级	可配置 4 级中断优先级
增强指令	单周期 32 位乘法器
调试接口	支持SWD 2线式调试接口，支持4个硬中断(break point)以及2个观察点(watch point)

### 3.3-1 Flash

嵌入式闪存存储器(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 64K 字节。

### 3.3-216K Bytes SRAM

根据客户选择不同的功耗模式，RAM 数据都会被保留。

### 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 192MHz 输出的 PLL。

### 3.5 工作模式

ZB32L032 支持 3 种工作模式：

1. 运行模式 Active: CPU 运行, 周边功能模块运行。
2. 休眠模式 Sleep: CPU 停止运行, 周边功能模块运行。
3. 深度休眠模式 Deep Sleep: CPU 停止运行, 系统主时钟关闭, 低功耗功能模块运行。

可以通过软件来选择运行在哪种工作模式。睡眠模式时 CPU 时钟关闭, 其他部分依然可以工作, 可以通过中断来唤醒 CPU。深度睡眠模式下, 系统主时钟关闭, 绝大部分模块停止工作, 系统工作在内置的 38.4KHz/32.768KHz 内置低速时钟上, 可以通过 RTC 中断, AWK 中断或外部中断来唤醒芯片。在正常工作模式下, 可以选择分频方式工作或停止一些不需要使用的模块的时钟来实现功耗和性能之间的灵活切换。

### 3.6 中断控制器(NVIC)

Cortex®-M0+ 处理器内置了嵌套向量中断控制器(NVIC), 支持最多 32 个中断请求(IRQ)输入, 有四个中断优先级, 可处理复杂逻辑, 能够进行实时控制和中断处理。

详情请参考 “ARM® Cortex®-M0+ Technical Reference Manual” 与 “ARM® v6-M Architecture Reference Manual”。

外部中断号 (IRQ#)	中断源	描述	Sleep 模式唤醒	Deep Sleep 模式唤醒	向量地址
0	GPIO_PA	GPIOA INT	Y	Y	0x0000 0040
1	GPIO_PB	GPIOB INT	Y	Y	0x0000 0044
2	GPIO_PC	GPIOC INT	Y	Y	0x0000 0048
3	GPIO_PD	GPIOD INT	Y	Y	0x0000 004C
4	FLASH	FLASH INT	N	N	0x0000 0050
5	PDMA	PDMA INT	Y	N	0x0000 0054
6	UART0	UART0 INT	Y	N	0x0000 0058
7	UART1	UART1 INT	Y	N	0x0000 005C
8	LPUART	LPUART INT	Y	Y	0x0000 0060
9	USART2,3	USART2,3 INT	Y	N	0x0000 0064
10	SPI/I2S,QSPI	SPI/I2S,QSPI INT	Y	N	0x0000 0068
11	TIM2C	TIM2C INT	Y	N	0x0000 0074
12	I2C0/I2C1	I2C0/I2C1 INT	Y	N	0x0000 0070
13	TIM1B	TIM1B INT	Y	N	0x0000 006C
14	TIM10	TIM10 INT	Y	N	0x0000 0078
15	TIM11	TIM11 INT	Y	N	0x0000 007C
16	LPTIM	LPTIM INT	Y	Y	0x0000 0080
17	TIM1A	TIM1A INT	Y	N	0x0000 0084

外部中断号 (IRQ#)	中断源	描述	Sleep 模式唤醒	Deep Sleep 模式唤醒	向量地址
18	TIM1	TIM1 INT	Y	N	0x0000 0088
19	TIM2	TIM2 INT	Y	N	0x0000 008C
20	TIM2A	TIM2A INT	Y	N	0x0000 0090
21	PCA	PCA INT	Y	N	0x0000 0094
22	WWDG	WWDG INT	Y	Y	0x0000 0098
23	IWDG	IWDG INT	Y	N	0x0000 009C
24	ADC	ADC INT	Y	Y	0x0000 00A0
25	LVD	LVD INT	Y	Y	0x0000 00A4
26	VC0/VC1	VC0/VC1 INT	Y	Y	0x0000 00A8
27	TIM2B	TIM2B INT	Y	N	0x0000 00AC
28	AWK	AWK INT	Y	Y	0x0000 00B0
29	ONEWIRE/ AES	ONEWIRE INT/ AES INT	Y	N	0x0000 00B4
30	RTC	RTC INT	Y	Y	0x0000 00B8
31	CLKTRIM/ TRNG/	CLKTRIM 中断/ TRNG INT/	Y	Y	0x0000 00BC

**Table 3 - 2 中断源**

注：只有在选择内部低速监控外部低速时钟功能时才能唤醒

### 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 CPURST 复位
9	寄存器 MCURST 复位

### 3.8 通用 IO 端口(GPIO)

最多可提供 40 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。40 个通用 IO 可支持外部异步中断。

### 3.9定时器和看门狗

ZB32L032 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 个系统滴答(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			
	TIM1A	16 位	1/2/4/8/16/64/256/1024	递增	4	4	3 对
				递减			
				递增/递减			
	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			
	TIM2A	16 位	1/2/4/8/16/64/256/1024	递增	4	4	0 对
				递减			
				递增/递减			
	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 - 3 定时器菜单

### 3.9-1 高级控制定时器(TIM1,1A,1B)

3个高级控制定时器(TIM1,1A,1B),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	<ul style="list-style-type: none"><li>计数比较匹配中断</li><li>计数周期匹配中断</li></ul>

### 3.9-2 通用定时器(TIM2,2A,2B,2C)

通用定时器(TIM2,2A,2B,2C),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。

### 3.9-8SysTick 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)

### 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(UART0/UART1/USART2/USART3)

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

2 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收发器 (Low Power Universal Asynchronous Receiver/Transmitter)

### 3.13 串行外设接口(SPI) / 内部集成音频接口(I2S)

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

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 One-Wire 接口(OWIRE)

支持 One-Wire 总线协议。

### 3.16蜂鸣器(BEEP)

蜂鸣器模块可以在 BEEP 引脚上产生一个 1KHz, 2KHz, 4KHz 的蜂鸣信号，用来驱动外部的蜂鸣器。2 个基础定时器 TIM10/TIM11 与 1 个 LPTIM 可以功能复用输出，为 Buzzer 提供可编程驱动频率。可以支持互补输出，不需要额外的三级管。

### 3.17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。

### 3.18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。

### 3.19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。  
ID 地址 0x180000F0-0x180000FF。

### 3.20循环冗余校验计算单元(CRC16/32)

符合 ISO/IEC13239 中给出的多项式：

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

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

### 3.21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 16MHz 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 VCAP( 2.5V),VREF(2.048V,1.5V) 参考电压

### 3.22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。

### 3.23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。

### 3.24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。

### 3.25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。

### 3.26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。

### 3.27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。

### 3.28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。

### 3.29 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：PA9、PA10 或 PD01、PC07。

SWD 协议编程接口：PD01、PC07。

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 (PD03)** 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。

### 3.30 内嵌 SRAM

ZB32L032 内置 8K 字节的 SRAM。它可以以字节(8 位)、半字(16 位)或字(32 位)进行访问。

### 3.31 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。

### 3.32 Flash 存储器

Flash 存储器有两个不同存储区域：

- 主存储(Main Array)区域，包括应用程序和用户数据区(若需要时)
- 非易失性存储区域(NVR)，包含三个部分：
  - 选项字节(Option Bytes)区域。参见 31 选项字节区(Option Bytes)。
  - 系统配置(System Configuration)区域
  - ISP 程序区域

Flash 存储器接口基于 AHB 协议执行指令和数据存取。

### 3.33 内嵌的自举程序(Bootloader)

内嵌的自举程序存放在 NVR 区域，在生产时写入。该程序可以通过 UART1 对 Flash 进行重新编程。

## 【4】引脚配置及功能

### 4.1 引脚定义

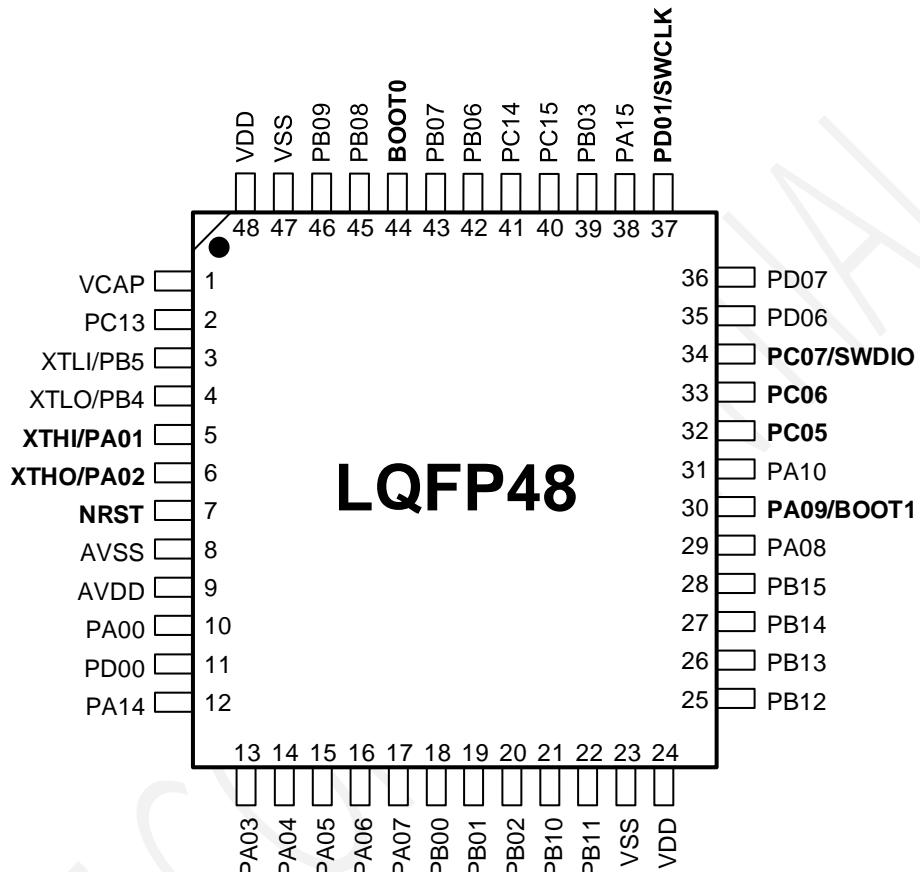


Figure 4 - 1 ZB32L032 LQFP48

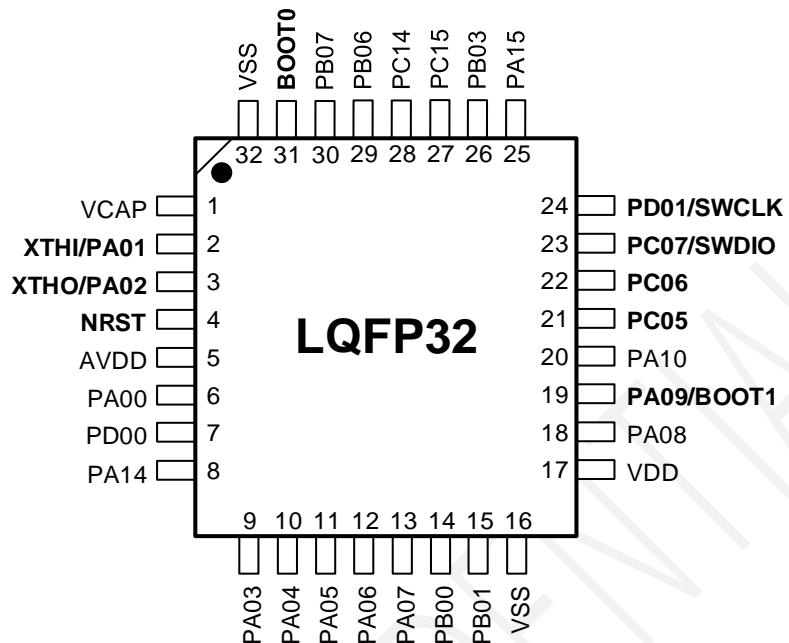


Figure 4 - 2 ZB32L032 LQFP32

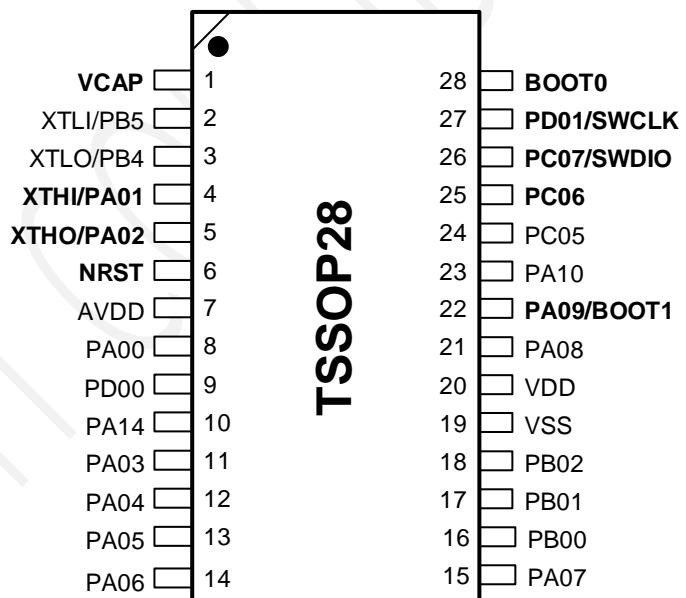


Figure 4 - 3 ZB32L032 TSSOP28

## 4.2 引脚功能配置

Pin Num			Pin Name	Power WKUP	UARTX4 LPUART	SPIX2 I2CX2	TIM2 PCA LPTIM	TIM2A TIM1	TIM2B TIM1A	TIM2C TIM1B	ADC VCX2 OPAX2	Clock IR LVD SWD
48	32	28										
1	1	1	VCAP	P								
2			PC13	WKUP					T2B_CH4	T2C_CH1		
3		2	PB05	XTLI								
4		3	PB04	XTLO								
5	2	4	PA01	XTHI	TXD_1 CKS_2	SDA_0 Q_CLK	T2_ETR	T1_ETR T2A_ETR	T1A_ETR T2B_ETR	TB1_ETR T2C_ETR		
6	3	5	PA02	XTHO	RXD_1 TXD_2	SCL_0 QBK1_NCS	T2_ETR	T1_CH1 T1_ETR T2A_ETR	T1A_ETR T2B_ETR	TB1_ETR T2C_ETR		
7	4	6	NRST									
8			AVSS	G								
9	5	7	AVDD	P								
10	6	8	PA00	WKUP	RXD_2	QBK1_IO0	T2_CH1 T2_ETR	T1_CH1 T2A_CH1 T1_ETR TA2_ETR	T1A_ETR T2B_ETR	T2C_ETR T1B_ETR T2C_ETR	AIN0 VC0_P4 VC0_N0 VC0_O VC1_P0 VC1_N4	
11	7	9	PD00	WKUP	CTS_2	QBK1_IO1	T2_CH2	T1_CH1N T2A_CH2 T2A_ETR			AIN1 VC0_P5 VC0_N1 VC1_P1 VC1_N5	
12	8	10	PA14	WKUP	TXD_1 RTS_2	QBK1_IO2	T2_CH3	T1_CH1 T2A_CH1	T2B_CH1		AIN2 VC0_P6 VC0_N2 VC1_P2 VC1_O	
13	9	11	PA03	WKUP	RXD_1	QBK1_IO3	T2_CH4	T1_CH1N T2A_CH2 T1_ETR	T1A_CH1 T2C_CH1	T1B_CH1 T2C_CH1 T1B_ETR	AIN3 VC0_P7 VC0_N3 VC1_P3	
14	10	12	PA04	WKUP	TXD_1	CS_0/WS QBK2_NCS	PCA_CH4 T2_ETR	T2A_CH3 T1_ETR T2A_ETR	T2B_ETR T1A_CH1 T1A_ETR T2B_ETR	T1B_CH1N T2C_CH2 T1B_ETR T2C_ETR	AIN4 VC0_P8 VC0_N4 VC1_P4	
15	11	13	PA05	WKUP		CLK_0/CK QBK2_IO0	T2_CH1 T2_ETR PCA_ECI T2_ETR	T2A_CH4 T1_ETR T2A_ETR	T1A_CH1N T1A_ETR T2B_ETR	T1B_ETR T2C_ETR	AIN5 VC0_P9 VC0_N5 VC1_P5	LVD_O

Pin Num			Pin Name	Power WKUP	UARTX4 LPUART	SPIX2 I2CX2	TIM2 PCA LPTIM	TIM2A TIM1	TIM2B TIM1A	TIM2C TIM1B	ADC VCX2 OPAX2	Clock IR LVD SWD
48	32	28										
16	12	14	PA06	WKUP	CKS_3	MISO_0 QBK2_IO0	PCA_CH0 T2_ETR	T2A_CH1 T1_ETR T2A_ETR	T1A_ETR T2B_ETR	T1B_CH1 T1B_ETR T2ACETR	AIN6 VC0_P10 VC0_N6 VC0_O	
17		15	PA07	WKUP	TXD_3	MOSI_0/SD QBK2_IO0	PCA_CH1	T1_CH1	T2B_CH1	T1B_CH1N T2C_CH1	AIN7 VC0_P11 VC0_N7 VC1_O	
18		16	PB00	WKUP	RXD_3	QBK2_IO0 MCK	PCA_CH2		T1A_CH1	T2C_CH2	AIN8 VC0_N10 VC1_N6	MCO
19		17	PB01	WKUP	CTS_3		PCA_CH3		T1A_CH1N	T2C_CH3	AIN9 EXVREF VC1_P6 VC1_N7	
20	13	18	PB02	WKUP	RTS_3		PCA_ECI LPT_TOG	T1_CH1 T1_BK	T1A_BK	T1B_BK	AIN16 VC1_P7 VC1_N8 OP2_INN	
21	14		PB10	WKUP	TX_LP	CLK_1 SCL_1		T1_CH1 T2A_CH1		T2C_CH1	AIN17 VC1_P8 OP2_INP	
22	15		PB11	WKUP	RX_LP CKS_2	SDA_1		T1_CH1N		T1B_CH1	AIN18 OP2_O	
23	16	19	VSS	G								
24	17	10	VDD	P								
25			PB12	WKUP	TXD_2	QBK1_NCS		T1_BK		T1B_CH1	AIN19 VC1_P9	
26	18		PB13	WKUP	RXD_2	Q_CLK SCL_1		T1A_CH1N		T2C_CH1 T1B_CH1N	AIN20 VC1_P10 OP1_INP	
27			PB14	WKUP	CTS_2	QBK1_IO0 SDA_1	T2_CH1	T1_CH1	T1A_BK	T2C_CH1	AIN21 VC1_P11 OP1_O	
28	19		PB15	WKUP	RTS_2	QBK1_IO1	T2_CH2	T1_CH1N		T2C_CH2	AIN22 OP1_INN	
29		21	PA08	WKUP		QBK1_IO2	T2_CH3	T1_CH1		T2C_CH3		
30		22	PA09	WKUP	TXD_0	QBK1_IO3 SCL_0	T2_CH4	T1_CH1N T1_BK	T1A_CH1	T2C_CH4		
31	20	23	PA10	WKUP	RXD_0	QBK2_NCS SDA_0		T1_CH2	T1A_CH1	T1B_CH1 T2C_CH1 T1B_BK		

Pin Num			Pin Name	Power WKUP XTAL ISP	UARTX4 LPUART	SPIX2 I2CX2	TIM2 PCA LPTIM	TIM2A TIM1	TIM2B TIM1A	TIM2C TIM1B	ADC VCX2 OPAX2	Clock IR LVD SWD
48	32	28										
32	22	25	PC05	WKUP		SCL_1 MISO_0 QBK2_IO0	T2_ETR	T1_CH2N T1_ETR T2A_ETR	T1A_CH1N T1A_ETR T2B_ETR	T1B_ETR T2C_ETR	VC0_O	MCO
33	22	25	PC06	WKUP		SDA_1 MOSI_0/(SD) QBK2_IO1	T2_ETR	T1_CH3 T1_ETR T2A_ETR	T1A_CH2 T2B_ETR T1A_ETR	T1B_CH1 T2C_ETR T1B_ETR T2C_ETR	VC1_O	
34	23	26	PC07	WKUP	RXD_0	QBK2_IO2		T1_CH3N T1_ETR	T1A_CH2N T1A_ETR	T1B_CH1N T2C_ETR T1B_ETR		SWDIO LVD_O
35			PD06	WKUP		QBK2_IO3 SCL_1		T1_CH4	T1A_CH3	T1B_CH2		
36			PD07			SDA_1			T1A_CH3N	T1B_CH2N		
37	24	27	PD01	WKUP	TXD_0 TXD_1	MCK		T2A_CH1 T1_ETR	T1A_CH4 T1A_ETR	T1B_CH3 T1B_ETR		SWCLK LVD_O MCO
38	25		PA15	WKUP	RXD_1	CS_0 (WS)	T2_ETR T2_CH1	T2A_CH2 T1_CH1		T1B_CH3N T2C_CH4		
39	26		PB03	WKUP		CLK_0 (CK)	T2_CH2 LPT_Gate	T2A_CH3 T1_CH1N		T2C_CH3 T1B_CH1	VC1_N9	
40	27		PC15	WKUP		MISO_0	PCA_CH0 LPT_ETR	T2A_CH4		T2C_CH2 T1B_CH1N T1B_BK	VC0_P12 VC1_P12 VC1_N10	
41	28		PC14	WKUP		MOSI_0/SD	PCA_CH1 LPT_Gate	T1_BK		T2C_CH1	VC0_P13 VC1_P13	
42	29		PB06	WKUP	TXD_0	SCL_0 Q_CLK	T2_CH1 LPT_ETR LPT_TOG	T1_CH1 T2A_CH2			VC0_P14 VC1_P14	
43	30		PB07	WKUP	RXD_0	SDA_0 QBK1_NCS	T2_CH2 LPT_TOGN	T1_CH1N	T2B_CH1	T1B_CH1	VC0_P15 VC1_P15	
44	31	28	PD03	<b>BOOT0</b>								
45			PB08	WKUP	TXD_0	SCL_0 Q_CLK		T2A_CH1	T2B_CH2	T2C_CH2 T1B_CH1N		
46			PB09	WKUP	RXD_0	SDA_0 QBK1_NCS			T2B_CH3	T1B_CH4		
47	32		VSS	G								
48			VDD	P								

Table 4 - 1 引脚功能说明表

#### 4.3选择芯片引脚复用功能说明

Pin Num			GPIOX_AFR[i+3:i]										
			0	1	2	3	4	5	6	7	8	9	
48	32	28										F/Config	
1	1	1	VCAP	P	P	P	P	P	P	P	P	P	
2			PC13			T2C_CH1		T2B_CH4					
3		2	PB05									XTLI	
4		3	PB04									XTLO	
5	2	4	PA01	SDA_0		TXD_1					CKS_2	Q_CLK	XTHI
6	3	5	PA02	SCL_0	T1_CH1	RXD_1					TXD_2	QBK1_NC	XTHO
7	4	6	NRST										
8			AVSS	G	G	G	G	G	G	G			G
9	5	7	AVDD	P	P	P	P	P	P	P			P
10	6	8	PA00	TIM10_TOG	T1_CH1	T2_CH1		T2A_CH1		RXD_2	VC0_O	QBK1_IO0	AIN0 VC0_P4 VC0_N0 VC1_P0 VC1_N4
11	7	9	PD00	TIM10_TOGN	T1_CH1N	T2_CH2	CTS_2	T2A_CH2		MOSI_1	TIM11_TOGN	QBK1_IO1	AIN1 VC0_P5 VC0_N1 VC1_P1 VC1_N5
12	8	10	PA14	TXD_1	T1_CH1	T2_CH3	T2A_CH1	T2B_CH1	TIM11_TOG	MISO_1	RTS_2	QBK1_IO2	AIN2 VC0_P6 VC0_N2 VC1_P2
13	9	11	PA03	RXD_1	T1_CH1N	T1A_CH1	T1B_CH1	T2A_CH2	T2_CH4	CS_1	T2C_CH1	QBK1_IO3	AIN3 VC0_P7 VC0_N3 VC1_P3
14	10	12	PA04	CS_0/WS	TXD_1	PCA_CH4	T1A_CH1	T1B_CH1N	T2A_CH3		T2C_CH2	QBK2_NCS	AIN4 VC0_P8 VC0_N4 VC1_P4
15	11	13	PA05	CLK_0/CK	PCA_ECI		T1A_CH1N	T2_CH1	T2A_CH4	LVDO	VC1_O	QBK2_IO0	AIN5 VC0_P9 VC0_N5 VC1_P5
16	12	14	PA06	MISO_0	PCA_CH0		T1B_CH1	T2A_CH1		VC0_O	CKS_3	QBK2_IO1	AIN6 VC0_P10 VC0_N6

Pin Num			GPIOX_AFR[i+3:i]										
	0	1	2	3	4	5	6	7	8	9	F/Config		
17		15	PA07	MOSI_0/SD	PCA_CH1	T1_CH1	T1B_CH1N	T2B_CH1	T2C_CH1	VC1_O	TXD_3	QBK2_IO2	AIN7 VC0_P11 VC0_N7
18		16	PB00	PCA_CH2	MCK	T1A_CH1	TIM11_G		T2C_CH2	MCO	RXD_3	QBK2_IO3	AIN8 VC0_N10 VC1_N6
19		17	PB01	PCA_CH3		T1A_CH1N	TIM11_EXT		T2C_CH3		CTS_3		AIN9 EXVREF VC1_P6 VC1_N7
20	13	18	PB02		PCA_ECI	LPT_TOG	T1_CH1	T1_BK	T1A_BK	T1B_BK	RTS_3		AIN16 VC1_P7 VC1_N8 OP2_INN
21	14		PB10	SCL_1		T1_CH1		T2A_CH1	T2C_CH1		TX_LP		AIN17 VC1_P8 OP2_INP
22	15		PB11	SDA_1		T1_CH1N		T1B_CH1		CKS_2	RX_LP		AIN18 OP2_O
23	16	19	VSS	G									
24	17	20	VDD	P									
25			PB12		T1B_CH1		T1_BK		LXT_out		TXD_2	QBK1_NCS	AIN19 VC1_P9
26	18		PB13		SCL_1	T1A_CH1N	T1B_CH1N		HXT_out	T2C_CH1	RXD_2	Q_CLK	AIN20 VC1_P10 OP1_INP
27			PB14		SDA_1	T1_CH1	T2_CH1	T1A_BK	SIRC_out	T2C_CH1	CTS_2	QBK1_IO0	AIN21 VC1_P11 OP1_O
28	19		PB15		T1_CH1N	T2_CH2	T2C_CH2		HIRC_out		RTS_2	QBK1_IO1	AIN22 OP0_INN
29		21	PA08	TXD_0		T1_CH1	T2_CH3		BEEP	T2C_CH3		QBK1_IO2	
30		22	PA09 (BOOT1)	TXD_0	T1_CH1N	T1A_CH1	T2_CH4	T1_BK	1-Wire	T2C_CH4	SCL_0	QBK1_IO3	
31	20	23	PA10	RXD_0	T1_CH2	T1A_CH1	T1B_CH1	T1B_BK	RTC_1Hz	T2C_CH1	SDA_0	QBK2_NC	
32	21	24	PC05	TIM10_EX	MISO_0	T1_CH2N	T1A_CH1N		MCO	VC0_O	SCL_1	QBK2_IO0	
33	22	25		TIM10_G	MOSI_0/SD	T1_CH3	T1A_CH2	T1B_CH1		VC1_O	SDA_1	QBK2_IO1	
34	23	26	PC07		RXD_0	T1_CH3N	T1A_CH2N	T1B_CH1N		LVD_O		QBK2_IO2	SWDIO (Config)
35			PD06	SCL_1		T1_CH4	T1A_CH3	T1B_CH2				QBK2_IO3	

Pin Num	GPIOX_AFR[i+3:i]											
	0	1	2	3	4	5	6	7	8	9	F/Config	
36		PD07	SDA_1		T1A_CH3N	T1B_CH2N						
37	24	27	PD01	TXD_1	TXD_0	T1A_CH4	T1B_CH3	T2A_CH1	LVD_O	MCO	MCK	SWCLK (Config)
38	25		PA15	CS_0/WS	RXD_1	T1_CH1	T1B_CH3N		T2_CH1	T2A_CH2	T2C_CH4	
39	26		PB03	CLK_0/CK	LPT_GATE		T1_CH1N	T1B_CH1	T2_CH2	T2A_CH3	T2C_CH3	VC1_N9
40	27		PC15	MISO_0		PCA_CH0	LPT_ETR	T1B_CH1N	T2A_CH4	T1B_BK	T2C_CH2	VC0_P12 VC1_P12 VC1_N10
41	28		PC14	MOSI_0/SD		LPT_GATE	PCA_CH1	T1_BK		T2C_CH1		VC0_P13 VC1_P13
42	29		PB06	SCL_0	TXD_0	LPT_ETR	LPT_TOG	T1_CH1	T2A_CH2	T2_CH1		Q_CLK
43	30		PB07	SDA_0	RXD_0	LPT_TOGN	T1_CH1N	T1B_CH1	T2_CH2	T2B_CH1		QBK1_NCS
44	31	28	PD03 (BOOT0)									
45			PB08	SCL_0	TXD_0	T1B_CH1N	T2A_CH1	T2B_CH2	T2C_CH2			Q_CLK
46			PB09	SDA_0	RXD_0	CS_1		T2B_CH3	T1B_CH4			QBK1_NCS
47	32		VSS	G								
48			VDD	P								

Table 4 - 2 选择芯片引脚表

#### 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 通用数字输入/输出引脚
NRST	NRST	复位输入端口, 低有效, 芯片复位
ADC	AIN0~AIN23	ADC 输入通道 0~23
	EXVREF	ADC 外部参考电压
OPA X=0, 1, 2	OPx_INN	OPA 负端输入
	OPx_INP	OPA 负端输出
	OPx_O	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	LVD_O	电压侦测输出
ISP	BOOT0 BOOT1	当复位时 BOOT0 (PD03) 管脚为高电平和 芯片工作于 ISP 编程模式, 可通过 ISP 协议对 FLASH 进行 编程。 当复位时 BOOT0 (PD03) 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	All GPIO	外部唤醒脚位
LPUART	TXD_LP	LPUART 数据发送端
	RXD_LP	LPUART 数据接收端
UART x=0,1,2,3 USART y=2,3	CKS_y	USART_y CKS
	RTS_x	USART_y RTS
	CTS_x	USART_y CTS
	TXD_x	UART_x 数据发送端
	RXD_x	UART_x 数据接收端

模块功能	引脚名称	说明
SPI x=0,1	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
	CS_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=2,2A,2B,2C	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 Timer1, 1A, 1B	TIM1_CH1,2,3,4	TIM1 PWM 输出 channel 1/2/3/4
	TIM1_CH1N,2N,3N	TIM1 PWM 输出 反相 channel 1N/2N/3N
	TIM1_BKIN	TIM1 刹车信号输入
	TIM1A_CH1,2,3,4	TIM1 PWM 输出 channel 1/2/3/4
	TIM1A_CH1N,2N,3N	TIM1 PWM 输出 反相 channel 1N/2N/3N
	TIM1A_BKIN	TIM1 刹车信号输入
	TIM1B_CH1,2,3,4	TIM1 PWM 输出 channel 1/2/3/4
	TIM1B_CH1N,2N,3N	TIM1 PWM 输出 反相 channel 1N/2N/3N
	TIM1B_BKIN	TIM1 刹车信号输入
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
	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 片选择使能

**Table 4 - 3** 模块与引脚信号说明表

**4.5串行引脚接口说明**

Number Serial	0	1	2	3	4	5	6
I2C	SCL_0 SDA_0	SCL_1 SDA_1					
SPI	CS_0 CLK_0 MISO_0 MOSI_0	CS_1 CLK_1 MISO_1 MOSI_1					
QSPI	BK1_IO0~IO3 BK2_IO0~IO3 Q_CLK BK1_nCS BK2_nCS						
UART	CTS_0 RTS_0 TXD_0 RXD_0	CTS_1 RTS_1 TXD_1 RXD_1	CTS_2 RTS_2 TXD_2 RXD_2	CTS_3 RTS_3 TXD_3 RXD_3			
LPUART	CTS_LP RTS_LP TXD_LP RXD_LP						
PWM 独立 输出	T1_CH1/ T1_CH2/ T1_CH3/ T1_CH4/ PCA_CH0/ PCA_CH1/ PCA_CH2/ PCA_CH3/ PCA_CH4/	T1A_CH1/ T1A_CH2/ T1A_CH3/ T1A_CH4/	T1B_CH1 T1B_CH2 T1B_CH3 T1B_CH4	T2_CH1 T2_CH2 T2_CH3 T2_CH4	T2A_CH1 T2A_CH2 T2A_CH3 T2A_CH4	T2B_CH1 T2B_CH2 T2B_CH3 T2B_CH4	T2C_CH1 T2C_CH2 T2C_CH3 T2C_CH4
PWM 互补 输出	T1_CH1, 1N T1_CH2, 2N T1_CH3, 3N	T1A_CH1, 1N T1A_CH2, 2N T1A_CH3, 3N	T1B_CH1, 1N T1B_CH2, 2N T1B_CH3, 3N				

**Table 4 - 4 串行接口说明表**
**Note :**

PWM 互补输出: Tx\_CHx → Positive, Tx\_CHxN → Negative

## 【5】系统和存储器概要

### 5.1 系统架构图

主要的系统构成:

- 2 个 AHB 总线系统 Master:

- Cortex®-M0+ 内核
- PDMA 控制器

- 8 AHB 总线 Slave

- 内部 SRAM
- 内部 Flash
- 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
- GPIO 接口
- RCC 模块
- CRC
- AES 等 AHB 接口模块

系统的模块框图如图 5-1 所示:

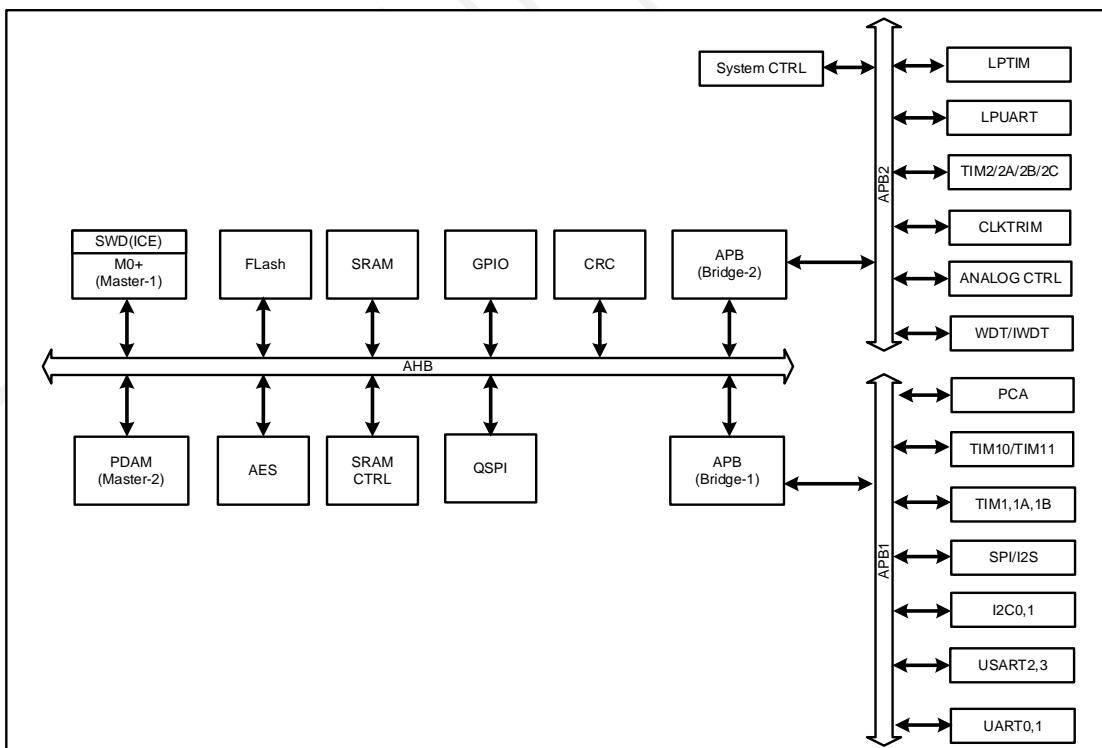


Figure 5 - 1 系统的模块框图

## 5.2 存储器映射

系统的地址空间总共有 4GB，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：

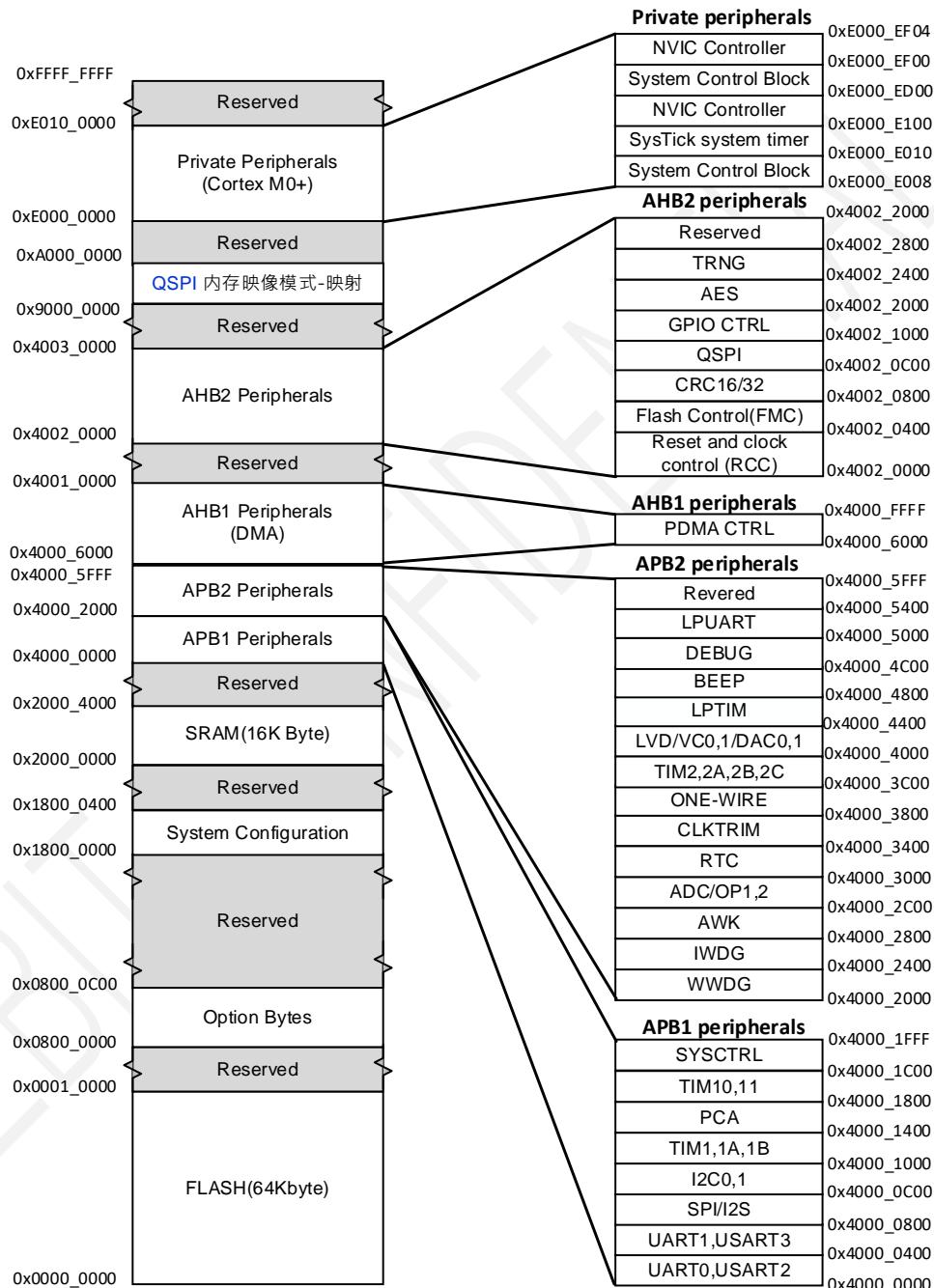


Figure 5 - 2 储存器映像图

### 5.3 存储空间和模块地址

下列表给出了 ZB32L032 chip 内部包含的各模块的地址空间和边界信息。

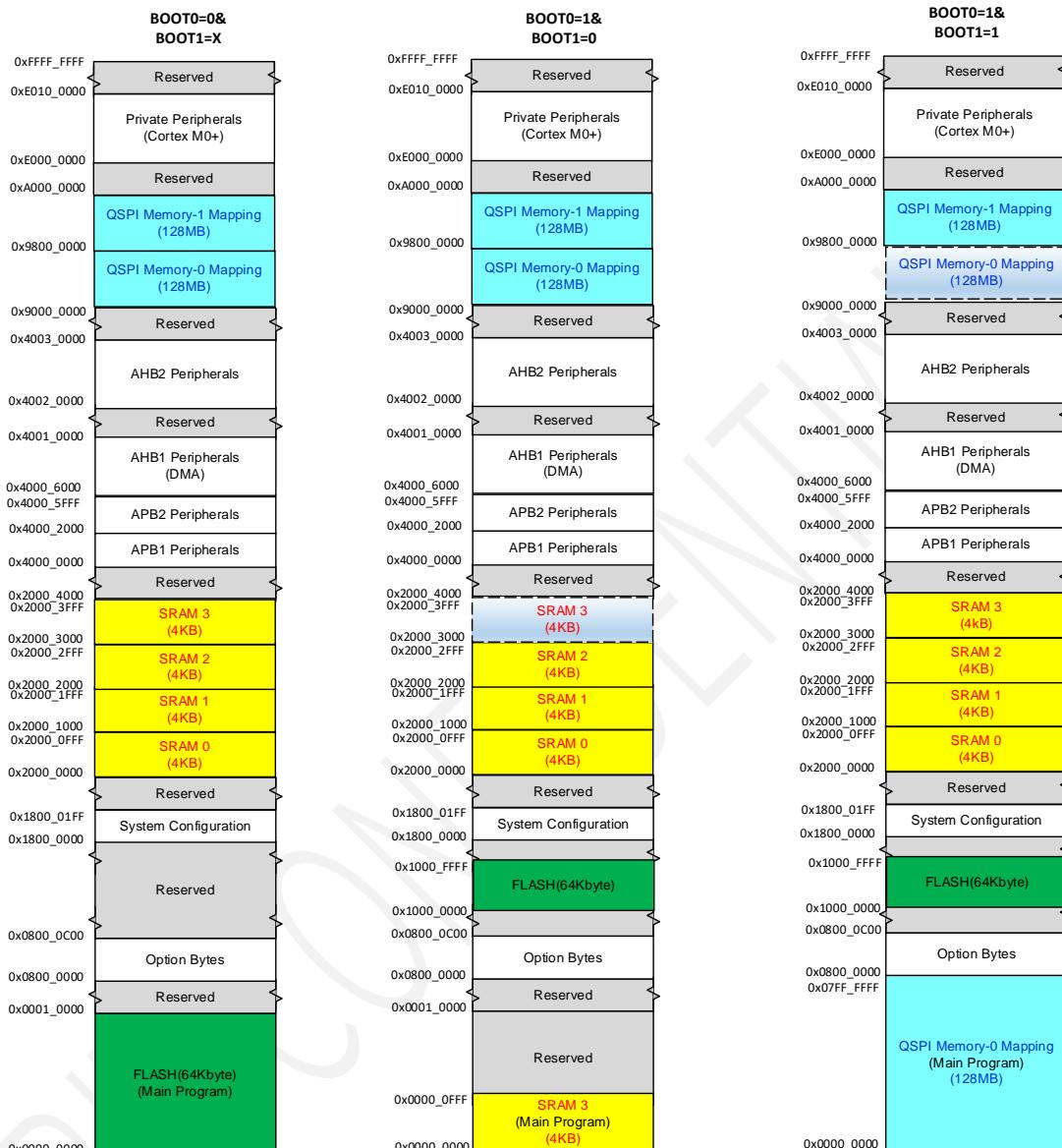
总线	边界地址	空间大小	Peripheral
	0xE000_0000 - 0xE00F_FFFF	1M	Coretex-M0 internal peripheral
<b>AHB</b>	0x9000_0000 - 0x9FFF_FFFF	256MB	QSPI-内存映像模式
	0x4002_2400 - 0x4002_27FF	1KB	TRNG
	0x4002_2000 - 0x4002_23FF	1KB	AES
	0x4002_1C00 - 0x4002_1FFF	1KB	GPIOD
	0x4002_1800 - 0x4002_1BFF	1KB	GPIOC
	0x4002_1400 - 0x4002_17FF	1KB	GPIOB
	0x4002_1000 - 0x4002_13FF	1KB	GPIOA
	0x4002_0C00 - 0x4002_0FFF	1KB	QSPI
	0x4002_0800 - 0x4002_0BFF	1KB	CRC16
	0x4002_0400 - 0x4002_07FF	1KB	FMC
	0x4002_0000 - 0x4002_03FF	1KB	RCC
	0x4001_0000 - 0x4FFF_FFFF		Reserved
<b>AHB</b>	0x4000_6000 - 0x4000_FFFF	16KB	PDMA
<b>APB</b>	0x4000_5800 - 0x4000_5FFF	2KB	Reserved
	0x4000_5400 - 0x4000_57FF	1KB	OP1,2
	0x4000_5000 - 0x4000_53FF	1KB	LPUART
	0x4000_4C00 - 0x4000_4FFF	1KB	DEBUG
	0x4000_4800 - 0x4000_4BFF	1KB	BEEP
	0x4000_4400 - 0x4000_47FF	1KB	LPTimer
	0x4000_4100 - 0x4000_43FF	256B	VC1(DAC1)
	0x4000_4080 - 0x4000_40FF	128B	VC0(DAC0)
	0x4000_4000 - 0x4000_407F	128B	LVD
	0x4000_3F00 - 0x4000_3FFF	256B	TIM2C
	0x4000_3E00 - 0x4000_3EFF	256B	TIM2B
	0x4000_3D00 - 0x4000_3DFF	256B	TIM2A
	0x4000_3C00 - 0x4000_3CFF	256B	TIM2
	0x4000_3800 - 0x4000_3BFF	1KB	1-WIER
	0x4000_3400 - 0x4000_37FF	1KB	CLKTRIM
	0x4000_3000 - 0x4000_33FF	1KB	RTC
	0x4000_2C00 - 0x4000_2FFF	1KB	ADC
	0x4000_2800 - 0x4000_2BFF	1KB	AWK
	0x4000_2400 - 0x4000_27FF	1KB	IWDT
	0x4000_2000 - 0x4000_23FF	1KB	WWDT
	0x4000_1C00 - 0x4000_1FFF	1KB	SYSCTRL

总线	边界地址	空间大小	Peripheral
AHB	0x4000_1800 - 0x4000_1BFF	1KB	Timer10/11
	0x4000_1400 - 0x4000_17FF	1KB	PCA
	0x4000_1300 - 0x4000_13FF	256B	Timer1B
	0x4000_1200 - 0x4000_12FF	256B	Timer1A
	0x4000_1000 - 0x4000_11FF	512B	Timer1
	0x4000_0E00 - 0x4000_0FFF	512B	I2C1
	0x4000_0C00 - 0x4000_0DFF	512B	I2C0
	0x4000_0800 - 0x4000_0BFF	512B	SPI/I2S
	0x4000_0600 - 0x4000_07FF	512B	USART3
	0x4000_0400 - 0x4000_05FF	512B	UART1
	0x4000_0200 - 0x4000_03FF	512B	USART2
	0x4000_0000 - 0x4000_01FF	512KB	UART0
APB	0x2000_2000 - 0x3FFF_FFFF		Reserved
	0x2000_0000 - 0x2000_3FFF	16KB	SRAM
	0x1800_0100 - 0x1FFF_FFFF		Reserved
	0x1800_0000 - 0x1800_01FF	512B	System Configuration
	0x0800_0200 - 0x17FF_FFFF		Reserved
	0x0800_0000 - 0x0800_01FF	512B	Option Bytes
	0x0001_0000 - 0x07FF_FFFF		Reserved
	0x0000_0000 - 0x0001_FFFF	64KB	eFlash

Table 5 - 1 地址对应表

#### 5.4BOOT PIN 地址映射

L032 当发生 POR,NRST,WDT,LVR 重置时,会 Latch BOOT0,BOOT1PIN 状态将储存器作地址重新映射,如下图所示([Timing Refer 11.3-8.2](#)):


**Figure 5 - 3 BOOT PIN 地址映射图**

## 【6】工作模式和电源管理

**ZB32ML032** 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 2.5~5.5V。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): CPU 停止运行，周边功能模块运行。
3. 深度休眠模式(Deep Sleep Mode): CPU 停止运行，高速时钟停止运行。

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

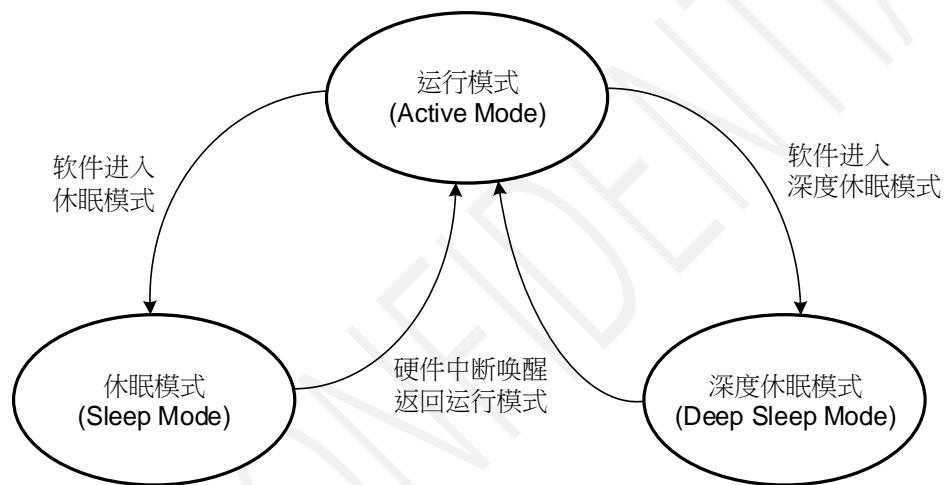


Figure 6 - 1 控制模式框图

在各种模式下，CPU 可响应的中断类型请参考表 3-2。

各模式下，CPU 可以响应的复位源类型

编号	复位源/中断源	运行模式	Sleep 模式唤醒	Deep Sleep 模式唤醒
1	上电/掉电复位	Y	Y	Y
2	外部 Reset Pin 复位	Y	Y	Y
3	IWDG 复位	Y	Y	Y
4	WWDG 复位	Y	Y	N
5	系统软件复位	Y	N	N
6	欠电压(LVD)复位	Y	Y	Y
7	LOCKUP 复位	Y	N	N
8	寄存器 CPURST 复位	Y	N	N
9	寄存器 MCURST 复位	Y	N	N
10	所有中断源	请参考表 3-2 中断源	请参考表 3-2 中断源	请参考表 3-2 中断源

Table 6 - 1 MCU 唤醒菜单

## 6.1 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-M0+	TIM10,TIM11	UART0,1,2,3	ADC	XTAL32
SWD	TIM1,TIM1A,TIM1B	SPI0,1	RNG	XTAH
Flash	TIM2,TIM2A,TIM2B,TIM2C	I2C0,1	OPA0,1	HIRC
SRAM	PCA	LPUART	Reset	SIRC
DMAC	IWDG	1-WIRE	POR/PDR	PLL
HMUL	WWDG	BEEP	BandGap	RTC
CRC	LPTIM	GPIO	Temp. Sensor	CLK_TRIM
TRAN	AWK		LVD	
AES			VC0,1	
			PIN Reset	
			DAC0,1	

Table 6 - 2 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 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。

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。

使用注意：

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

休眠模式(Sleep Mode)				
MCU Core	Timer	Serial Interface	Analog	Clock
Cortex-M0+ (1*)	TIM10,TIM11	UART0,1,2,3	ADC	XTAL32
SWD	TIM1,1A,1B	SPI0,1	RNG	XTAH
Flash	TIM2,2A,2B,2C	I2C0,1	OPA0,1	HIRC
SRAM	PCA	LPUART	Reset	SIRC
CRC	IWDG	1-WIRE	POR/PDR	PLL
TRAN	WWDG	BEEP	BandGap	RTC
DMA	LPTIM	GPIO	Temp. Sensor	CLK_TRIM
	AWK		LVD	
			VC0,1	
			PIN Reset	
			DAC	

1\* : 灰色的模块在当前状态下不工作。

Table 6 - 3 休眠模式下的模块工作表

### 6.3深度休眠模式(Deep Sleep Mode)

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

- (1) 系统从高速时钟进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度睡眠前的状态。
- (2) 系统从低速时钟进入深度休眠模式，低速时钟保持运行，除了低功耗模块可以运行，其他模块自动关闭。
- (3) 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。
- (4) 系统进入深度休眠状态，不会改变端口状态，在进入深度休眠前根据需要更改 IO 的状态为深度休眠模式下的状态。

如何进入深度休眠模式：

首先设置 **Cortex®-M0+** 系统控制寄存器中的 SLEEPDEEP 位，通过执行 WFI 指令进入深度睡眠状态。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深度睡眠模式进入机制：

- (1) **SLEEP-NOW:** 如果 SLEEPONEXIT=0，当 WFI 或 WFE 被执行时，微控制器立即进入深度睡眠模式。
- (2) **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深度睡眠模式。

如何退出深度休眠模式：

如果执行 WFI 指令进入睡眠模式，任意一个被嵌套向量中断控制器(NVIC)响应的外设中断(Deep Sleep 模式下可运行的周边模块中断)都能将系统从睡眠模式唤醒。唤醒设置参考 8.5 中断唤醒控制器 WIC。

深度休眠模式(Deep Sleep Mode)				
MCU	Timer	Serial Interface	Analog	Clock
Cortex-M0+	TIM10,TIM11	UART0,1,2,3	ADC	XTAL32
SWD (1*,2*)	TIM1,1A,1B	SPI0,1	RNG	XTAH
Flash	TIM2,2A,2B,2C	I2C	OPA0,1	HIRC
SRAM	PCA	LPUART	Reset	SIRC
DMAC	IWDG (1*)	1-WIRE	POR/PDR	--
HMUL	WWDG	BEEP	BandGap	RTC
CRC	LPTIM(1*)	GPIO(1*)	--	CLK_TRIM(1*,3*)
TRAN	AWK(1*)		LVD	
AES			VC0,1	
DMA			PIN Reset(1*)	
			DAC	

Note : 1\*: 除了灰色的模块在当前状态下可(或不)工作, 其他模块不工作。

2\*: 在 Deep Sleep 模式下, 芯片重新复位后, 可以通过 SWD 接口唤醒。

3\*: 只有在选择内部低速监控外部低速时钟功能时才能唤醒

**Table 6 - 4 深度休眠模式下的模块工作表**

#### 6.4 Cortex®-M0+内核系统控制寄存器(SCR)

地址 : 0xE000\_ED10

复位值: 0x0000 0000

Bits	标记	功能	复位值	读写
31:5	RESERVED	保留	0x0	-
4	SEVONPEND	1 = 启用的事件和所有中断（包括禁用的中断）都可以唤醒处理器。每次新的中断挂起都会产生一个事件，如果使用了WFE 休眠，它可用于唤醒处理器 0 = 只有使能的中断或事件才能唤醒处理器，不包括禁用的中断	0	R/W
3	RESERVED	保留	0	-
2	SLEEPDEEP	1 = 执行 WFI 进入深度休眠，本产品进入 <b>Deep Sleep</b> 模式 0 = 执行 WFI 进入休眠，本产品进入 <b>Sleep/Idle</b> 模式	0	R/W
1	SLEEPONEXIT	1 = 当退出异常处理并返回程序线程时，处理器自动进入休眠模式(WFI) 0 = 该特性就会被自动禁止	0	R/W
0	RESERVED	保留	0	-

Note : 进入深度休眠后，唤醒后系统时钟有两种选择，默认使用进入深度休眠的时钟，配置寄存器 RCC\_SYSCLKCR.WKBYHIRC 为 1 后不管进入深度休眠前是什么时钟，唤醒后都使用内部高速时钟 HIRC。如果进入深度睡眠前系统使用外部晶体振荡这样设置可以加速系统唤醒。

## 【7】系统复位与时钟(RCC)

### 7.1 复位

#### 7.1-1 复位控制器介绍

ZB32L032 具有 10 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序计数器 PC 会被复位成复位地址。

- (1) 数字区域上电掉电复位 POR
- (2) 外部引脚复位(NRST PAD)，低电平复位
- (3) IWDG 复位
- (4) WWDG 复位
- (5) PCA 复位
- (6) 低电压复位(LVD)
- (7) 软件复位(Cortex®-M0+ SYSRESETREQ)
- (8) Cortex®-M0+ LOCKUP 硬件复位
- (9) 寄存器复位(CPURST)
- (10) 寄存器复位(MCUREST)

每个复位源都有一个专用的复位标志位来表示，由硬件置“1”，软件清除。除了数字区域的 POR 复位标志位，其他的复位标志位都可以被数字区域的 POR 来清除。

下图描述了个复位的来源：

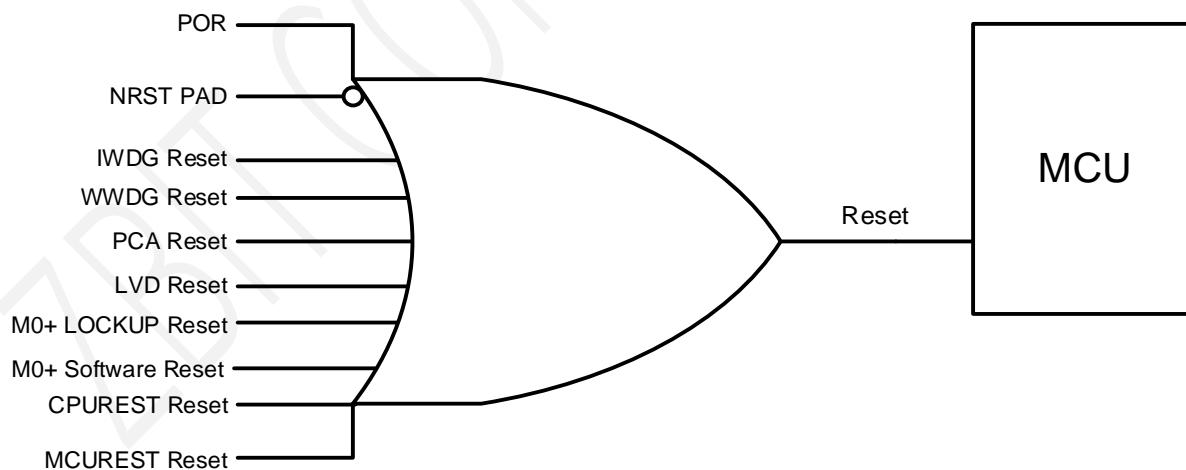


Figure 7 - 1 复位来源示意图

## 7.1-2复位源

### 7.1-2.1上电/下电复位 POR

当芯片上电以及掉电时，如果电源低于一个阈值电压(2.2V)，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。

### 7.1-2.2外部引脚复位

把外部复位引脚拉到低电平就会产生一个系统复位。这个复位引脚内部带有上拉电阻，另外，在内部集成了一个毛刺滤波电路，系统会自动过滤小于 10us(典型值)的毛刺信号，因此，用户在使用复位引脚产生复位信号时，必须产生大于 10us 的低电平以保证芯片可以正确接收到复位信号。

### 7.1-2.3IWDG 复位

独立看门狗复位，请参看 [IWDG 一章说明](#)。

### 7.1-2.4WWDG 复位

窗口看门狗复位，请参看 [WWDG 一章说明](#)。

### 7.1-2.5PCA 低电压复位

PCA 复位，请参考 [PCA 一章说明](#)。

### 7.1-2.6Cortex®-M0+软件复位

通过将 Cortex®-M0+ 系统控制寄存器中的 SYSRESETREQ 位置 1，可实现软件复位。请参考 Cortex®-M0+ 技术参考手册获得进一步信息。

### 7.1-2.7寄存器复位

可以通过写 RCC\_RSTCR 寄存器的 MCURST 和 CPURST 来复位芯片。

### 7.1-2.8Cortex®-M0+ LOCKUP 硬件复位

当 Cortex®-M0+ 遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，并在几个时钟周期延时之后复位整个内核区域。

ZBIT CONFIDENTIAL

## 7.2 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 : 64MHz)

Note :

- (1\*) XTLI、XTHI 可以通过端子 PC14、PD00 从外部输入。使用外部振荡输入时，需要使能相应的振荡。选择外部振荡控制选择在 **RCC\_SYSCLKCR**、**RCC\_LXTCR** 寄存器中。
- (2\*) 每种时钟源都可以单独的打开或关断，当它们不用时，可以关断它们来降低功耗。
- (3\*) 有多个分频器可用于配置 AHB 和 APB 时钟域，AHB 和 APB 域的最大时钟频率为 64MHz。
- (4\*) Cortex M0+ SysTick 定时器由 AHB 时钟驱动，其可由 AHB/4 或 AHB 时钟频率直接驱动(通过 SYST\_CSR.CLKSOURCE 来配置)。RCC 可以使用 AHB 时钟(HCLK)的 4 分频作为 SysTick 定时器的外部时钟，通过对 SysTick 控制与状态寄存器(SYST\_CSR.CLKSOURCE)的设置，可选择上述时钟或 Cortex 时钟(HCLK)作为 SysTick 时钟。
- (5\*) 需要特别注意的是：在切换系统时钟源的过程中需要按照正确的流程，具体流程参见“7.2.7 系统时钟切换”章节。

### 7.2-1 系统时钟模块图

ZB32L032 系统时钟控制模块图如下：

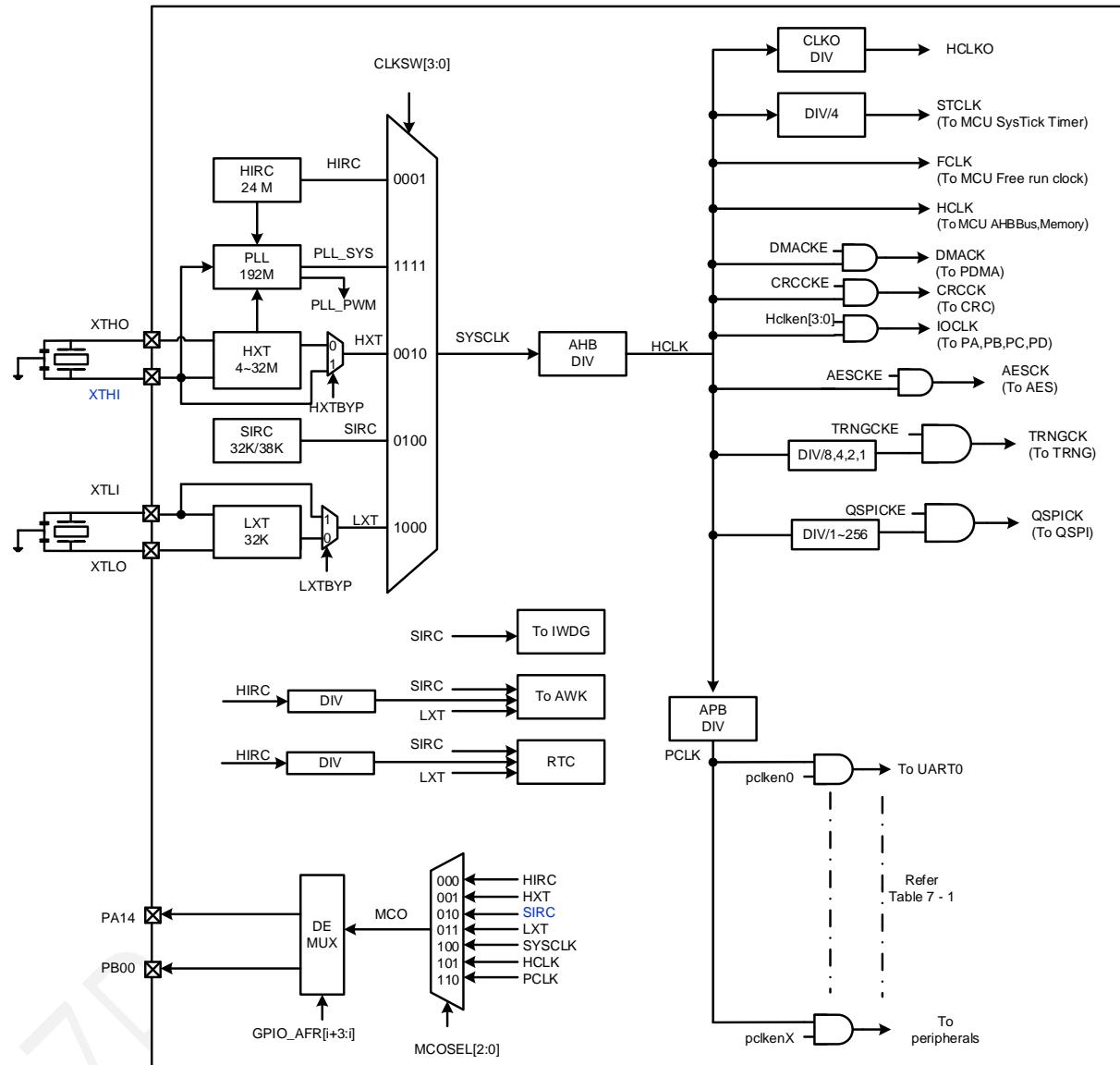


Figure 7 - 2 系统时钟控制模块图

控制信号	模块功能	控制讯号	模块功能
Pclk[0]	UART0	Pclk[16]	CLKTRIM
Pclk[1]	UART1	Pclk[17]	IWDG
Pclk[2]	I2C	Pclk[18]	LVD/VC0,VC0
Pclk[3]	LPUART	Pclk[19]	BEEP
Pclk[4]	---	Pclk[20]	MCUFBG
Pclk[5]	LPTIM	Pclk[21]	USART2
Pclk[6]	TIM10/11	Pclk[22]	USART3
Pclk[7]	SYSCON	Pclk[23]	SPI1/I2S
Pclk[8]	PCA	Pclk[24]	I2C1
Pclk[9]	1-Wire	Pclk[25]	TIM1A
Pclk[10]	TIM1	Pclk[26]	TIM1B
Pclk[11]	TIM2	Pclk[27]	TIM2A
Pclk[12]	WWDG	Pclk[28]	TIM2B
Pclk[13]	ADC	Pclk[29]	TIM2C
Pclk[14]	AWG	Pclk[30]	OPA1,2
Pclk[15]	RTC	Pclk[31]	RNC

**Table 7 - 1 模块时钟对应表**

## 7.2-2 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。

## 7.2-3 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。

## 7.2-4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。

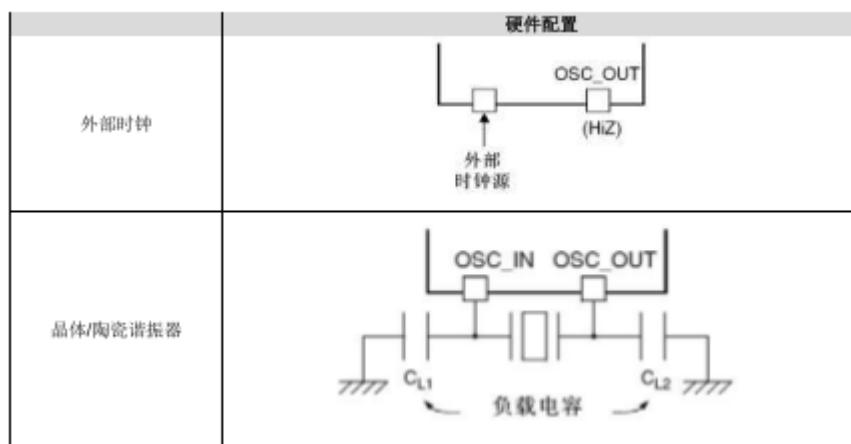


Figure 7 - 3 HXT/LXT 时钟源

## 7.2-5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。

## 7.2-6PLL 时钟

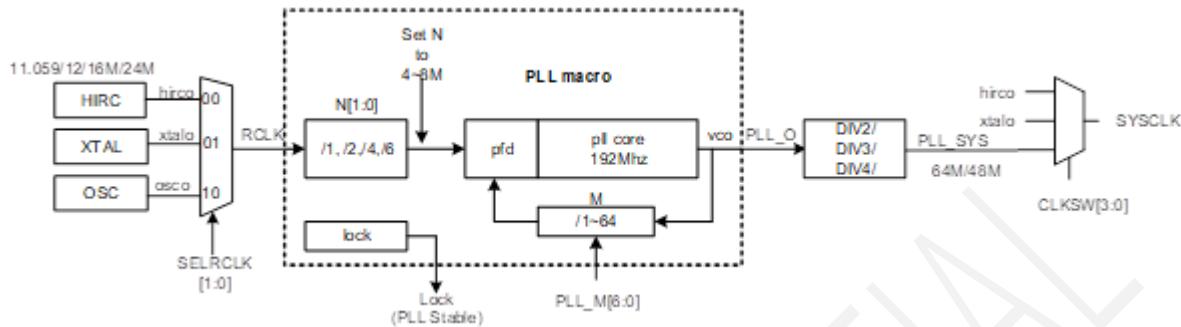


Figure 7 - 4 PLL 时钟源

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.
- 需设定 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	2' b11(6)	48	192MHz
22.118MHz	2' b11(6)	52	191.69 MHz
16MHz	2' b10(4)	48	192MHz
8 MHz	2' b01(2)	48	192MHz
4 MHz	2' b00(1)	48	192MHz

RCLK (HIRC/XTAL/OSC)	PLL_N [1:0]	PLL_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 7 - 2 PLL 参数表

Note: 1\* 建议设置  $PLL\_M=48$

### 7.2-7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。

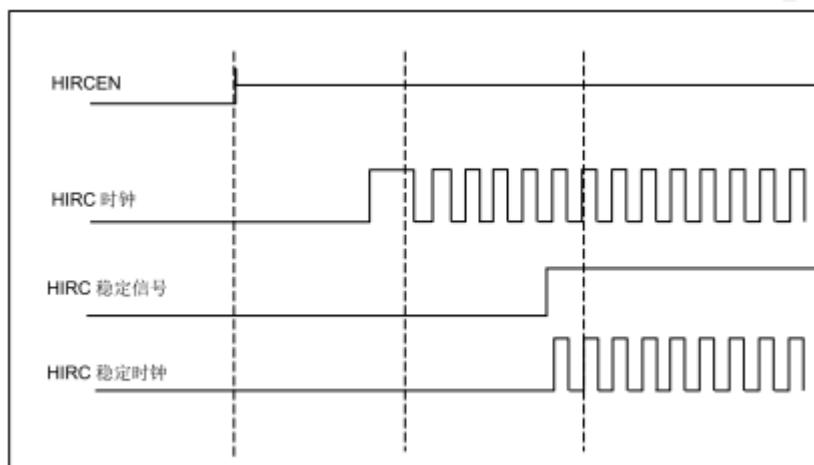


Figure 7 - 5 内部高速时钟启动示意图

### 7.2-8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。

#### 7.2-8.1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：

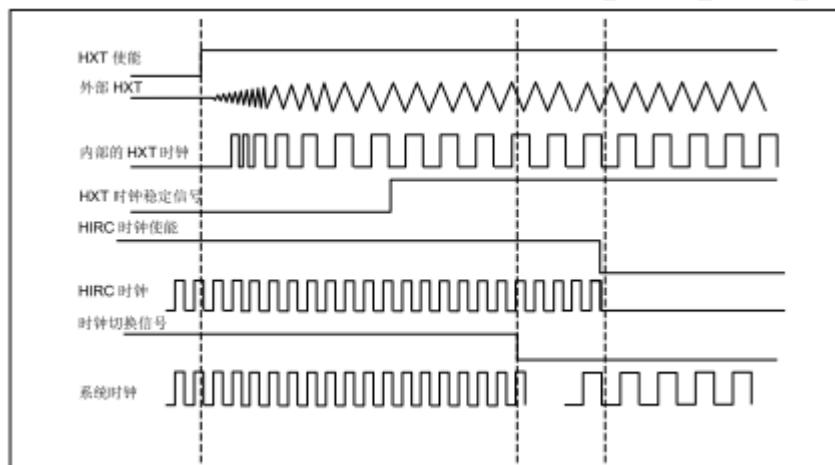
- (1) 通过 RCC\_LXTCR.LXTPORT 位来配置要切换的时钟 LXT 使用的引脚为模拟引脚，或者通过对于引脚的 GPIOx\_AFR=0x0F 来配置为模拟功能
- (2) 写 RCC\_LXTCR.LXTEN 使能 LXT 时钟
- (3) 等待寄存器 RCC\_LXTCR.LXTRDY 位被硬件置“1”
- (4) 写寄存器 RCC\_SYSCLKSEL.CLKSW[3:0]来切换时钟
- (5) 根据需要关闭 HIRC 时钟

### 7.2-8.2 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 7 - 6 时钟切换示意图**

### 7.2-8.3 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

### 7.2-8.4 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

### 7.2-9 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]
HIRC	3' b000
HXT	3' b001
SIRC	3' b010
LXT	3' b011
SYSCLK	3' b100
HCLK_O	3' b101

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

### 7.2-10 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。

### 7.2-11IWDG 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。

### 7.2-12RTC 时钟

RTCCLK 时钟源可以由 HXT 分频、LXT 或 SIRC 时钟提供。

### 7.2-13AWK 时钟

AWKCLK 时钟源可以由 HXT 分频、LXT 或 SIRC 时钟提供。

### 7.2-14低功耗模式

APB 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。

### 7.3 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 0000**

偏移地址	名称	描述	复位值
0x00	RCC_HCLKDIV	AHB 时钟分频寄存器	0x0000 0000
0x04	RCC_PCLKDIV	APB 时钟分频寄存器	0x0000 0000
0x08	RCC_HCLKEN	AHB 周边模块时钟使能寄存器	0x0000 0100
0x0C	RCC_PCLKEN	APB 周边模块时钟使能寄存器	0x0000 0000
0x10	RCC_MOCR	时钟输出控制寄存器	0x0000 0000
0x14	Revered	Revered	0x0000 0000
0x18	RCC_RSTCR	系统Reset 控制寄存器	0x0000 0000
0x1C	RCC_RSTSR	Reset 状态寄存器	0x0000 00A0
0x20	RCC_SYSCLKCR	时钟源设置寄存器	0x0000 0001
0x24	RCC_SYSCLKSEL	系统时钟源选择寄存器	0x0000 0001
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	RCC_IRQLATENCY	Cortex M0+ IRQ 延时控制	0x0000 0000
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_PERIRST1	周边模块复位控制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 7 - 3 RCC 寄存器列表和复位值**

## 7.4寄存器说明

### 7.4-1AHB 时钟分频寄存器(RCC\_HCLKDIV)

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								AHBCKDIV[7:0]							
--								R/W							

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	-
7:0	AHBCKDIV[7:0]	系统HCLK 时钟分频 0: HCLK=SYSCLK 1~255: Divide by 2×DIV (HCLK = SYSCLK/(2×AHBCKDIV))	0x0	R/W

#### 7.4-2APB 时钟分频寄存器(RCC\_PCLKDIV)

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								APBCKDIV[7:0]							
--								R/W							

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	-
7:0	APBCKDIV[7:0]	系统 PCLK 时钟分频(最大 1/16) 0: PCLK=HCLK 1~255: Divide by 2×DIV (PCLK = HCLK /(2×APBCKDIV))	0x0	R/W

### 7.4-3AHB 周边模块时钟使能寄存器(RCC\_HCLKEN)

地址偏移: 0x08 复位值: 0x0000 0100

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						QSPI CKE	Flash CKE	DMA CKE	TRNG CKE	AES CKE	CRC CKE	GPIOD CKE	GPIOC CKE	GPIOB CKE	GPIOA CKE
--						R/W									

位	标记	功能描述	复位值	读写
31:10	-	保留	0x0	-
9	QSPICKE	QSPI 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
8	FlashCKE	Flash 控制器模块时钟使能。关闭后Flash 配置寄存器不可写， Flash中的程序仍然可以运行。 0: 关闭 1: 使能	1	R/W
7	DMACKE	DMA 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
6	TRNGCKE	TRNG 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
5	AESCCKE	AES 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
4	CRCCCKE	CRC 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
3	GPIODCKE	GPIOD 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

位	标记	功能描述	复位值	读写
2	GPIOCKE	GPIOC 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
1	GPIOBCKE	GPIOB 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
0	GPIOACKE	GPIOA 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

### 7.4-4APB 周边模块时钟使能寄存器(RCC\_PCLKEN)

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
RNC EN	OPA12 EN	TIM2C CKEN	TIM2B CKEN	TIM2A CKEN	TIM1B CKEN	TIM1A CKEN	I2C1 CKEN	SPI/I2S CKEN	UART3 CKEN	UART2 CKEN	DBG CKEN	BEEP CKEN	LVDVC CKEN	IWDG CKEN	CLKTRIM CKEN
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC CKEN	AWK CKEN	ADC CKEN	WWDG CKEN	TIM2 CKEN	TIM1 CKEN	OWIRE CKEN	PCA CKEN	SYSCON CKEN	BASETIM CKEN	LPTIM CKEN	保留	LPUART CKEN	I2C CKEN	UART0 CKEN	UART0 CKEN
R/W															

位	标记	功能描述	复位值	读写
31	RNCEN	RNC PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
30	OPA12EN	OPA1,OP2 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
29	TIM2CCKEN	TIM2C PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
28	TIM2BCKEN	TIM2B PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
27	TIM2ACKEN	TIM2A PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
26	TIM1BCKEN	TIM1B PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
25	TIM1ACKEN	TIM1A PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
24	I2C1CKEN	I2C1 PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

位	标记	功能描述	复位值	读写
23	SPI/I2SCKEN	SPI/I2S PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
22	USART3CKEN	USART3 PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
21	UART2CKEN	USART2 PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
20	DBGCKEN	Debug PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
19	BEEPCKEN	BEEP PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
18	LVDVCCKEN	LVD/VC PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
17	IWDGCKEN	IWDG PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
16	CLKTRIMCKEN	CLKTRIM PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
15	RTCCKEN	RTC PCLK 时钟使能 1: 时钟使能 0: 时钟关闭	0	R/W
14	AWKCKEN	AWK PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
13	ADCCKEN	ADC PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

位	标记	功能描述	复位值	读写
12	WWDGCKEN	WWDG PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
11	TIM2CKEN	TIM2 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
10	TIM1CKEN	TIM1 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
9	OWIRECKEN	1-WIRE PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
8	PCACKEN	PCA PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
7	SYSCONCKEN	SYSCON PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
6	BASETIMCKEN	TIM10/11 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
5	LPTIMCKEN	Low Power Timer PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
4	保留	--	0	R/W
3	LPUARTCKEN	Low Power UART PCLK 寄存器配置时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
2	I2CCKEN	I2C PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
1	UART1CKEN	UART1 PCLK 模块时钟使能 1: 时钟使能 0: 时钟关闭	0	R/W
0	UART0CKEN	UART0 PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

### 7.4-5时钟输出控制寄存器(RCC\_MCOCR)

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			MCOEN	MCOSEL[2:0]				MCODIV[7:0]							
--	R/W	R/W													

位	标记	功能描述	复位值	读写
31:13	-	保留	0x0	-
12	MCOEN	MCO 输出使能写: 0: MCO 输出禁止 1: MCO 输出使能读: 0: MCO 未开始输出 1: MCO 开始输出 注意, 读该位时是输出使能信号通过输出时钟同步后的信号。	0	R/W
11:8	MCOSEL[3:0]	时钟输出源选择 0000: HIRC 0001: HXT 0010: SIRC 0011: LXT 0100: SYSCLK 0101: HCLK 以及分频输出 1xxx: PCLK 0110, 0111: 保留, 设定禁止	0x0	R/W
7:0	MCODIV[7:0]	HCLK 时钟分频系数 0: HCLK 1~255: Divide by 2×DIV(HCLK_O = HCLK / (2×DIV))	0x0	R/W

### 7.4-6系统复位控制寄存器(RCC\_RSTCR)

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
RSTKEY															
WO															

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

位	标记	功能描述	复位值	读写
31:2	RSTKEY	必须在写 RCC_RSTCR[1:0]时 写入 0x156A99A6 (0x55AA6699>>2)才有效。 写入其他值都无效。	0x0	WO
1	CPURST	寄存器CPU 复位，该复位发生时，不会重新装载选项字节区中的ISP 设定 0: Normal 1: Reset CPU	0	R/W
0	MCURST	寄存器MCU 复位，该复位发生时，会重新装载选项字节区中的ISP 设定 0: Normal 1: Reset MCU	0	R/W

Note:

- (1) MCU reset by writing 0x55AA6699 to RCC\_RSTCR
- (2) CPU reset by writing 0x55AA669A to RCC\_RSTCR
- (3) 只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。

#### 7.4-7 系统复位状态寄存器(RCC\_RSTSR)

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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							SFT RST	PAD RST	LOCKUP RST	POR RST	LVD RST	IWDG RST	WWDG RST	CPU RST	MCU RST
--							R/W								

位	标记	功能描述	复位值	读写
31:9	-	保留	0x0	-
8	SFTRST	Cortex-M0+ CPU 软件复位标志, 需要软件清除, 0: 无Cortex-M0+ CPU 软件复位发生 1: 发生Cortex-M0+ CPU 软件复位	0	R/W
7	PADRST	RESET 端口复位标志, 能够被POR 复位 0: 无端口复位发生 1: 发生端口复位	1	R/W
6	LOCKUPRST	Cortex-M0+ CPU Lockup 复位标志, 0: 无Cortex-M0+ CPU Lockup 复位发生 1: 发生Cortex-M0+ CPU Lockup 复位	0	R/W
5	PORRST	Vcore 域POR 复位标志 0: Vcore 域POR 无复位发生 1: Vcore 域POR 发生复位	1	R/W
4	LVDRST	LVD 复位标志 0: LVD 无复位发生 1: LVD 发生复位	0	R/W
3	IWDGRST	IWDG 复位标志 0: IWDG 无复位发生 1: IWDG 发生复位	0	R/W
2	WWDGRST	WWDG 复位标志 0: WWDG 无复位发 1: WWDG 发生复位	0	R/W

1	CPURST	寄存器 CPU 复位标志，该复位发生时，不会重新装载选项字节区中的ISP 和IAP 设定 0: 寄存器CPURST 无复位发生 1: 寄存器CPURST 复位发生	0	R/W
0	MCURST	寄存器 MCU 复位标志，该复位发生时，会重新装载选项字节区中的ISP 和IAP 设定 0: 寄存器MCURST 无复位发生 1: 寄存器MCURST 复位发生	0	R/W

Note: 只受 POR 控制，只能由硬件置“1”，软件清“0”。

### 7.4-8系统时钟源配置寄存器(RCC\_SYSCLKCR)

地址偏移: 0x20

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
WKBY HIRC	保留				CLKFAIL EN	保留	HXT PORT	HXT BYP	保留	PLL EN	SIRC EN	HXT EN	HIRC EN		
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	WKBYHIRC	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	-	保留	0	-
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
4	-	保留	0x0	-
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 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

位	标记	功能描述	复位值	读写
1	HXTEN	<p>外部 4M~24M 晶振 HXTOSC 使能信号。</p> <p>0: 关闭</p> <p>1: 使能</p> <p>注意:</p> <p>1. 当系统进入 Deep Sleep, 此高速时钟会自动关闭。</p> <p>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)。</p> <p>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。</p>	0	R/W
0	HIRCEN	<p>内部高速时钟 HIRC 使能信号。</p> <p>0: 关闭</p> <p>1: 使能</p> <p>注意:</p> <p>1. 当系统进入 deep Sleep, 此高速时钟会自动关闭。</p> <p>2. 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。</p> <p>3. 当系统时钟选择该时钟时, 不能关闭</p>	1	R/W

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

### 7.4-9系统时钟源选择寄存器(RCC\_SYSCLKSEL)

地址偏移: 0x20 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										HIRCDIV[1:0]		CLKSW[3:0]			
--										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
15:4	保留	--	0x0	-
5:4	HIRCDIV[1:0]	HIRC 除频后时钟为 HIRCO 2' b00 : HIRCO = HIRC 2' b01 : HIRCO = HIRC/2 2' b10 : HIRCO = HIRC/3 2' b11 : HIRCO = HIRC/6	0x00	R/W
3:0	CLKSW[3:0]	System Clock Source Select 0001: HIRCO 选择 0010: HXT 选择 0100: LIRC 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。	0x1	R/W

Note : 该位写受 RCC\_UNLOCK 保护

#### 7.4-10 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		HIRCRDY		HIRCTRIM[11:0]											
--		RO		R/W											

此寄存器不受 Cortex-M0+ 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 0xA650 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	HIRCRDY	<p>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。</p>	1	RO
11:0	HIRCTRIM [11:0]	<p>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[11:0] 即可配置精准主频时钟。</p> <p><b>对于封装片(Package):</b></p> <ul style="list-style-type: none"> <li>(1) 24M 校准值地址: 0x1800_00A0</li> <li>(2) 22.12M 校准值地址: 0x1800_00A2</li> <li>(3) 16M 校准值地址: 0x1800_00A4</li> <li>(4) 8M 校准值地址: 0x1800_00A6</li> <li>(5) 4M 校准值地址: 0x1800_00A8</li> </ul> <p><b>对于裸片(KGD):</b></p> <ul style="list-style-type: none"> <li>(1) 24M 校准值地址: 0x1800_00C0</li> <li>(2) 22.12M 校准值地址: 0x1800_00C2</li> <li>(3) 16M 校准值地址: 0x1800_00C4</li> <li>(4) 8M 校准值地址: 0x1800_00C6</li> <li>(5) 4M 校准值地址: 0x1800_00C8</li> </ul>	0x312	R/W

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

#### 7.4-11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]		保留	HXTDRV[2:0]			
--								RO	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	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	HXTSTARTUP [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	保留	--	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。

#### 7.4-12 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 RDY	SIRC STARTUP[1:0]		保留	SIRCTRIM[8:0]									
--		RO	R/W		--	R/W									

此寄存器 LVD 和 Cortex-M0+ 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	SIRCSTARTUP [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	-	保留	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。 <b>对于封装片(Package):</b> (1) 32.768KHz 校准值地址: 0x1800_00B0 (2) 38.4KHz 校准值地址: 0x1800_00B4 <b>对于裸片(KGD):</b> (1) 32.768KHz 校准值地址: 0x1800_00D0 (2) 38.4KHz 校准值地址: 0x1800_00D4	0x07F	R/W

注意：只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。

#### 7.4-13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 PORT	LXT AON	LXT BYP	LXT EN	保留	LXT RDY	LXT STARTUP[1:0]		LXTDRV[3:0]					
--		R/W				--	RO	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: LSE 振荡器未被旁路 1: LSE 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	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
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。

#### 7.4-14 Cortex M0+ IRQ 延时控制寄存器(RCC\_IRQLATENCY)

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															
--															

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	IRQLATENCY[7:0]	IRQLATENCY[7:0] 具体使用方法参考 ARM Cortex M0+ 文档	0x0	R/W

注意：只有 RCC\_UNLOCK 寄存器保护解除后，才能写该寄存器。

### 7.4-15SysTick 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						NOREF	SKEW	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
31:26	保留	--	0x0	--
25	NOREF	<p>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 </p>	0	R/W
24	SKEW	<p>10ms STCALIB 值是否准确  0: 准确  1: 不准确 </p>	1	R/W
23:0	STCALIB [23:0]	<p>SysTick 10ms 校准值  此值为使用外部参考时钟HCLK/4(4MHz)的 10ms 校准值。</p>	0x009C3F	R/W

#### 7.4-16SWDIO 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															
--															

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
15:1	保留	--	0x0	--
0	SWDPOR	配置PC7 和PD1 的端子功能模式 0: 周边模块功能模式 1: SWD 端子功能	1	R/W

注意: 该位写受 RCC\_UNLOCK 保护

#### 7.4-17 周边模块复位控制寄存器(RCC\_PERIRST)

地址偏移: 0x44

复位值: 0x0000 00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMXRST[31:29]	CRC RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST	OPA12 RST	UART3 RST	UART2 RST	DBG RST	BEEP RST	LVDVC RST	SPI1 RST	CLKTRIM RST		
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1 RST	AWK RST	ADC RST	WWDG RST	TIM2 RST	TIM1 RST	OWRIE RST	PCA RST	SYSCON RST	BASETIM RST	LPTIM RST	SPI0 RST	LPUART RST	I2C0 RST	UART1 RST	UART0 RST
R/W															

位	标记	功能描述	复位值	读写																		
31:29	TIMXRST	TIM1A,TIM1B,TIM2A,TIM2B,TIM2C 的模块复位 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">TIMXRST[31:29]</td><td style="padding: 2px;">说明</td></tr> <tr> <td style="padding: 2px;">3' b000</td><td style="padding: 2px;">All Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b001</td><td style="padding: 2px;">TIM1A复位,其他Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b010</td><td style="padding: 2px;">TIM1B复位,其他Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b011</td><td style="padding: 2px;">TIM2A复位, 其他Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b100</td><td style="padding: 2px;">TIM2B复位, 其他Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b101</td><td style="padding: 2px;">TIM2C复位, 其他Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b110</td><td style="padding: 2px;">All Timer 正常工作</td></tr> <tr> <td style="padding: 2px;">3' b111</td><td style="padding: 2px;">All Timer 正常工作</td></tr> </table>	TIMXRST[31:29]	说明	3' b000	All Timer 正常工作	3' b001	TIM1A复位,其他Timer 正常工作	3' b010	TIM1B复位,其他Timer 正常工作	3' b011	TIM2A复位, 其他Timer 正常工作	3' b100	TIM2B复位, 其他Timer 正常工作	3' b101	TIM2C复位, 其他Timer 正常工作	3' b110	All Timer 正常工作	3' b111	All Timer 正常工作	3' b000	R/W
TIMXRST[31:29]	说明																					
3' b000	All Timer 正常工作																					
3' b001	TIM1A复位,其他Timer 正常工作																					
3' b010	TIM1B复位,其他Timer 正常工作																					
3' b011	TIM2A复位, 其他Timer 正常工作																					
3' b100	TIM2B复位, 其他Timer 正常工作																					
3' b101	TIM2C复位, 其他Timer 正常工作																					
3' b110	All Timer 正常工作																					
3' b111	All Timer 正常工作																					
28	CRCRST	CRC 模块复位 0: 正常工作 1: 复位	0	R/W																		
27	GPIODRST	GPIOD 模块复位 0: 正常工作 1: 复位	0	R/W																		
26	GPIOCRST	GPIOC 模块复位 0: 正常工作 1: 复位	0	R/W																		
25	GPIOBRST	GPIOB 模块复位 0: 正常工作 1: 复位	0	R/W																		

24	GPIOARST	GPIOA 模块复位 0: 正常工作 1: 复位	0	R/W
23	OPA12RST	OPA1/2 模块复位 0: 正常工作 1: 复位	0	R/W
22	UART3RST	UART3 模块复位 0: 正常工作 1: 复位	0	R/W
21	UART2RST	UART2 模块复位 0: 正常工作 1: 复位	0	R/W
20	DBGRST	MCU DEBUG 模块复位 0: 正常工作 1: 复位	0	R/W
19	BEEPRST	BEEP 模块复位 0: 正常工作 1: 复位	0	R/W
18	LVDVCRST	LVD 模块复位 0: 正常工作 1: 复位	0	R/W
17	SPI/I2SRST	SPI/I2S 模块复位 0: 正常工作 1: 复位	0	R/W
16	CLKTRIMRST	Clock TRIM 模块复位 0: 正常工作 1: 复位	0	R/W
15	I2C1RST	I2C1 模块复位 0: 正常工作 1: 复位	0	R/W
14	AWKRST	AWK 模块复位 0: 正常工作 1: 复位	0	R/W
13	ADCRST	ADC 模块复位 0: 正常工作 1: 复位	0	R/W
12	WWDGRST	WWDG 模块复位 0: 正常工作 1: 复位	0	R/W

11	TIM2RST	TIM2 模块复位 0: 正常工作 1: 复位	0	R/W
10	TIM1RST	TIM1 模块复位 0: 正常工作 1: 复位	0	R/W
9	OWIREST	1-Wire 模块复位 0: 正常工作 1: 复位	0	R/W
8	PCARST	PCA 模块复位 0: 正常工作 1: 复位	0	R/W
7	SYSCONRST	SYSCON 模块复位 1: 复位 0: 正常工作	0	R/W
6	BASETIMRST	Base Timer10/11 复位 0: 正常工作 1: 复位	0	R/W
5	LPTIMRST	Low Power Timer0/1 复位 1: 复位 0: 正常工作	0	R/W
4	QSPIRST	QSPI 模块复位 0: 正常工作 1: 复位	0	R/W
3	LPUARTRST	LPUART 模块复位 0: 正常工作 1: 复位	0	R/W
2	I2C0RST	I2C0 模块复位 0: 正常工作 1: 复位	0	R/W
1	UART1RST	UART1 模块复位 0: 正常工作 1: 复位	0	R/W
0	UART0RST	UART0 模块复位 0: 正常工作 1: 复位	0	R/W

#### 7.4-18RTC 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															
--															

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	RTCRST	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。

### 7.4-19MASK 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。

### 7.4-20LPM 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															
--															

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_HANDLE	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。

### 7.4-21周边模块复位控制 1 寄存器(RCC\_PERIRST1)

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															
--															

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

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
2	TRNGRST	TRNG 模块复位 0: 正常工作 1: 复位	0	R/W
1	AESRST	AES 模块复位 0: 正常工作 1: 复位	0	R/W
0	DMARST	DMA 模块复位 0: 正常工作 1: 复位	0	R/W

### 7.4-22PLL 控制寄存器 1(RCC\_PLLCR1)

地址偏移: 0x58

复位值: 0x0000 9210

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]							保留	PLLSYS [1:0]	PLLSTARTUP [1:0]		PLL RDY	保留			
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，其它值无效。	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.HIRCSTRIM[2:0]即可配置精准主频时钟。 对于封装片(Package): 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	保留	--	0	--
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 128M~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	保留	--	0	--

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

### 7.4-23PLL 控制寄存器 2(RCC\_PLLCR2)

地址偏移: 0x5C

复位值: 0x0000 1A00

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		LDO_PLL_TR [1:0]	PLL_M[6:0]						PLL_N[1:0]			SELRCLK [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																								
15:13	保留	--	0	--																								
12:11	LDO_PLL_TR [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' b11	R/W																								
10:4	PLL_M[6:0]	PLL_M参考选择: 建议 PLL_M = 32 <table border="1" data-bbox="473 1170 1111 1500"> <tr> <th>RCLK (HIRC/ XTAL/OSC)</th> <th>PLL_N [1:0]</th> <th>M [6:0]</th> <th>PLL_O =RCLK*(M/N)</th> </tr> <tr> <td>24MHz</td> <td>2' b11(6)</td> <td>7' d32</td> <td>128MHz</td> </tr> <tr> <td>22.118MHz</td> <td>2' b11(6)</td> <td>7' d35</td> <td>129.02 MHz</td> </tr> <tr> <td>4MHz</td> <td>2' b00(1)</td> <td>7' d32</td> <td>128MHz</td> </tr> <tr> <td>24MHz</td> <td>2' b11(6)</td> <td>7' d48</td> <td>192MHz</td> </tr> <tr> <td>22.118MHz</td> <td>2' b11(6)</td> <td>7' d52</td> <td>191.69 MHz</td> </tr> </table>	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	7' b0100000 (7' d32)	R/W
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																									
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' b00	R/W																								
1:0	SELRCLK [1:0]	PLL参考时钟(RCLK)选择: 00: HIRC 01: XTAL 10: OSC(XIN) 11: 保留	0x0	R/W																								

Note: 只有 RCC\_UNLOCK 寄存器保护解除后, 才能写该寄存

#### 7.4-24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
KEY															UNLOCK
WO															R/W

位	标记	功能描述	复位值	读写
31:1	Key	只有高位写 0x2AD5334C 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。

## 【8】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

### 8.1 寄存器列表

SYSCON Address : 0x4000 1C00

偏移地址	名称	描述	复位值
0x00	SYSCON_CFGR0	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	TIM2 通道输入源选择	0x0000 0000
0x18	SYSCON_TIM1ACR	TIM1A 通道输入源选择	0x0000 0000
0x1C	SYSCON_TIM1BCR	TIM1B 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
0x100	SYSCON_CIDL	CHIP ID Low word	0x0000 1000
0x104	SYSCON_CIDH	CHIP ID High word	0x0000 0005
0x230	SYSCON_TST_REG0	只有power on (por) 可以reset	0x0000 0000
0x234	SYSCON_TST_REG1	只有power on (por) 可以reset	0x0000 0000
0x238	SYSCON_TST_REG2	只有power on (por) 可以reset	0x0000 0000
0x3C0	SYSCON_BOOT	BOOT状态寄存器	0x0000 0000

Table 8- 1 SYSCON 寄存器列表和复位值

## 8.2寄存器说明

### 8.2-1系统配置寄存器 0(SYSCON\_CFRG0)

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 _IREN		VDS _SEL		DBGDLSP _DIS		LOCKUP EN	
--							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 10ns TREEDLY[2] 0 : Pulse End at 12ns 1 : Pulse End at 11ns	0x0	R/W
3	VD12_IREN	1: LDO Inrush 补偿 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	DBGDLSP_DIS	Debug 模式, 进入Deep Sleep 模式禁止控制位 0: 允许在Debug 模式进入Deep Sleep 1: 不允许在Debug 模式进入Deep Sleep	0	R/W
0	LOCKUPEN	Cortex-M0+ LookUp 功能使能 0: 关闭 1: 使能 注: 当 Cortex-M0+读取错误的指令时, MCU 会复位, 以增强系统可靠性。	0	R/W

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

### 8.2-2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										RET1N EN	PADDLSP CON	PADINT SEL			
--										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。

### 8.2-3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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					LPTIM_GATE_SEL [1:0]		TIM11_GATE_SEL [1:0]		TIM10_GATE_SEL [1:0]		SPI0SSN_SEL [3:0]				
R/W															

位	标记	功能描述	复位值	读写
31:15	保留	--	0	--
14	保留	必须写入 0。	0	R/W
13:10	保留	--	0	R/W
9:8	LPTIM_GATE_SEL[1:0]	Low Power Timer 门控输入信号来源选择 00: LPTIM_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
7:6	TIM11_GATE_SEL[1:0]	Timer11 门控输入信号来源选择 00: TIM11_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
5:4	TIM10_GATE_SEL[1:0]	Timer10 门控输入信号来源选择 00: TIM10_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W

位	标记	功能描述	复位值	读写																																		
3:0	SPISSN_SEL[3:0]	<p>SPI/I2S Slave SSN 信号来源选择:</p> <table border="1"><thead><tr><th>SPISSN_SEL[3:0]</th><th>SSN</th></tr></thead><tbody><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></tbody></table>	SPISSN_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
SPISSN_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																																					

### 8.2-4PCA 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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				PCA_CAP4_SEL [1:0]		PCA_CAP3_SEL [1:0]		PCA_CAP2_SEL [1:0]		PCA_CAP1_SEL [1:0]		PCA_CAP0_SEL [1:0]			
--				R/W											

位	标记	功能描述	复位值	读写
31:10	保留	--	0x0	--
9:8	PCA_CAP4_SEL[1:0]	PCA 捕获通道 4 信号来源选择 00: PCA_CH4 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
7:6	PCA_CAP3_SEL[1:0]	PCA 捕获通道 3 信号来源选择 00: PCA_CH3 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
5:4	PCA_CAP2_SEL[1:0]	PCA 捕获通道 2 信号来源选择 00: PCA_CH2 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
3:2	PCA_CAP1_SEL[1:0]	PCA 捕获通道 1 信号来源选择 00: PCA_CH1 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
1:0	PCA_CAP0_SEL[1:0]	PCA 捕获通道 0 信号来源选择 00: PCA_CH0 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W

### 8.2-5TIM1/1A/1B 通道输入源选择(SYSCon\_TIM1xCR)

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
保留									TIM1x CLKFAILBRK EN	TIM1x DSLPBRK EN	TIM1x BRKOUT CFG	TIM1x ETR_SEL[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	
保留	TIMXCH4IN_SEL [2:0]				TIMXCH3IN_SEL [2:0]				TIMXCH2IN_SEL [2:0]				TIMXCH1IN_SEL [2:0]			
--	R/W				--				R/W				--			

位	标记	功能描述	复位值	读写																																
31:23	-	保留	0x0	-																																
22	TIM1x CLKFAILBRKEN	系统时钟停止检出时TIMX Break 使能 0: 无效 1: 使能	0	R/W																																
21	TIM1x DSLPBRKEN	Deep Sleep 模式时TIMX Break 使能 0: 无效 1: 使能	0	R/W																																
20	TIM1x BRKOUTCFG	0: ocxp/ocxnp 在break 模式的输出由 TIMX控制	0	R/W																																
19:16	TIM1x ETR_SEL[3:0]	TIMX ETR 信号来源选择 <table border="1"> <tr><td>0000</td><td>固定电平</td></tr> <tr><td>0001</td><td>PA01</td></tr> <tr><td>0010</td><td>PA02</td></tr> <tr><td>0011</td><td>PA03</td></tr> <tr><td>0100</td><td>PB04</td></tr> <tr><td>0101</td><td>PB05</td></tr> <tr><td>0110</td><td>PA00</td></tr> <tr><td>0111</td><td>PD00</td></tr> <tr><td>1000</td><td>PC05</td></tr> <tr><td>1001</td><td>PC06</td></tr> <tr><td>1010</td><td>PC07</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	固定电平	0001	PA01	0010	PA02	0011	PA03	0100	PB04	0101	PB05	0110	PA00	0111	PD00	1000	PC05	1001	PC06	1010	PC07	1011	PD01	1100	PA03	1101	PA04	1110	PA05	1111	PA06	0x0	R/W
0000	固定电平																																			
0001	PA01																																			
0010	PA02																																			
0011	PA03																																			
0100	PB04																																			
0101	PB05																																			
0110	PA00																																			
0111	PD00																																			
1000	PC05																																			
1001	PC06																																			
1010	PC07																																			
1011	PD01																																			
1100	PA03																																			
1101	PA04																																			
1110	PA05																																			
1111	PA06																																			

15	-	保留	0	-
14:12	TIM1x CH4IN_SEL[2:0]	<p>TIMX CH4 输入通道信号来源选择</p> <p>000 : TIMX_CH4 001 : UART0_RXD 010 : UART1_RXD 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) 111 : PA14(可连接霍尔传感器)</p>	0x0	R/W
11	-	保留	0	-
10:8	TIM1x CH3IN_SEL[2:0]	<p>TIMX CH3 输入通道信号来源选择</p> <p>000: TIMX_CH3 001: UART0_RXD 010: UART1_RXD 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)</p>	0x0	R/W
7	-	保留	0	-
6:4	TIM1x CH2IN_SEL[2:0]	<p>TIMX CH2 输入通道信号来源选择</p> <p>000: TIMX_CH2 001: UART0_RXD 010: UART1_RXD 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)</p>	0x0	R/W
3	-	保留	0	-
2:0	TIM1x CH1IN_SEL[2:0]	<p>TIMX CH1 输入通道信号来源选择</p> <p>000: TIMX_CH1 001: UART0_RXD 010: UART1_RXD 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)</p>	0x0	R/W

### 8.2-6TIM2/2A/2B/2C 通道输入源选择(SYSCON\_TIM2XCR)

SYSCON\_TIM2CR：地址偏移： 0x14

复位值： 0x0000 0000

SYSCON\_TIM2ACR：地址偏移： 0x20

复位值： 0x0000 0000

SYSCON\_TIM2BCR：地址偏移： 0x24

复位值： 0x0000 0000

SYSCON\_TIM2CCR：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														TIM2x ETR_SEL[3:0]	
--														R/W	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIM2x CH4IN_SEL[2:0]		保留	TIM2x CH3IN_SEL[2:0]		保留	TIM2x CH2IN_SEL[2:0]		保留	TIM2x CH1IN_SEL[2:0]		保留	TIM2x CH1IN_SEL[2:0]		
--	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	TIM2x ETR_SEL[3:0]	TIM2X ETR信号来源选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>0000</td><td>固定电平</td></tr> <tr><td>0001</td><td>PA01</td></tr> <tr><td>0010</td><td>PA02</td></tr> <tr><td>0011</td><td>PA03</td></tr> <tr><td>0100</td><td>PB04</td></tr> <tr><td>0101</td><td>PB05</td></tr> <tr><td>0110</td><td>PA00</td></tr> <tr><td>0111</td><td>PD00</td></tr> <tr><td>1000</td><td>PC05</td></tr> <tr><td>1001</td><td>PC06</td></tr> <tr><td>1010</td><td>PC07</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	固定电平	0001	PA01	0010	PA02	0011	PA03	0100	PB04	0101	PB05	0110	PA00	0111	PD00	1000	PC05	1001	PC06	1010	PC07	1011	PD01	1100	PA03	1101	PA04	1110	PA05	1111	PA06	0x0	R/W
0000	固定电平																																			
0001	PA01																																			
0010	PA02																																			
0011	PA03																																			
0100	PB04																																			
0101	PB05																																			
0110	PA00																																			
0111	PD00																																			
1000	PC05																																			
1001	PC06																																			
1010	PC07																																			
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	TIM2x CH4IN_SEL[2:0]	TIM2X CH4 输入通道信号来源选择 000: TIMX_CH4 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: SIRC 101: PB13 110: PC14 111: PB02	0x0	R/W
11	保留	--	0	--
10:8	TIM2x CH3IN_SEL[2:0]	TIM2X CH3 输入通道信号来源选择 000: TIMX_CH3 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: SIRC 101: PB12 110: PC15 111: PB01	0x0	R/W
7	保留	--	0	--
6:4	TIM2x CH2IN_SEL[2:0]	TIM2X CH2 输入通道信号来源选择 000: TIMX_CH2 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: SIRC 101: PB11 110: PB03 111: PB00	0x0	R/W
3	保留	--	0	--
2:0	TIM2x CH1IN_SEL[2:0]	TIM2X CH1 输入通道信号来源选择 000: TIMX_CH1 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: SIRC 101: PB10 110: PA15 111: PA07	0x0	R/W

### 8.2-7SYS CON 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 0x2AD5334C 时配置该寄存器才有效，写其它值时无效。	0x0	WO
0	UNLOCK	0: 保护有效 1: 解除保护	0	R/W

Note : 写 0x55AA6699, 解除保护。

### 8.2-8SYS CON CIDL 寄存器(SYSCON\_CIDL)

地址偏移: 0x100

复位值: 0x0000 1000

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

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

位	标记	功能描述	复位值	读写
31:0	CIDL	固定值 0x00001000。	0x00001000	RO

### 8.2-9SYS CON CIDH 寄存器(SYSCON\_CIDH)

地址偏移: 0x104

复位值: 0x0000 0002

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

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

位	标记	功能描述	复位值	读写
31:0	CIDH	固定值 0x00000002。	0x00000002	RO

### 8.2-10SYS CON TST0 寄存器(SYSCON\_TST\_REG0)

地址偏移: 0x230

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TST_REG0[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TST_REG0[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	TST_REG0	只有power on (por) 可以reset。	0x00	R/W

### 8.2-11SYS CON TST1 寄存器(SYSCON\_TST\_REG1)

地址偏移: 0x234

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TST_REG1[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TST_REG1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	TST_REG1	只有power on (por) 可以reset。	0x00	R/W

### 8.2-12SYS CON TST2 寄存器(SYSCON\_TST\_REG2)

地址偏移: 0x238

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TST_REG2[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TST_REG2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	TST_REG2	只有power on (por) 可以reset。	0x00	R/W

### 8.2-13SYS CON BOOT 状态寄存器(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
保留															
--															

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

位	标记	功能描述	复位值	读写
31:3	保留	禁用	0	--
2	BOOT1ST	{BOOT1ST, BOOT0ST}： 2' b00 : 主要程序从eflash 开始,映射位址: 0x0000_0000~0x0000_3FFF (64KB) 2' b01 : 主要程序从SRAM 开始, 映射位址: 0x0000_0000~0x0000_0BFF (4KB) 2' b10 : 主要程序从eflash 开始,eflash 映射位址: 0x0000_0000~0x0000_3FFF (64KB) 2' b11 : 主要程序从external Flash(QSPI) 開始, 映射位址: 0x0000_0000~0x07FF_FFFF (128MB)	0	RO
1	BOOT0ST		0	RO
0	保留	禁用	--	--

## 【9】中断控制器(NVIC)

### 9.1概述

Cortex®-M0+提供中断控制器，用于总体管理中断和异常，称之为“嵌套向量中断控制器(NVIC)”。

NVIC 和处理器内核紧密相连，可以实现低延迟的中断处理和高效的晚到中断处理。

NVIC 支持最多 32 个中断请求(IRQ)输入，以及 1 个不可屏蔽中断(NMI)输入(在本产品系统中并未使用)。另外，处理器还支持多个内部异常。

本章节只对处理器的 32 个外部中断请求(中断 0 到中断 31)做详细介绍，处理器内部异常的具体情况可参考其他相关文档。详情请参考“ARM® Cortex®-M0+ Technical Reference Manual”与“ARM® v6-M Architecture Reference Manual”。

### 9.2特征

1. 32 个可屏蔽中断通道(不包含 16 个 Cortex®-M0+的中断线)
2. 4 个可编程的优先级(使用了 2 位的中断优先级)
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

### 9.3中断优先级

软件可以对外部中断设置 4 级优先级。最高优先级为“0”，最低优先级为“3”，所有用户可配置的优先级的默认值为“0”。

如果处理器正在运行一个中断处理，而新中断的优先级大于正在运行的，这时就会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。

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

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

## 9.4 中断向量表

Cortex®-M0+响应中断时，处理器自动从存储器的中断向量表中取出中断服务例程(ISR)的起始地址。

向量表包括复位后堆栈(MSP)的初始值以及所有异常处理的入口地址。中断号表示处理异常的先后次序。

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

**Table 9 - 1 中断向量表**

中断号	外部中断号 (IRQ#)	优先级	优先级类型	中断源	说明	地址
0	-	-	-	-	MSP 初始值	0x0000 0000
1		-3	固定	Reset	复位向量(RESET)	0x0000 0004
2		-2	固定	NMI	不可屏蔽中断	0x0000 0008
3		-1	固定	HardFault	所有类型的错误(fault)	0x0000 000C
4-10	-	-	-	-	保留	0x0000 000C- 0x0000 002B
11			可配置	SVCall	通用 SWI 指令 调用的系统服务	0x0000 002C
12-13	-	-	-	-	保留	0x0000 0030- 0x0000 0037
14			可配置	PendSV	可挂起的系统服务	0x0000 0038
15			可配置	SysTick	系统滴答定时器	0x0000 003C
16	0		可配置	GPIO_PA	GPIOA INT	0x0000 0040
17	1		可配置	GPIO_PB	GPIOB INT	0x0000 0044
18	2		可配置	GPIO_PC	GPIOC INT	0x0000 0048
19	3		可配置	GPIO_PD	GPIOD INT	0x0000 004C
20	4		可配置	FLASH	FLASH INT	0x0000 0050
21	5		可配置	PDMA	PMA INT	0x0000 0054
22	6		可配置	UART0	UART0 INT	0x0000 0058
23	7		可配置	UART1	UART1 INT	0x0000 005C
24	8		可配置	LPUART	LPUART INT	0x0000 0060
25	9		可配置	USART2,3	USART2,3 INT	0x0000 0064
26	10		可配置	SPI/I2S QSPI	SPI/I2S QSPI INT	0x0000 0068
27	11		可配置	TIM2C	TIM2C INT	0x0000 006C
28	12		可配置	I2C0/I2C1	I2C0/I2C1 INT	0x0000 0070
29	13		可配置	TIM1B	TIM1B INT	0x0000 0074
30	14		可配置	TIM10	TIM10 INT	0x0000 0078

31	15		可配置	TIM11	TIM11 INT	0x0000 007C
32	16		可配置	LPTIM	LPTIM INT	0x0000 0080
33	17		可配置	TIM1A	TIM1A INT	0x0000 0084
34	18		可配置	TIM1	TIM1 INT	0x0000 0088
35	19		可配置	TIM2	TIM2 INT	0x0000 008C
36	20		可配置	TIM2A	TIM2A INT	0x0000 0090
37	21		可配置	PCA	PCA INT	0x0000 0094
38	22		可配置	WWDG	WWDG INT	0x0000 0098
39	23		可配置	IWDG	IWDG INT	0x0000 009C
40	24		可配置	ADC	ADC INT	0x0000 00A0
41	25		可配置	LVD	LVD INT	0x0000 00A4
42	26		可配置	VC0/VC1	VC0/VC1 INT	0x0000 00A8
43	27		可配置	TIM2B	TIM2B INT	0x0000 00AC
44	28		可配置	AWK	AWK INT	0x0000 00B0
45	29		可配置	ONEWIRE /AES	ONEWIRE INT/AES INT	0x0000 00B4
46	30		可配置	RTC	RTC INT	0x0000 00B8
47	31		可配置	CLKTRIM /TRNG	CLKTRIM INT/TRNG INT/	0x0000 00BC

## 9.5 中断唤醒控制 WIC

当处理器利用退出休眠(SLEEP-ON-EXIT)特性或者执行 WFI 指令进入休眠之后，就会停止指令执行，并且当发生了中断请求(更高优先级)并且需要处理时，处理器就会被唤醒。

WFI 行为	唤醒	ISR 执行
PRIMASK 清除		
IRQ 优先级 > 当前等级	Y	Y
IRQ 优先级 ≡ 当前等级	N	N
PRIMASK 置位(中断禁止)		
IRQ 优先级 > 当前等级	Y	N
IRQ 优先级 ≡ 当前等级	N	N

### 9.5-1 NVIC 从深度休眠模式唤醒设置进入中断 ISR 设置

1. 使能深度休眠需要唤醒的中断 NVIC
2. 使能模块中断使能
3. 设置 SCR.DEEPSLEEP 为 1
4. 使用 WFI 指令进入深度休眠模式
5. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序

例程：

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

### 9.5-2 NVIC 从深度休眠模式唤醒设置不执行中断 ISR 设置

1. 使能深度休眠需要唤醒的中断 NVIC
2. 使用 PRIMASK 寄存器屏蔽中断
3. 使能模块中断使能
4. 设置 SCR.DEEPSLEEP 为 1
5. 使用 WFI 指令进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令
7. 清除中断标志，清除中断挂起状态

### 9.5-3 使用退出休眠特性

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

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

1. 使能深度休眠需要唤醒的中断 NVIC
2. 使能模块中断使能
3. 设置 SCR.DEEPSLEEP 为 1
4. 设置 SCR.SLEEPONEXIT 为 1
5. 使用 WFI 指令进入深度休眠模式
6. 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序
7. 退出中断服务时自动进入休眠模式

例程：

```
SCR |= 0x00000004u;  
SCR |= 0x00000002u;  
_asm("nop");  
_asm("nop");  
_asm("nop");  
while(1){  
    _asm("WFI");  
}
```

## 9.6 软件基本操作

### 9.6-1 外部中断使能

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

### 9.6-2 NVIC 中断使能和清除使能

Cortex-M0+处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零位。这样就有了 32 位的中断使能置位寄存器 NVIC\_ISER 和 32 位的中断使能清零寄存器 NVIC\_ICER。如果想使能某一个中断，则对 NVIC\_ISER 寄存器的相应位置 1。如果想清零某一个中断使能，则对 NVIC\_ICER 寄存器的相应位置 1。

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

### 9.6-3 NVIC 中断挂起和清除挂起

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

当处理器开始进入中断处理，硬件会自动清除相应的中断挂起状态。

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

### 9.6-4 NVIC 中断优先级

设置 NVIC\_IPR0- NVIC\_IPR7 寄存器决定 IRQ0-IRQ32 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优先级，这种情况的结果不可预知，并且不被 Cortex-M0+ 处理器支持。

## 9.6-5NVIC 中断屏蔽

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

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

```
void _enable_irq(void); //清除 PRIMASK  
void _disable_irq(void); //设置 PRIMASK
```

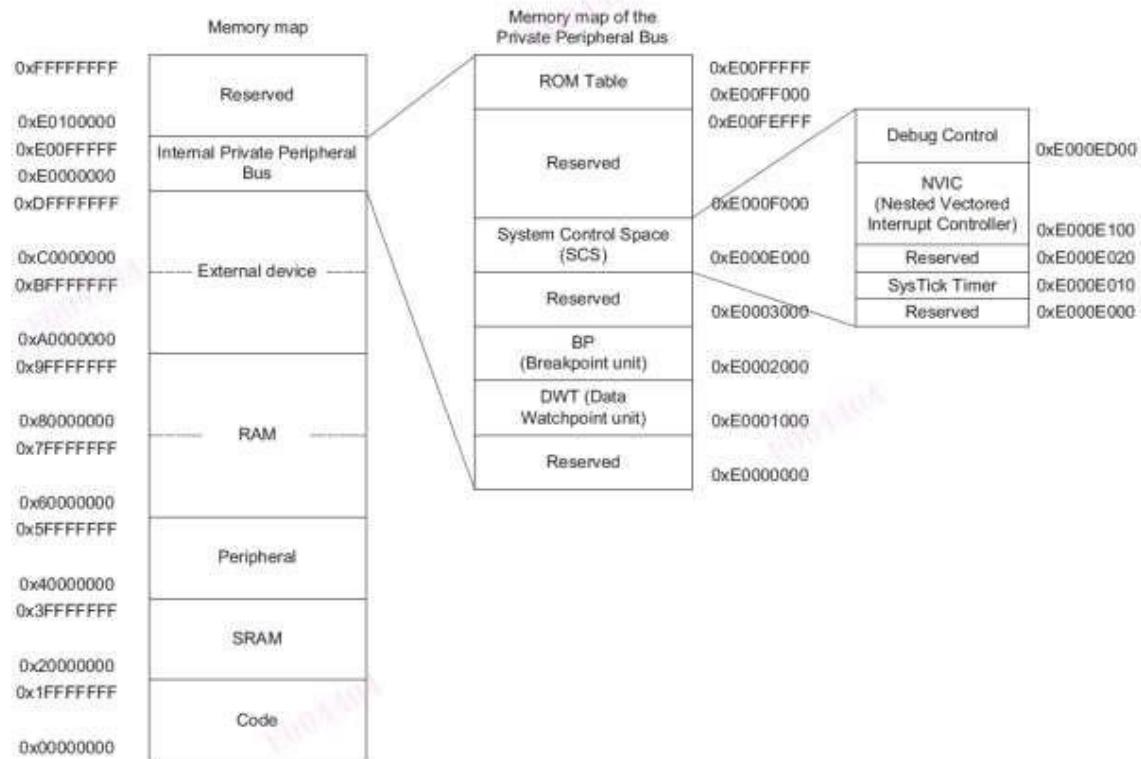
## 9.7 寄存器列表

SCS 基地址: 0xE000 E000

偏移地址	名称	描述	复位值
0x100	NVIC_ISER	IRQ0~IRQ31 使能设置控制寄存器	0x0000 0000
0x180	NVIC_ICER	IRQ0~IRQ31 使能清除控制寄存器	0x0000 0000
0x200	NVIC_ISPR	IRQ0~IRQ31 挂起设置控制寄存器	0x0000 0000
0x280	NVIC_ICPR	IRQ0~IRQ31 挂起清除控制寄存器	0x0000 0000
0x400	NVIC_IPR0	IRQ0~IRQ3 优先级控制寄存器 0	0x0000 0000
0x404	NVIC_IPR1	IRQ4~IRQ7 优先级控制寄存器 1	0x0000 0000
0x408	NVIC_IPR2	IRQ8~IRQ11 优先级控制寄存器 2	0x0000 0000
0x40C	NVIC_IPR3	IRQ12~IRQ15 优先级控制寄存器 3	0x0000 0000
0x410	NVIC_IPR4	IRQ16~IRQ19 优先级控制寄存器 4	0x0000 0000
0x414	NVIC_IPR5	IRQ20~IRQ23 优先级控制寄存器 5	0x0000 0000
0x418	NVIC_IPR6	IRQ24~IRQ27 优先级控制寄存器 6	0x0000 0000
0x41C	NVIC_IPR7	IRQ28~IRQ31 优先级控制寄存器 7	0x0000 0000

Table 9 - 2 NVIC 寄存器列表和复位值

Note : NVIC 寄存器是 SCS 寄存器的一部分，SCS 基地址是 0xE000E000，如下图所示：



## 9.8寄存器说明

### 9.8-1中断使能设置寄存器(NVIC\_ISER)

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
SETENA[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	SETENA	设置使能外部中断 0 到中断 31; 写 “1” 置位，写 “0” 无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 ..... [31]: IRQ31	0x0	R/W

### 9.8-2 中断使能清除寄存器(NVIC\_ICER)

地址偏移: 0x180 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:0	CLRENA	清除使能外部中断 0 到中断 31; 写 “1” 清除, 写 “0” 无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 ..... [31]: IRQ31	0x0	R/W

### 9.8-3 中断挂起设置寄存器(NVIC\_ISPR)

地址偏移: 0x200 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:0	SETPEND	设置外部中断 0 到中断 31 的挂起状态; 写 “1” 置位, 写 “0” 无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 ..... [31]: IRQ31	0x0	R/W

### 9.8-4 中断挂起清除寄存器(NVIC\_ICPR)

地址偏移: 0x280 复位值: 0x0000 0000

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

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

位	标记	功能描述	复位值	读写
31:0	CLRPEND	清除外部中断 0 到中断 31 的挂起状态; 写 “1” 清零, 写 “0” 无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 ..... [31]: IRQ31	0x0	R/W

### 9.8-5 中断优先级控制寄存器 0(NVIC\_IPR0)

地址偏移: 0x400      复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR0[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR0[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR0	中断 0 到中断 3 的优先级; [31:30]: 中断 3 的优先级 [23:22]: 中断 2 的优先级 [15:14]: 中断 1 的优先级 [7:6]: 中断 0 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-6 中断优先级控制寄存器 1(NVIC\_IPR1)

地址偏移: 0x404      复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR1[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR1	中断 4 到中断 7 的优先级; [31:30]: 中断 7 的优先级 [23:22]: 中断 6 的优先级 [15:14]: 中断 5 的优先级 [7:6]: 中断 4 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-7 中断优先级控制寄存器 2(NVIC\_IPR2)

地址偏移: 0x408

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR2[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR2	中断 8 到中断 11 的优先级; [31:30]: 中断 11 的优先级 [23:22]: 中断 10 的优先级 [15:14]: 中断 9 的优先级 [7:6]: 中断 8 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-8 中断优先级控制寄存器 3(NVIC\_IPR3)

地址偏移: 0x40C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR3[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR3	中断 12 到中断 15 的优先级; [31:30]: 中断 15 的优先级 [23:22]: 中断 14 的优先级 [15:14]: 中断 13 的优先级 [7:6]: 中断 12 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-9 中断优先级控制寄存器 4(NVIC\_IPR4)

地址偏移: 0x410      复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR4[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR4[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR4	中断 16 到中断 19 的优先级; [31:30]: 中断 19 的优先级 [23:22]: 中断 18 的优先级 [15:14]: 中断 17 的优先级 [7:6]: 中断 16 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-10 中断优先级控制寄存器 5(NVIC\_IPR5)

地址偏移: 0x414      复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR5[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR5[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR5	中断 20 到中断 23 的优先级; [31:30]: 中断 23 的优先级 [23:22]: 中断 22 的优先级 [15:14]: 中断 21 的优先级 [7:6]: 中断 20 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-11 中断优先级控制寄存器 6(NVIC\_IPR6)

地址偏移: 0x418 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR6[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR6[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR6	中断 24 到中断 27 的优先级; [31:30]: 中断 27 的优先级 [23:22]: 中断 26 的优先级 [15:14]: 中断 25 的优先级 [7:6]: 中断 24 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

### 9.8-12 中断优先级控制寄存器 7(NVIC\_IPR7)

地址偏移: 0x41C 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR7	中断 28 到中断 31 的优先级; [31:30]: 中断 31 的优先级 [23:22]: 中断 30 的优先级 [15:14]: 中断 29 的优先级 [7:6]: 中断 28 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

## 【10】通用输入输出口(GPIO)

### 10.1 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(GPIOx\_AFR)配置。

每个通用 I/O 口都有 5 个配置寄存器(GPIOx\_DIRCR、GPIOx\_OTYPER、GPIOx\_PUPDR、GPIOx\_SLEWCR 和 GPIOx\_DRVCR)，2 个数据寄存器(GPIOx\_IDR 和 GPIOx\_ODR)，1 个输出置位寄存器(GPIOx\_ODSET)，1 个输出复位寄存器(GPIOx\_ODCLR)和 1 个复用功能寄存器(GPIOx\_AFR)。

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”。

所有端口都可以提供外部中断，并且每个中断都可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发或者任意边沿触发，支持边沿模式下的输入抖动。支持在工作模式/睡眠模式/深度睡眠模式下产生中断。

### 10.2 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(GPIOx\_ODR)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(GPIOx\_IDR)或外设输入数据(复用功能输出)
6. 输出置位/复位寄存器(GPIOx\_ODSET, GPIOx\_ODCLR)为对 GPIOx\_ODR 寄存器提供位改写能力
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

### 10.3 GPIO 功能描述

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 10-1 给出了一个标准 I/O 端口的基本结构。

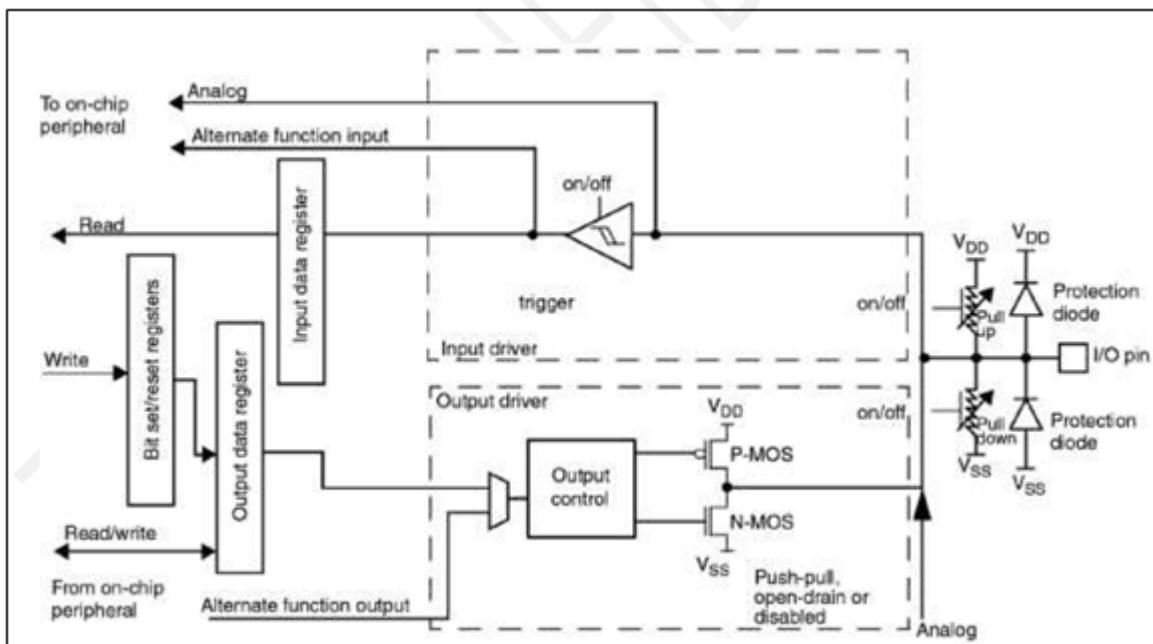


Figure 10 - 1 标准 I/O 端口的位基本结构

表 10-1 给出了 GPIO 口和外设引脚的复用功能映射

AFRi[3:0]	DIRi	OTYPi	DRVi	PUPDi		IO Configuration	
0000	1	0	DRVi	0	0	GPIO output	PP
		0		0	1	GPIO output	PP+PU
		0		1	0	GPIO output	PP+PD
		0		1	1	Reserved	
		1		0	0	GPIO output	OD
		1		0	1	GPIO output	OD+PU
		1		1	0	GPIO output	OD+PD
		1		1	1	Reserved(output OD)	
	0	x		x	0	Input	Floating
		x		x	0	Input	PU
		x		x	1	Input	PD
		x		x	1	Reserved(input floating)	
0001~1110	x	0	DRVi	0	0	AF	PP
	x	0		0	1	AF	PP+PU
	x	0		1	0	AF	PP+PD
	x	0		1	1	Reserved	
	x	1		0	0	AF	OD
	x	1		0	1	AF	OD+PU
	x	1		1	0	AF	OD+PD
	x	1		1	1	Reserved	
	x	x		x	0	Input/output	Analog
	x	x		x	0	Forbidden	Forbidden
1111	x	x		x	1		
	x	x		x	1		

Table 10 - 1 端口位配置表

i = 0...15

GP=通用， PP=推挽输出， PU=上拉， PD=下拉， OD=开漏， AF=复用功能。

### 10.3-1通用 I/O(GPIO)

复位期间和复位后，复用功能未开启，并且除了调试引脚以外的所有 I/O 端口都被配置为浮空输入模式。复位后，调试引脚被置为复用功能的上拉/下拉模式：

- PD01: SWDCLK 置于下拉模式
- PC07: SWDIO 置于上拉模式

当作为输出配置时，写到输出数据寄存器(GPIOx\_ODR)的值输出到相应的引脚上。可以以推挽模式或开漏模式(仅低电平被驱动，高电平表现为高阻)输出。

输入数据寄存器(GPIOx\_IDR)在每个 AHB 时钟周期捕捉 I/O 引脚上的数据。

所有 GPIO 引脚都有一个内部弱上拉和弱下拉电阻，它们被激活或断开依赖于 GPIOx\_PUPD 寄存器的值。

### 10.3-2I/O 端口控制寄存器

每个通用 I/O 口都有 5 个控制寄存器(GPIOx\_DIRCR、GPIOx\_OTYPER、GPIOx\_PUPDR、GPIOx\_SLEWCR 和 GPIOx\_DRVCR)用来配置 I/O 口。GPIOx\_DIRCR 寄存器用来选择方向(输入/输出)。GPIOx\_OTYPER、GPIOx\_SLEWCR 和 GPIOx\_DRVCR 寄存器来选择输出类型(推挽或开漏)、电压转换速率和驱动强度。GPIOx\_PUPDR 寄存器用来选择上拉/下拉方式。

### 10.3-3I/O 端口控制寄存器

每个 GPIO 口有两个数据寄存器：输入和输出数据寄存器(GPIOx\_IDR 和 GPIOx\_ODR)。GPIOx\_ODR 用于存储输出数据，其可进行读/写访问。从 I/O 线的输入数据存放在(GPIOx\_IDR)寄存器中，该寄存器为只读寄存器。

### 10.3-4 I/O 数据位处理

输出置位寄存器(GPIOx\_ODSET)和输出清零寄存器(GPIOx\_ODCLR)寄存器，允许应用对输出数据寄存器(GPIOx\_ODR)的每个位进行置位和清零操作。

当对位 GPIOx\_ODSET[i]写 1 时，则置位相应的 ODR[i]位。当对 GPIOx\_ODCLR[i]写 1 时，则清零相应的 ODR[i]位。

对 GPIOx\_ODSET, GPIOx\_ODCLR 中的任意位写 0 都不会影响 GPIOx\_ODR 寄存器的值。不仅可以用 GPIOx\_ODSET, GPIOx\_ODCLR 寄存器来改变 GPIOx\_ODR 的相应位，也可以对 GPIOx\_ODR 寄存器直接进行访问。GPIOx\_ODSET, GPIOx\_ODCLR 寄存器提供对 GPIOx\_ODR 寄存器原子位操作处理。

GPIOx\_ODR 用 GPIOx\_ODSET, GPIOx\_ODCLR 置位或清零的访问机制不需要软件去关闭中断来访问 GPIOx\_ODR：在一个 AHB 写访问周期改变 1 位或多为数据是可能的。

### 10.3-5 输入配置

当 I/O 编程配置为输入时：

- 该端口的输出缓冲区禁用
- 由 GPIOx\_PUPD 寄存器的值来激活上拉和下拉电阻
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器。

用对输入数据寄存器的读访问来获取同步后的输入数据图 10-2 给出了 I/O 端口的输入配置。

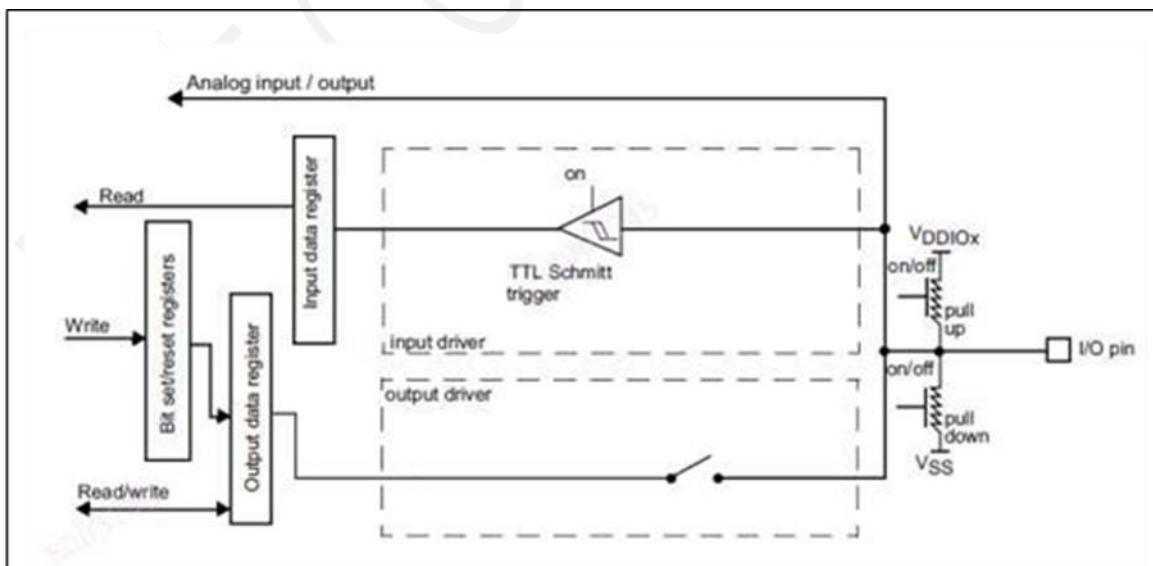


Figure 10 - 2 浮空输入/上拉/下拉配置

### 10.3-6 输出配置

当 I/O 口配置为输出时：

● 输出缓冲开启：

- 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。

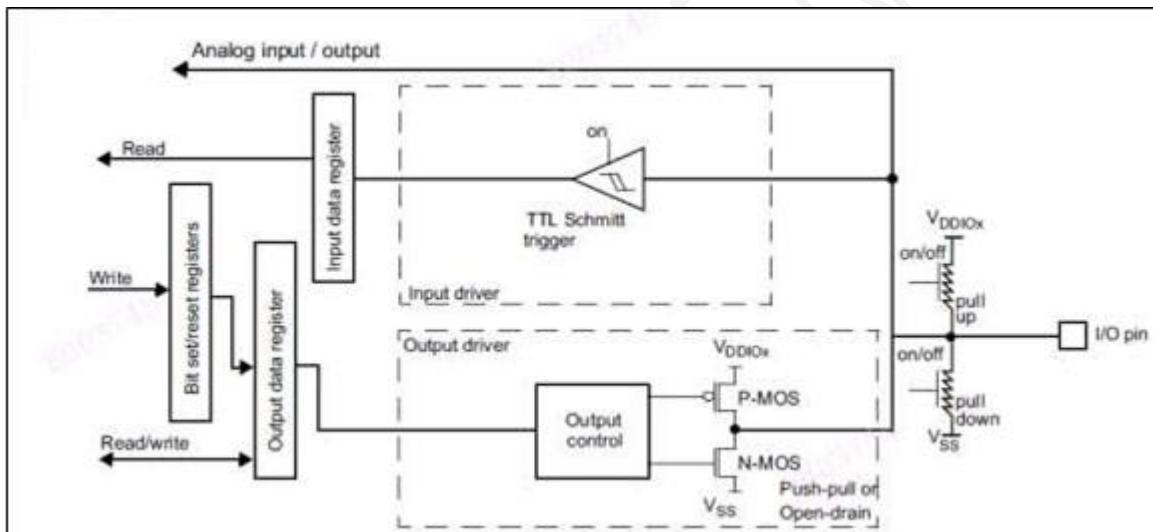
● 施密特触发输入被激活

● 弱上拉和弱下拉电阻是否激活取决于 GPIOx\_PUPDR 寄存器的值

● 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器

● 用对输入数据寄存器的读访问来获取同步后的输入数据

用对输出寄存器的读访问来获取最后写进该寄存器的值图 10-3 给出了 I/O 端口位的输出配置。



**Figure 10 - 3** 输出配置

### 10.3-7 外部中断/唤醒线

所有端口都有中断产生和唤醒能力。进入 Sleep 或者 Deep Sleep 模式后，如果检测到中断产生，WIC 模块会自动唤醒芯片。

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，当中断触发时，通过查询中断状态寄存器就可以判是哪一个端口触发了中断，通过置位中断清除寄存器就可以清除对应的中断状态标志位。

### 10.3-8I/O 引脚的复用功能和重映射

器件 I/O 通过多路复用器连接到内嵌的外设模块。微观上，同一时刻仅允许外设的复用功能的一个引脚连接到一个 I/O 接口上。因此，同一根线上不能有冲突的外设引脚分配。

每个 I/O 引脚都有一个复用功能多路选择器，可通过配置 GPIOx\_AFR 寄存器(从引脚 0 到引脚 7)来实现。

复位后，所有的 I/O 口都连接到 GPIO 功能。每个外设还有复用功能映射到不同的 I/O 引脚上，这种通用输入输出口(GPIO)方法用于在小封装器件上优化更多的可用外设。

有关每个引脚的具体复用功能请参考：表 10-2 GPIO 口和外设引脚的复用功能映射

为了使用一个给定的 I/O 口配置，你必须按如下的原则执行：

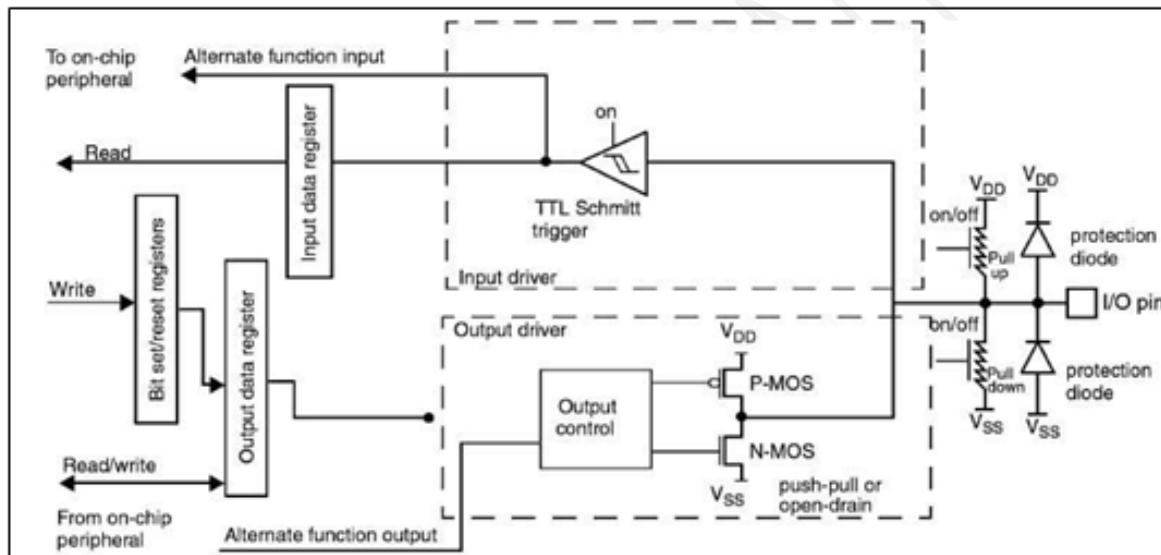
- 调试功能：每个器件复位后，这些引脚立即配置为复用功能用来支持调用。
- GPIO：在 GPIOx\_DIRCR 寄存器中配置所需的 I/O 为输出/输入。
- 外设的复用功能：
  - 连接 I/O 到所需的 AFRx，AFRx 定义在 GPIOx\_AFR 寄存器中；
  - 通过对 GPIOx\_PUPDR 和 GPIOx\_SLEWCR, GPIOx\_DRVCR 寄存器来配置相应引脚的上拉/下拉、输出速度及输出能力；
  - 通过对 GPIOx\_OTYPER 寄存器来配置输出类型：0 表示 push-pull, 1 表示 open drain;
- 附加功能：
  - 对于 ADC/VC，配置 GPIOx\_AFR 为 0x0F，配置所需的 I/O 口线为模拟方式并在 ADC 或 VC 寄存器中配置所需的功能。
  - 对于附加功能振荡器，在关联的 RCC 寄存器配置相应所需的功能。

**Table 10 - 2 GPIO 口和外设引脚的复用功能映射,请参考 4.3**

当 I/O 端口被配置为复用功能时:

- 在开漏或推挽模式下输出缓冲器可被配置
- 外设的信号驱动输出缓冲器
- 施密特触发输入被激活
- 弱上拉和弱下拉电阻是否激活取决于 GPIOx\_PUPDR 寄存器的值
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器
- 用对输入数据寄存器的读访问来获取同步后的输入数据

图 10-4 给出了 I/O 端口位的复用功能配置。


**Figure 10 - 4 复用功能配置**

### 10.3-9 模拟配置

当 I/O 端口编程为模拟配置时(GPIOx\_AFR.AFR=0x0F):

- 输出缓冲器关闭
- 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

图 10-5 给出了 I/O 端口位的高阻抗模拟配置。

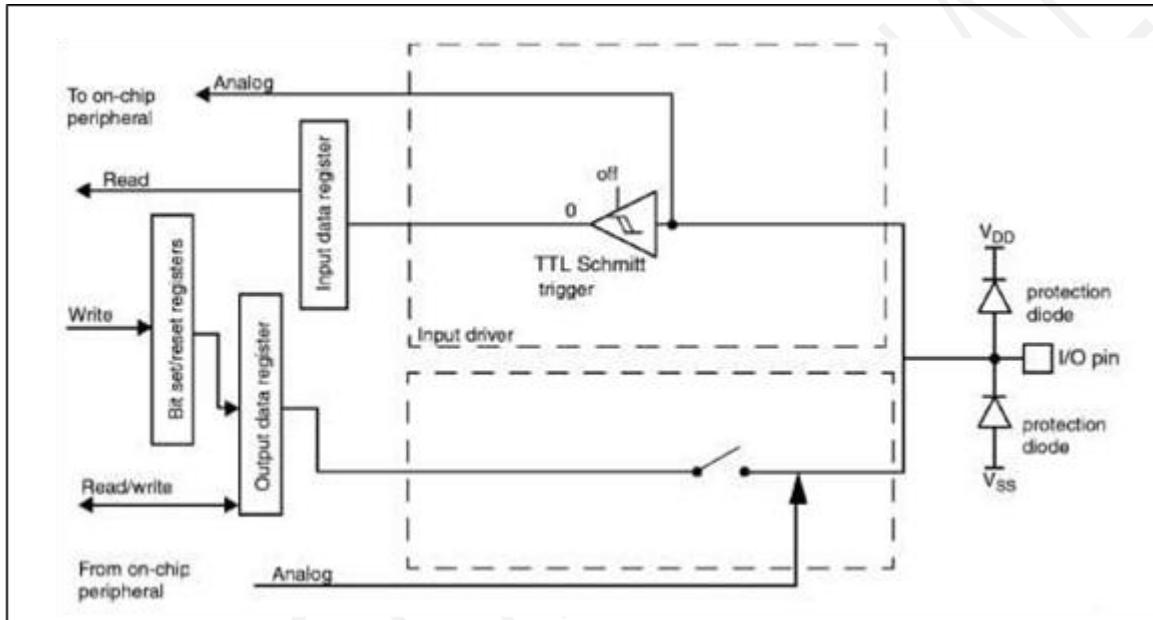


Figure 10 - 5 高阻抗模拟配置

### 10.3-10 HXT 或 LXT 引脚用作 GPIO

当 HXT 或 LXT 振荡器关断时(复位后的缺省状态)，相关振荡器引脚可以用做普通的 GPIO 口。

当 HXT 或 LXT 振荡器开启(设置 RCC\_SYSCLKCR.HXTEN 或 RCC\_LXTCR.LXTEN 位来开启)时，必须配置相应的管脚为模拟功能，振荡器控制其相关引脚且相关引脚的 GPIO 配置无效。

配置振荡器引脚为模拟功能的方法：

- 通过 RCC\_SYSCLKCR.HXTPORT=1 或 RCC\_LXTCR.LXTPORT=1 来配置
- 通过设定相应的引脚的 GPIOx\_AFR=0xF 来配置

当 振荡器配置为用户外部时钟输入方式 (RCC\_SYSCLKCR.HXTBYP=1 或 RCC\_LXTCR.LXTBYP=1)，不需要配置相应的引脚为模拟功能，仅使用 OSC\_IN 或 X32K\_IN 引脚做为时钟输入处理，OSC\_OUT 或 X32K\_OUT 引脚仍然可配置为正常的 GPIO 引脚。

#### 10.4 GPIOx 寄存器列表

GPIOx 基地址(x=A、B、C、D)

GPIOx	偏移地址	描述
GPIOA	0x4002 1000	GPIOA 偏移地址
GPIOB	0x4002 1400	GPIOB 偏移地址
GPIOC	0x4002 1800	GPIOC 偏移地址
GPIOD	0x4002 1C00	GPIOD 偏移地址

Table 10 - 3 GPIOx 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	GPIOx_DIRCR	输入输出模式寄存器	0x0000 0000
0x04	GPIOx_OTYPER	输出类型寄存器	0x0000 0000
0x08	GPIOx_ODR	输出数据寄存器	0x0000 0000
0x0C	GPIOx_IDR	输入数据寄存器	0x0000 XXXX
0x10	GPIOx_INTEN	中断使能寄存器	0x0000 0000
0x14	GPIOx_RAWINTSR	中断原始状态寄存器, 只读。 不论中断是否使能, 都可以读到中断状态。	0x0000 0000
0x18	GPIOx_MSKINTCR	中断状态寄存器, 只读	0x0000 0000
0x1C	GPIOx_INTCLR	中断清除寄存器	0x0000 0000
0x20	GPIOx_INTPOLPCR	中断类型寄存器	0x0000 0000
0x24	GPIOx_INTPOLCR	中断类型值寄存器	0x0000 0000
0x28	GPIOx_INTANY	任意边沿触发中断寄存器	0x0000 0000
0x2C	GPIOx_ODSET	输出置位寄存器	0x0000 0000
0x30	GPIOx_ODCLR	输出清除寄存器	0x0000 0000
0x34	GPIOx_INDBEN	输入去抖动和同步使能寄存器	0x0000 0000
0x38	GPIOx_DBCLKCR	输入去抖动时钟配置寄存器	0x0000 0000
0x3C	GPIOx_PUPDR	上拉/下拉寄存器	0x0000 0000
0x40	GPIOx_SLEWCR	电压转换速率控制	GPIOA : 0x0000 000E GPIOB : 0x0000 0030 GPIOC : 0x0000 00F8 GPIOD : 0x0000 007E
0x44	GPIOx_DRVCR	驱动强度配置	0x0000 0000
0x48	GPIOx_AFRL(1*)	复用功能寄存器(1*)	0x0000 0000
0x4C	GPIOx_AFRH(2*)	复用功能寄存器(2*)	0x0000 0000
0x50	GPIOx_STEN	输入触发器始能寄存器	0xFFFF FFFF

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

1\* : PORTA, B, C, D(7~0 bit); 2\* : PORTA, B, C, D(17~10 bit);

### 10.5GPIO 寄存器说明

#### 10.5-1GPIO 端口方向寄存器(GPIOx\_DIRCR)(x = A,B,C,D)

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
Px DIR15	Px DIR14	Px DIR13	Px DIR12	Px DIR11	Px DIR10	Px DIR9	Px DIR8	Px DIR7	Px DIR6	Px DIR5	Px DIR4	Px DIR3	Px DIR2	Px DIR1	Px DIR0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxDIR15	端子Px15 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
14	PxDIR14	端子Px14 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
13	PxDIR13	端子Px13 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
12	PxDIR12	端子Px12 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
11	PxDIR11	端子Px11 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
10	PxDIR10	端子Px10 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
9	PxDIR9	端子Px9 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W

8	PxDIR8	端子Px8 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
7	PxDIR7	端子Px7 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
6	PxDIR6	端子Px6 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
5	PxDIR5	端子Px5 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
4	PxDIR4	端子Px4 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
3	PxDIR3	端子Px3 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
2	PxDIR2	端子Px2 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
1	PxDIR1	端子Px1 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W
0	PxDIR0	端子Px0 输入/输出方向选择位 0: 输入模式; 1: 输出模式	0	R/W

### 10.5-2GPIO 端口输出类型寄存器(GPIOx\_OTYPER)(x = A,B,C,D)

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
Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px
OTYP15	OTYP14	OTYP13	OTYP12	OTYP11	OTYP10	OTYP9	OTYP8	OTYP7	OTYP6	OTYP5	OTYP4	OTYP3	OTYP2	OTYP1	OTYP0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxOTYP15	端子Px15 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
14	PxOTYP14	端子Px14 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
13	PxOTYP13	端子Px13 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
12	PxOTYP12	端子Px12 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
11	PxOTYP11	端子Px11 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
10	PxOTYP10	端子Px10 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
9	PxOTYP9	端子Px9 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
8	PxOTYP8	端子Px8 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W

7	PxOTYP7	端子Px7 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
6	PxOTYP6	端子Px6 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
5	PxOTYP5	端子Px5 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
4	PxOTYP4	端子Px4 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
3	PxOTYP3	端子Px3 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
2	PxOTYP2	端子Px2 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
1	PxOTYP1	端子Px1 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W
0	PxOTYPO	端子Px0 输出类型控制位 0: 推挽输出(复位状态); 1: 开漏输出	0	R/W

### 10.5-3GPIO 端口输出数据寄存器(GPIOx\_ODR)(x = A,B,C,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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px OD15	Px OD14	Px OD13	Px OD12	Px OD11	Px OD10	Px OD9	Px OD8	Px OD7	Px OD6	Px OD5	Px OD4	Px OD3	Px OD2	Px OD1	Px OD0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxOD15	端子 Px15 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
14	PxOD14	端子Px14 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
13	PxOD13	端子Px13 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
12	PxOD12	端子Px12 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
11	PxOD11	端子Px11 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
10	PxOD10	端子Px10 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
9	PxOD9	端子Px9 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
8	PxOD8	端子Px8 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W

位	标记	功能描述	复位值	读写
7	PxOD7	端子Px7 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
6	PxOD6	端子Px6 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
5	PxOD5	端子Px5 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
4	PxOD4	端子Px4 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
3	PxOD3	端子Px3 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
2	PxOD2	端子Px2 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
1	PxOD1	端子Px1 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
0	PxOD0	端子Px0 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W

### 10.5-4GPIO 端口输入数据寄存器(GPIOx\_IDR)(x = A,B,C,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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px ID15	Px ID14	Px ID13	Px ID12	Px ID11	Px ID15	Px ID9	Px ID8	Px ID7	Px ID6	Px ID5	Px ID4	Px ID3	Px ID2	Px ID1	Px ID0
RO															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxID15	端子Px15 输入值 0: 输入低电平 1: 输入高电平	0	RO
14	PxID14	端子Px14 输入值 0: 输入低电平 1: 输入高电平	0	RO
13	PxID13	端子Px13 输入值 0: 输入低电平 1: 输入高电平	0	RO
12	PxID12	端子Px12 输入值 0: 输入低电平 1: 输入高电平	0	RO
11	PxID11	端子Px11 输入值 0: 输入低电平 1: 输入高电平	0	RO
10	PxID10	端子Px10 输入值 0: 输入低电平 1: 输入高电平	0	RO
9	PxID9	端子Px9 输入值 0: 输入低电平 1: 输入高电平	0	RO
8	PxID8	端子Px8 输入值 0: 输入低电平 1: 输入高电平	0	RO

位	标记	功能描述	复位值	读写
7	PxID7	端子Px7 输入值 0: 输入低电平 1: 输入高电平	0	RO
6	PxID6	端子Px6 输入值 0: 输入低电平 1: 输入高电平	0	RO
5	PxID5	端子Px5 输入值 0: 输入低电平 1: 输入高电平	0	RO
4	PxID4	端子Px4 输入值 0: 输入低电平 1: 输入高电平	0	RO
3	PxID3	端子Px3 输入值 0: 输入低电平 1: 输入高电平	0	RO
2	PxID2	端子Px2 输入值 0: 输入低电平 1: 输入高电平	0	RO
1	PxID1	端子Px1 输入值 0: 输入低电平 1: 输入高电平	0	RO
0	PxID0	端子Px0 输入值 0: 输入低电平 1: 输入高电平	0	RO

### 10.5-5GPIO 端口中断使能寄存器(GPIOx\_INTEN)(x = A,B,C,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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px IEN15	Px IEN14	Px IEN13	Px IEN12	Px IEN11	Px IEN10	Px IEN9	Px IEN8	Px IEN7	Px IEN6	Px IEN5	Px IEN4	Px IEN3	Px IEN2	Px IEN1	Px IENO
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxIEN15	端子 Px15 中断屏蔽解除位 0: Px15 端子中断屏蔽 1: Px15 端子中断有效	0	R/W
14	PxIEN14	端子 Px14 中断屏蔽解除位 0: Px14 端子中断屏蔽 1: Px14 端子中断有效	0	R/W
13	PxIEN13	端子 Px13 中断屏蔽解除位 0: Px13 端子中断屏蔽 1: Px13 端子中断有效	0	R/W
12	PxIEN12	端子 Px12 中断屏蔽解除位 0: Px12 端子中断屏蔽 1: Px12 端子中断有效	0	R/W
11	PxIEN11	端子 Px11 中断屏蔽解除位 0: Px11 端子中断屏蔽 1: Px11 端子中断有效	0	R/W
10	PxIEN10	端子 Px10 中断屏蔽解除位 0: Px10 端子中断屏蔽 1: Px10 端子中断有效	0	R/W
9	PxIEN9	端子 Px9 中断屏蔽解除位 0: Px9 端子中断屏蔽 1: Px9 端子中断有效	0	R/W
8	PxIEN8	端子 Px8 中断屏蔽解除位 0: Px8 端子中断屏蔽 1: Px8 端子中断有效	0	R/W

位	标记	功能描述	复位值	读写
7	PxIEN7	端子 Px7 中断屏蔽解除位 0: Px7 端子中断屏蔽 1: Px7 端子中断有效	0	R/W
6	PxIEN6	端子 Px6 中断屏蔽解除位 0: Px6 端子中断屏蔽 1: Px6 端子中断有效	0	R/W
5	PxIEN5	端子 Px5 中断屏蔽解除位 0: Px5 端子中断屏蔽 1: Px5 端子中断有效	0	R/W
4	PxIEN4	端子 Px4 中断屏蔽解除位 0: Px4 端子中断屏蔽 1: Px4 端子中断有效	0	R/W
3	PxIEN3	端子 Px3 中断屏蔽解除位 0: Px3 端子中断屏蔽 1: Px3 端子中断有效	0	R/W
2	PxIEN2	端子 Px2 中断屏蔽解除位 0: Px2 端子中断屏蔽 1: Px2 端子中断有效	0	R/W
1	PxIEN1	端子 Px1 中断屏蔽解除位 0: Px1 端子中断屏蔽 1: Px1 端子中断有效	0	R/W
0	PxIEN0	端子 Px0 中断屏蔽解除位 0: Px0 端子中断屏蔽 1: Px0 端子中断有效	0	R/W

### 10.5-6GPIO 端口中断原始状态寄存器(GPIOx\_RAWINTSR)(x = A,B,C,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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px RIS15	Px RIS14	Px RIS13	Px RIS12	Px RIS11	Px RIS10	Px RIS9	Px RIS8	PX RIS7	Px RIS6	Px RIS5	Px RIS4	Px RIS3	Px RIS2	Px RIS1	Px RIS0
RO															

不论中断是否使能，都可以读到中断状态。

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxRIS15	端子Px15 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
14	PxRIS14	端子Px14 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
13	PxRIS13	端子Px13 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
12	PxRIS12	端子Px12 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
11	PxRIS11	端子Px11 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
10	PxRIS10	端子Px10 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
9	PxRIS9	端子Px9 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
8	PxRIS8	端子Px8 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO

位	标记	功能描述	复位值	读写
7	PxRIS7	端子Px7 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
6	PxRIS6	端子Px6 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
5	PxRIS5	端子Px5 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
4	PxRIS4	端子Px4 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
3	PxRIS3	端子Px3 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
2	PxRIS2	端子Px2 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
1	PxRIS1	端子Px1 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
0	PxRIS0	端子Px0 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO

### 10.5-7GPIO 端口中断状态寄存器(GPIOx\_MSKINTSR)(x = A,B,C,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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px MIS15	Px MIS14	Px MIS13	Px MIS12	Px MIS11	Px MIS10	Px MIS9	Px MIS8	Px MIS7	Px MIS6	Px MIS5	Px MIS4	Px MIS3	Px MIS2	Px MIS1	Px MISO
RO															

这些位只读，由硬件置位。只有中断使能，才可以读到中断状态。

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxMIS15	端子Px15 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
14	PxMIS14	端子Px14 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
13	PxMIS13	端子Px13 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
12	PxMIS12	端子Px12 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
11	PxMIS11	端子Px11 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
10	PxMIS10	端子Px10 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
9	PxMIS9	端子Px9 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
8	PxMIS8	端子Px8 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO

位	标记	功能描述	复位值	读写
7	PxMIS7	端子Px7 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
6	PxMIS6	端子Px6 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
5	PxMIS5	端子Px5 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
4	PxMIS4	端子Px4 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
3	PxMIS3	端子Px3 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
2	PxMIS2	端子Px2 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
1	PxMIS1	端子Px1 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
0	PxMIS0	端子Px0 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO

### 10.5-8GPIO 端口中断清除寄存器(GPIOx\_INTCLR)(x = A,B,C,D)

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
Px ICLR5	Px ICLR14	Px ICLR13	Px ICLR12	Px ICLR11	Px ICLR10	Px ICLR9	Px ICLR8	Px ICLR7	Px ICLR6	Px ICLR5	Px ICLR4	Px ICLR3	Px ICLR2	Px ICLR1	Px ICLR0
WO															

这些位只写，用于软件清除中断。

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxICLR15	写 1 清除端子Px15 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
14	PxICLR14	写 1 清除端子Px14 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
13	PxICLR13	写 1 清除端子Px13 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
12	PxICLR12	写 1 清除端子Px12 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
11	PxICLR11	写 1 清除端子Px11 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
10	PxICLR10	写 1 清除端子Px10 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
9	PxICLR9	写 1 清除端子Px9 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
8	PxICLR8	写 1 清除端子Px8 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO

位	标记	功能描述	复位值	读写
7	PxICLR7	写 1 清除端子Px7 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
6	PxICLR6	写 1 清除端子Px6 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
5	PxICLR5	写 1 清除端子Px5 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
4	PxICLR4	写 1 清除端子Px4 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
3	PxICLR3	写 1 清除端子Px3 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
2	PxICLR2	写 1 清除端子Px2 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
1	PxICLR1	写 1 清除端子Px1 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
0	PxICLR0	写 1 清除端子Px0 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO

### 10.5-9GPIO 端口中断类型寄存器(GPIOx\_INTTYPCR)(x = A,B,C,D)

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
Px ITYPE1 5	Px ITYPE1 4	Px ITYPE 13	Px ITYPE 12	Px ITYPE 11	Px ITYPE 10	Px ITYPE 9	Px ITYPE 8	Px ITYPE 7	Px ITYPE 6	Px ITYPE 5	Px ITYPE 4	Px ITYPE 3	Px ITYPE 2	Px ITYPE 1	Px ITYPE 0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxITYPE15	端子Px15 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
14	PxITYPE14	端子Px14 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
13	PxITYPE13	端子Px13 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
12	PxITYPE12	端子Px12 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
11	PxITYPE11	端子Px11 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
10	PxITYPE10	端子Px10 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
9	PxITYPE9	端子Px9 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W

位	标记	功能描述	复位值	读写
8	PxI TYPE8	端子Px8 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
7	PxI TYPE7	端子Px7 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
6	PxI TYPE6	端子Px6 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
5	PxI TYPE5	端子Px5 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
4	PxI TYPE4	端子Px4 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
3	PxI TYPE3	端子Px3 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
2	PxI TYPE2	端子Px2 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
1	PxI TYPE1	端子Px1 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
0	PxI TYPE0	端子Px0 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W

### 10.5-10GPIO 端口中断类型寄存器(GPIOx\_INTPOLCR)(x = A,B,C,D)

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
Px IVAL15	Px IVAL14	Px IVAL13	Px IVAL12	Px IVAL11	Px IVAL10	Px IVAL9	Px IVAL8	Px IVAL7	Px IVAL6	Px IVAL5	Px IVAL4	Px IVAL3	Px IVAL2	Px IVAL1	Px IVAL0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxIVAL15	端子Px15 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
14	PxIVAL14	端子Px14 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
13	PxIVAL13	端子Px13 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
12	PxIVAL12	端子Px12 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
11	PxIVAL11	端子Px11 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
10	PxIVAL10	端子Px10 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
9	PxIVAL9	端子Px9 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
8	PxIVAL8	端子Px8 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W

位	标记	功能描述	复位值	读写
7	PxIVAL7	端子Px7 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
6	PxIVAL6	端子Px6 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
5	PxIVAL5	端子Px5 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
4	PxIVAL4	端子Px4 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
3	PxIVAL3	端子Px3 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
2	PxIVAL2	端子Px2 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
1	PxIVAL1	端子Px1 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
0	PxIVAL0	端子Px0 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W

### 10.5-11GPIO 端口任意边沿触发中断寄存器(GPIOx\_INTANY)(x = A,B,C,D)

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
Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px	Px
IANY15	IANY14	IANY13	IANY12	IANY11	IANY10	IANY9	IANY8	IANY7	IANY6	IANY5	IANY4	IANY3	IANY2	IANY1	IANY0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxIANY15	端子Px15 任意沿触发中断配置位 0: 中断触发沿由PxIVAL15 决定 1: 上升/下降沿都触发中断	0	R/W
14	PxIANY14	端子Px14 任意沿触发中断配置位 0: 中断触发沿由PxIVAL14 决定 1: 上升/下降沿都触发中断	0	R/W
13	PxIANY13	端子Px13 任意沿触发中断配置位 0: 中断触发沿由PxIVAL13 决定 1: 上升/下降沿都触发中断	0	R/W
12	PxIANY12	端子Px12 任意沿触发中断配置位 0: 中断触发沿由PxIVAL12 决定 1: 上升/下降沿都触发中断	0	R/W
11	PxIANY11	端子Px11 任意沿触发中断配置位 0: 中断触发沿由PxIVAL11 决定 1: 上升/下降沿都触发中断	0	R/W
10	PxIANY10	端子Px10 任意沿触发中断配置位 0: 中断触发沿由PxIVAL10 决定 1: 上升/下降沿都触发中断	0	R/W
9	PxIANY9	端子Px9 任意沿触发中断配置位 0: 中断触发沿由PxIVAL9 决定 1: 上升/下降沿都触发中断	0	R/W
8	PxIANY8	端子Px8 任意沿触发中断配置位 0: 中断触发沿由PxIVAL8 决定 1: 上升/下降沿都触发中断	0	R/W

位	标记	功能描述	复位值	读写
7	PxIANY7	端子Px7 任意沿触发中断配置位 0: 中断触发沿由PxIVAL7 决定 1: 上升/下降沿都触发中断	0	R/W
6	PxIANY6	端子Px6 任意沿触发中断配置位 0: 中断触发沿由PxIVAL6 决定 1: 上升/下降沿都触发中断	0	R/W
5	PxIANY5	端子Px5 任意沿触发中断配置位 0: 中断触发沿由PxIVAL5 决定 1: 上升/下降沿都触发中断	0	R/W
4	PxIANY4	端子Px4 任意沿触发中断配置位 0: 中断触发沿由PxIVAL4 决定 1: 上升/下降沿都触发中断	0	R/W
3	PxIANY3	端子Px3 任意沿触发中断配置位 0: 中断触发沿由PxIVAL3 决定 1: 上升/下降沿都触发中断	0	R/W
2	PxIANY2	端子Px2 任意沿触发中断配置位 0: 中断触发沿由PxIVAL2 决定 1: 上升/下降沿都触发中断	0	R/W
1	PxIANY1	端子Px1 任意沿触发中断配置位 0: 中断触发沿由PxIVAL1 决定 1: 上升/下降沿都触发中断	0	R/W
0	PxIANY0	端子Px0 任意沿触发中断配置位 0: 中断触发沿由PxIVAL0 决定 1: 上升/下降沿都触发中断	0	R/W

### 10.5-12GPIO 端口输出置位寄存器(GPIOx\_ODSET)(x = A,B,C,D)

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

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxODSET15	端子Px15 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
14	PxODSET14	端子Px14 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
13	PxODSET13	端子Px113 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
12	PxODSET12	端子Px2 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
11	PxODSET11	端子Px11 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
10	PxODSET10	端子Px10 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
9	PxODSET9	端子Px9 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO

位	标记	功能描述	复位值	读写
8	PxODSET8	端子Px8 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
7	PxODSET7	端子Px7 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
6	PxODSET6	端子Px6 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
5	PxODSET5	端子Px5 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
4	PxODSET4	端子Px4 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
3	PxODSET3	端子Px3 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
2	PxODSET2	端子Px2 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
1	PxODSET1	端子Px1 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
0	PxODSET0	端子Px0 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO

### 10.5-13GPIO 端口输出清除寄存器(GPIOx\_ODCLR)(x = A,B,C,D)

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
Px ODCLR15	Px ODCLR14	Px ODCLR13	Px ODCLR12	Px ODCLR11	Px ODCLR10	Px ODCLR9	Px ODCLR8	Px ODCLR7	Px ODCLR6	Px ODCLR5	Px ODCLR4	Px ODCLR3	Px ODCLR2	Px ODCLR1	Px ODCLR0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WO															

若 ODSETx 和 ODCLRx 同时设置，ODSETx 有优先权。

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxODCLR15	端子Px15 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
14	PxODCLR14	端子Px14 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
13	PxODCLR13	端子Px13 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
12	PxODCLR12	端子Px12 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
11	PxODCLR11	端子Px11 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
10	PxODCLR10	端子Px10 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
9	PxODCLR9	端子Px9 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO

位	标记	功能描述	复位值	读写
8	PxODCLR8	端子Px8 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
7	PxODCLR7	端子Px7 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
6	PxODCLR6	端子Px6 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
5	PxODCLR5	端子Px5 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
4	PxODCLR4	端子Px4 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
3	PxODCLR3	端子Px3 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
2	PxODCLR2	端子Px2 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
1	PxODCLR1	端子Px1 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO
0	PxODCLR0	端子Px0 输出清 0 控制位 0: 输出保持 1: 复位相应的ODRx 位	0	WO

### 10.5-14GPIO 端口输入去抖动寄存器(GPIOx\_INDBEN)(x = A,B,C,D)

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
SYNC_EN	保留														
WO	--														

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

位	标记	功能描述	复位值	读写
31	SYNC_EN	消抖未使能时通过设定该寄存器来配置输入是否使用 2 级同步来消除 亚稳态(只对中断有效)。 0: 不使用两级同步 1: 使用两级同步	0	R/W
30:16	保留	--	0x0	--
15:8	PxDIDBnn nn=(8~15)	端子 Px <sub>n</sub> (nn=8~15)消抖使能配置位, 如果输入信号不能被连续 2 个的 去抖采样周期采样, 则输入信号被视为信号抖动, 而不会触发中断. 仅用于边沿触发 “edge-trigger” 中断, 不能用于电平触发( “level trigger” )中断。电平模式时为两级同步输入 0: 禁止端口消抖功能 1: 使能端口消抖功能	0x0	R/W
7:0	PxDIDBn n=(0~7)	端子 Px <sub>n</sub> (n=0~7)消抖使能配置位, 如果输入信号不能被连续 2 个的 去抖采样周期采样, 则输入信号被视为信号抖动, 而不会触发中断. 仅 用于边沿触发 “edge-trigger” 中断, 不能用于电平触发( “level trigger” )中断。电平模式时为两级同步输入 0: 禁止端口消抖功能 1: 使能端口消抖功能	0	R/W

### 10.5-15GPIO 端口输入去抖动时钟配置寄存器(GPIOx\_DBCLKCR)(x = A,B,C,D)

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
保留										DBCLKEN	DBCLK_DIV[3:0]				
--											R/W				

位	标记	功能描述	复位值	读写
31:5	保留	--	0x0	--
4	DBCLKEN	是否使能去抖动时钟 0: 不使能去抖动时钟 1: 使能去抖动时钟	0	R/W
3:0	DBCLK_DIV	去抖动采样周期选择 Debounce 时钟是hclk 的( $2^{\text{dbclk\_div}[3:0]}$ )分频。 消抖采样周期选择: 0x0 : 消抖采样 1 个 HCLK 周期 1 次 0x1 : 消抖采样 2 个 HCLK 周期 1 次 0x2 : 消抖采样 4 个 HCLK 周期 1 次 0x3 : 消抖采样 8 个 HCLK 周期 1 次 0x4 : 消抖采样 16 个 HCLK 周期 1 次 0x5 : 消抖采样 32 个 HCLK 周期 1 次 0x6 : 消抖采样 64 个 HCLK 周期 1 次 0x7 : 消抖采样 128 个 HCLK 周期 1 次 0x8 : 消抖采样 256 个 HCLK 周期 1 次 0x9 : 消抖采样 512 个 HCLK 周期 1 次 0xA : 消抖采样 1024 个 HCLK 周期 1 次 0xB : 消抖采样 2*1024 个 HCLK 周期1 次 0xC : 消抖采样 4*1024 个 HCLK 周期 1 次 0xD : 消抖采样 8*1024 个 HCLK 周期 1 次 0xE : 消抖采样 16*1024 个 HCLK 周期 1 次 0xF : 消抖采样 32*1024 个 HCLK 周期 1 次	0	R/W

### 10.5-16GPIO 端口上拉/下拉寄存器(GPIOx\_INDBEN)(x = A,B,C,D)

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
PxPUPD15 [1:0]		PxPUPD14 [1:0]		PxPUPD13 [1:0]		PxPUPD12 [1:0]		PxPUPD11 [1:0]		PxPUPD10 [1:0]		PxPUPD9 [1:0]		PxPUPD8 [1:0]	
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
PxPUPD7 [1:0]		PxPUPD6 [1:0]		PxPUPD5 [1:0]		PxPUPD4 [1:0]		PxPUPD3 [1:0]		PxPUPD2 [1:0]		PxPUPD1 [1:0]		PxPUPD0 [1:0]	
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
31:30	PxPUPD15	端子Px15 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
29:28	PxPUPD14	端子Px14 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
27:26	PxPUPD13	端子Px13 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
25:24	PxPUPD12	端子Px12 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
23:22	PxPUPD11	端子Px11 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W

位	标记	功能描述	复位值	读写
21:20	PxPUPD10	端子Px10 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
19:18	PxPUPD9	端子Px9 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
17:16	PxPUPD8	端子Px8 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
15:14	PxPUPD7	端子Px7 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
13:12	PxPUPD6	端子Px6 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
11:10	PxPUPD5	端子Px5 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
9:8	PxPUPD4	端子Px4 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W

位	标记	功能描述	复位值	读写
7:6	PxPUPD3	端子Px3 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
5:4	PxPUPD2	端子Px2 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
3:2	PxPUPD1	端子Px1 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
1:0	PxPUPD0	端子Px0 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W

### 10.5-17GPIO 端口电压转换速率配置(GPIOx\_SLEWCR)(x = A,B,C,D)

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
Px SR15	Px SR14	Px SR13	Px SR12	Px SR11	Px SR10	Px SR9	Px SR8	Px SR7	Px SR6	Px SR5	Px SR4	Px SR3	Px SR2	Px SR1	Px SR0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxSR15	端子Px15 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
14	PxSR14	端子Px14 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
13	PxSR13	端子Px13 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
12	PxSR12	端子Px12 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
11	PxSR11	端子Px11 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
10	PxSR10	端子Px10 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
9	PxSR9	端子Px9 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
8	PxSR8	端子Px8 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W

位	标记	功能描述	复位值	读写
7	PxSR7	端子Px7 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
6	PxSR6	端子Px6 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
5	PxSR5	端子Px5 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
4	PxSR4	端子Px4 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
3	PxSR3	端子Px3 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
2	PxSR2	端子Px2 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
1	PxSR1	端子Px1 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W
0	PxSR0	端子Px0 电压转换速率配置控制位 0: 高电压转换速率 1: 低电压转换速率	注 1	R/W

注 1: 复位值请参考表 9-3 GPIOx 寄存器列表和复位值

### 10.5-18GPIO 端口驱动强度配置寄存器(GPIOx\_DRVCR)(x = A,B,C,D)

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
Px DRV15	Px DRV14	Px DRV13	Px DRV12	Px DRV11	Px DRV10	Px DRV9	Px DRV8	Px DRV7	Px DRV6	Px DRV5	Px DRV4	Px DRV3	Px DRV2	Px DRV1	Px DRV0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxDVR15	端子Px15 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
14	PxDVR14	端子Px14 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
13	PxDVR13	端子Px13 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
12	PxDVR12	端子Px12 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
11	PxDVR11	端子Px11 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
10	PxDVR10	端子Px10 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
9	PxDVR9	端子Px9 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
8	PxDVR8	端子Px8 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W

位	标记	功能描述	复位值	读写
7	PxDRV7	端子Px7 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
6	PxDRV6	端子Px6 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
5	PxDRV5	端子Px5 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
4	PxDRV4	端子Px4 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
3	PxDRV3	端子Px3 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
2	PxDRV2	端子Px2 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
1	PxDRV1	端子Px1 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
0	PxDRV0	端子Px0 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W

### 10.5-19GPIOA 端口复用功能寄存器(GPIOA\_AFR)

边界地址: 0x4002 1048 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			

边界地址: 0x4002 104C 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]				PAAFR8[3:0]			
R/W				R/W				R/W				R/W			

### 10.5-19.1PORTA 端口复用配置

请参考 4.3 Table

### 10.5-20GPIOB 端口复用功能寄存器(GPIOB\_AFR)

边界地址: 0x4002 1448 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			

边界地址: 0x4002 144C 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			

### 10.5-20.1PORTB 端口复用配置

请参考 4.3 Table

### 10.5-21GPIOC 端口复用功能寄存器(GPIOC\_AFR)

边界地址: 0x4002 1848 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			

边界地址: 0x4002 184C 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			

### 10.5-21.1PORTC 端口复用配置

请参考 4.3 Table

### 10.5-22GPIOD 端口复用功能寄存器(GPIOD\_AFR)

边界地址: 0x4002 1C48 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			

### 10.5-22.1PORTD 端口复用配置配置

请参考 [4.3 Table](#)

### 10.5-23GPIO 端口输入触发器始能(GPIOx\_DRVCR)(x = A,B,C,D)

偏移地址: 0x50

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Px STEN15	Px STEN14	Px STEN13	Px STEN12	Px STEN11	Px STEN10	Px STEN9	Px STEN8	Px STEN7	Px STEN6	Px STEN5	Px STEN4	Px STEN3	Px STEN2	Px STEN1	Px STEN0
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	PxSTEN15	端子Px15输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
14	PxSTEN14	端子Px14输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
13	PxSTEN13	端子Px13输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
12	PxSTEN12	端子Px12输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
11	PxSTEN11	端子Px11输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
10	PxSTEN10	端子Px10输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
9	PxSTEN9	端子Px9输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
8	PxSTEN8	端子Px8输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W

位	标记	功能描述	复位值	读写
7	PxSTEN7	端子Px7输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
6	PxSTEN6	端子Px6输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
5	PxSTEN5	端子Px5输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
4	PxSTEN4	端子Px4输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
3	PxSTEN3	端子Px3输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
2	PxSTEN2	端子Px2输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
1	PxSTEN1	端子Px1输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W
0	PxSTEN0	端子Px0输入触发器(schmitter Trigger)始能 0: 禁用 1: 始能	1	R/W

## 【11】Flash 控制器(Flash)

### 11.1 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。

### 11.2 Flash 结构框图

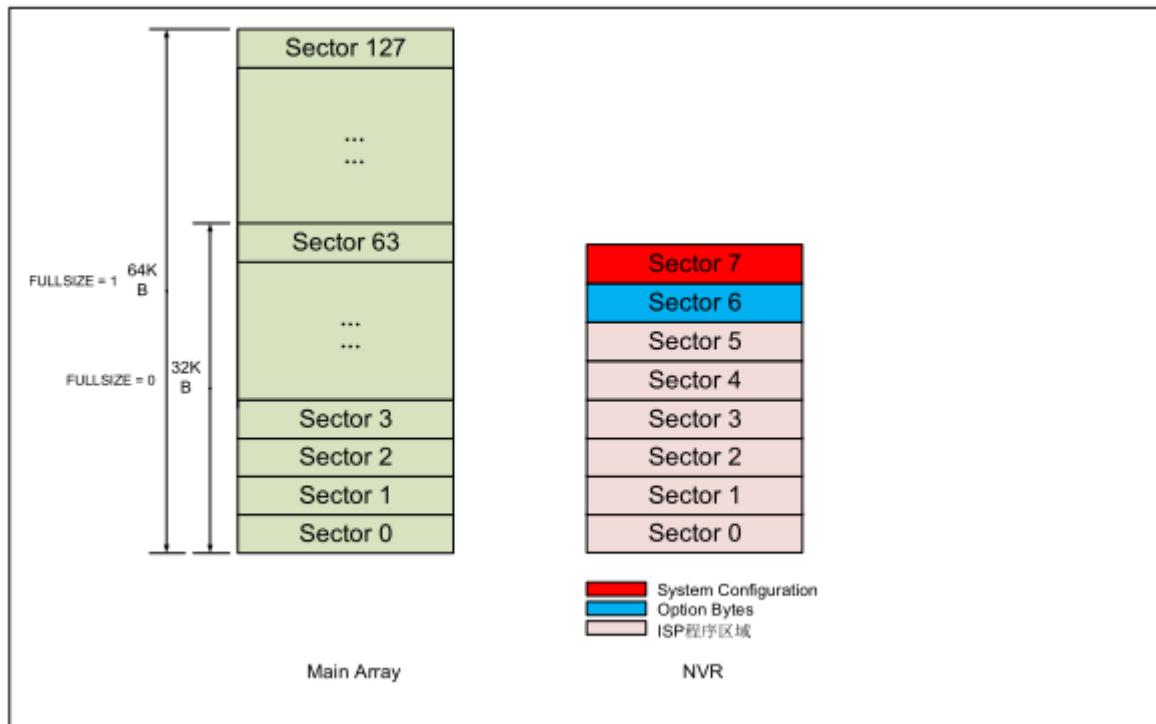


Figure 11- 1Flash 结构框图

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。

### 11.3 功能描述

本控制器支持对 Flash 的 byte(8bits)、half-word(16bits)、word(32bits)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。

#### 11.3-1 擦除操作

##### 11.3-1.1 Sector 擦除

Sector 擦除操作步骤如下图所示：

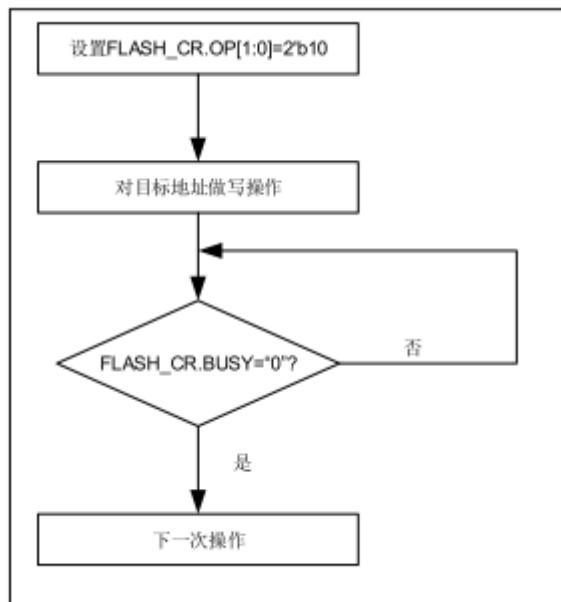


Figure 11- 2 擦除操作步骤

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。

### 11.3-1.2Chip 擦除

Chip 擦除操作步骤如下图所示：

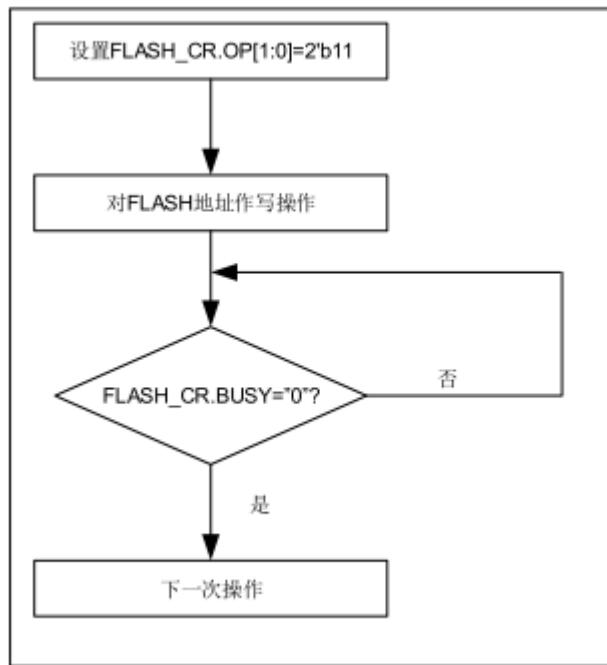


Figure 11- 3 Chip 擦除操作步骤

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。

### 11.3-2写操作

写操作步骤如下图所示：

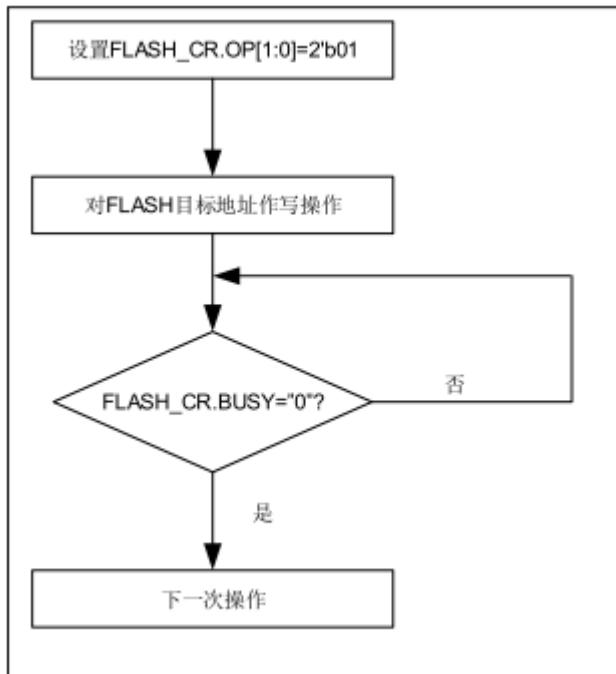


Figure 11- 4 写操作步骤

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。

### 11.3-3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  $32\text{MHz} < \text{SYSCLK} \leq 64\text{MHz}$

### 11.3-4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。

### 11.3-5擦写保护

#### 11.3-5.1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”。

#### 11.3-5.2PC 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。

### 11.3-5.3 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：

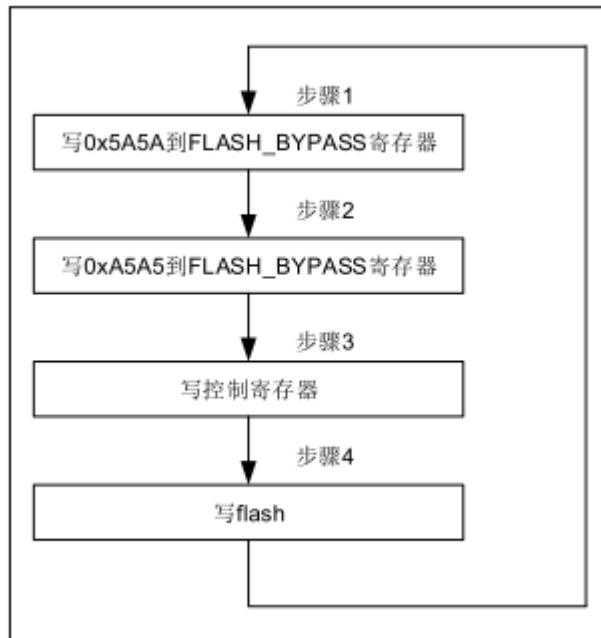


Figure 11- 5 写寄存器 BYPASS 序列

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。

### 11.3-6系统 BOOT 地址映像

#### 11.3-6.1APP 程序 boot

下图是 APP 程序区 boot 的地址映像图。

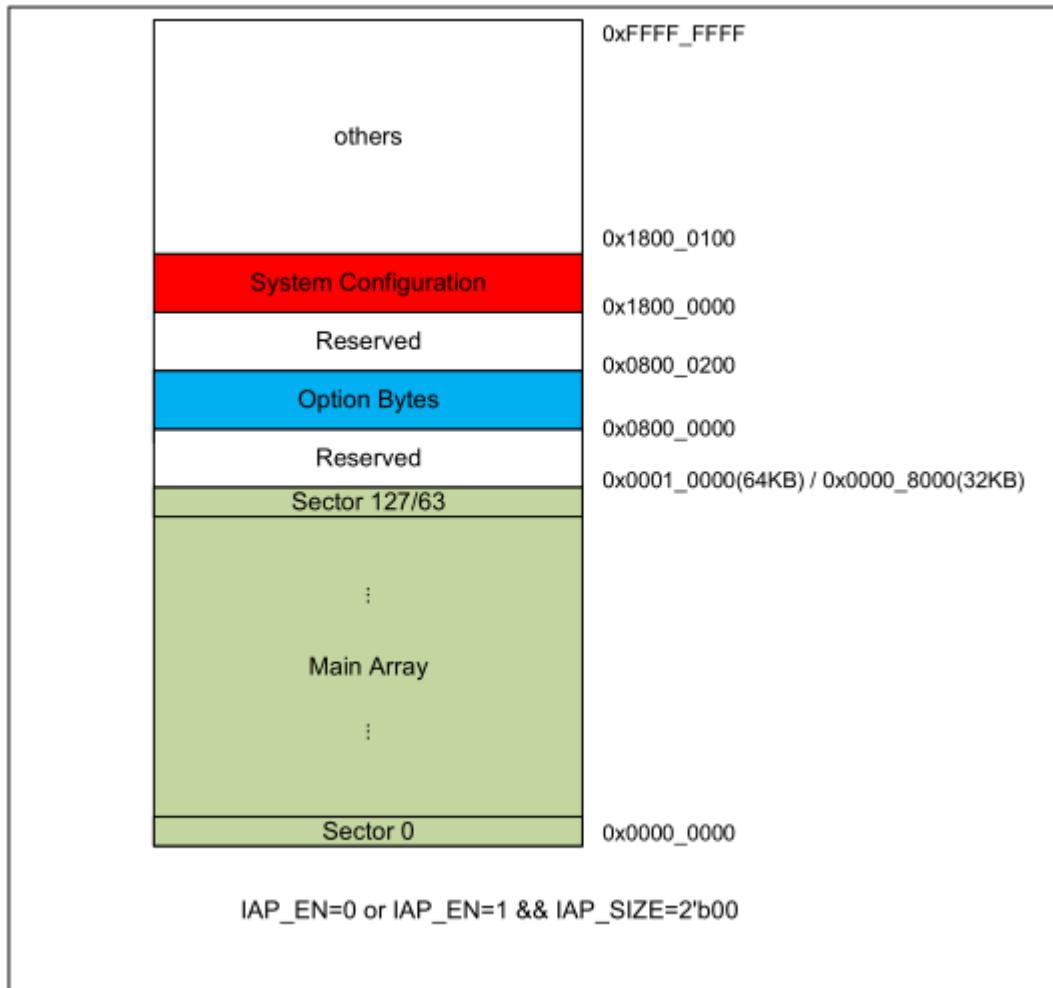


Figure 11- 6 APP 程序区 boot 地址映像图

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 512Byte 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。

### 11.3-7系选项字节区(Option Bytes)

具体内容请参考第 33 章选项字节区。

### 11.3-8 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。

#### 11.3-8.1 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 )

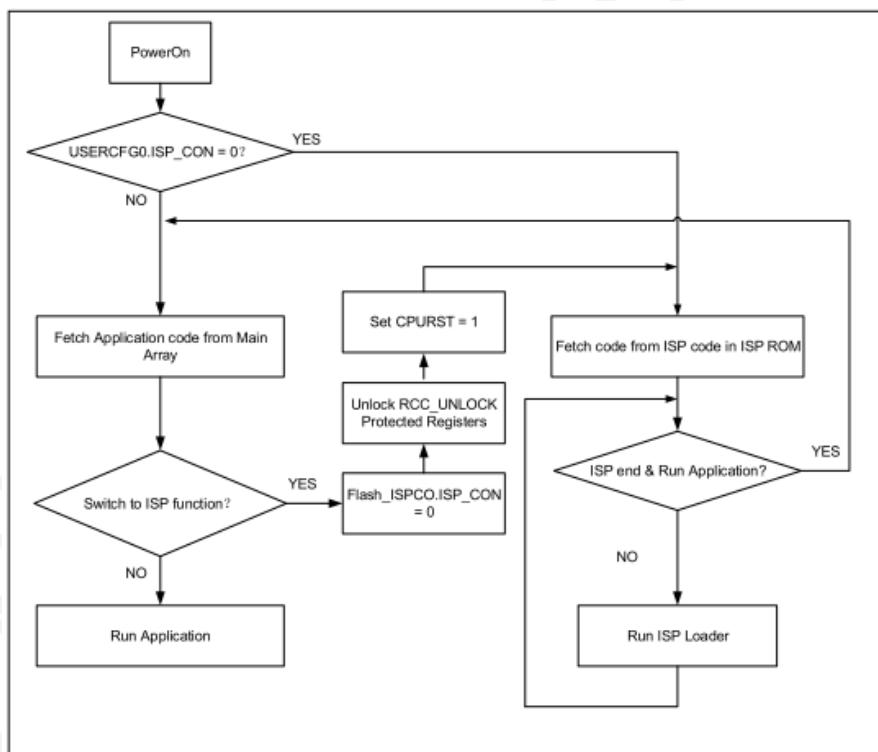


Figure 11- 7 内部配置与软件进入 ISP 流程图

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。

### 11.3-8.2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。

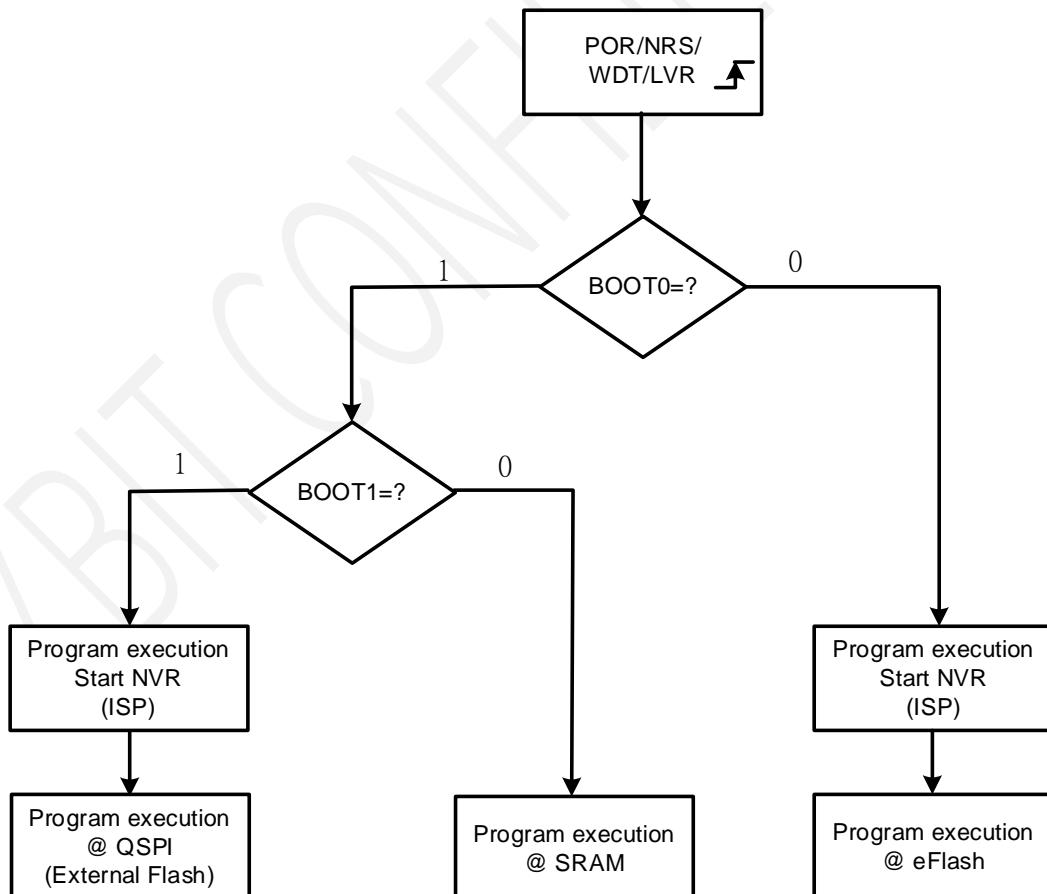


Figure 11- 8 外部 BOOT0,BOOT1 PIN 选择流程图

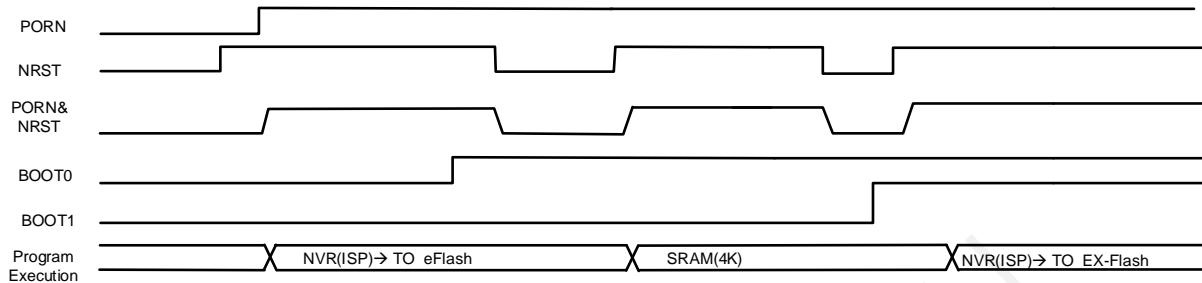


Figure 11- 9 BOOT0,BOOT1 PIN 时序图

#### 11.4 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 11- 1 Flash 寄存器列表和复位值

注: 所有 Flash 寄存器只能按 word 方式读写。

## 11.5 寄存器说明

### 11.5-1 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															
--															

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

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 rase 2' b11: Chip erase	2' b00	R/W

### 11.5-2Flash\_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															
--															

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

### 11.5-3Flash\_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															
--															

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

### 11.5-4Flash\_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															
--															

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

### 11.5-5Flash\_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 0: 不允许擦写 1: 允许擦写 SLOCK0[31] 对应: sector 62-63 SLOCK0[30] 对应: sector 60-61 SLOCK0[29] 对应: sector 58-59 ... ... ... SLOCK0[1] 对应: sector 2-3 SLOCK0[0] 对应: sector 0-1</p>	32' h0000_0000	R/W

### 11.5-6Flash\_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	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>SLOCK1[31] 对应: sector 126-127</p> <p>SLOCK1[30] 对应: sector 124-125</p> <p>SLOCK1[29] 对应: sector 122-123</p> <p>...</p> <p>...</p> <p>...</p> <p>SLOCK1[1] 对应: sector 66-67</p> <p>SLOCK1[0] 对应: sector 64-65</p>	32' h0000_0000	R/W

### 11.5-7Flash\_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															
--															

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

### 11.5-8Flash 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												Fdo_mask_en	WCYCLE		
--												R/W			

位	标记	功能描述	复位值	读写
31:16	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)

## 【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：

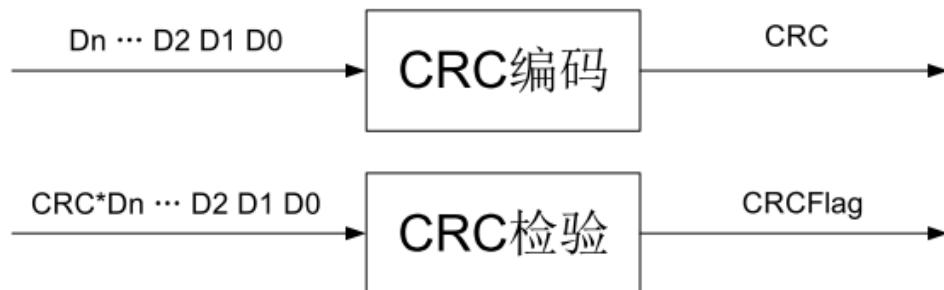


Figure 12- 1CRC 应用示意图

### 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

### 12.2-1CRC 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。

### 12.2-2CRC 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。

### 12.3CRC 寄存器列表

基址址: 0x4002 0800

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

### 12.4-1CRC 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	17	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

### 12.4-2CRC 控制寄存器寄存器(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	17	16
保留															
--															

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

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 $\rightarrow (x^{16} + x^{12} + x^5 + 1)$ 0 : CRC32 $\rightarrow (x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1)$	0x0	R/W

### 12.4-3CRC 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	18	17	16
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(TIM1/1A/1B)

### 13.1 Advanced Timer 简介

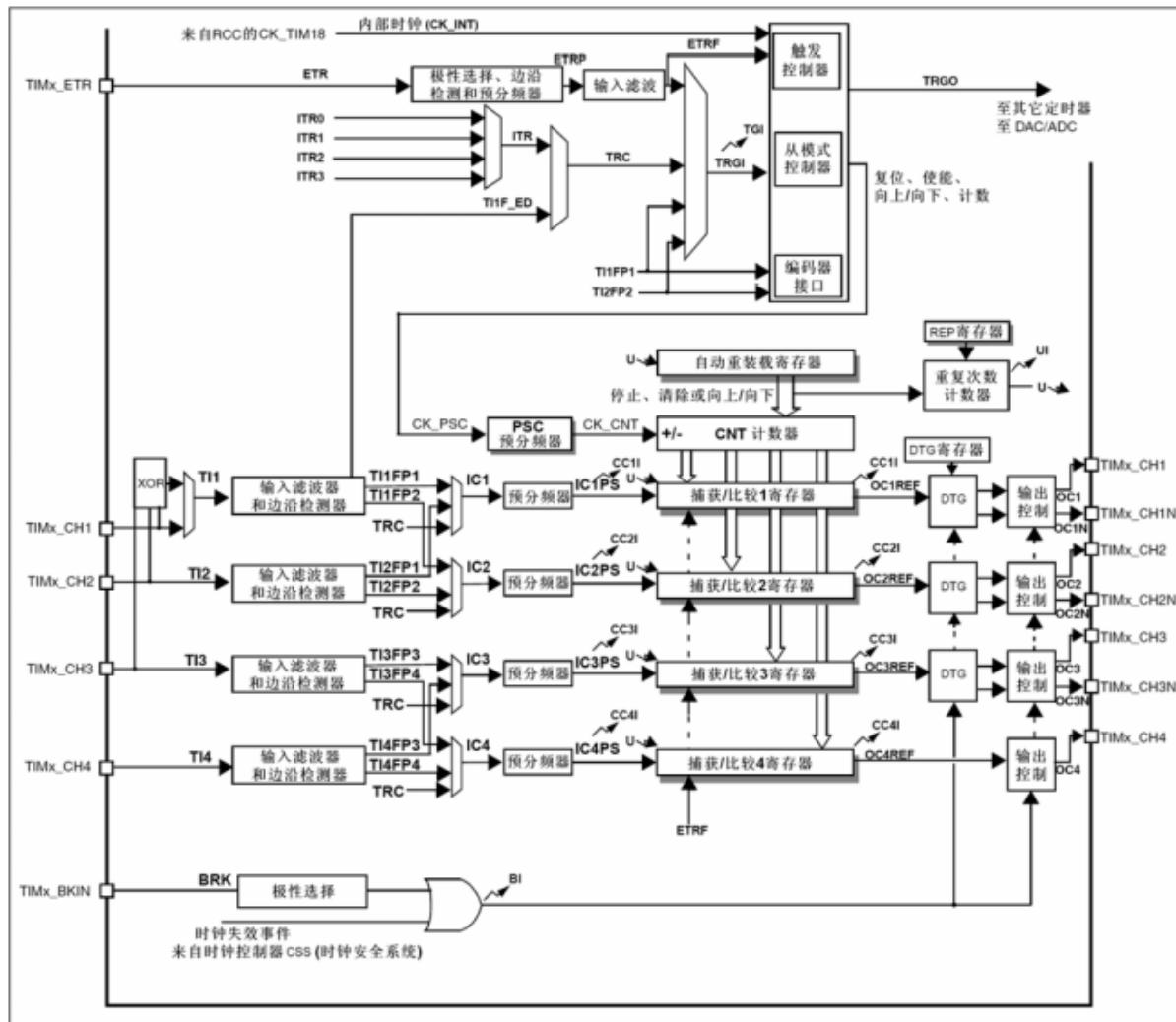
高级控制定时器(TIM1/1A/1B)由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较、PWM、嵌入死区时间的互补 PWM 等)。

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 :

- Reg** 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。

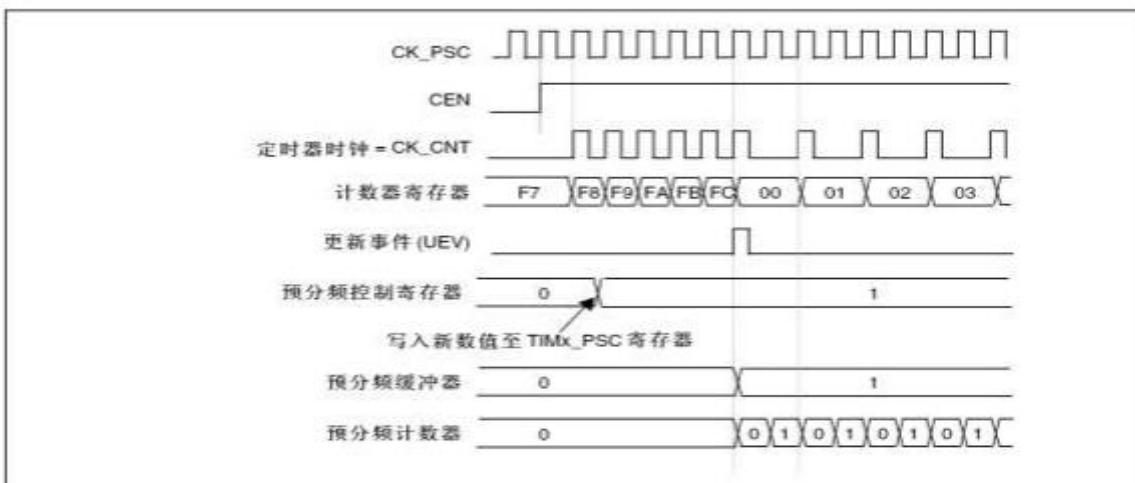


Figure 13- 2 当预分频器的参数从 1 变到 2 时，计数器的时序图

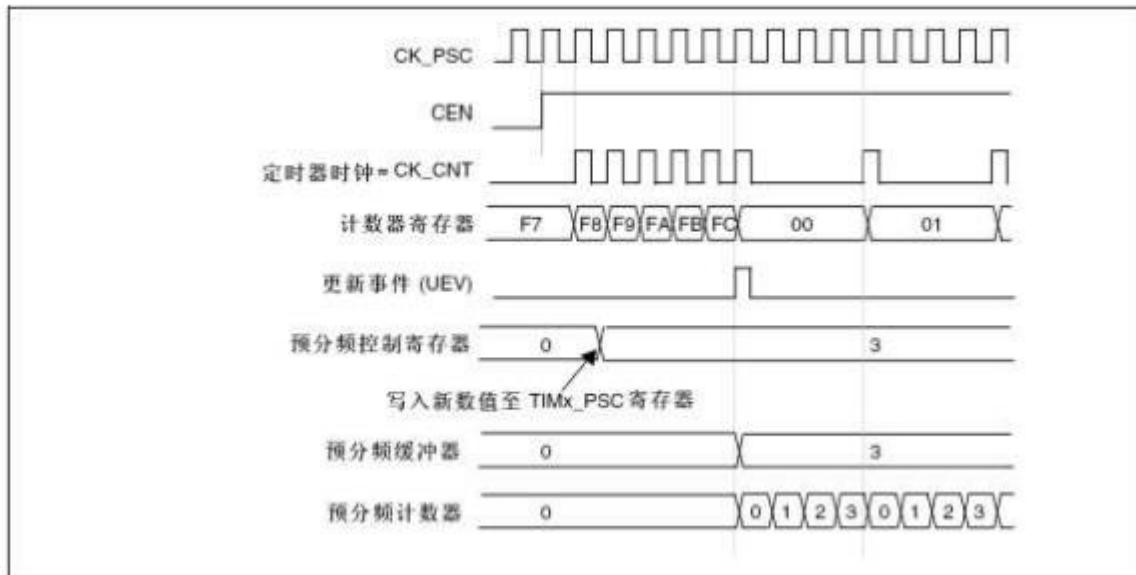


Figure 13- 3 当预分频器的参数从 1 变到 4 时，计数器的时序图

### 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 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。

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  $\text{TIM1X\_ARR}=0x36$  时计数器在不同时钟频率下的动作。

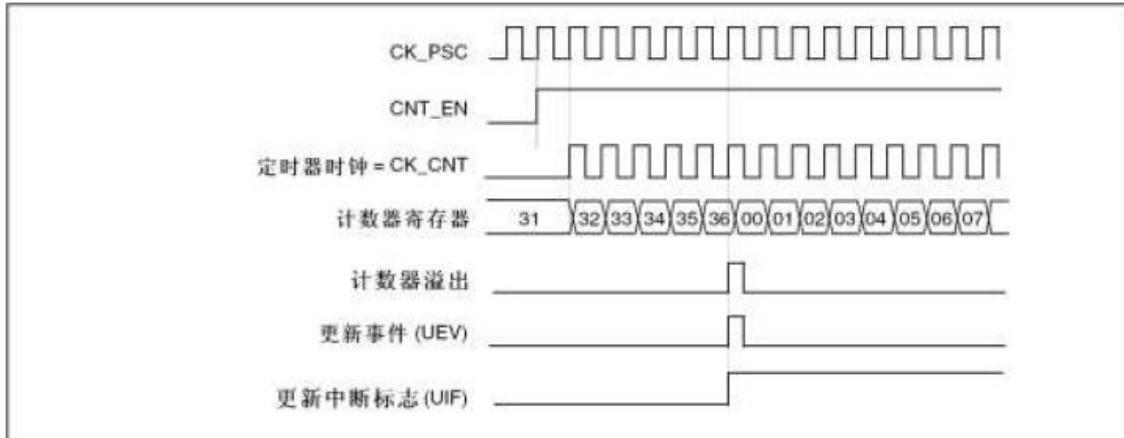


Figure 13- 4 计数器时序图：内部时钟分频因子为 1

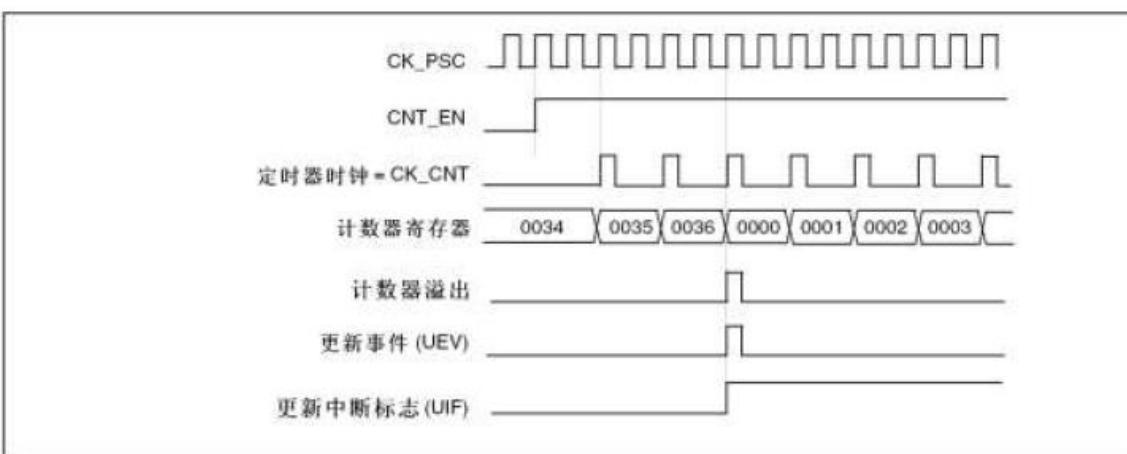


Figure 13- 5 计数器时序图：内部时钟分频因子为 2

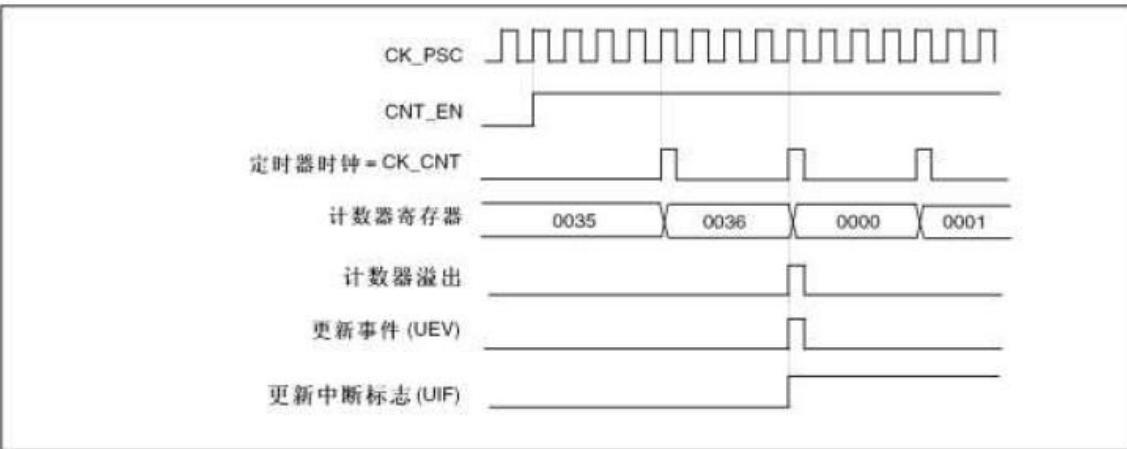


Figure 13- 6 计数器时序图：内部时钟分频因子为 4

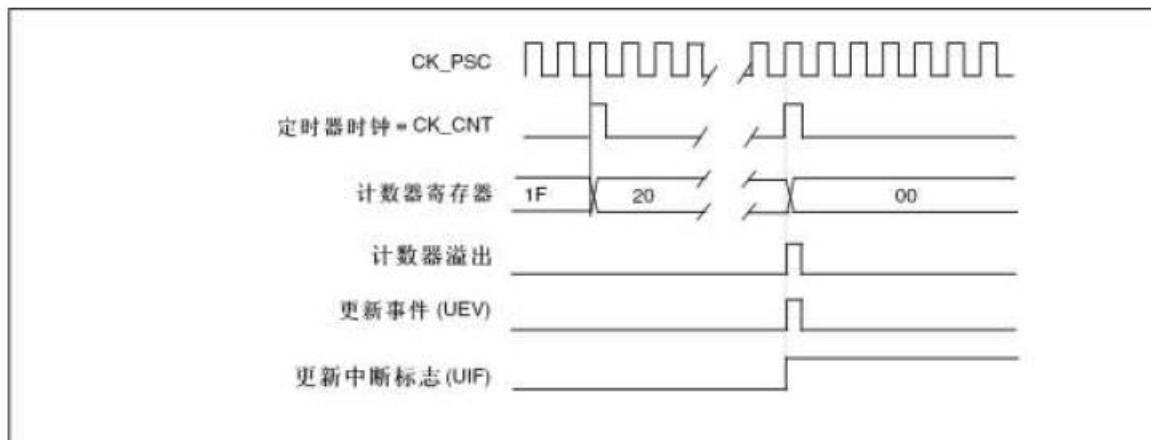


Figure 13- 7 计数器时序图：内部时钟分频因子为 N

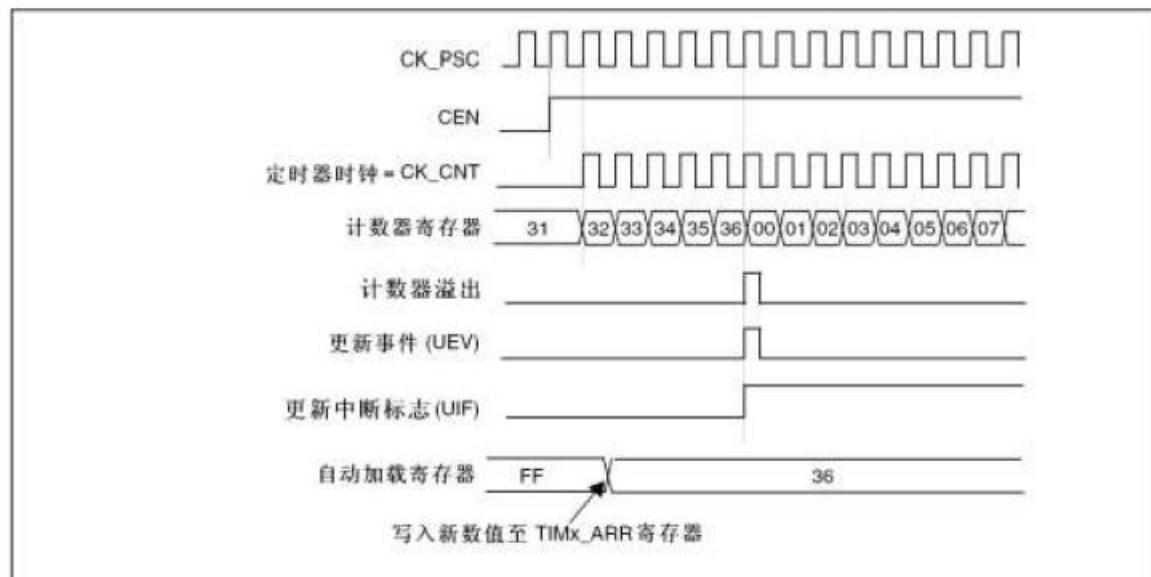


Figure 13- 8 计数器时序图：当 ARPE=0 时的更新事件(TIM1X\*\_ARR 没有预装入)

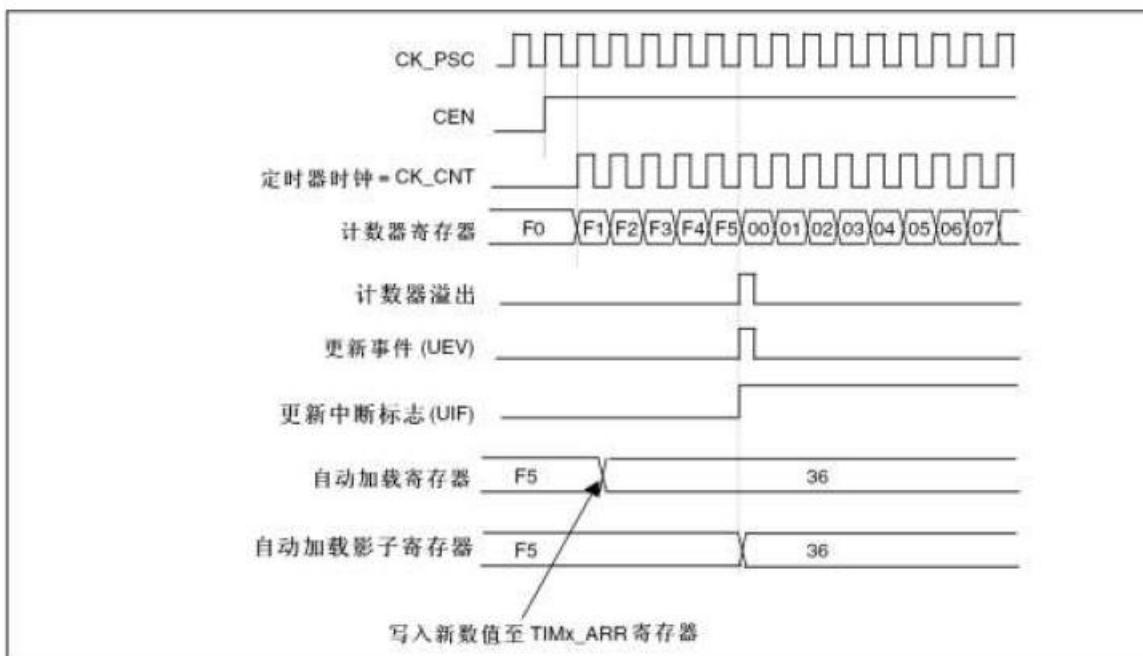


Figure 13- 9 计数器时序图：当 ARPE=1 时的更新事件(预装入了 TIM1X\*\_ARR)

### 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。

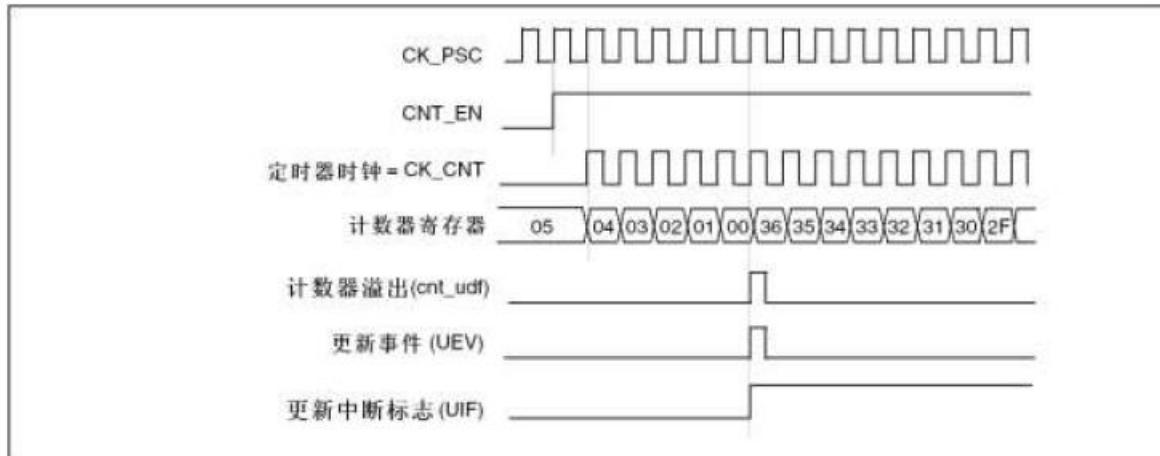


Figure 13- 10 计数器时序图：内部时钟分频因子为 1

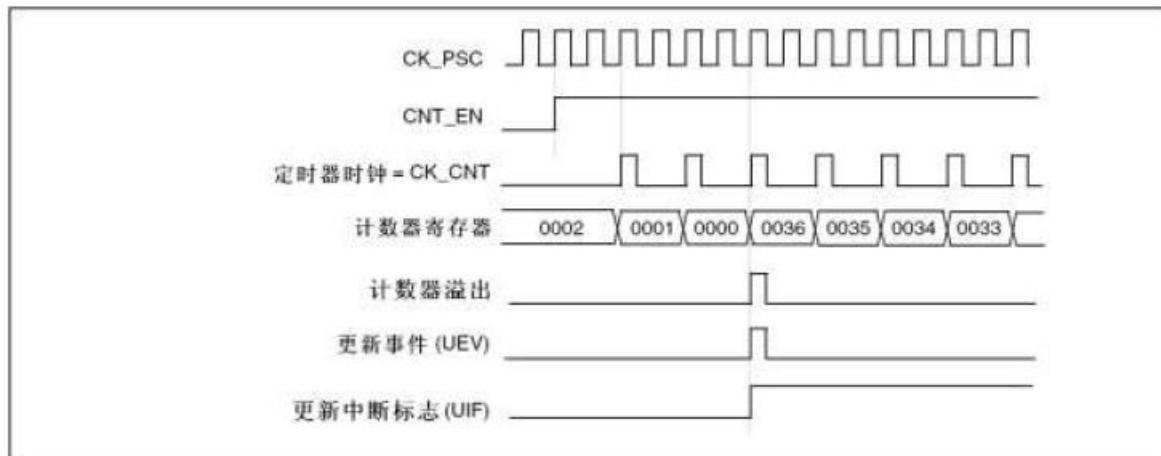


Figure 13- 11 计数器时序图：内部时钟分频因子为 2

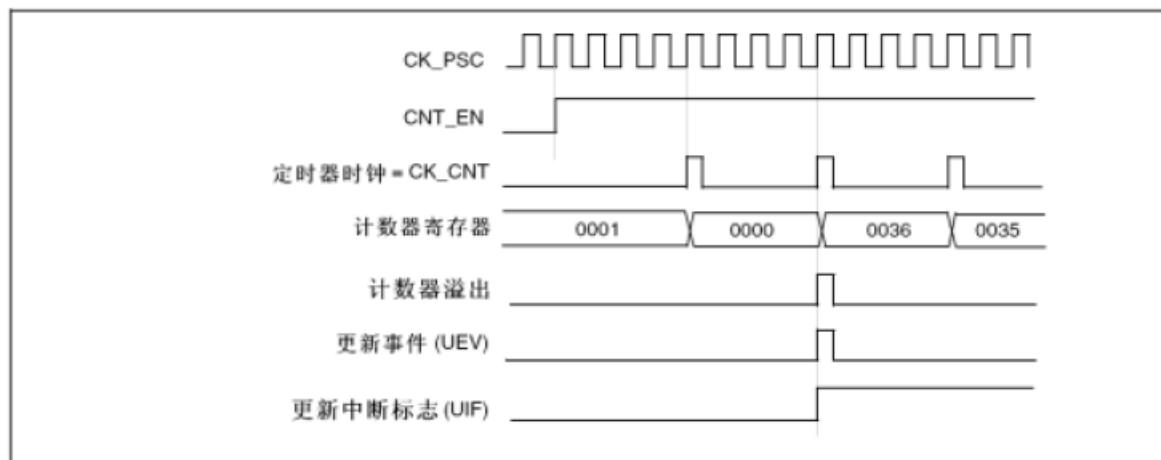


Figure 13- 12 计数器时序图：内部时钟分频因子为 4

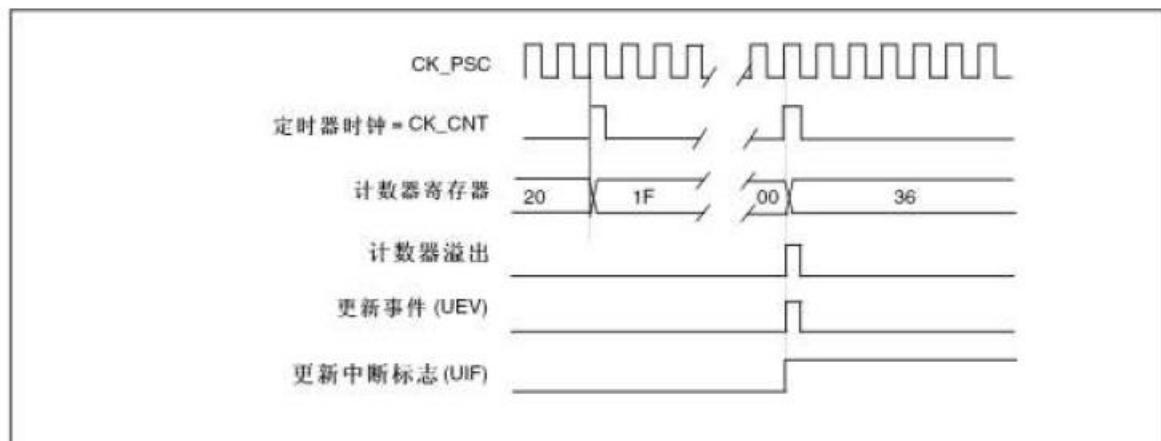


Figure 13- 13 计数器时序图：内部时钟分频因子为 N

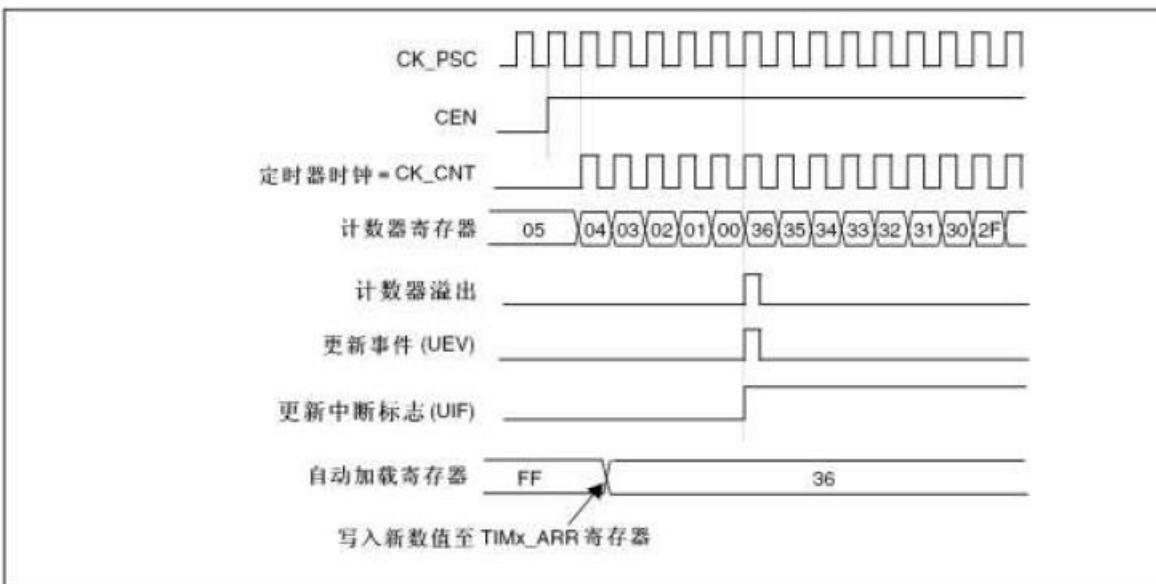


Figure 13- 14 计数器时序图：当没有使用重复计数器时的更新事件

### 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：

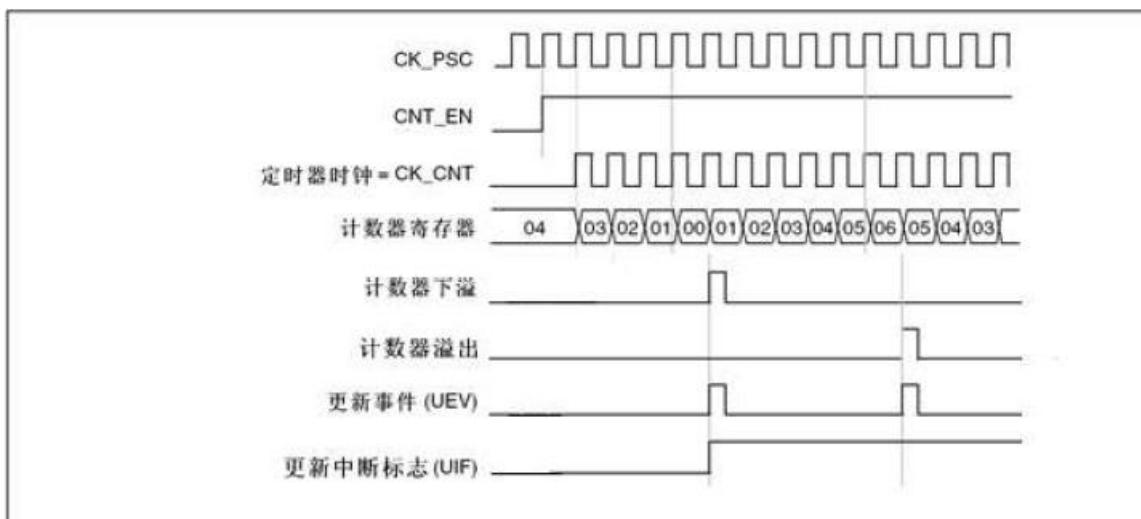


Figure 13- 15 计数器时序图：内部时钟分频因子为 1(**TIM1X\*\_ARR=0x6**)

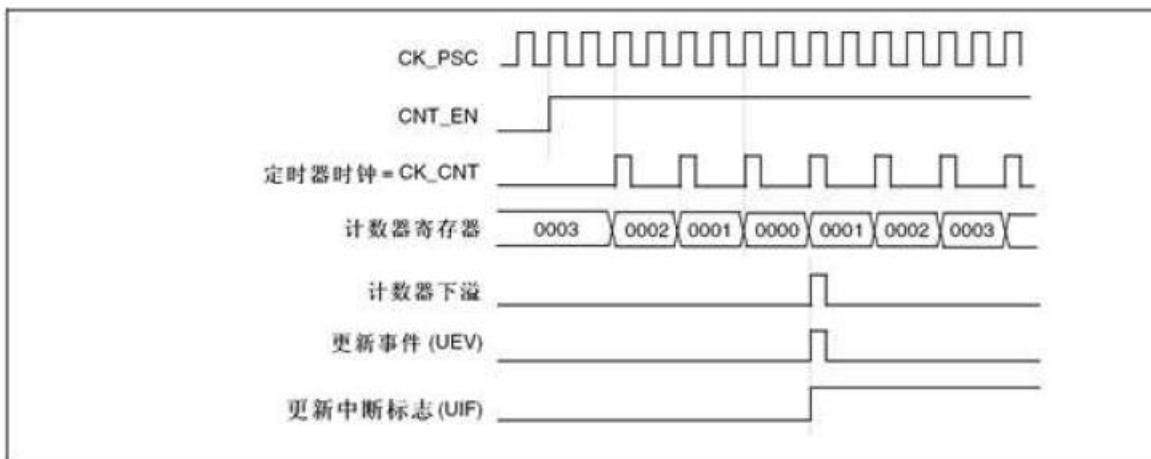


Figure 13- 16 计数器时序图：内部时钟分频因子为 2

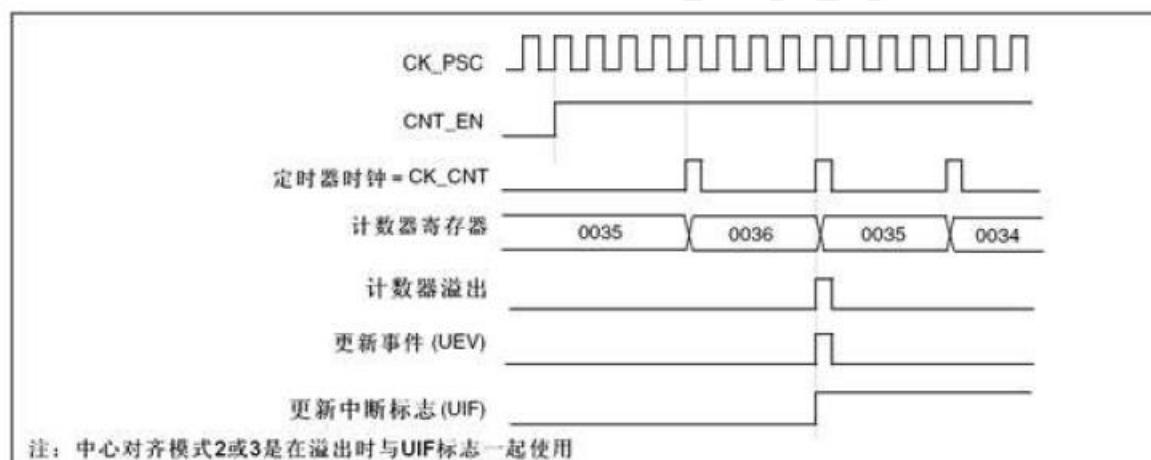


Figure 13- 17 计数器时序图：内部时钟分频因子为 4(TIM1X\*\_ARR=0x36)

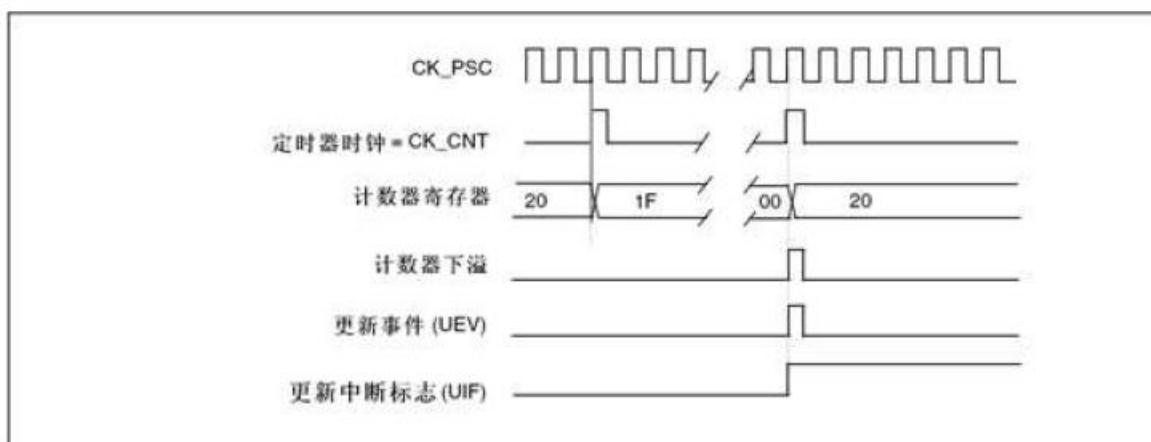


Figure 13- 18 计数器时序图：内部时钟分频因子为 N

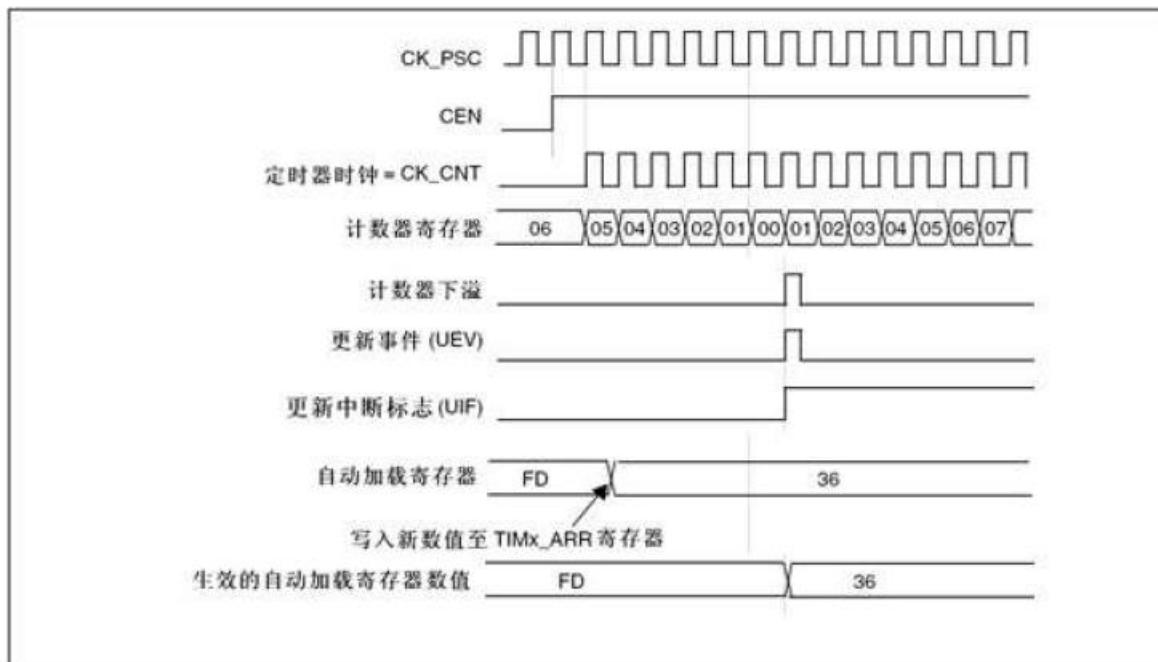


Figure 13- 19 计数器时序图：ARPE=1 时的更新事件(计数器下溢)

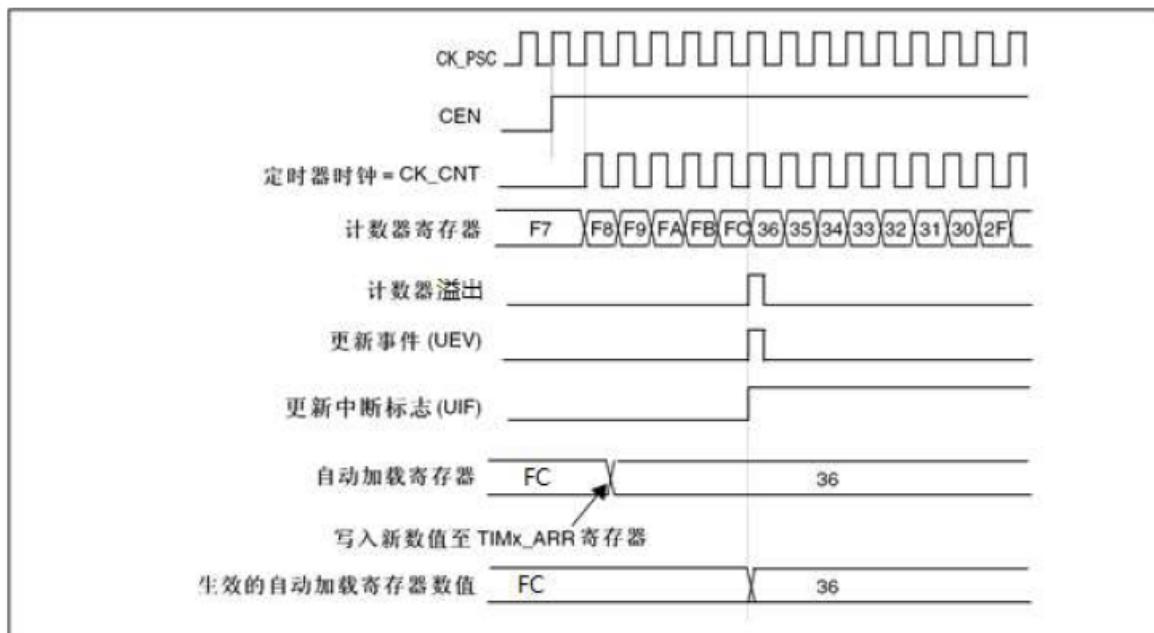


Figure 13- 20 计数器时序图：ARPE=1 时的更新事件(计数器溢出)

### 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。

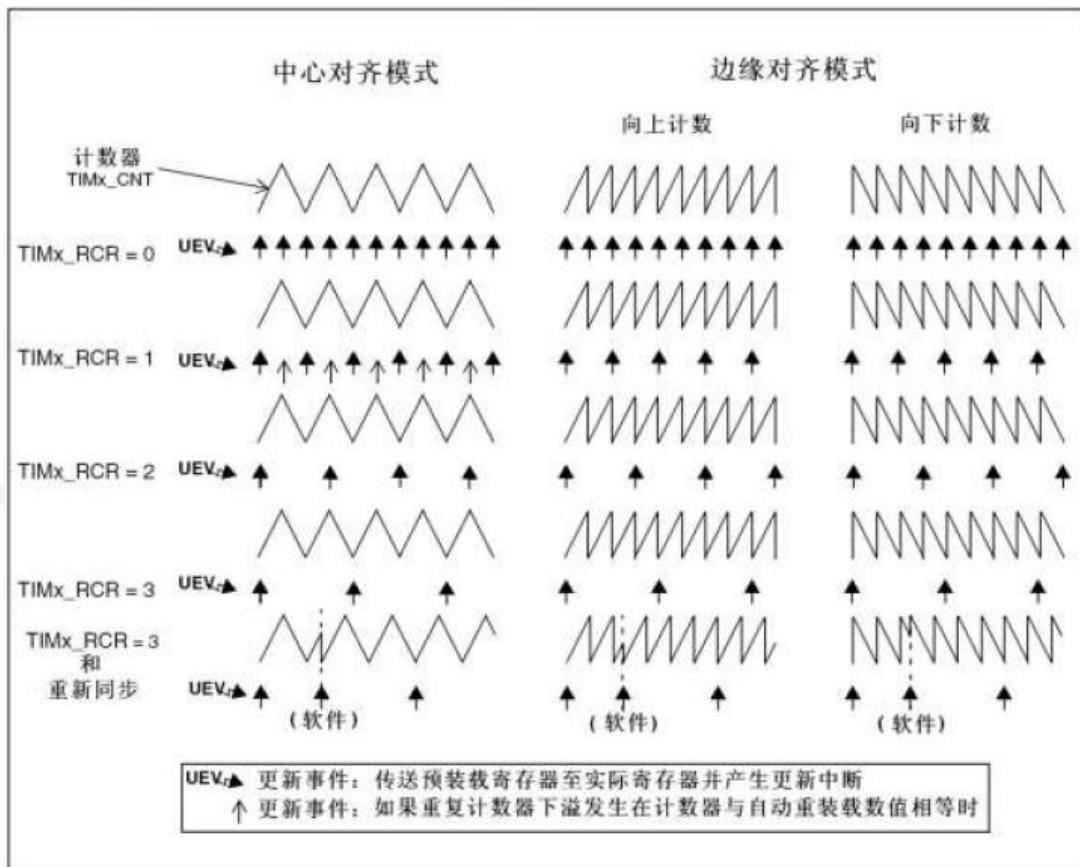


Figure 13- 21 不同模式下更新速率的例子，及 TIM1X\*\_RCR 的寄存器设置

### 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。

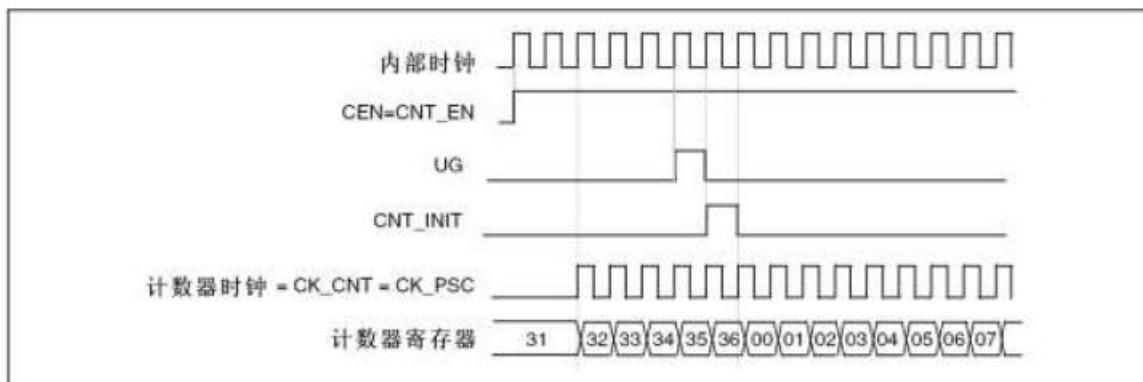


Figure 13- 22 一般模式下的控制电路，内部时钟分频因子为 1

### 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降沿计数。

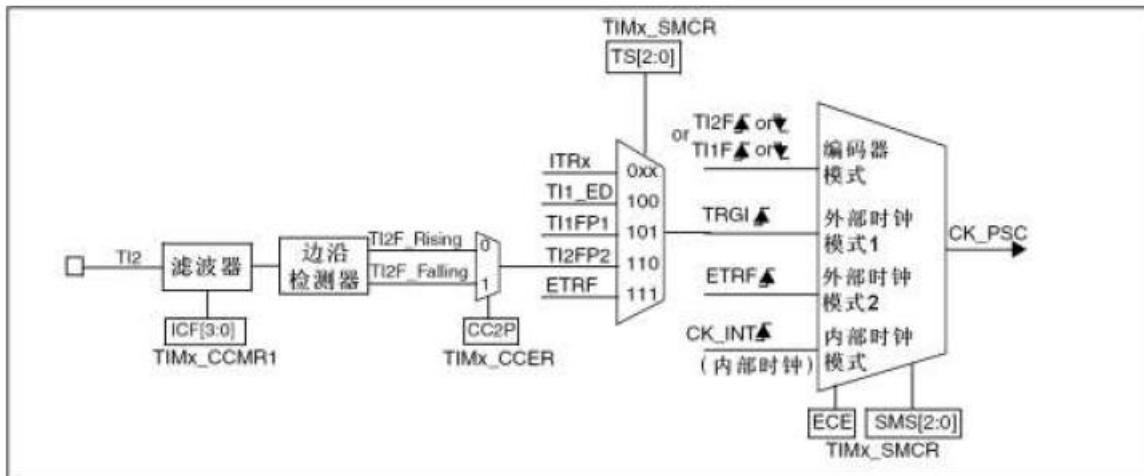


Figure 13- 23 TI2 外部时钟连接例子

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。

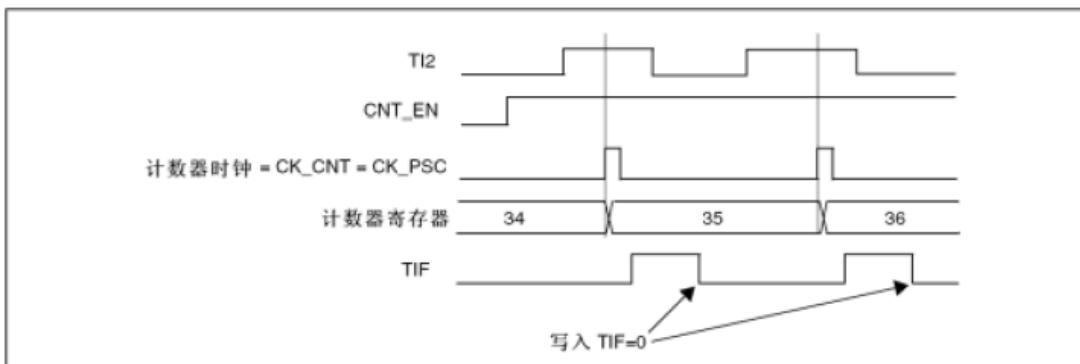


Figure 13- 24 外部时钟模式 1 下的控制电路

### 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

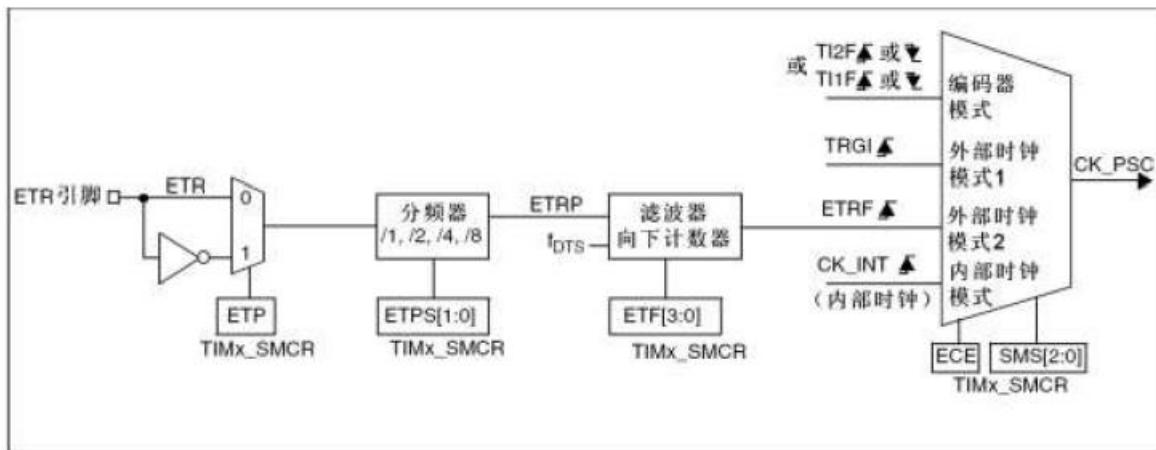


Figure 13- 25 外部触发输入框图

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。

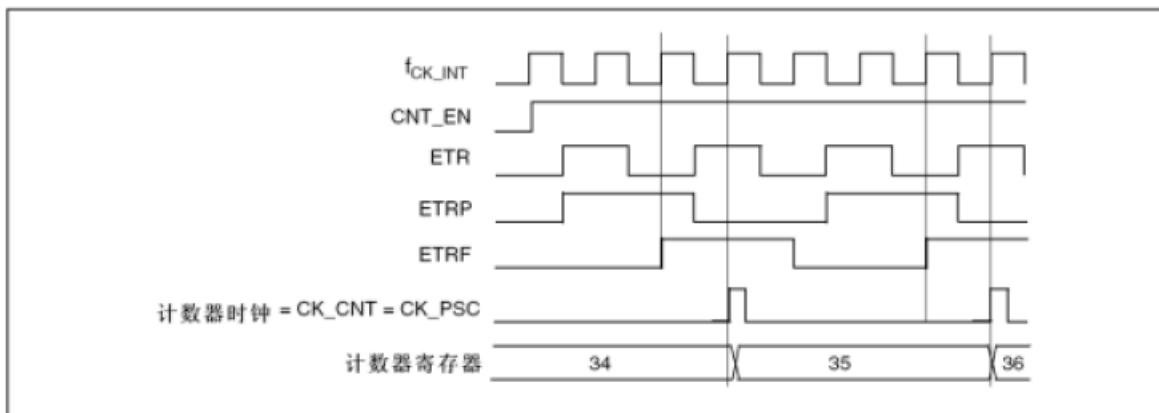


Figure 13- 26 外部时钟模式 2 下的控制电路

### 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)。

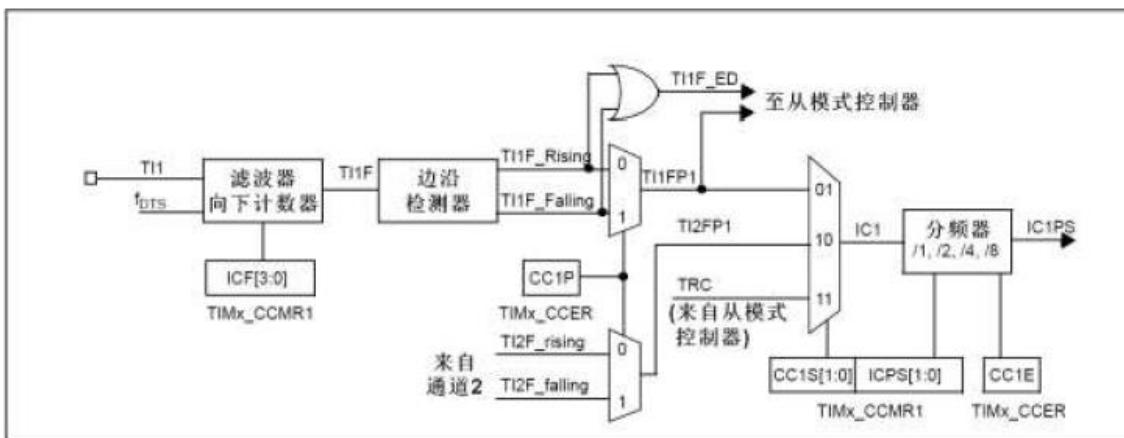


Figure 13- 27 捕获/比较通道(如：通道 1 输入部分)

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。

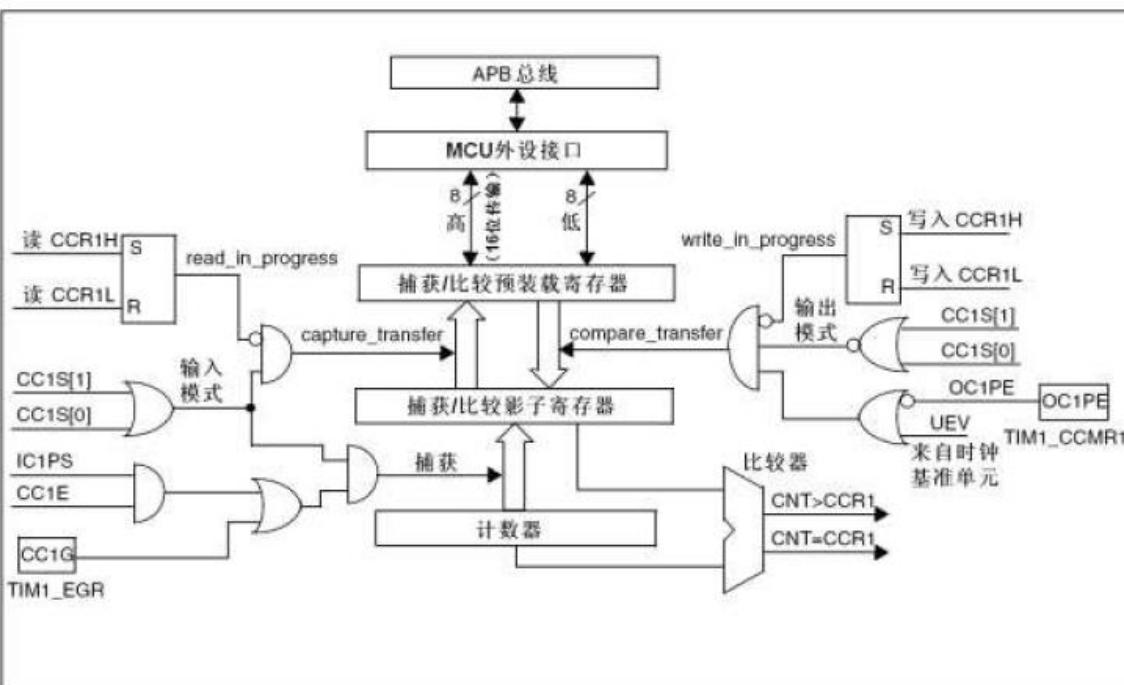


Figure 13- 28 捕获/比较通道 1 的主电路

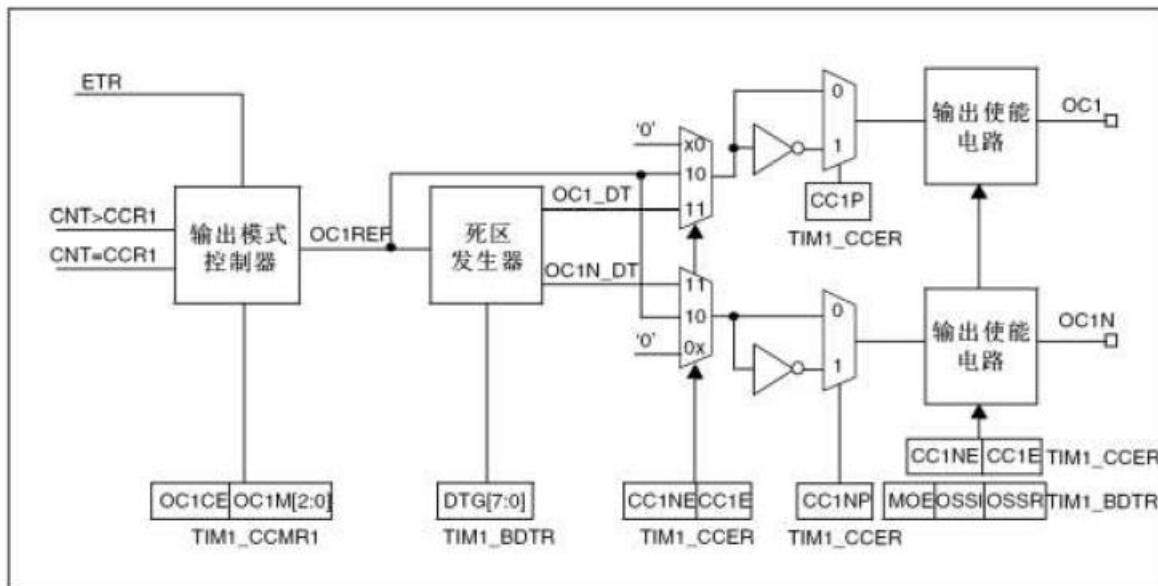


Figure 13- 29 捕获/比较通道的输出部分(通道 1 至 3)

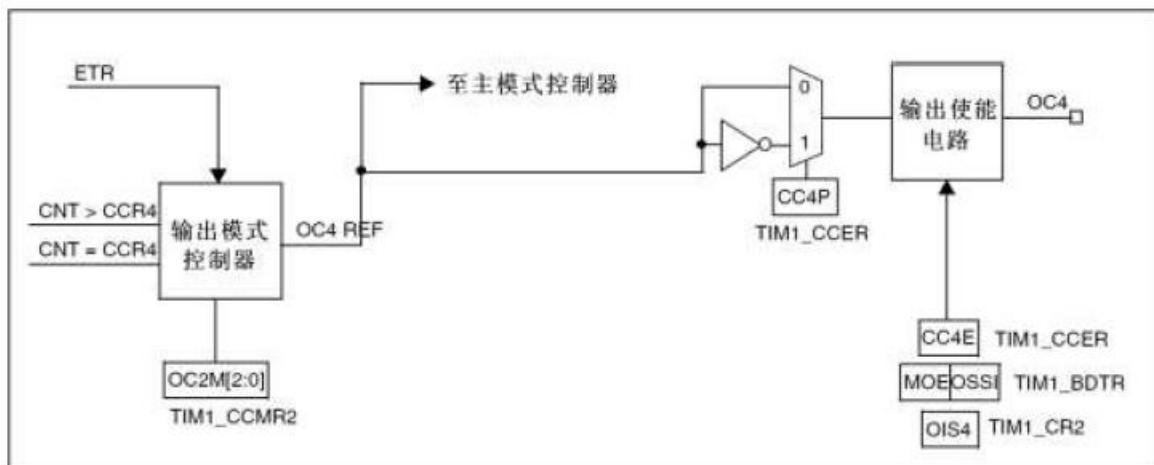


Figure 13- 30 捕获/比较通道的输出部分(通道 4)

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，如果开放了中断或者 DMA 操作，则将产生中断或者 DMA 操作。如果发生捕获事件时 CC<sub>x</sub>IF 标志已经为高，那么重复捕获标志 CC<sub>x</sub>OF (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，我们必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<sub>DTS</sub> 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 CC<sub>x</sub>G 位，可以通过软件产生输入捕获中断/或 DMA 请求。

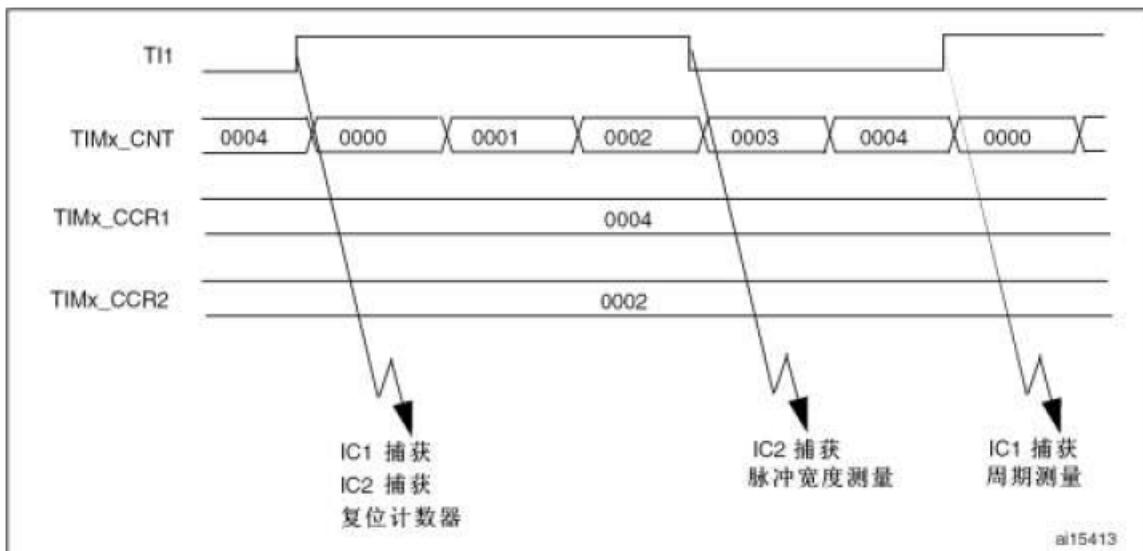
### 13.3-7PWM 输入模式

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(TIM1X\*\_CCMRx 寄存器中 CCxS=00)下，输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIM1X\*\_CCMRx 寄存器中相应的 OCxM=101，即可强置输出比较信号(OCxREF/OCx)为有效状态。

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

例如：CCxP=0(OCx 高电平有效)，则 OCx 被强置为高电平。

置 TIM1X\*\_CCMRx 寄存器中的 OCxM=100，可强置 OCxREF 信号为低。

该模式下，在 TIM1X\*\_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(TIM1X\*\_CCMRx 寄存器中的 OCxM 位)和输出极性(TIM1X\*\_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(TIM1X\*\_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIM1X\*\_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(TIM1\_DIER 寄存器中的 CCxDE 位，TIM1\_CR2 寄存器中的 CCDS 位选择 DMA 请求功能)，则产生一个 DMA 请求。

TIM1X\*\_CCMRx 中的 OCxPE 位选择 TIM1X\*\_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤：

- (1) 选择计数器时钟(内部，外部，预分频器)。
- (2) 将相应的数据写入 TIM1X\*\_ARR 和 TIM1X\*\_CCRx 寄存器中。
- (3) 如果要产生一个中断请求，设置 CCxIE 位。
- (4) 选择输出模式，例如：
  - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚，设置 OCxM=011
  - 置 OCxPE = 0 禁用预装载寄存器
  - 置 CCxP = 0 选择极性为高电平有效
  - 置 CCxE = 1 使能输出
- (5) 设置 TIM1X\*\_CR1 寄存器的 CEN 位启动计数器

TIM1X\*\_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器

( $OCxPE = '0'$ ，否则  $TIM1X^*_CCRx$  的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

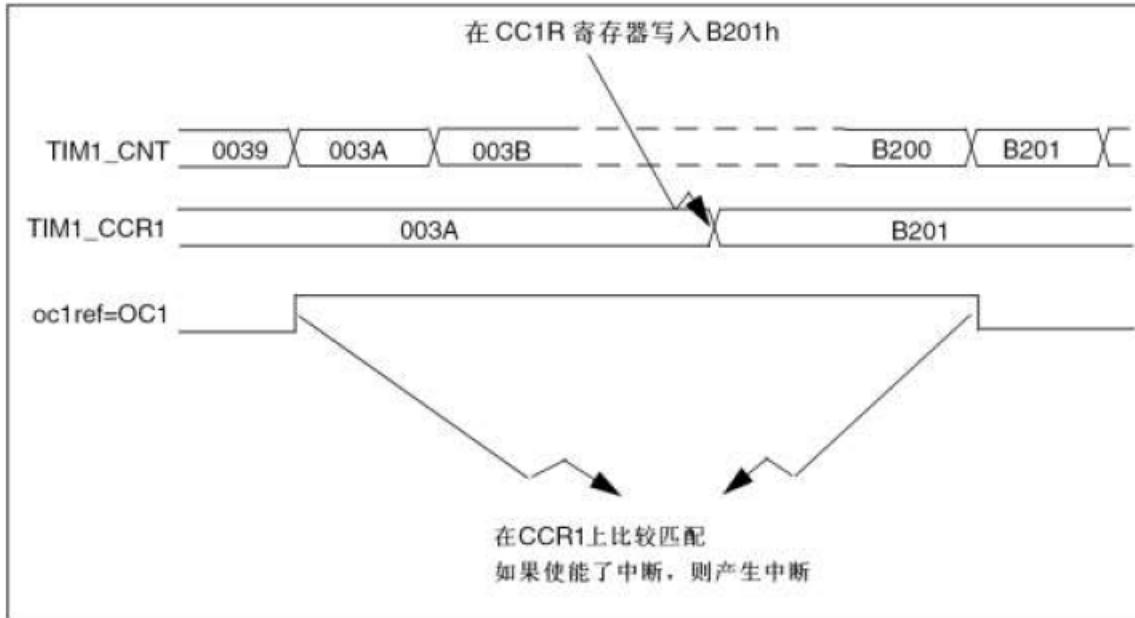


Figure 13- 32 输出比较模式, 翻转 OC1

### 13.3-10PWM 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  $TIM1X^*_CCRx \leq TIM1X^*_CNT$  或者  $TIM1X^*_CNT \leq TIM1X^*_CCRx$ 。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\*\_CCR<sub>x</sub> 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIM1X\*\_CCR<sub>x</sub> 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。

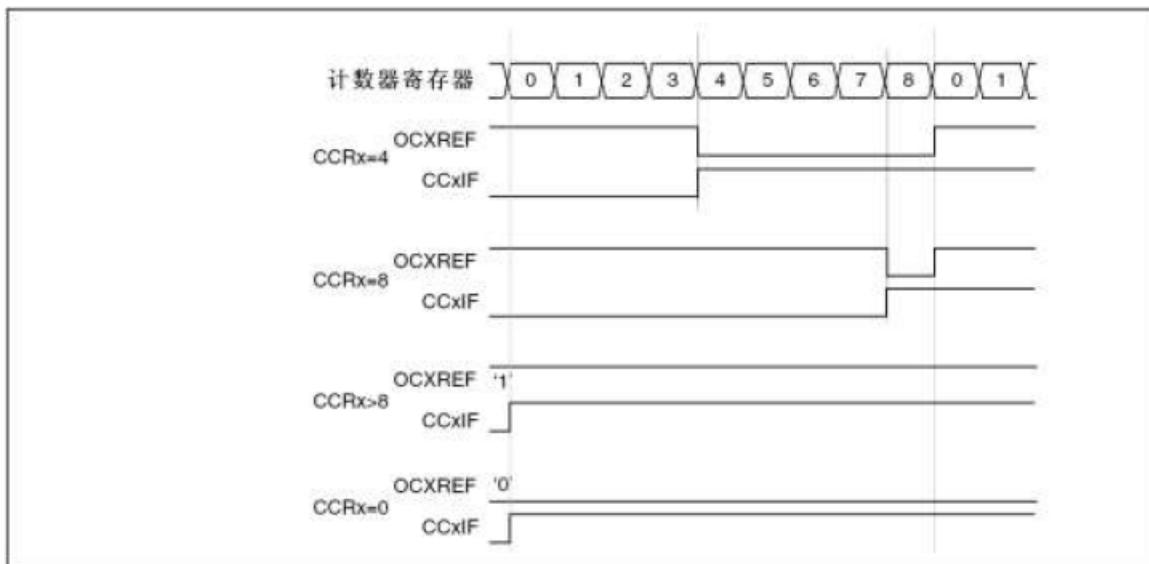


Figure 13- 33 边沿对齐的 PWM 波形(ARR=8)

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\*\_CCR<sub>x</sub> 时参考信号 OCxREF 为低，否则为高。如果 TIM1X\*\_CCR<sub>x</sub> 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 `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 `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。`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

- `TIM1X*_ARR=8`
- `PWM 模式 1`
- `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。

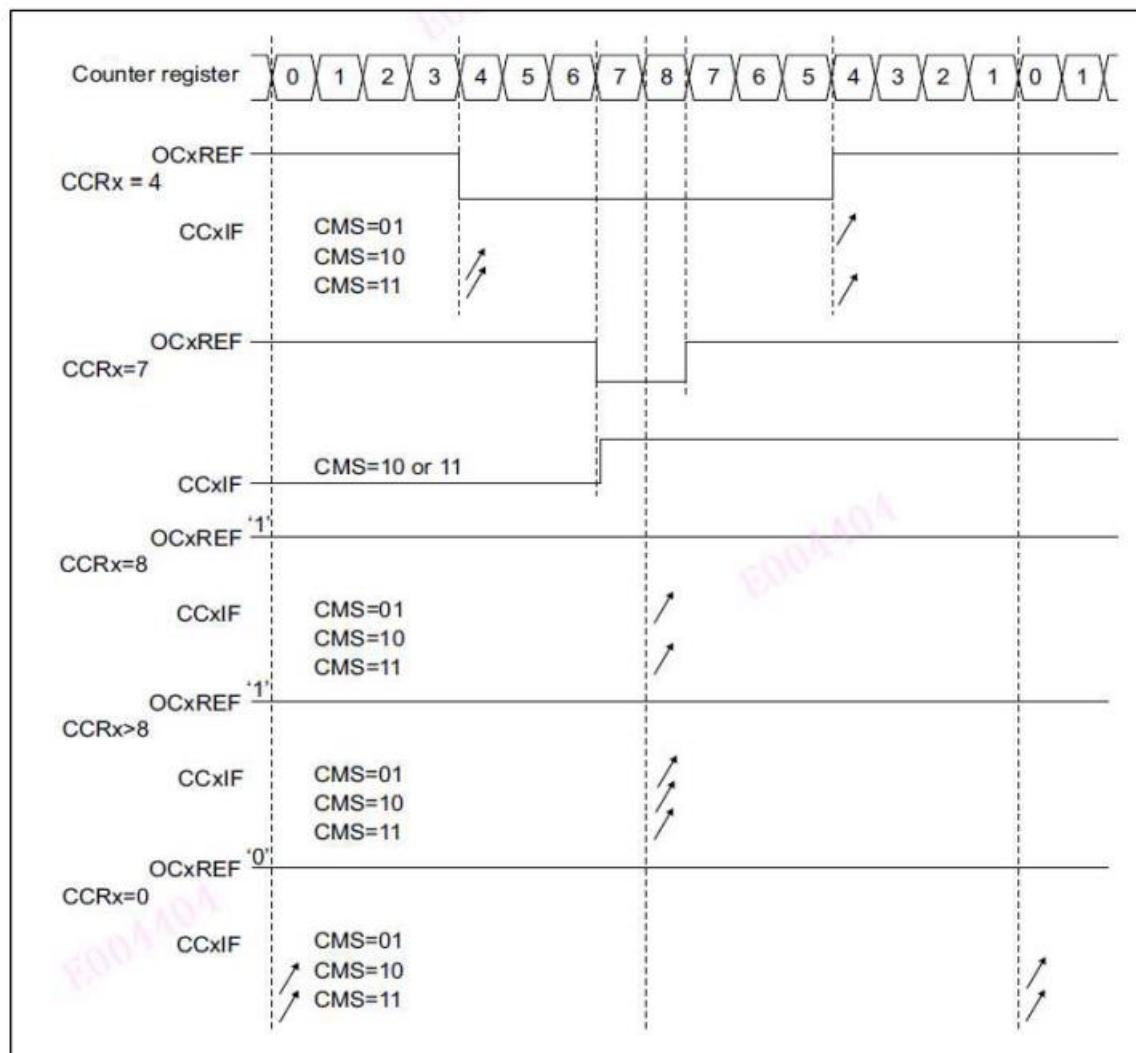


Figure 13- 34 中央对齐的 PWM 波形(APR=8)

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)

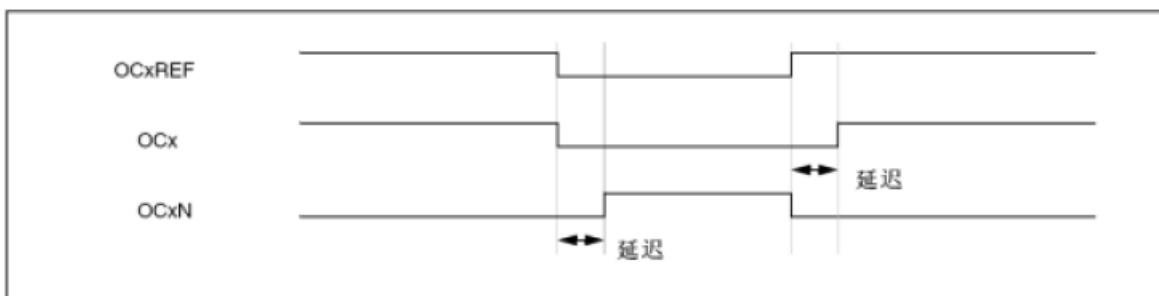


Figure 13- 35 带死区插入的互补输出

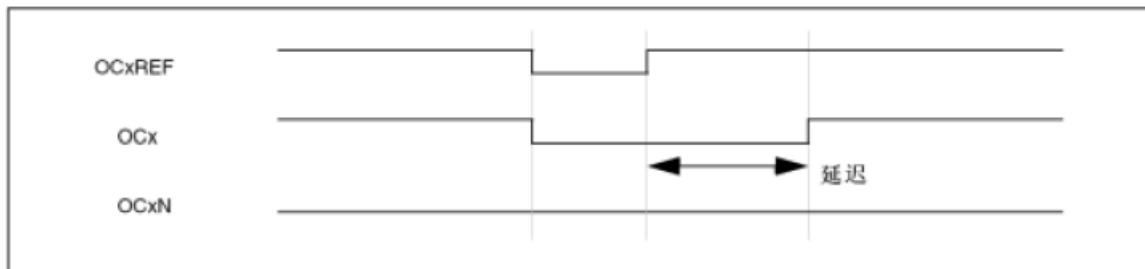


Figure 13- 36 死区波形延迟大于负脉冲

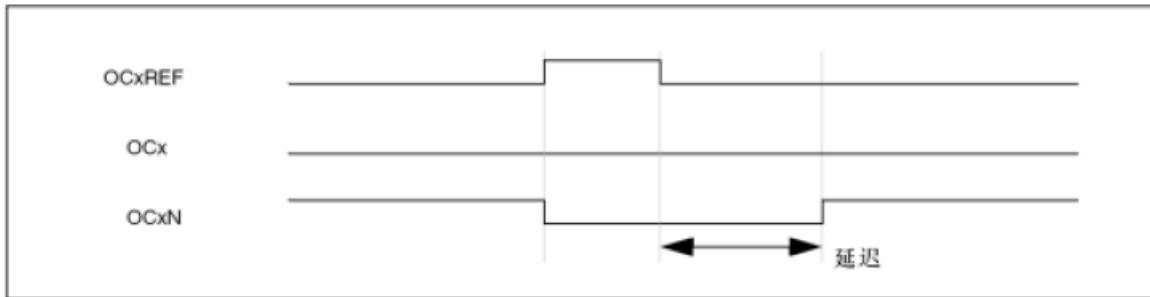


Figure 13-37 死区波形延迟大于正脉冲

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。

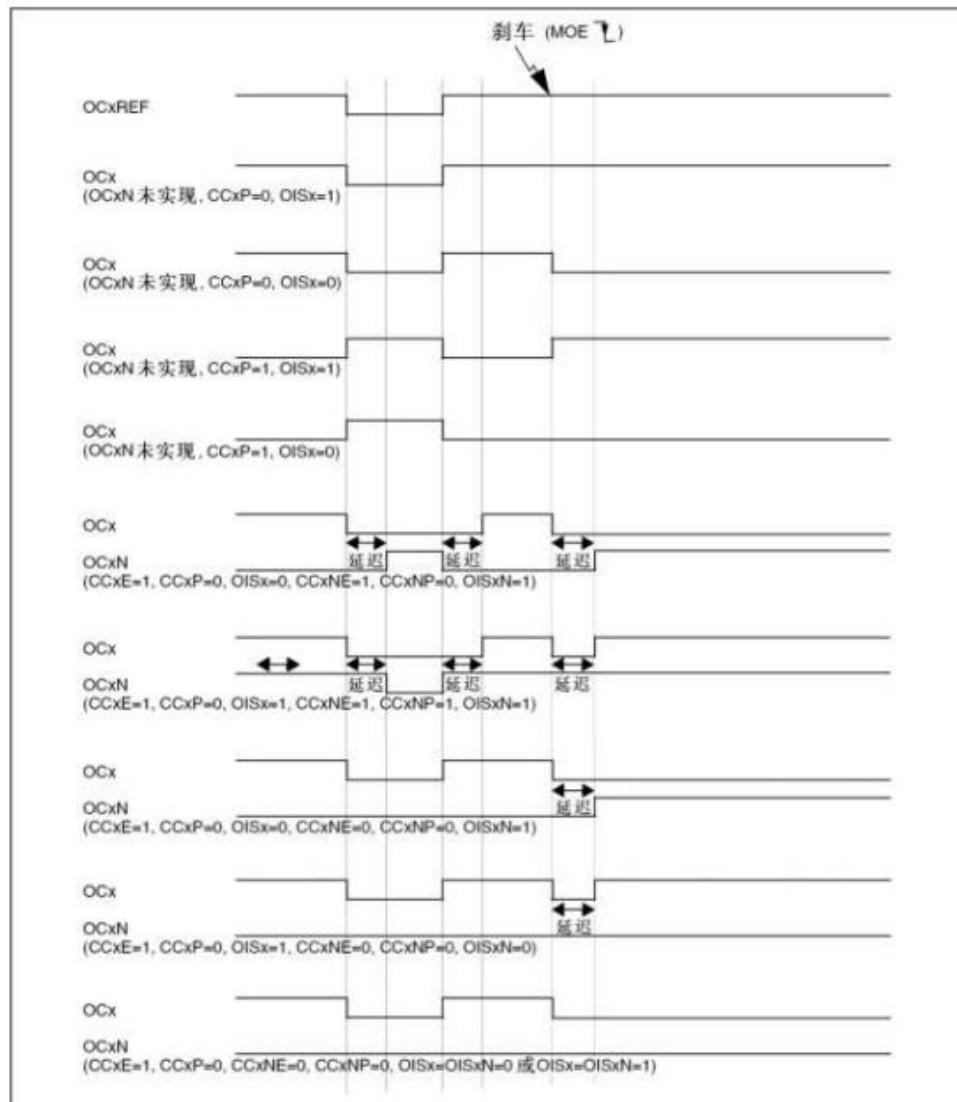


Figure 13- 38 响应刹车的输出

### 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。

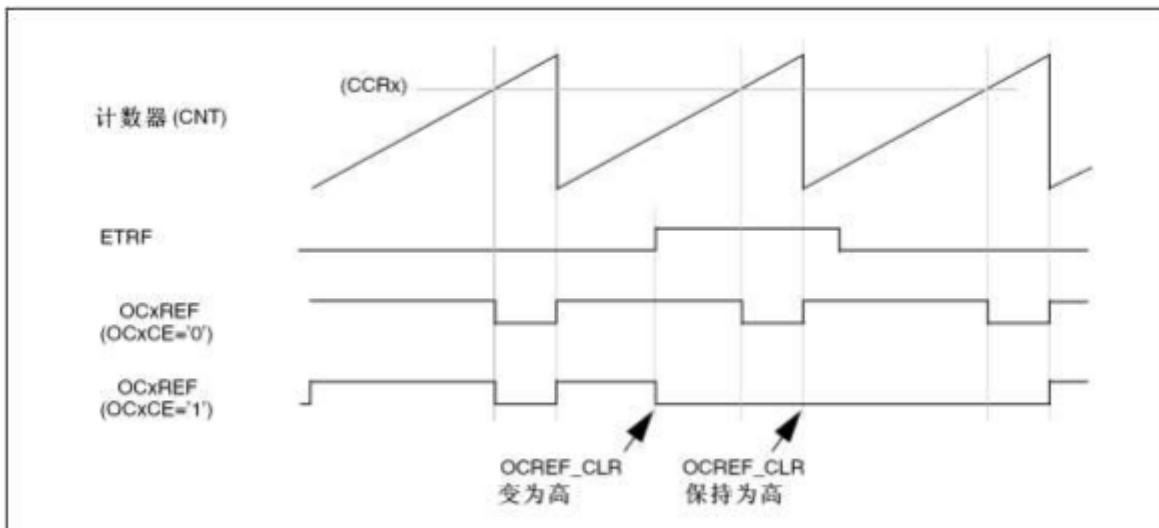


Figure 13- 39 清除 TIM1X\* 的 OCxREF

### 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。

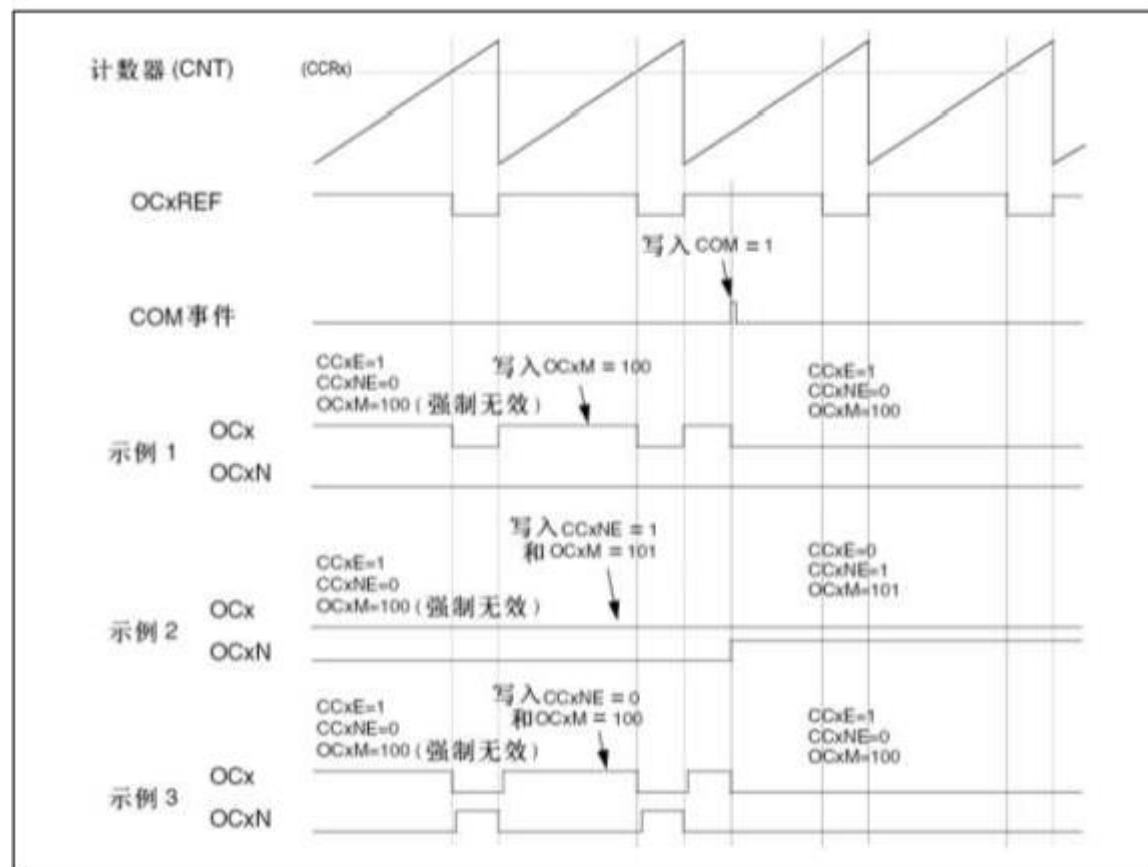


Figure 13- 40 产生六步 PWM，使用 COM 的例子(OSSR=1)

### 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  $CNT > CCRx$

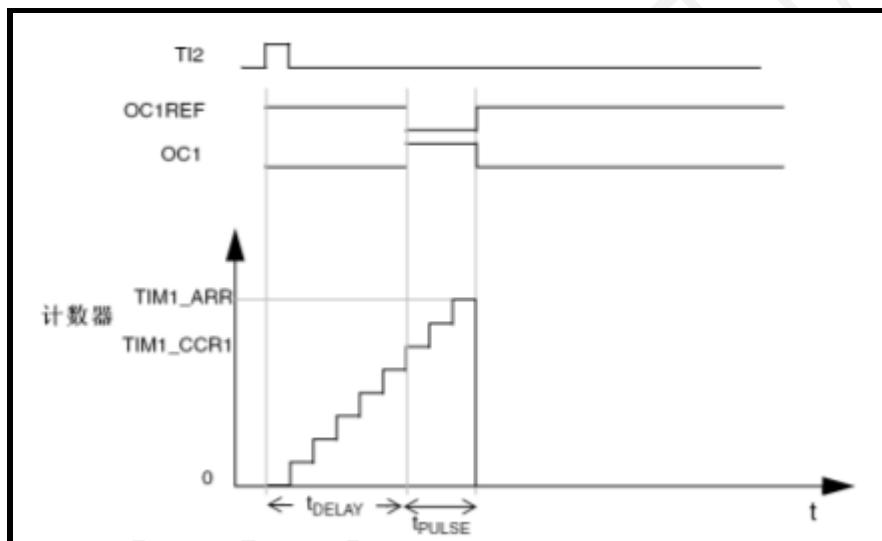


Figure 13-41 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟  $tDELAY$  之后，在 OC1 上产生一个长度为  $tPULSE$  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)

- $tDELAY$  由 TIM1X\*\_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( $TIM1X*_ARR - TIM1X*_CCR1$ )。
- 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

选择编码器接口模式的方法是：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。

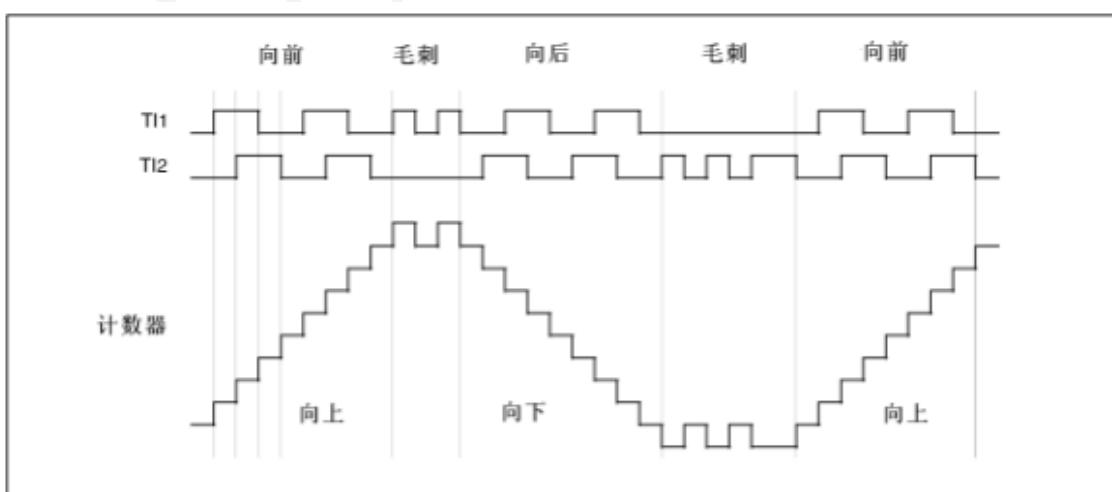
有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

**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)

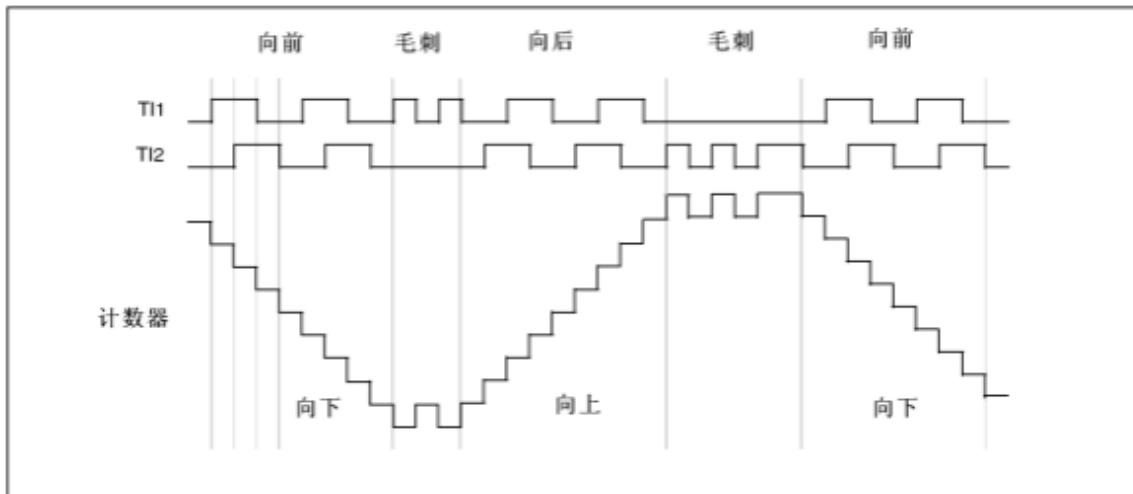


Figure 13- 43 IC1FP1 反相的编码器接口模式实例

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。

### 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。

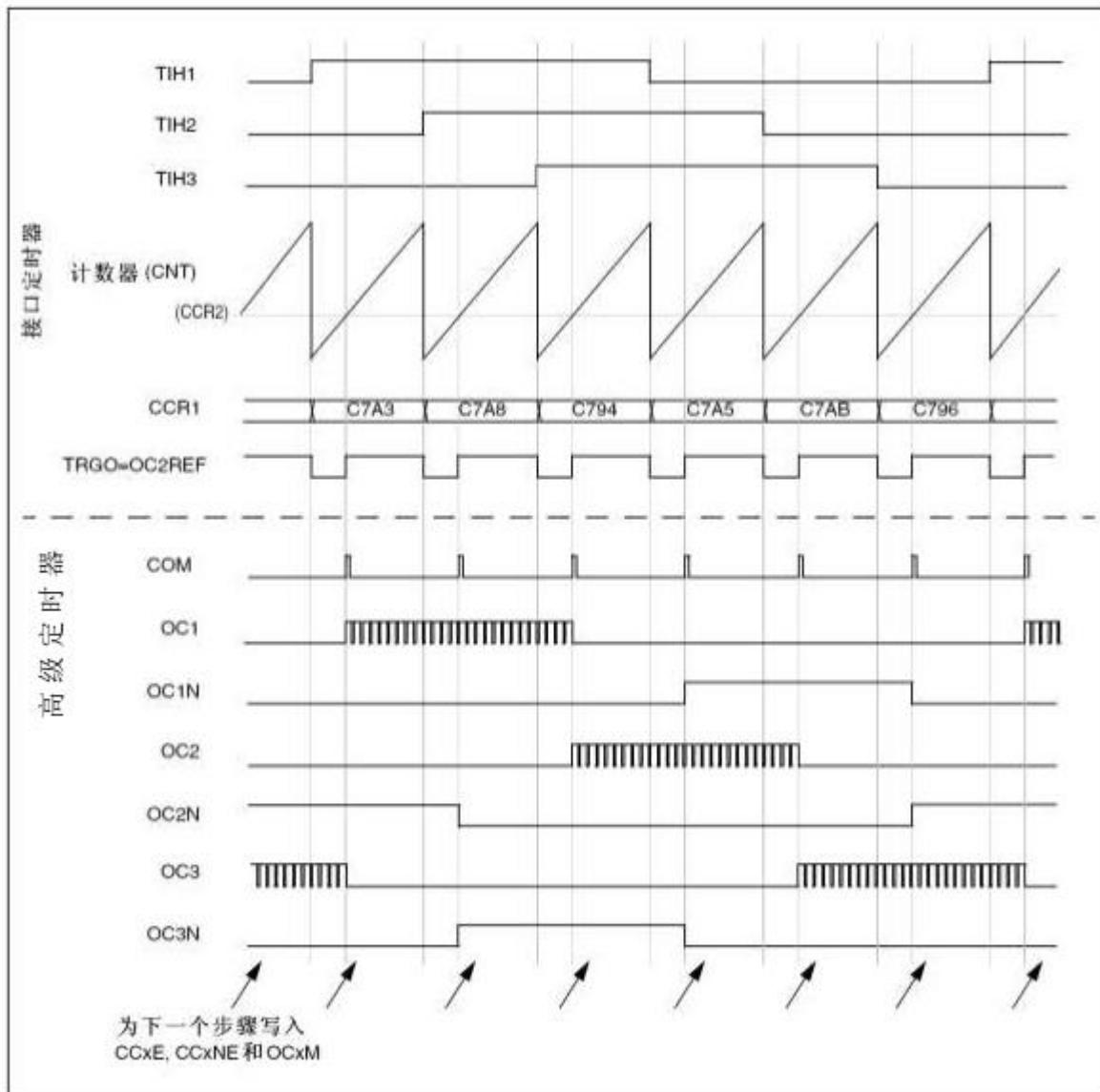
“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级定时器 TIM1X\* 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1X\*。

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;

### 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。

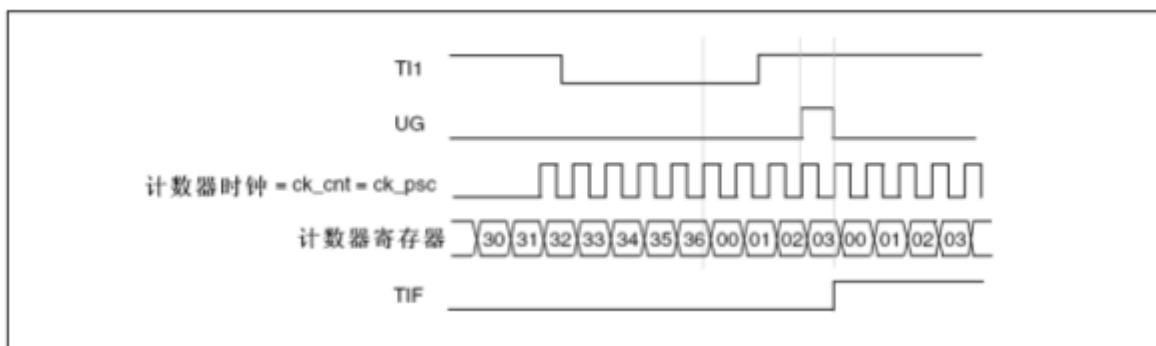


Figure 13- 45 复位模式下的控制电路

### 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。

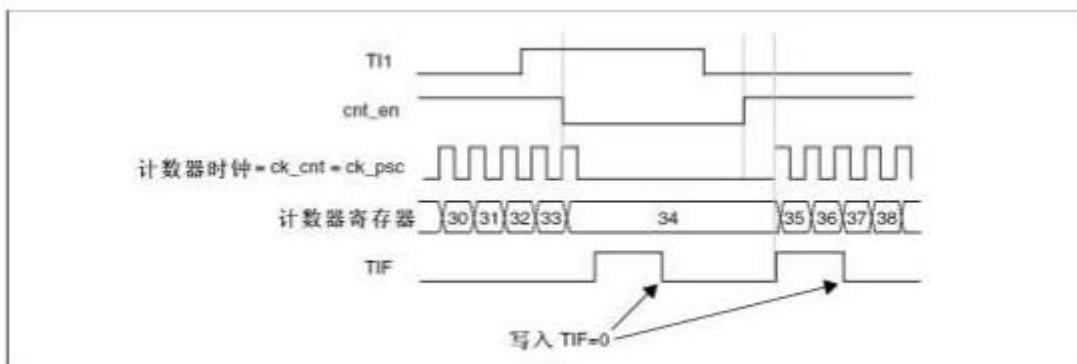


Figure 13-46 门控模式下的控制电路

### 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。

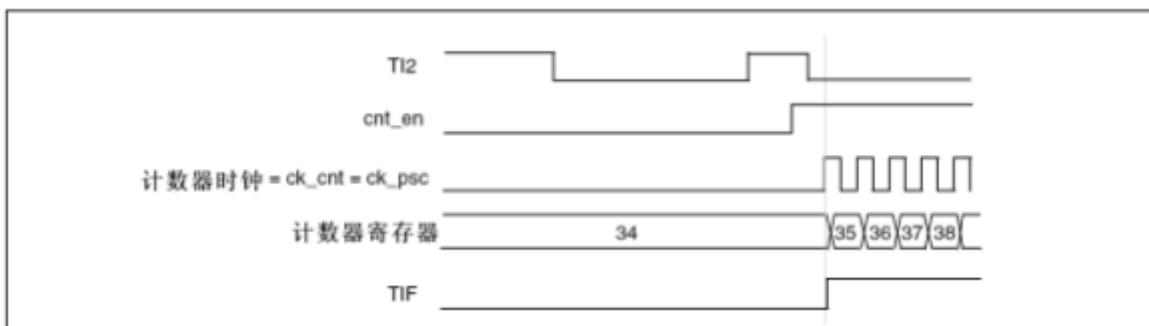


Figure 13- 47 触发器模式下的控制电路

### 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。

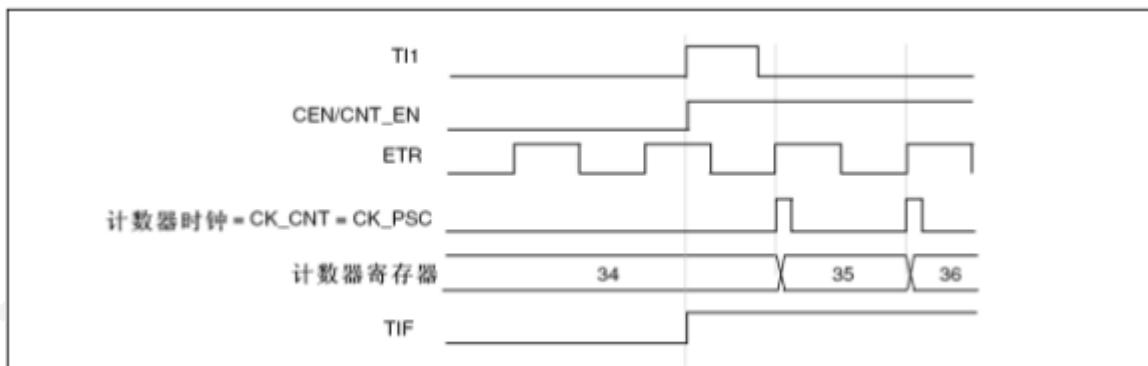


Figure 13- 48 外部时钟模式 2 + 触发模式下的控制电路

### 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(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** 基址 **0x4000 1000**

**TIM1A** 基址 **0x4000 1200**

**TIM1B** 基址 **0x4000 1300**

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

Note : TIM1X\* → TIM1 , TIM1A ,TIM1B

Table 13- 2 TIM1,TIM1A,TIM1B 寄存器列表和复位值

### 13.5 TIM1X\*寄存器说明

#### 13.5-1TIM1X\* 控制寄存器 1(TIMX\*\_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	16
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS		CEN	
--				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]	<p>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。</p> <p>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</p>	0x0	R/W
7	ARPE	<p>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。</p>	0	R/W
6:5	CMS[1:0]	<p>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。</p>	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

### 13.5-2TIM1X\* 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	16
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[1:0]	CCDS	CCUS	保留	CCPC	--	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
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

位	标记	功能描述	复位值	读写
6:4	MMS[2:0]	<p>主模式选择(Master mode selection)</p> <p>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：</p> <p>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。</p> <p>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)。</p> <p>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。</p> <p>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)。</p> <p>100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。</p> <p>101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。</p> <p>110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。</p> <p>111: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。</p>	0x0	R/W
3	CCDS	<p>捕获/比较 DMA 请求选择</p> <p>0 : 当CCX事件发生时会产生CCX DMA请求</p> <p>1 : 当CCX更新事件发生时会产生CCX DMA请求</p>	0	R/W
2	CCUS	<p>捕获/比较控制更新选择(Capture/compare control update selection)</p> <p>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；</p> <p>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。</p> <p>注：该位只对具有互补输出的通道起作用。</p>	0	R/W
1	保留	始终读为 0。	0	--
0	CCPC	<p>捕获/比较预装载控制位(Capture/compare preloaded control)</p> <p>0: CCxE, CCxNE 和 OCxM 位不是预装载的；</p> <p>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。</p> <p>注：该位只对具有互补输出的通道起作用。</p>	0	R/W

### 13.5-3TIM1X\* 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	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]		保留		SMS[2:0]			
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	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

位	标记	功能描述	复位值	读写
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。</p> <p>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  0001: 采样频率fSAMPLING=fCK_INT, N=2  0010: 采样频率fSAMPLING=fCK_INT, N=4  0011: 采样频率fSAMPLING=fCK_INT, N=8  0100: 采样频率fSAMPLING=fDTS/2, N=6  0101: 采样频率fSAMPLING=fDTS/2, N=8  0110: 采样频率fSAMPLING=fDTS/4, N=6  0111: 采样频率fSAMPLING=fDTS/4, N=8  1000: 采样频率fSAMPLING=fDTS/8, N=6  1001: 采样频率fSAMPLING=fDTS/8, N=8  1010: 采样频率fSAMPLING=fDTS/16, N=5  1011: 采样频率fSAMPLING=fDTS/16, N=6  1100: 采样频率fSAMPLING=fDTS/16, N=8  1101: 采样频率fSAMPLING=fDTS/32, N=5  1110: 采样频率fSAMPLING=fDTS/32, N=6  1111: 采样频率fSAMPLING=fDTS/32, N=8 </p>	0	R/W
7	MSM	<p>主/从模式(Master/slave mode)</p> <p>0: 无作用；  1: 触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。</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)  001: 内部触发 1(ITR1); 101: 滤波后的定时器输入 1(TI1FP1)  010: 内部触发 2(ITR2); 110: 滤波后的定时器输入 2(TI2FP2)  011: 内部触发 3(ITR3); 111: 外部触发输入(ETRF)</p> <p>更多有关ITRx 的细节，参见 Table13- 3。  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>	0	R/W
3	保留	始终读为 0。	0	--
2:0	SMS[2:0]	从模式选择(Slave mode selection) 当选择了外部信号，触发信号(TRGI)的有效边沿与选中的外部输入极	0	R/W

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

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

### 13.5-4TIM1X\* 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
保留	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
--	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

位	标记	功能描述	复位值	读写
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

### 13.5-5TIM1X\* 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
保留	CC4OF	CC3OF	CC1OF	CC1OF	保留	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF	--	--
--	RC W0		-	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	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 TIM1X*_CCR1 寄存器时，CC1IF 的状态已经为' 1' 。	0	RCW0
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	RCW0
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

位	标记	功能描述	复位值	读写
5	COMIF	<p>COM 中断标记(COM interrupt flag)</p> <p>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'。</p> <p>0: 无COM 事件产生； 1: COM 中断等待响应。</p>	0	RCW0
4	CC4IF	<p>捕获/比较 4 中断标记(Capture/Compare 4 interrupt flag)</p> <p>参考CC1IF 描述。</p>	0	RCW0
3	CC3IF	<p>捕获/比较 3 中断标记(Capture/Compare 3 interrupt flag)</p> <p>参考CC1IF 描述。</p>	0	RCW0
2	CC2IF	<p>捕获/比较 2 中断标记(Capture/Compare 2 interrupt flag)</p> <p>参考CC1IF 描述。</p>	0	RCW0
1	CC1IF	<p>捕获/比较 1 中断标记(Capture/Compare 1 interrupt flag)</p> <p>如果信道 CC1 配置为输出模式：</p> <p>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'。</p> <p>0: 无匹配发生； 1: TIM1X*_CNT 的值与TIM1X*_CCR1 的值匹配。</p> <p>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。</p> <p>如果信道 CC1 配置为输入模式：</p> <p>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'。</p> <p>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)。</p>	0	RCW0
0	UIF	<p>更新中断标记(Update interrupt flag)</p> <p>当产生更新事件时该位由硬件置' 1'。它由软件清' 0'。</p> <p>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 <a href="#">13.5-3 TIM1X* 从模式控制寄存器(TIM1X*_SMCR)</a>).</p>	0	RCW0

### 13.5-6TIM1X\* 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
保留								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
--								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

位	标记	功能描述	复位值	读写
1	CC1G	<p>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'。</p> <p>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。</p>	0	WO
0	UG	<p>产生更新事件(Update generation) 该位由软件置' 1'，由硬件自动清' 0'。</p> <p>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。</p>	0	WO

### 13.5-7TIM1X\* 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。

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		
OC2CE	OC2M[1:0]		OC2PE	OC2FE	CC2S[1:0]	OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]					
IC2F[3:0]		IC2PSC[1:0]				IC1F[3:0]			IC1PSC[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	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; 1: 一旦检测到ETRF 输入高电平, 清除OC1REF=0。	0	R/W

位	标记	功能描述	复位值	读写
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;TIM1X*_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
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

位	标记	功能描述	复位值	读写
2	OC1FE	<p>输出比较 1 快速使能(Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。</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) 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)。 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]	<p>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。</p>	0x0	R/W
7:4	IC1F[3:0]	<p>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  0001 : 采样频率fSAMPLING=fCK_INT, N=2  0010 : 采样频率fSAMPLING=fCK_INT, N=4  0011 : 采样频率fSAMPLING=fCK_INT, N=8  0100 : 采样频率fSAMPLING=fDTS/2, N=6  0101 : 采样频率fSAMPLING=fDTS/2, N=8  0110 : 采样频率fSAMPLING=fDTS/4, N=6  0111 : 采样频率fSAMPLING=fDTS/4, N=8  1000 : 采样频率fSAMPLING=fDTS/8, N=6  1001 : 采样频率fSAMPLING=fDTS/8, N=8  1010 : 采样频率fSAMPLING=fDTS/16, N=5  1011 : 采样频率fSAMPLING=fDTS/16, N=6  1100 : 采样频率fSAMPLING=fDTS/16, N=8  1101 : 采样频率fSAMPLING=fDTS/32, N=5  1110 : 采样频率fSAMPLING=fDTS/32, N=6  1111 : 采样频率fSAMPLING=fDTS/32, N=8</p>	0x0	R/W

3:2	IC1PSC[1:0]	<p>输入/捕获 1 预分频器(Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。</p> <p>一旦CC1E=0(TIM1X*_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；</p> <p>01: 每 2 个事件触发一次捕获；</p> <p>10: 每 4 个事件触发一次捕获；</p> <p>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)</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

### 13.5-8TIM1X\* 捕获/比较模式寄存器 2(TIM1X\*\_CCMR2)

地址偏移: 0x1C 复位值: 0x0000 0000

参考以上 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[1:0]		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]	<p>捕获/比较 3 选择。(Capture/Compare 3 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: CC3 信道被配置为输出;</p> <p>01: CC3 信道被配置为输入, IC3 映射在 TI3 上;</p> <p>10: CC3 信道被配置为输入, IC3 映射在 TI4 上;</p> <p>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)。</p> <p>注: CC3S 仅在通道关闭时(TIM1X*_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。	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]	<p>捕获/比较 4 选择(Capture/Compare 4 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: CC4 信道被配置为输出;</p> <p>01: CC4 信道被配置为输入, IC4 映射在 TI4 上;</p> <p>10: CC4 信道被配置为输入, IC4 映射在 TI3 上;</p> <p>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)。</p> <p>注: CC4S 仅在通道关闭时(TIM1X*_CCER 寄存器的CC4E=0)才是可写的。</p>	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]	<p>捕获/比较 3 选择(Capture/Compare 3 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: CC3 信道被配置为输出;</p> <p>01: CC3 信道被配置为输入, IC3 映射在 TI3 上;</p> <p>10: CC3 信道被配置为输入, IC3 映射在 TI4 上;</p> <p>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)。</p> <p>注: CC3S 仅在通道关闭时(TIM1X*_CCER 寄存器的CC3E=0)才是可写的。</p>	0x0	R/W

### 13.5-9TIM1X\* 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
保留	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E	--
--															R/W

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 outputpolarity) 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 output enable) 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 outputpolarity) 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 output enable) 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)</p> <p>0: OC1N 高电平有效; 1: OC1N 低电平有效。</p> <p>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)</p> <p>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:</p> <p>0: OC1 高电平有效; 1: OC1 低电平有效。</p> <p>CC1 信道配置为输入:</p> <p>该位选择是IC1 还是IC1 的反相信号作为触发或捕获信号。</p> <p>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。</p> <p>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:</p> <p>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。</p> <p>CC1 信道配置为输入:</p> <p>该位决定了计数器的值是否能捕获入TIM1X*_CCR1 寄存器。</p> <p>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。	
		0	1	1		
		1	0	0	关闭状态(输出使能且为无效电平)	
		1	0	1	异步地: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1;	
		1	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		

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。

### 13.5-10TIM1X\* 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

### 13.5-11TIM1X\* 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

### 13.5-12TIM1X\* 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。	0x0	--
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。	0x0	R/W

### 13.5-13TIM1X\* 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。	0x0	--
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；	0x0	R/W

### 13.5-14TIM1X\* 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。	0x0	--
15:0	CCR1[15:0]	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。 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。	0x0	R/W

### 13.5-15TIM1X\* 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。	0x0	--
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。	0x0	R/W

### 13.5-16TIM1X\* 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。	0x0	--
15:0	CCR3[15:0]	<p>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。</p>	0x0	R/W

### 13.5-17TIM1X\* 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]	<p>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。</p>	0x0	R/W

### 13.5-18TIM1X\* 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															

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	<p>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)。</p>	0	R/W
14	AOE	<p>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' 1' , 则该位不能被修改。</p>	0	R/W
13	BKP	<p>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' 1' , 则该位不能被修改。  注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。</p>	0	R/W

位	标记	功能描述	复位值	读写
12	BKE	<p>刹车功能使能(Break enable)</p> <p>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)。</p> <p>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。</p> <p>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)</p> <p>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。</p> <p>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))。</p> <p>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。</p> <p>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)</p> <p>该位用于当MOE=0 且通道设为输出时。</p> <p>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))。</p> <p>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。</p> <p>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

位	标记	功能描述	复位值	读写
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:</p> <p>DTG[7:5]=0xx =&gt; DT=DTG[7:0] × Tdtg, Tdtg = TDTS;</p> <p>DTG[7:5]=10x =&gt; DT=(64+DTG[5:0])×Tdtg, Tdtg=2×TDTS;</p> <p>DTG[7:5]=110 =&gt; DT=(32+DTG[4:0])×Tdtg, Tdtg=8×TDTS;</p> <p>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:</p> <p>0 到 15875ns, 若步长时间为 125ns;</p> <p>16us 到 31750ns, 若步长时间为 250ns;</p> <p>32us 到 63us, 若步长时间为 1us;</p> <p>64us 到 126us, 若步长时间为 2us;</p> <p>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。</p>	0x0	R/W

### 13.5-19TM1X\*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	16
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]						
--		R/W				--			R/W						

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]	<p>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:</p> <p>00000: 1次传输  00001: 2次传输  .....  00010: 3次传输  .....  10001: 18次传输</p> <p>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。</p>	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]	<p>DBA[4:0]: DMA基地址 (DMA base address)</p> <p>这些位定义了DMA在连续模式下的地址(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:</p> <p>00000: TIM1x_CR1, 00001: TIM1x_CR2, 00010: TIM1x_SMCR, .....</p>	0x0	R/W

### 13.5-20TM1X\*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	16
DMAB[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	DMAB[31:0]	<p>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:</p> <p>TIM1X*_CR1 Address + (DBA + DMA index)×4, 其中:</p> <p>“TIM1X*_CR1地址”是控制寄存器1的地址;</p> <p>“DBA”是TIM1X*_DCR寄存器中定义的地址;</p> <p>“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。</p>	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

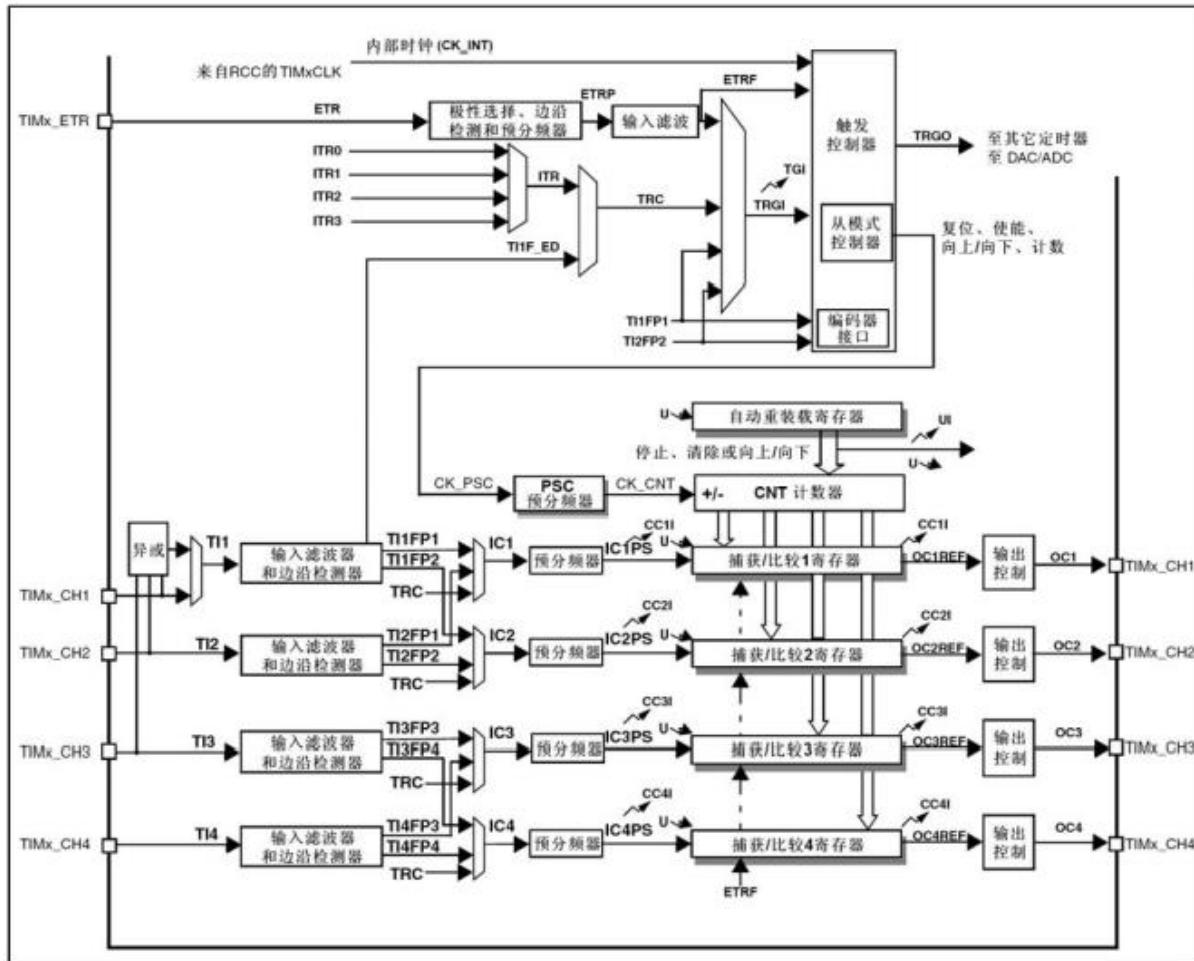


Figure 14- 1 通用定时器框图

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。

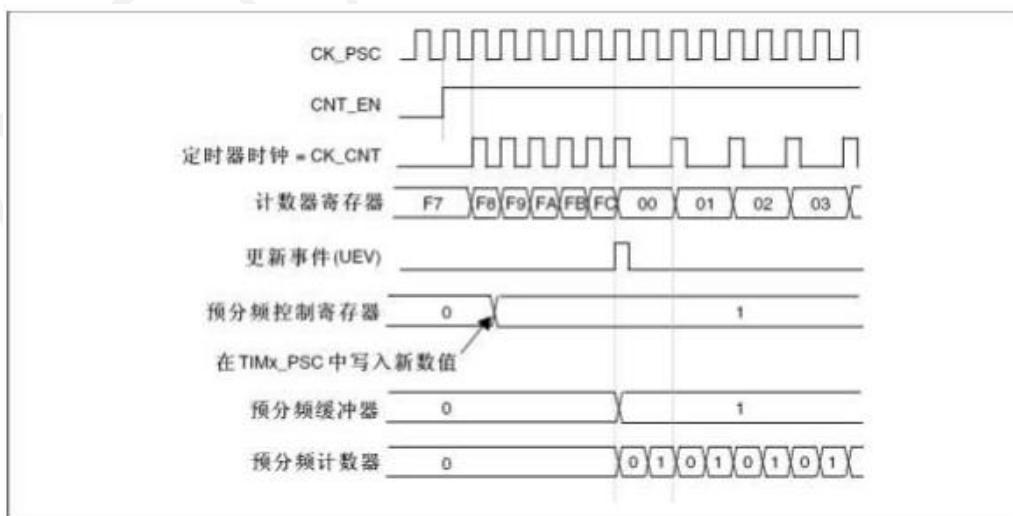


Figure 14- 2 当预分频器的参数从 1 变到 2 时，计数器的时序图

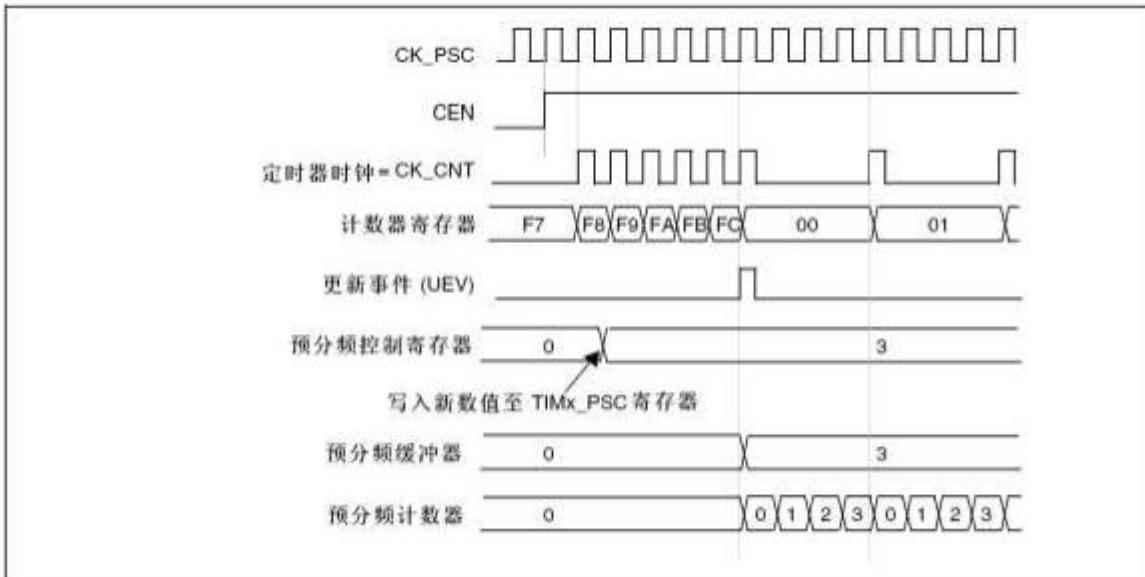


Figure 14- 3 当预分频器的参数从 1 变到 4 时，计数器的时序图

### 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(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 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，但硬件不设置 UIF 标志(即不产生中断或者 DMA 请求)；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时(依据 URS 位)设置更新标志位(TIM2X\*\_SR 寄存器中的 UIF 位)。

- 预分频器的缓冲区被置入预装载寄存器的值(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(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。

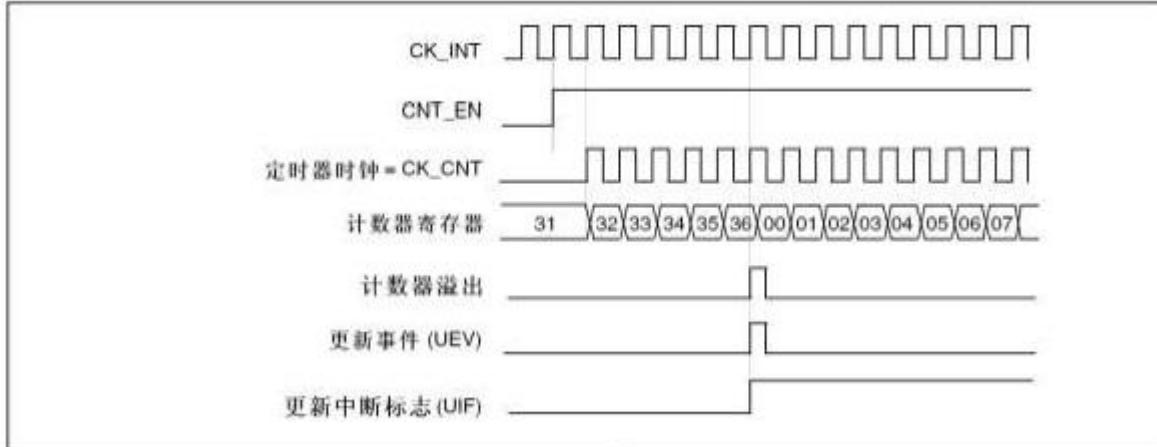


Figure 14- 4 计数器时序图：内部时钟分频因子为 1

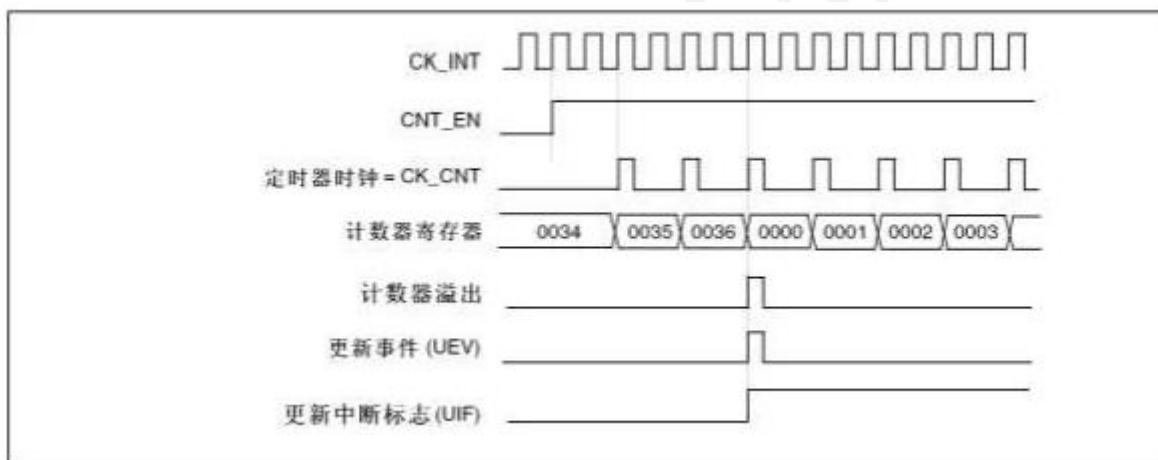


Figure 14- 5 计数器时序图：内部时钟分频因子为 2

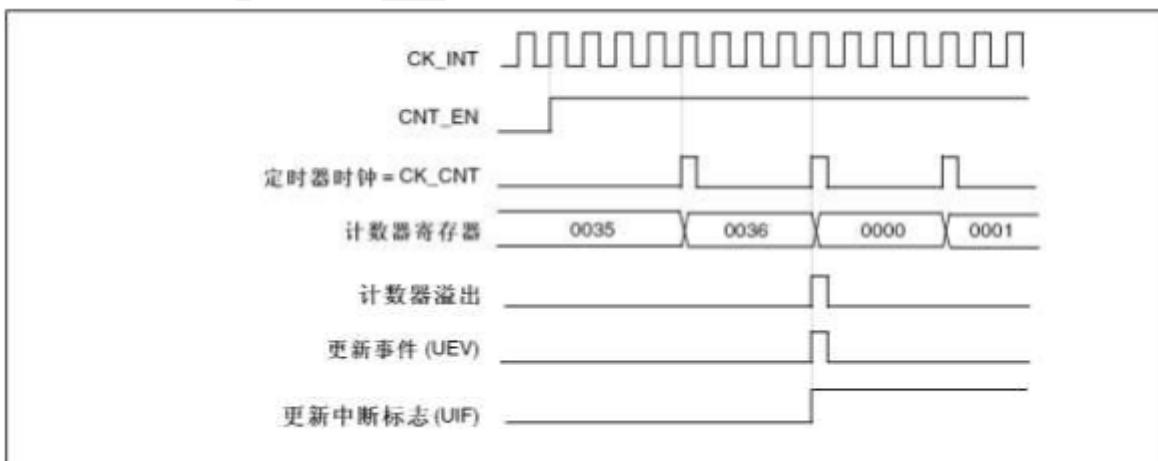
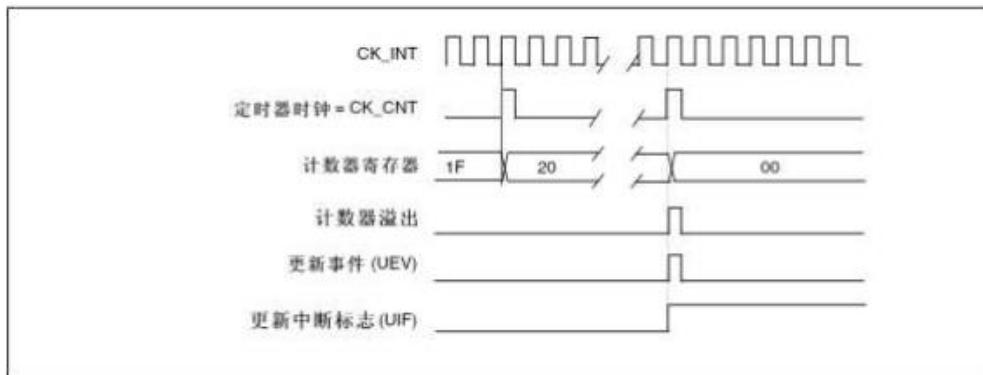
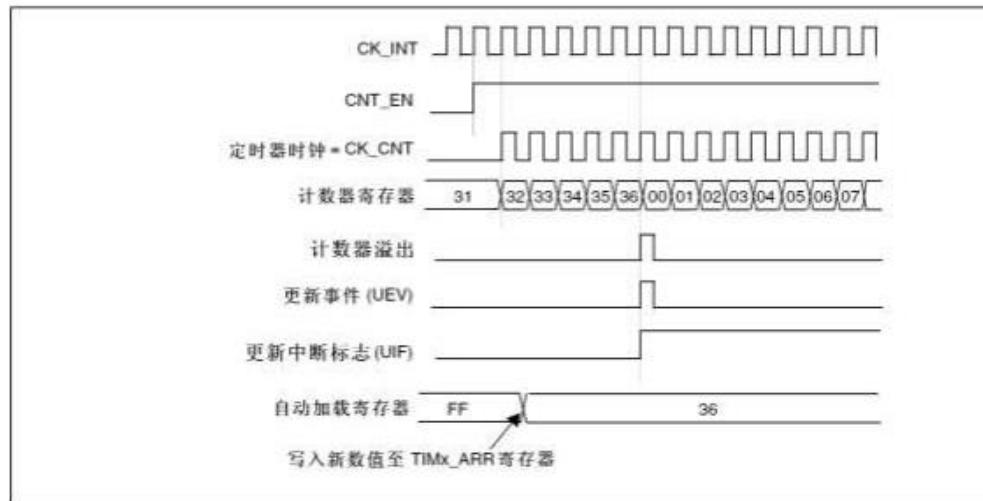


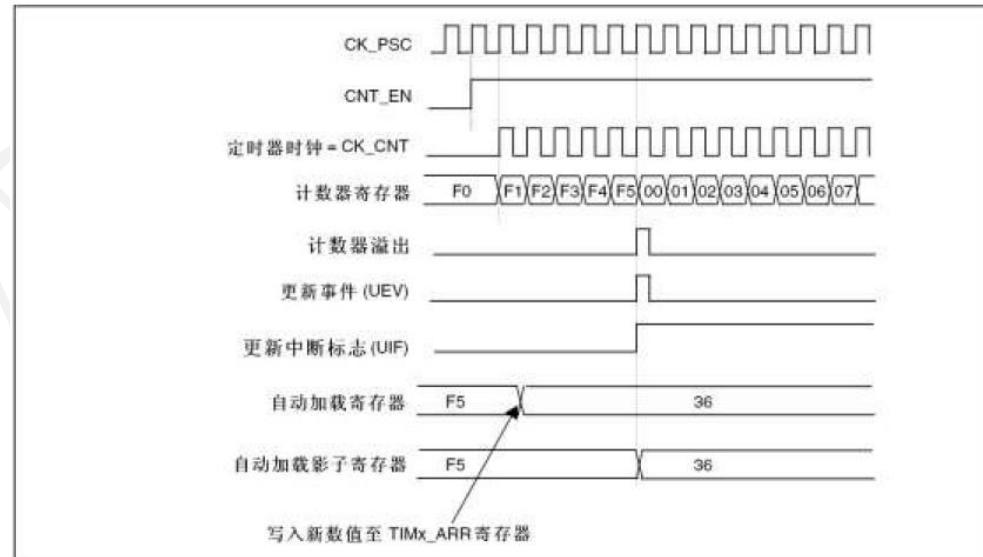
Figure 14- 6 计数器时序图：内部时钟分频因子为 4



**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。

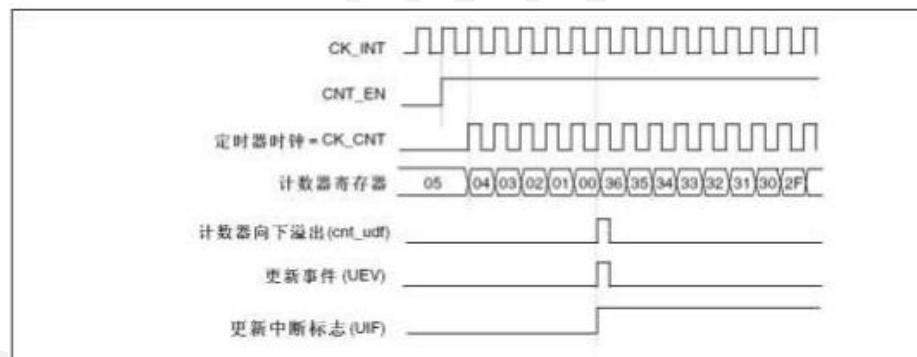


Figure 14- 10 计数器时序图：内部时钟分频因子为 1

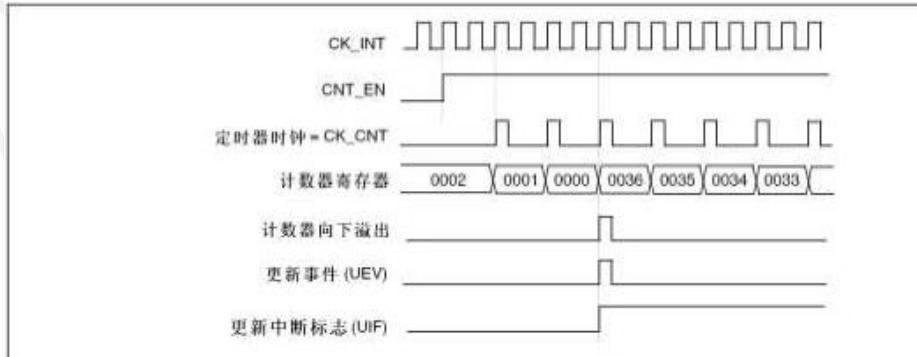


Figure 14- 11 计数器时序图：内部时钟分频因子为 2

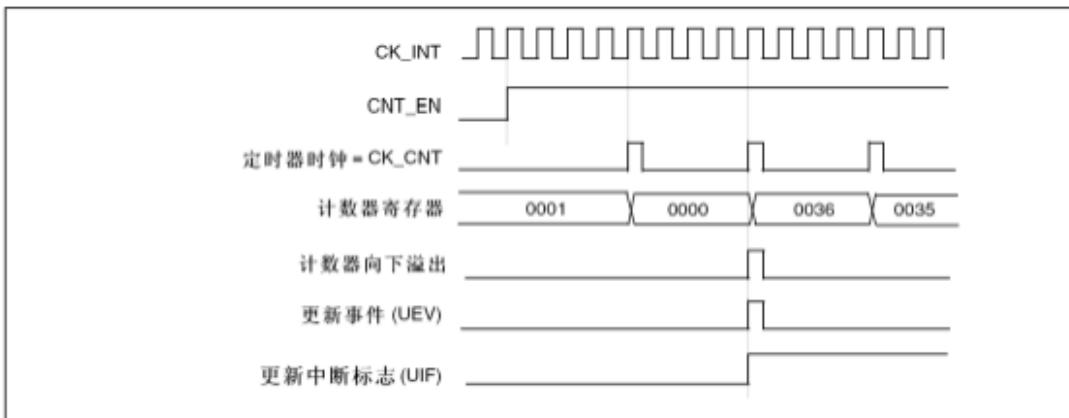


Figure 14- 12 计数器时序图：内部时钟分频因子为 4

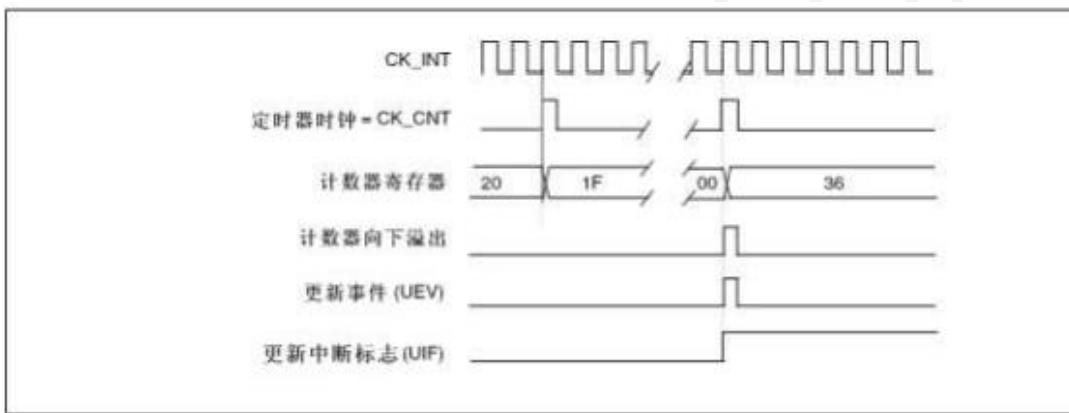


Figure 14- 13 计数器时序图：内部时钟分频因子为 N

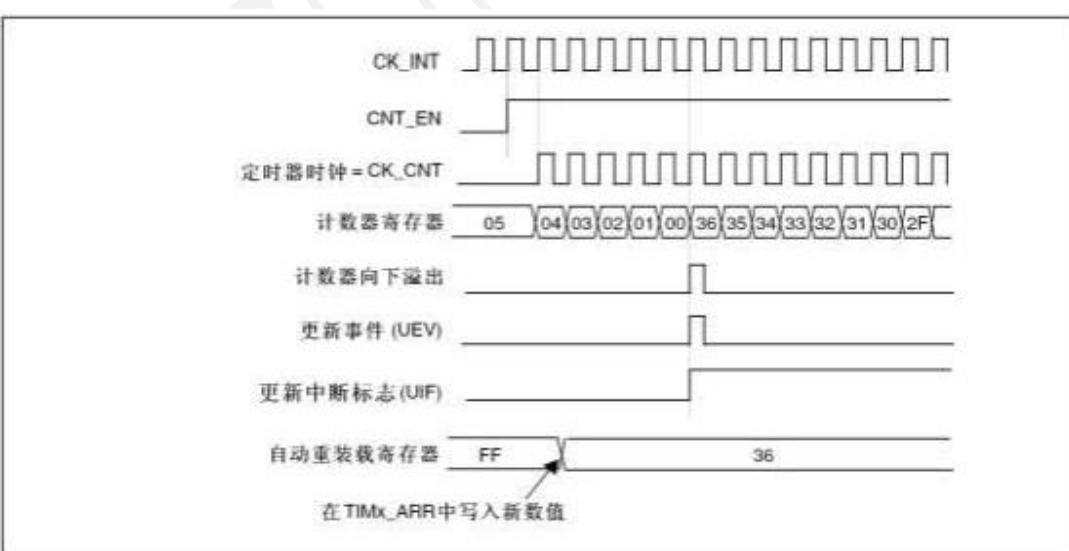


Figure 14- 14 计数器时序图：当没有使用重复计数器时的更新事件

### 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：

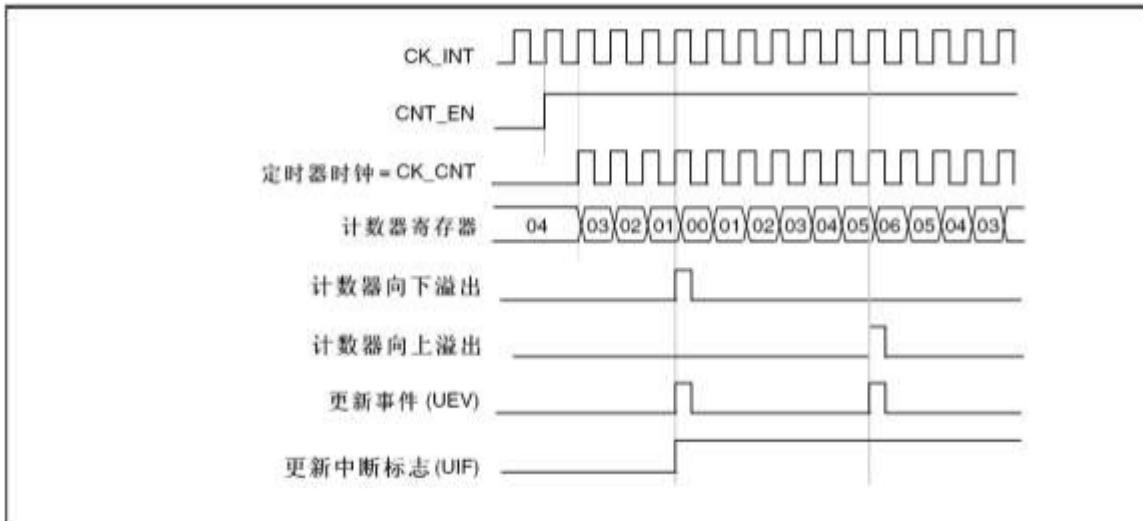


Figure 14- 15 计数器时序图：内部时钟分频因子为 1，**TIM2X\*\_ARR=0x6**

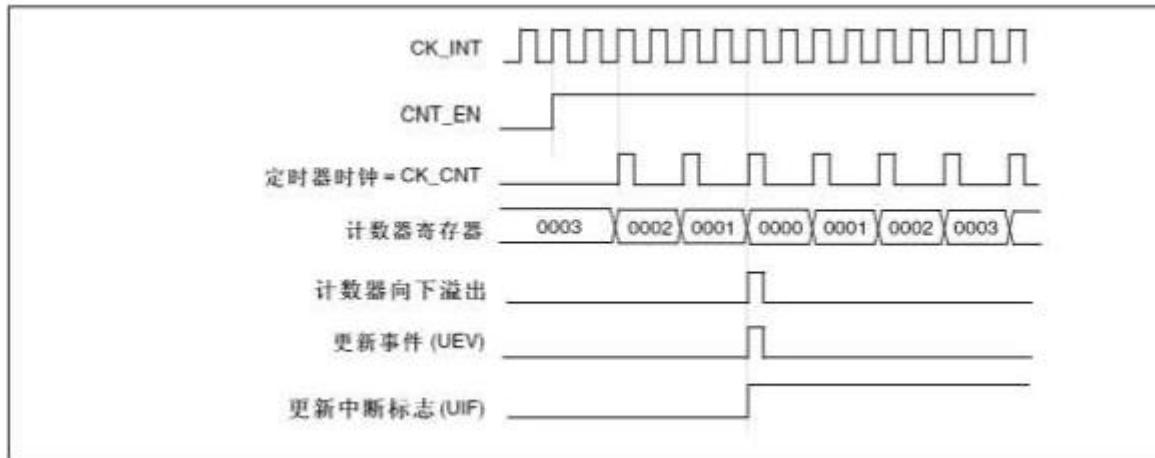


Figure 14- 16 计数器时序图：内部时钟分频因子为 2

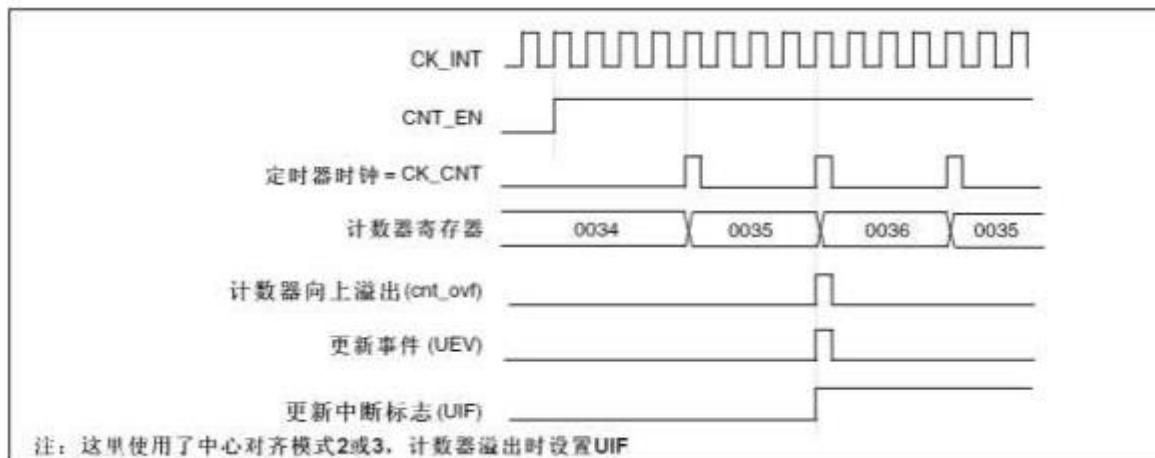


Figure 14- 17 计数器时序图：内部时钟分频因子为 4, TIM2X\*\_ARR=0x36

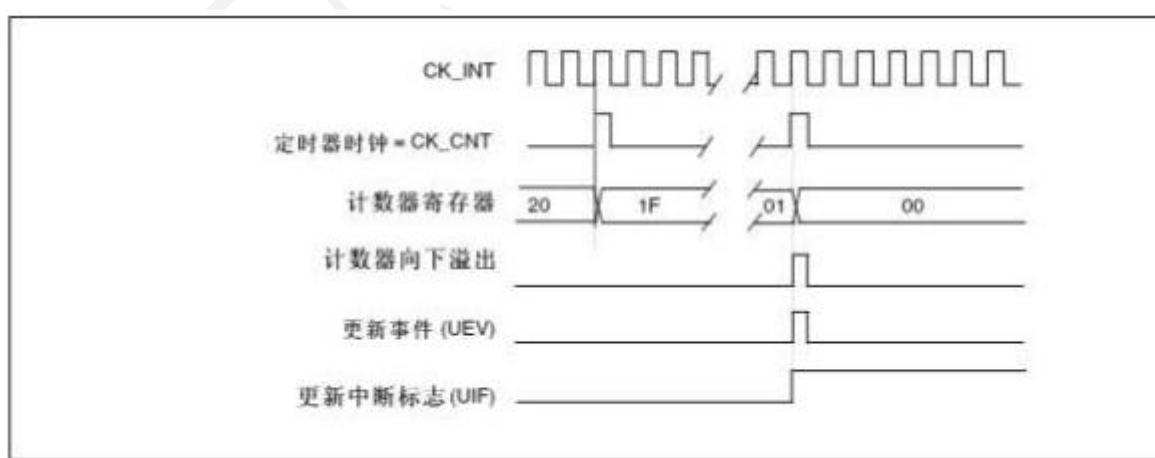


Figure 14- 18 计数器时序图：内部时钟分频因子为 N

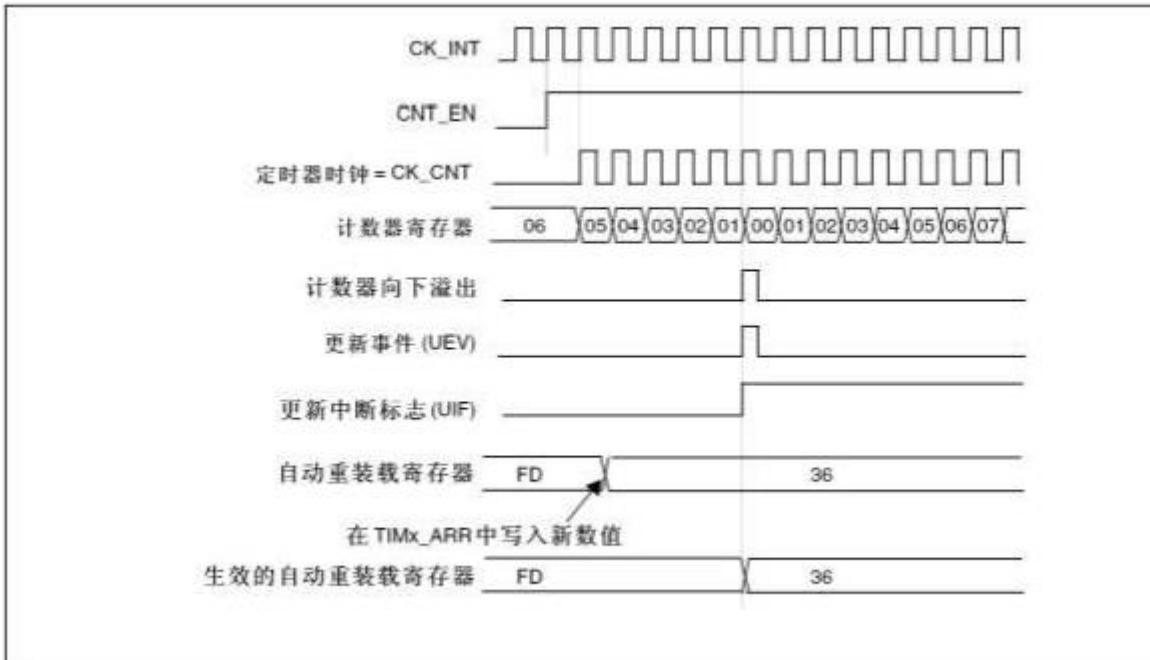


Figure 14- 19 计数器时序图：ARPE=1 时的更新事件(计数器下溢)

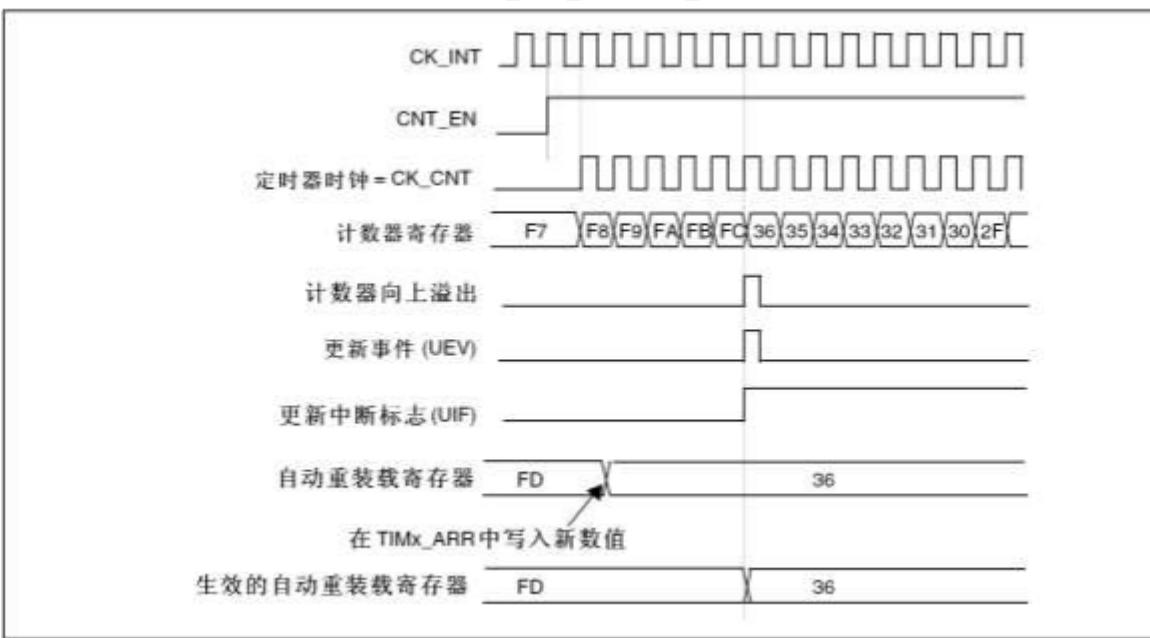


Figure 14- 20 计数器时序图：ARPE=1 时的更新事件(计数器溢出)

### 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。

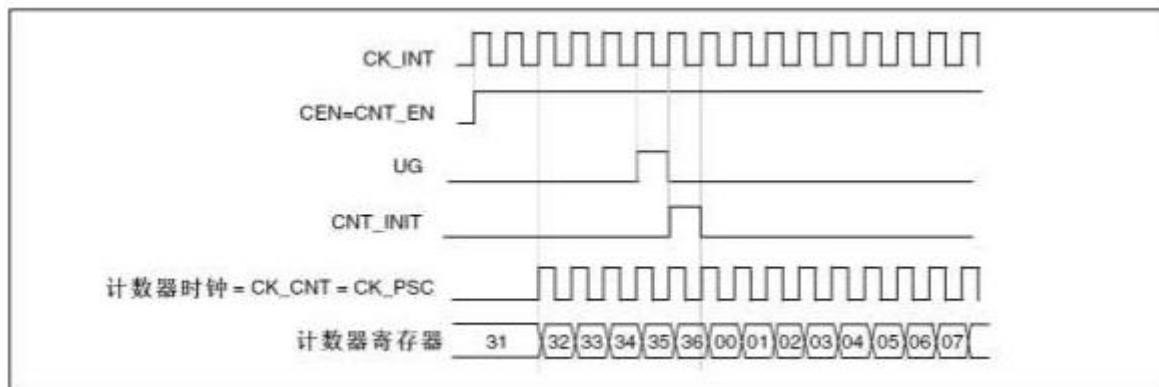


Figure 14- 21 一般模式下的控制电路，内部时钟分频因子为 1

### 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降沿计数。

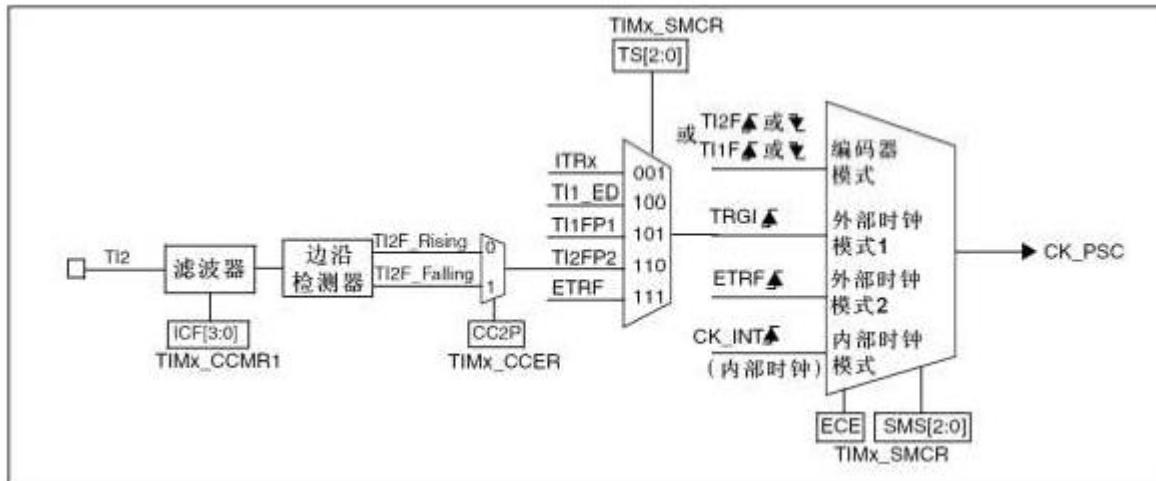


Figure 14- 22 TI2 外部时钟连接例子

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(如果不需要滤波器，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。

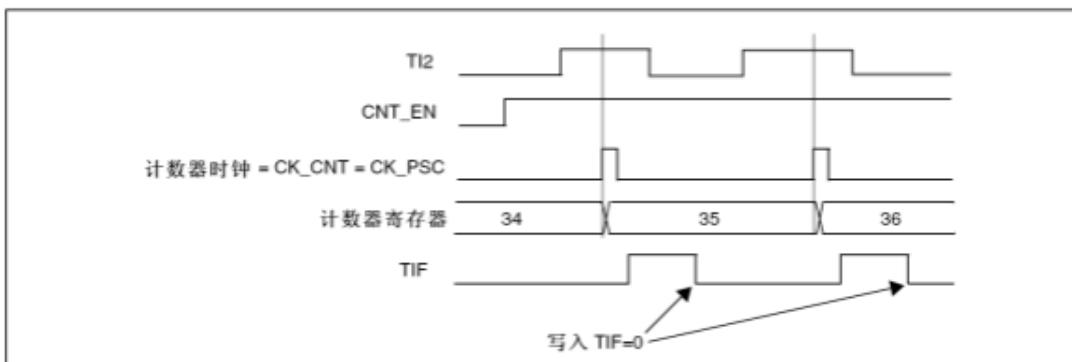


Figure 14- 23 外部时钟模式 1 下的控制电路

### 14.3-3.3 外部时钟源模式 2

选定此模式的方法为：令 `TIM2X*_SMCR` 寄存器中的 `ECE=1`

计数器能够在外部触发 `ETR` 的每一个上升沿或下降沿计数。

下图是外部触发输入的框图：

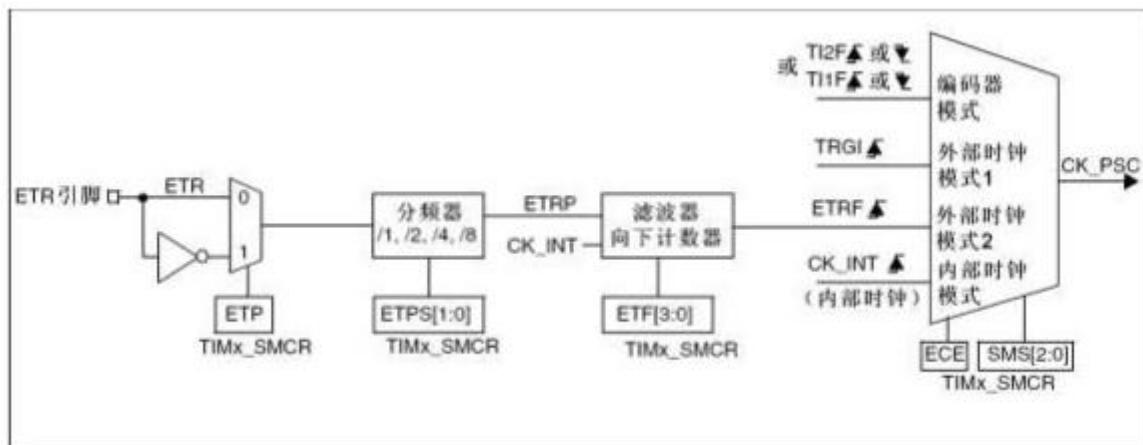


Figure 14- 24 外部触发输入框图

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：

1. 本例中不需要滤波器，置 `TIM2X*_SMCR` 寄存器中的 `ETF[3:0]=0000`
2. 设置预分频器，置 `TIM2X*_SMCR` 寄存器中的 `ETPS[1:0]=01`
3. 设置在 `ETR` 的上升沿检测，置 `TIM2X*_SMCR` 寄存器中的 `ETP=0`
4. 开启外部时钟模式 2，置 `TIM2X*_SMCR` 寄存器中的 `ECE=1`
5. 启动计数器，置 `TIM2X*_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。

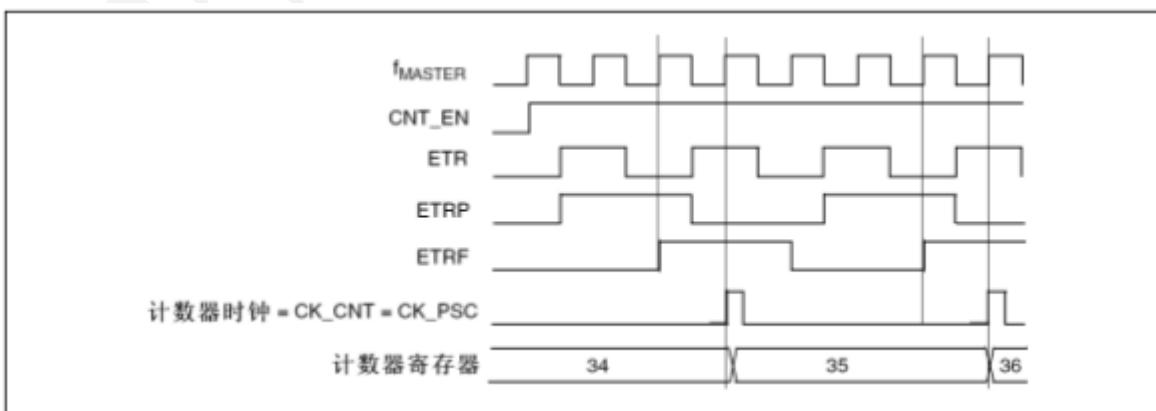


Figure 14- 25 外部时钟模式 2 下的控制时序

### 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  $TIx_F$ 。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( $ICxPS$ )。

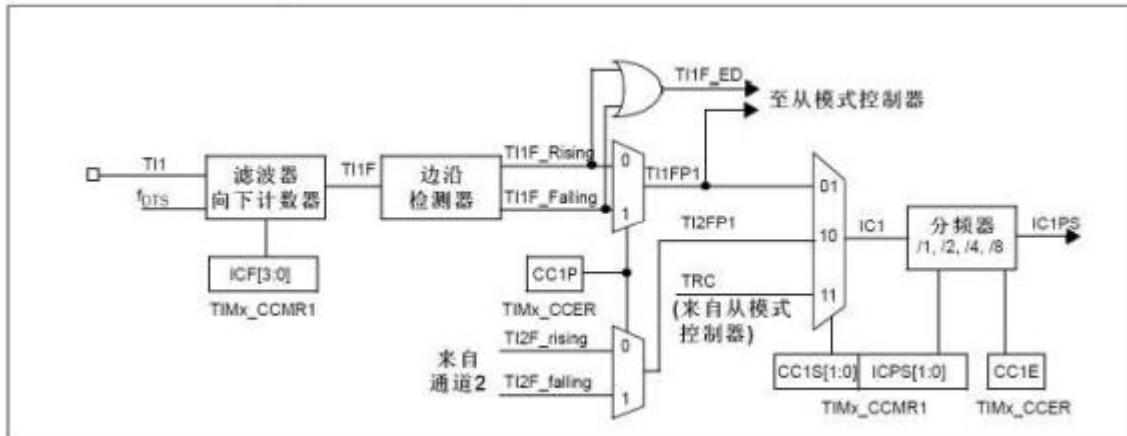


Figure 14- 26 捕获/比较通道(如：通道 1 输入部分)

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。

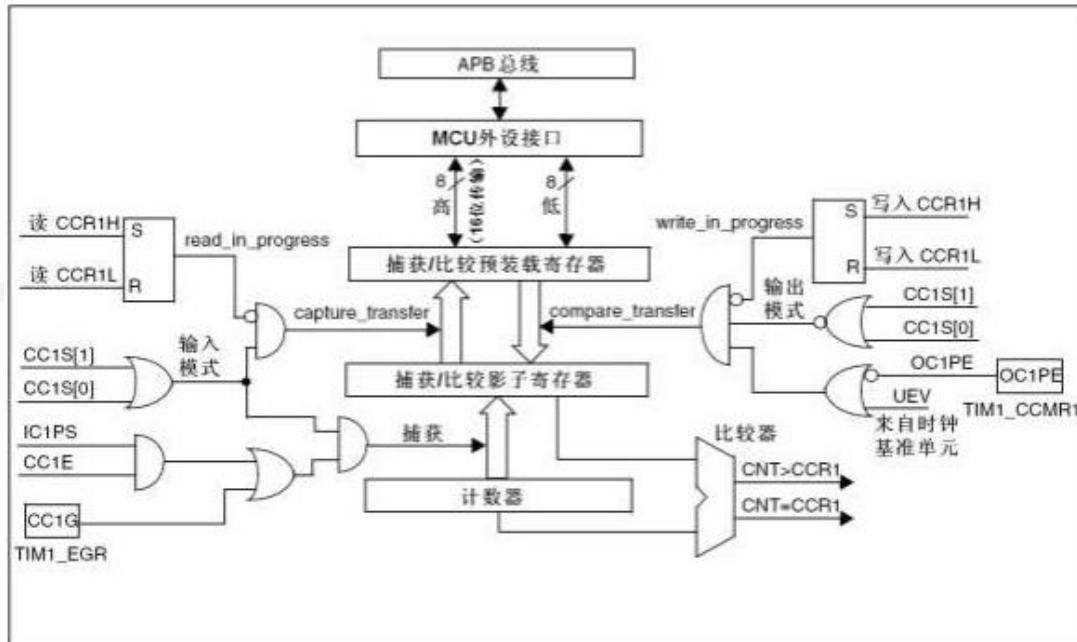


Figure 14- 27 捕获/比较通道 1 的主电路

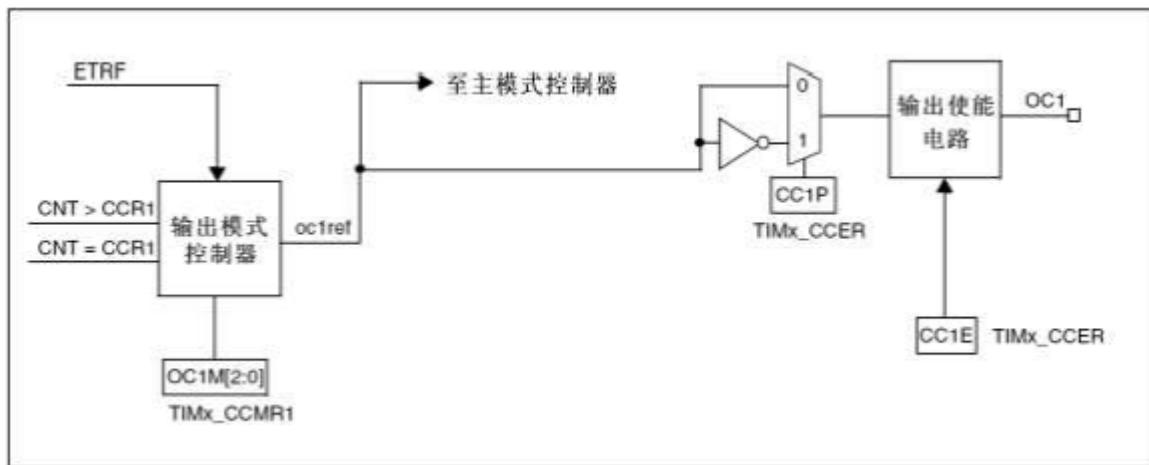


Figure 14- 28 捕获/比较通道的输出部分(通道 1)

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 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以(以 f<sub>DTS</sub> 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 TIMx\_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。

### 14.3-6PWM 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。

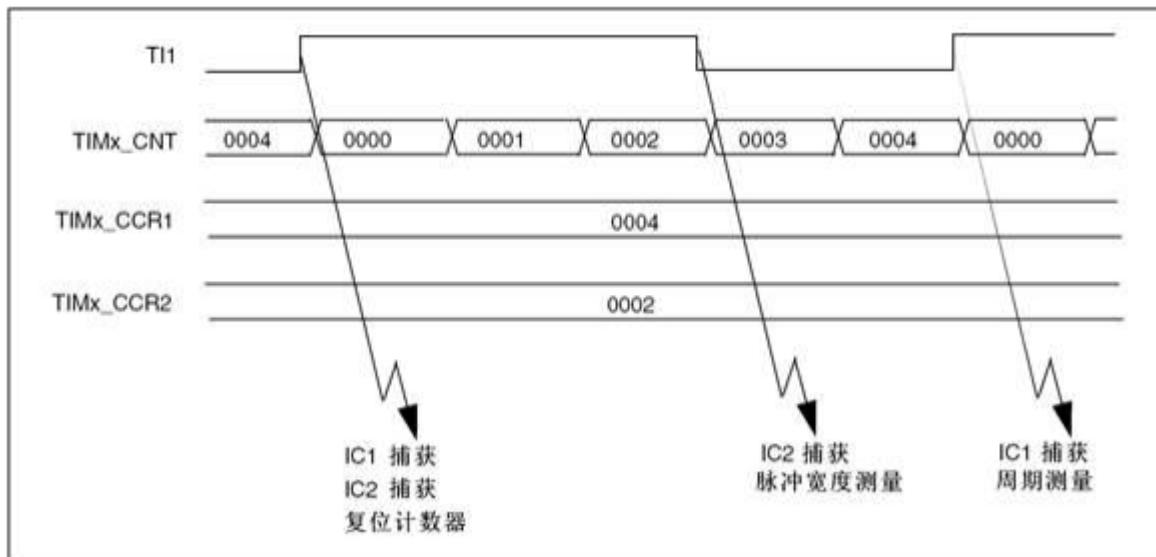


Figure 14- 29 PWM 输入模式时序

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。

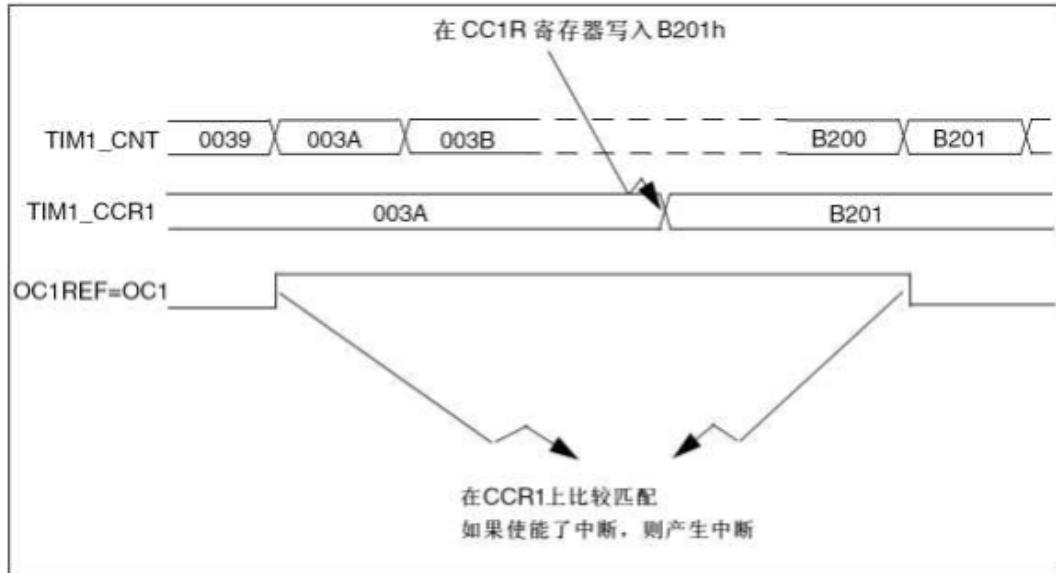


Figure 14- 30 输出比较模式，翻转 OC1

### 14.3-9PWM 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  $\text{TIM2X}^*\text{_CR1}$  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  $\text{TIM2X}^*\text{_CNT} < \text{TIM2X}^*\text{_CCR}_x$  时 PWM 信号参考 OCxREF 为高，否则为低。如果  $\text{TIM2X}^*\text{_CCR}_x$  中的比较值大于自动重装载值( $\text{TIM2X}^*\text{_ARR}$ )，则 OCxREF 保持为' 1'。如果比较值为 0，则 OCxREF 保持为' 0'。下图为  $\text{TIM2X}^*\text{_ARR}=8$  时边沿对齐的 PWM 波形实例。

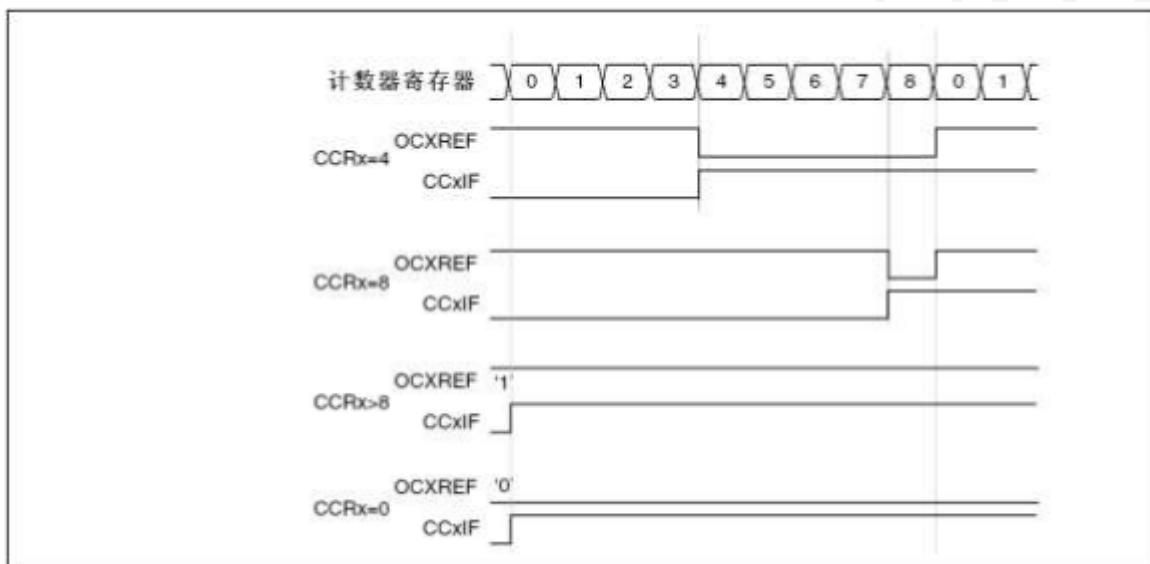


Figure 14- 31 边沿对齐的 PWM 波形(ARR=8)

#### 14.3-9.1.2 向下计数的配置

当  $\text{TIM2X}^*\text{_CR1}$  寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当  $\text{TIM2X}^*\text{_CNT} > \text{TIM2X}^*\text{_CCR}_x$  时参考信号 OCxREF 为低，否则为高。如果  $\text{TIM2X}^*\text{_CCR}_x$  中的比较值大于  $\text{TIM2X}^*\text{_ARR}$  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  $\text{TIM2X}^*\text{_CR1}$  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'。 $\text{TIM2X}^*\text{_CR1}$  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。

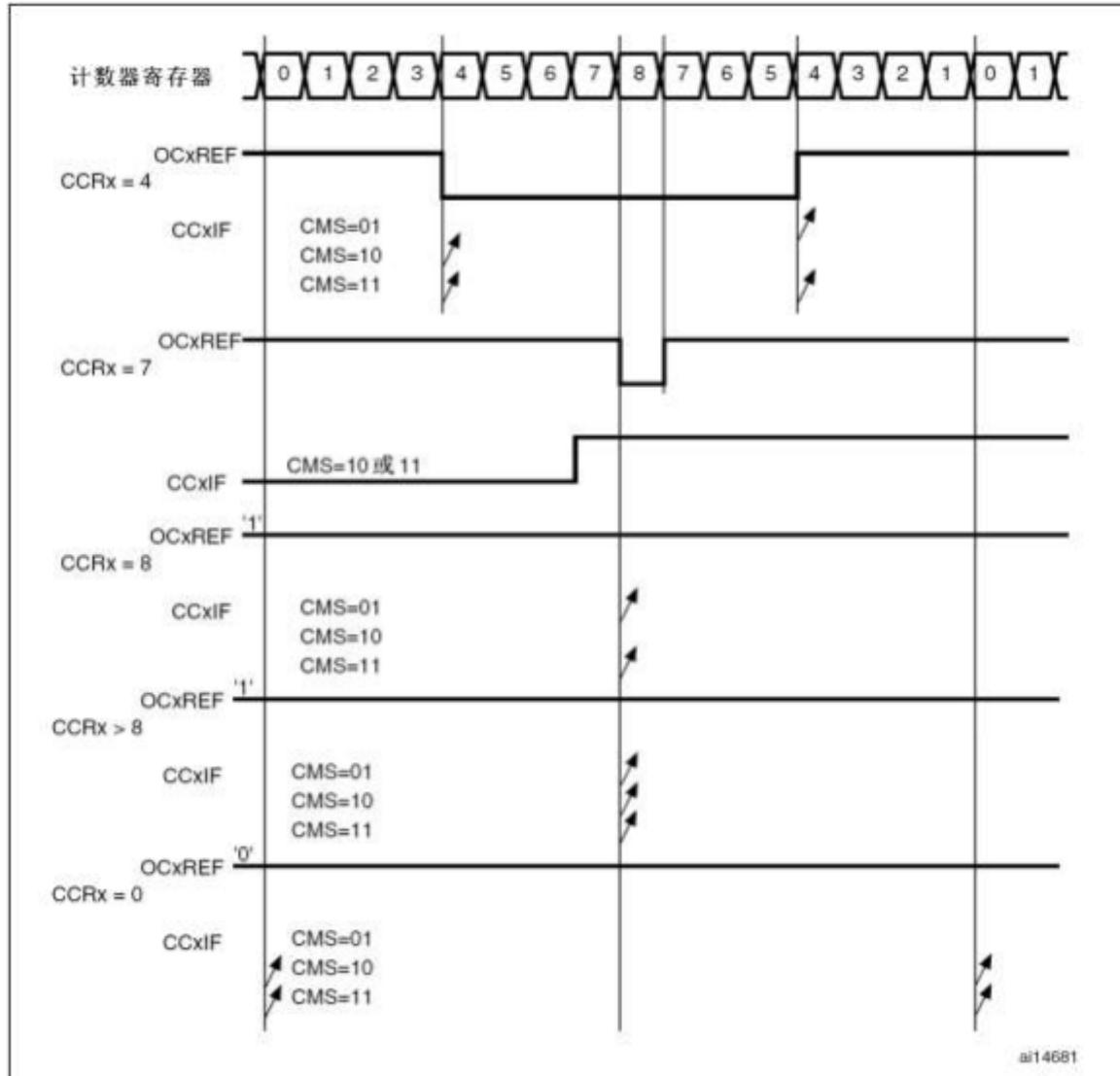


Figure 14- 32 中央对齐的 PWM 波形(APR=8)

#### 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  $\text{TIM2X*}_\text{ARR}$  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  $\text{TIM2X*}_\text{EGR}$  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  $\text{TIM2X*}_\text{CR1}$  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： $\text{CNT} < \text{CCR}_x \leq \text{ARR}$ (特别地， $0 < \text{CCR}_x$ )，

向下计数方式： $\text{CNT} > \text{CCR}_x$ 。

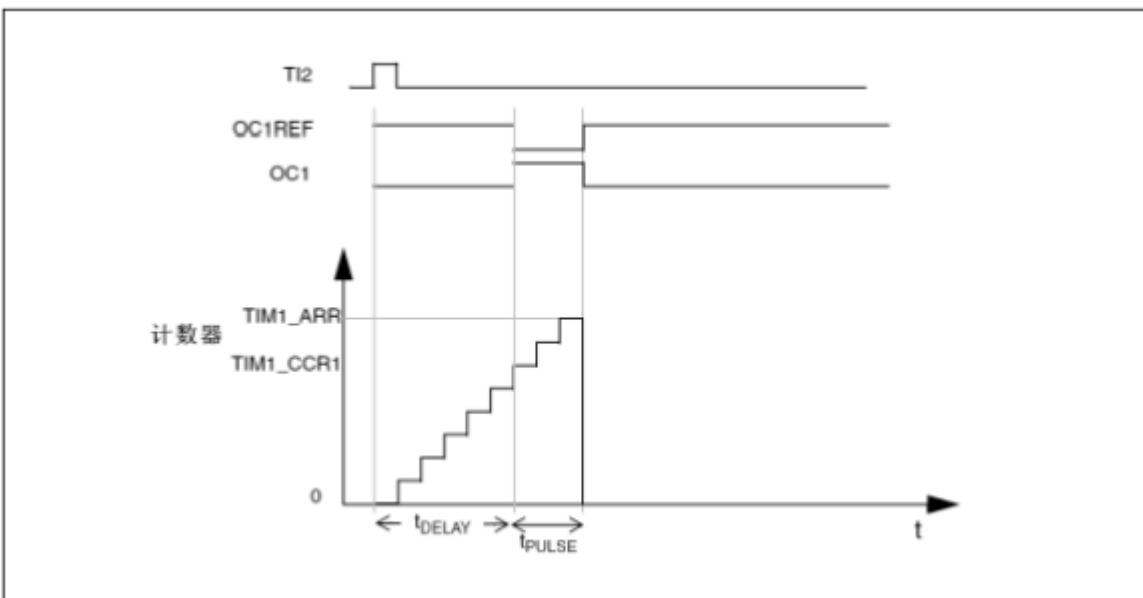


Figure 14- 33 单脉冲模式的例子

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

假定 TI2FP2 作为触发 1：

- 置  $\text{TIM2X*}_\text{CCMR1}$  寄存器中的  $\text{CC2S} = '01'$ ，把 TI2FP2 映像到 TI2。
- 置  $\text{TIM2X*}_\text{CCER}$  寄存器中的  $\text{CC2P} = '0'$ ，使 TI2FP2 能够检测上升沿。
- 置  $\text{TIM2X*}_\text{SMCR}$  寄存器中的  $\text{TS} = '110'$ ，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)
  - `tDELAY` 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(`TIM2X*_ARR - TIM2X*_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 `TIM2X*_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 `TIM2X*_CCMR1` 中的 `OC1PE= '1'` 和 `TIM2X*_CR1` 寄存器中的 `ARPE`;
- 然后在 `TIM2X*_CCR1` 寄存器中填写比较值, 在 `TIM2X*_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 `TI2` 上的一个外部触发事件。本例中, `CC1P= '0'`。  
在这个例子中, `TIM2X*_CR1` 寄存器中的 `DIR` 和 `CMS` 位应该置低。  
因为只需一个脉冲, 所以必须设置 `TIM2X*_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 `TIM2X*_CCMRx` 寄存器中的 `OCxFE` 位; 此时 `OCxREF`(和 `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。`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。

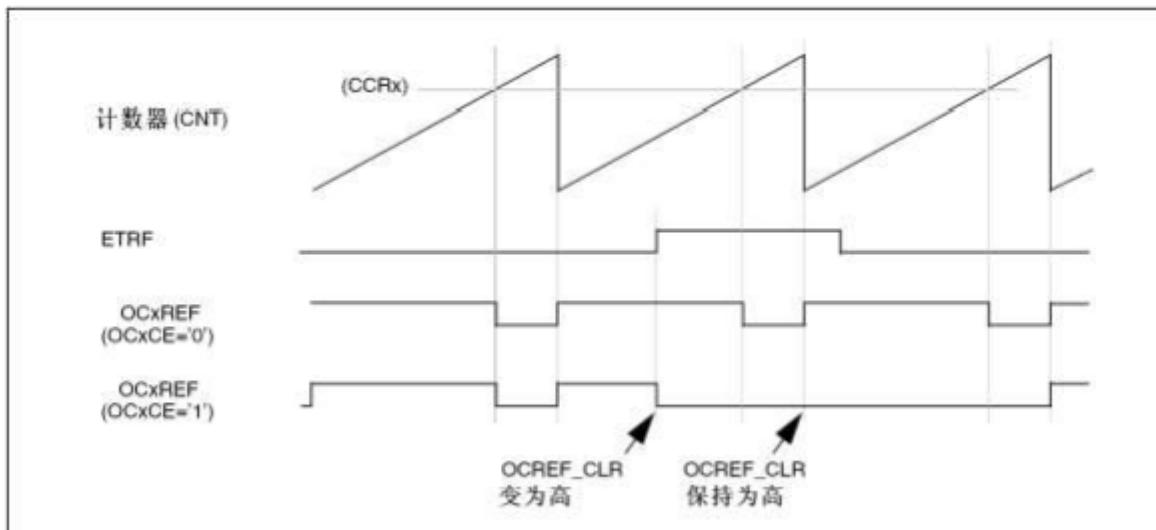


Figure 14- 34 清除 TIM2X\* 的 OCxREF

### 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。

Table 14- 1 计数方向与编码器信号的关系

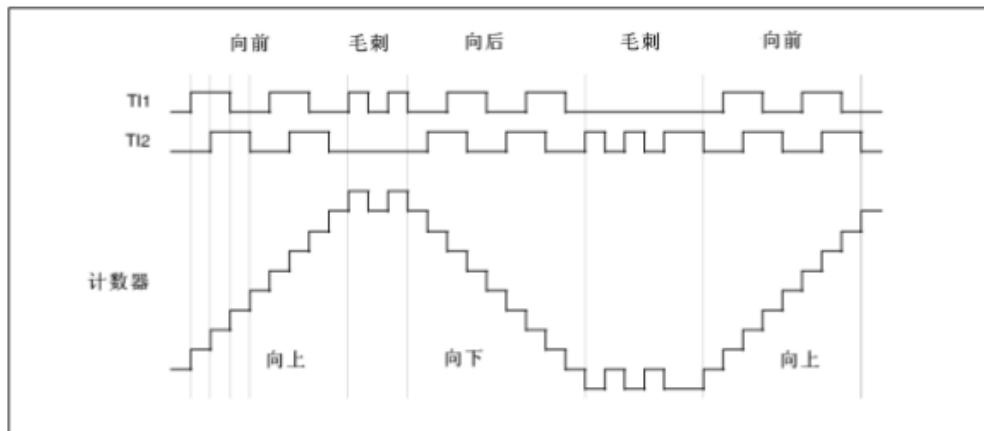
有效边沿	相对信号的电平 (TI1FP1 对应 TI2, TI2FP2 对应 TI1)	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 TI2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 TI1 和 TI2 上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

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

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

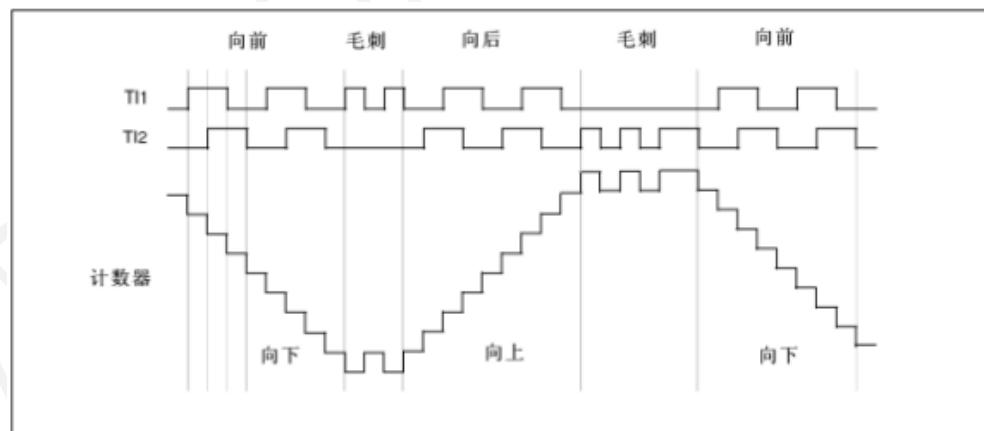
在这个例子中，我们假定配置如下：

- CC1S= '01' (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。

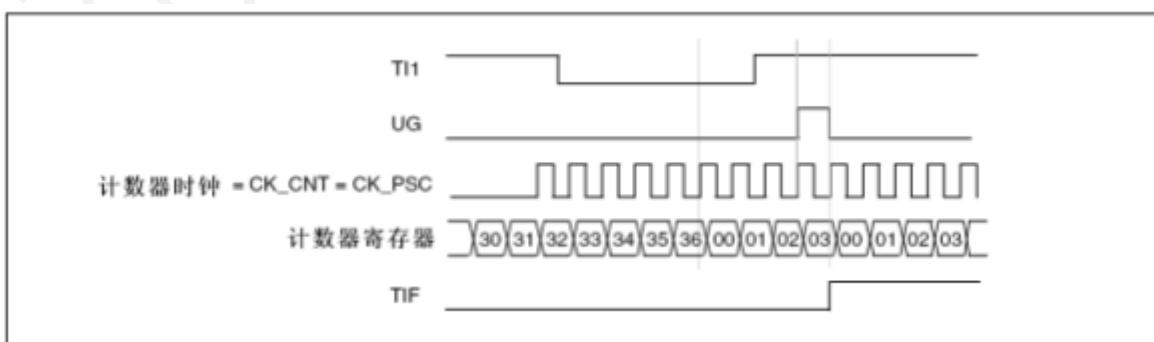


Figure 14- 37 复位模式下的控制电路

### 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。

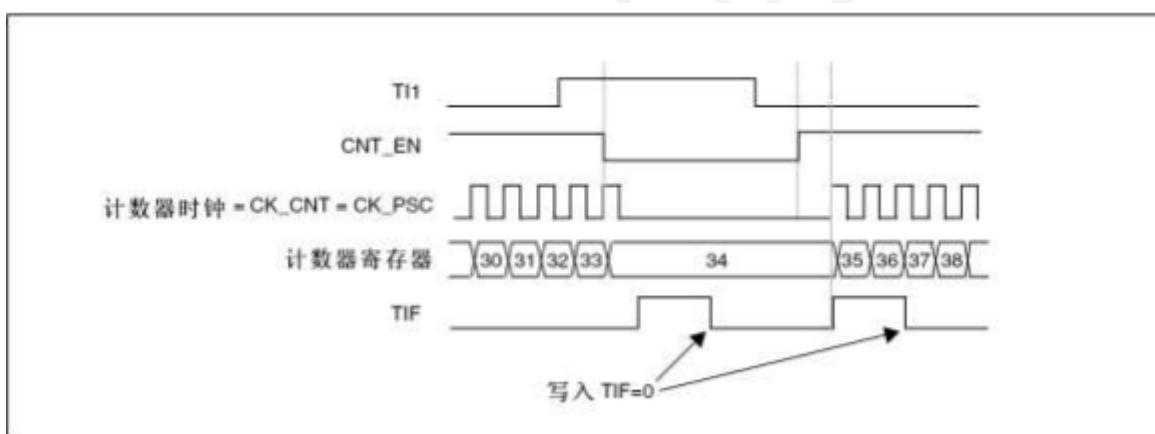


Figure 14- 38 门控模式下的控制电路

### 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。

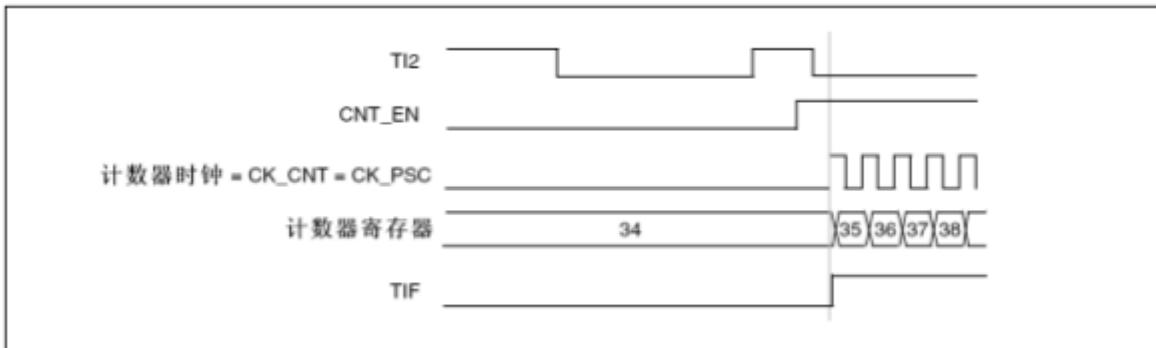


Figure 14- 39 触发器模式下的控制电路

### 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：

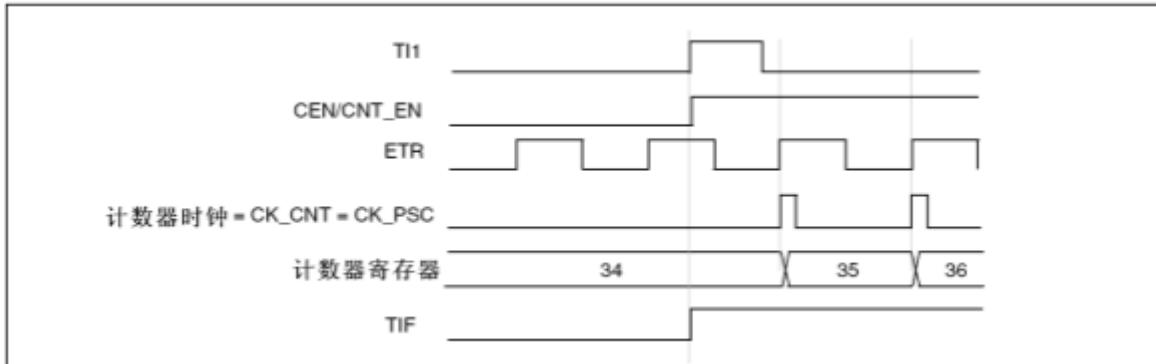
通过 TIM2X\*\_SMCR 寄存器配置外部触发输入电路：

- 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。



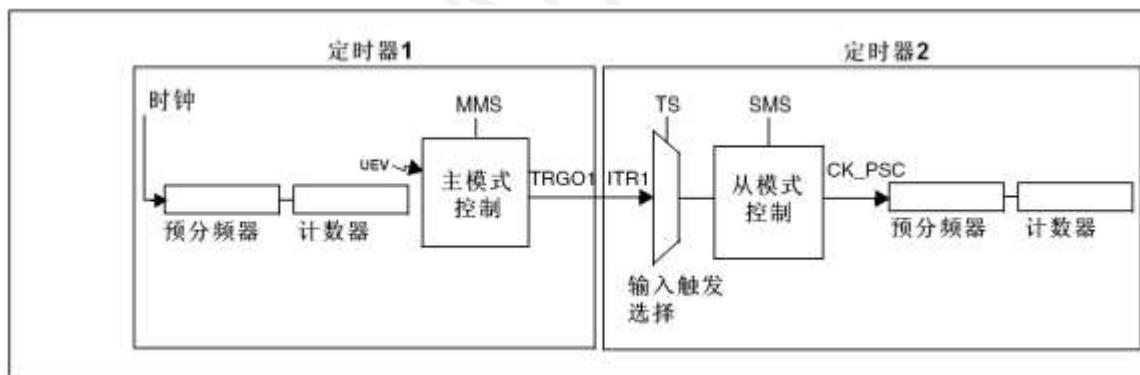
**Figure 14- 40 外部时钟模式 2 + 触发模式下的控制电路**

### 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 ,1A,1B,2C
- 置 TIM1X\*\_CR1 寄存器的 CEN=1 以启动定时器 1 ,1A,1B

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。

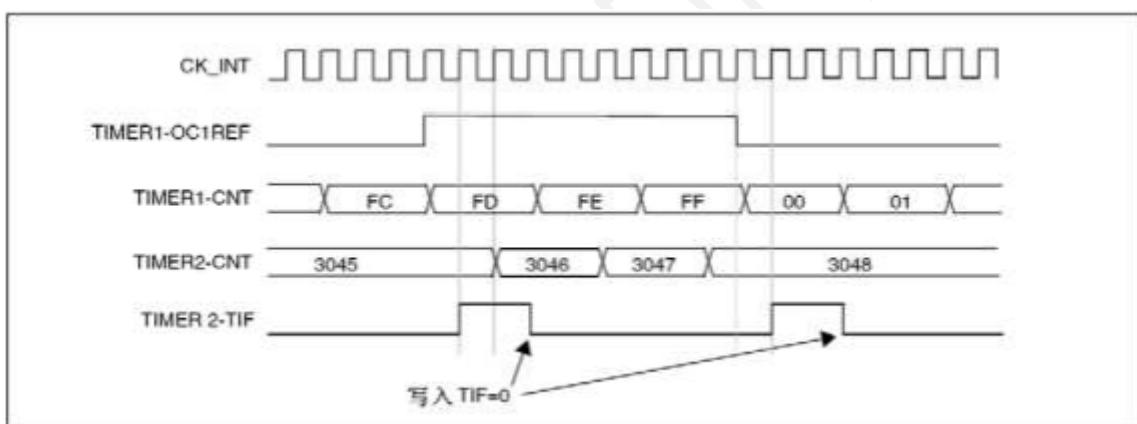


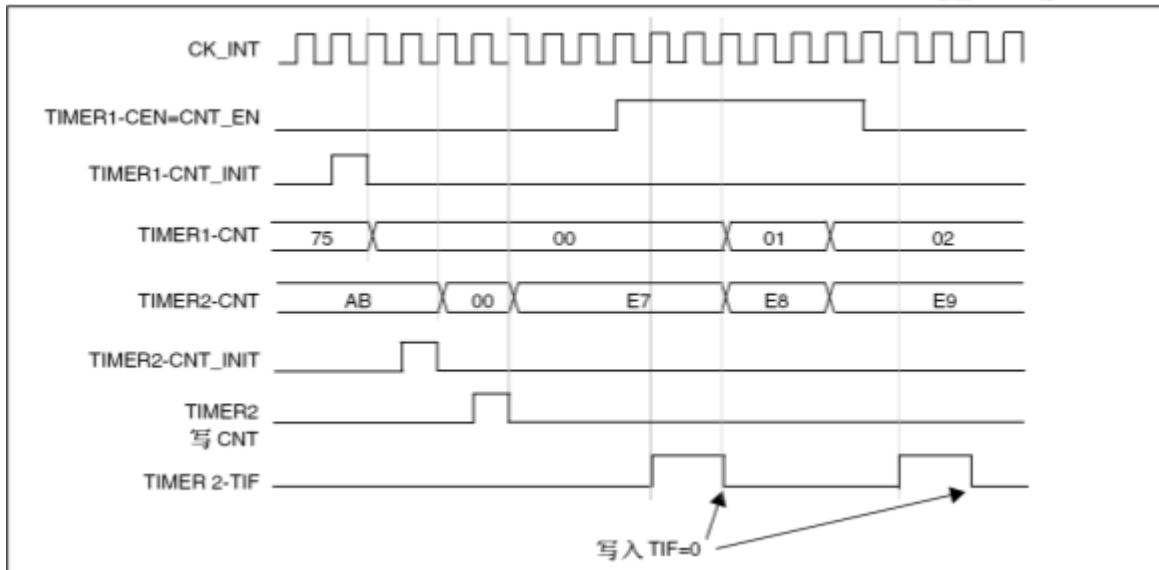
Figure 14- 42 的 OC1REF 控制定时器 2,2A,2B,2C

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。

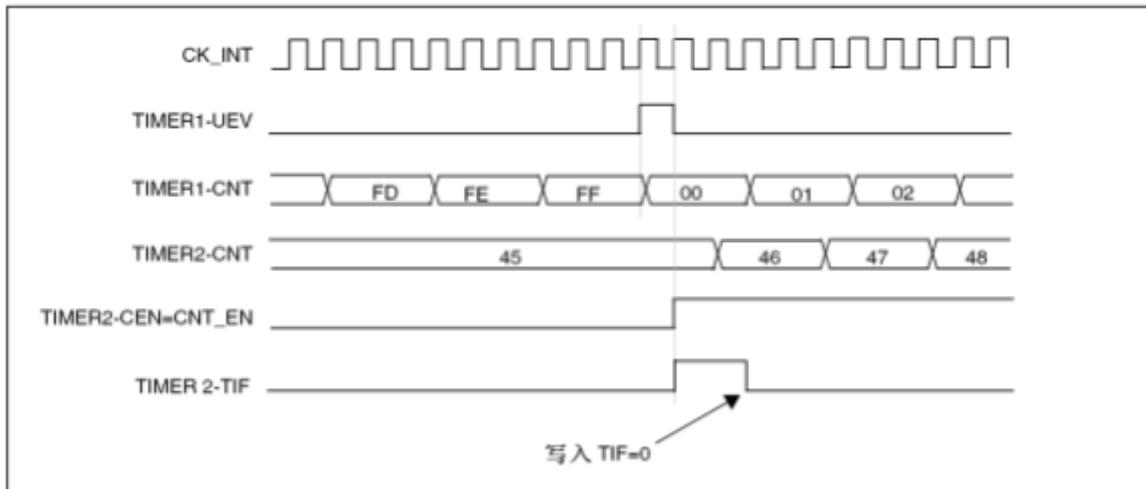


**Figure 14- 43 通过使能定时器 1 可以控制定时器 2**

#### 14.3-15.3 使用一个定时器去启动另一个定时器

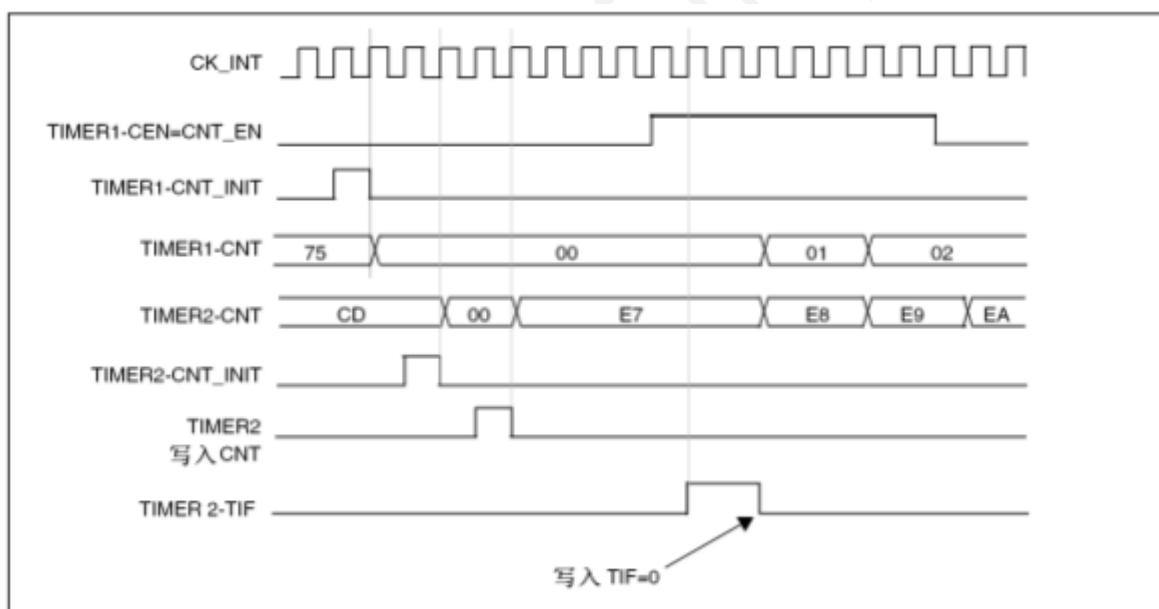
在这个例子中，使用定时器 1 的更新事件使能定时器 2。参考图 13-41 的连接。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CEN 位被自动地置' 1'，同时计数器开始计数直到写' 0' 到 TIM2\_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK\_INT 除以 3( $f_{CK\_CNT}=f_{CK\_INT}/3$ )。

- 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。

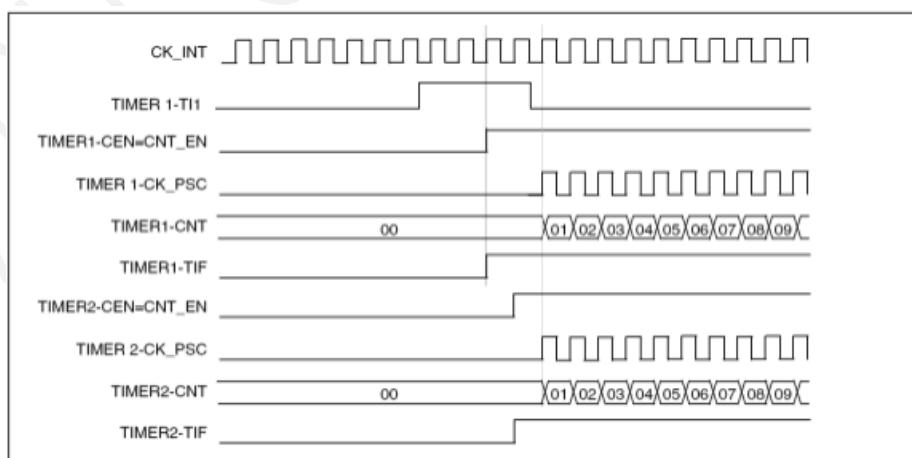


Figure 14- 46 利用定时器 1 的使能触发定时器 2

#### 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。

ZBIT CONFIDENTIAL

#### 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 3C00**

**TIM2A 基地址 : 0x4000 3D00**

**TIM2B 基地址 : 0x4000 3E00**

**TIM2C 基地址 : 0x4000 3F00**

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		
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		
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
保留				CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN		
--				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]	<p>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。</p> <p>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</p>	0	R/W
7	ARPE	<p>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。</p>	0	R/W
6:5	CMS[1:0]	<p>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。</p>	0	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

### 14.5-2TIM2X\* 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	OIS3N	OIS3	OIS2N	OIS2	OIS1	OIS1	TI1S	MMS[2:0]	保留					
--	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
7	TI1S	TI1 选择(TI1 selection) 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

位	标记	功能描述	复位值	读写
6:4	MMS[2:0]	<p>主模式选择(Master mode selection)</p> <p>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：</p> <ul style="list-style-type: none"> <li>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。</li> <li>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)。</li> <li>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。</li> <li>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)。</li> <li>100: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。</li> <li>101: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。</li> <li>110: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。</li> </ul>	0x0	R/W
3:0	保留	始终读为 0。	0	--

### 14.5-3TIM2X\* 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]		保留		SMS[2:0]			
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	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

位	标记	功能描述	复位值	读写
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</p> <p>0001: 采样频率fSAMPLING=fCK_INT, N=2</p> <p>0010: 采样频率fSAMPLING=fCK_INT, N=4</p> <p>0011: 采样频率fSAMPLING=fCK_INT, N=8</p> <p>0100: 采样频率fSAMPLING=fDTS/2, N=6</p> <p>0101: 采样频率fSAMPLING=fDTS/2, N=8</p> <p>0110: 采样频率fSAMPLING=fDTS/4, N=6</p> <p>0111: 采样频率fSAMPLING=fDTS/4, N=8</p> <p>1000: 采样频率fSAMPLING=fDTS/8, N=6</p> <p>1001: 采样频率fSAMPLING=fDTS/8, N=8</p> <p>1010: 采样频率fSAMPLING=fDTS/16, N=5</p> <p>1011: 采样频率fSAMPLING=fDTS/16, N=6</p> <p>1100: 采样频率fSAMPLING=fDTS/16, N=8</p> <p>1101: 采样频率fSAMPLING=fDTS/32, N=5</p> <p>1110: 采样频率fSAMPLING=fDTS/32, N=6</p> <p>1111: 采样频率fSAMPLING=fDTS/32, N=8</p>	0x0	R/W
7	MSM	<p>主/从模式(Master/slave mode)</p> <p>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。

### 14.5-4TIM2X\* 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
保留	TDE	保留	CC4DE	CC3DE	CC2DE	CC1DE	UDE	保留	TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE
--	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	保留	--	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
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	保留	--	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

位	标记	功能描述	复位值	读写
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

### 14.5-5TIM2X\* 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	CC1OF	CC1OF	保留	TIF	保留	CC4IF	CC3IF	CC2IF	CC1IF	UIF	--	--	--
--	RC W0	--	RCW0	--	RC W0	--	RCW0	--	RC W0	--	RCW0	--	RC W0	--	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	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TIM2X*_CCR1 寄存器时，CC1IF 的状态已经为' 1' 。	0	RC W0
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	RC W0
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 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

位	标记	功能描述	复位值	读写
2	CC2IF	捕获/比较 2 中断标记(Capture/Compare 2 interrupt flag) 参考CC1IF 描述。	0	RC WO
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 WO
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 WO

#### 14.5-6TIM2X\* 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															
--															

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

位	标记	功能描述	复位值	读写
0	UG	<p>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。</p>	0	WO

### 14.5-7TIM2X\* 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。

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		
OC2CE	OC2M[1:0]		OC2PE	OC2FE	CC2S[1:0]	OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]					
IC2F[3:0]		IC2PSC[1:0]				IC1F[3:0]			IC1PSC[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。	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]	<p>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;</p> <p>01: CC2 信道被配置为输入, IC2 映射在 TI2 上;</p> <p>10: CC2 信道被配置为输入, IC2 映射在 TI1 上;</p> <p>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)。</p> <p>注: CC2S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC2E=0)才是可写的。</p>	0	R/W
7	OC1CE	<p>输出比较 1 清' 0' 使能(Output Compare 1 clear enable)</p> <p>0: OC1REF 不受 ETRF 输入的影响;</p> <p>1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。</p>	0	R/W

位	标记	功能描述	复位值	读写
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 TIM2X*_CCR1 与计数器 TIM2X*_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 TIM2X*_CNT 的值与捕获/比较寄存器 1(TIM2X*_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 TIM2X*_CNT 的值与捕获/比较寄存器 1(TIM2X*_CCR1) 相同时，强制 OC1REF 为低。</p> <p>011: 翻转。当 TIM2X*_CCR1 = TIM2X*_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 TIM2X*_CNT &lt; 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 &gt; TIM2X*_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 TIM2X*_CNT &lt; 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 &gt; TIM2X*_CCR1 时通道 1 为有效电平，否则为无效电平。</p> <p>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。</p>	0	R/W
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

位	标记	功能描述	复位值	读写
2	OC1FE	<p>输出比较 1 快速使能(Output Compare 1 fast enable) 该位用于加快 CC 输出对触发输入事件的响应。</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) 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。	0	--
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]	<p>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:</p> <p>00: CC2 信道被配置为输出;</p> <p>01: CC2 信道被配置为输入, IC2 映射在 TI2 上;</p> <p>10: CC2 信道被配置为输入, IC2 映射在 TI1 上;</p> <p>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)。</p> <p>注: CC2S 仅在通道关闭时(TIM2X*_CCER 寄存器的 CC2E=0)才是可写的。</p>	0x0	R/W

位	标记	功能描述	复位值	读写
7:4	IC1F[3:0]	<p>输入捕获 1 滤波器(Input capture 1 filter)</p> <p>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：</p> <ul style="list-style-type: none"> <li>0000 : 无滤波器，以 fDTS 采样</li> <li>0001 : 采样频率fSAMPLING=fCK_INT, N=2</li> <li>0010 : 采样频率fSAMPLING=fCK_INT, N=4</li> <li>0011 : 采样频率fSAMPLING=fCK_INT, N=8</li> <li>0100 : 采样频率fSAMPLING=fDTS/2, N=6</li> <li>0101 : 采样频率fSAMPLING=fDTS/2, N=8</li> <li>0110 : 采样频率fSAMPLING=fDTS/4, N=6</li> <li>0111 : 采样频率fSAMPLING=fDTS/4, N=8</li> <li>1000 : 采样频率fSAMPLING=fDTS/8, N=6</li> <li>1001 : 采样频率fSAMPLING=fDTS/8, N=8</li> <li>1010 : 采样频率fSAMPLING=fDTS/16, N=5</li> <li>1011 : 采样频率fSAMPLING=fDTS/16, N=6</li> <li>1100 : 采样频率fSAMPLING=fDTS/16, N=8</li> <li>1101 : 采样频率fSAMPLING=fDTS/32, N=5</li> <li>1110 : 采样频率fSAMPLING=fDTS/32, N=6</li> <li>1111 : 采样频率fSAMPLING=fDTS/32, N=8</li> </ul>	0x0	R/W
3:2	IC1PSC[1:0]	<p>输入/捕获 1 预分频器(Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。</p> <p>一旦CC1E=0(TIM2X*_CCER 寄存器中)，则预分频器复位。</p> <ul style="list-style-type: none"> <li>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；</li> <li>01: 每 2 个事件触发一次捕获；</li> <li>10: 每 4 个事件触发一次捕获；</li> <li>11: 每 8 个事件触发一次捕获。</li> </ul>	0x0	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> <ul style="list-style-type: none"> <li>00: CC1 信道被配置为输出；</li> <li>01: CC1 信道被配置为输入， IC1 映射在TI1 上；</li> <li>10: CC1 信道被配置为输入， IC1 映射在TI2 上；</li> <li>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)。</li> </ul> <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

#### 14.5-8TIM2X\* 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[1:0]		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]	<p>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)。</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>	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)</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: CC3 信道被配置为输出;</p> <p>01: CC3 信道被配置为输入, IC3 映射在 TI3 上;</p> <p>10: CC3 信道被配置为输入, IC3 映射在 TI4 上;</p> <p>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)</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: CC4 信道被配置为输出;</p> <p>01: CC4 信道被配置为输入, IC4 映射在 TI4 上;</p> <p>10: CC4 信道被配置为输入, IC4 映射在 TI3 上;</p> <p>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)</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: CC3 信道被配置为输出;</p> <p>01: CC3 信道被配置为输入, IC3 映射在 TI3 上;</p> <p>10: CC3 信道被配置为输入, IC3 映射在 TI4 上;</p> <p>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>	0	R/W

#### 14.5-9TIM2X\* 捕获/比较使能寄存器(TIM2X\*\_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
保留		CC4P	CC4E	保留		CC3P	CC3E	保留		CC2P	CC2E	保留		CC1P	CC1E
--		R/W		--		R/W		--		R/		--		R/W	

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	<p>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</p> <p>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。</p>	0	R/W
0	CC1E	<p>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。</p> <p>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。</p>	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

### 14.5-10TIM2X\* 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

### 14.5-11TIM2X\* 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

### 14.5-12TIM2X\* 自动重装载寄存器(TIM2X\*\_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。	0x0	--
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。	0x0	R/W

#### 14.5-13TIM2X\* 捕获/比较寄存器 1(TIM2X\*\_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。	0x0	--
15:0	CCR1[15:0]	<p>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。 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。</p>	0x0	R/W

#### 14.5-14TIM2X\* 捕获/比较寄存器 2(TIM2X\*\_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。	0x0	--
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。	0x0	R/W

#### 14.5-15TIM2X\* 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。	0x0	--
15:0	CCR3[15:0]	<p>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。</p>	0x0	R/W

#### 14.5-16TIM2X\* 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。	0x0	--
15:0	CCR4[15:0]	<p>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。</p>	0x0	R/W

### 14.5-17TM2X\*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	16
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]						
--		R/W				--			R/W						

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:</p> <p>00000: 1次传输  00001: 2次传输  .....  00010: 3次传输  .....  10001: 18次传输</p> <p>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:  (TIM2x_CR1的地址) + DBA + (DMA索引), 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(TIM2x_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	--

位	标记	功能描述	复位值	读写
4:0	DBA[4:0]	<p>DBA[4:0]: DMA基地址 (DMA base address)</p> <p>这些位定义了DMA在连续模式下的地址(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:</p> <ul style="list-style-type: none"> <li>00000: TIM2x_CR1,</li> <li>00001: TIM2x_CR2,</li> <li>00010: TIM2x_SMCR,</li> <li>.....</li> </ul>	0x0	R/W

#### 14.5-18TM2X\*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	16
DMAB[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	DMAB[31:0]	<p>DMAB[31:0]: DMA连续传送寄存器(DMA register for burst accesses)</p> <p>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:</p> <p>TIM2X*_CR1 Address + (DBA + DMA index)x4, 其中:</p> <ul style="list-style-type: none"> <li>"TIM2X*_CR1地址" 是控制寄存器1的地址;</li> <li>"DBA" 是TIM2X*_DCR寄存器中定义的基地址;</li> <li>"DMA指针" 是由DMA自动控制的偏移量, 它取决于TIM2*_DCR寄存器中定义的DBL。</li> </ul>	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。

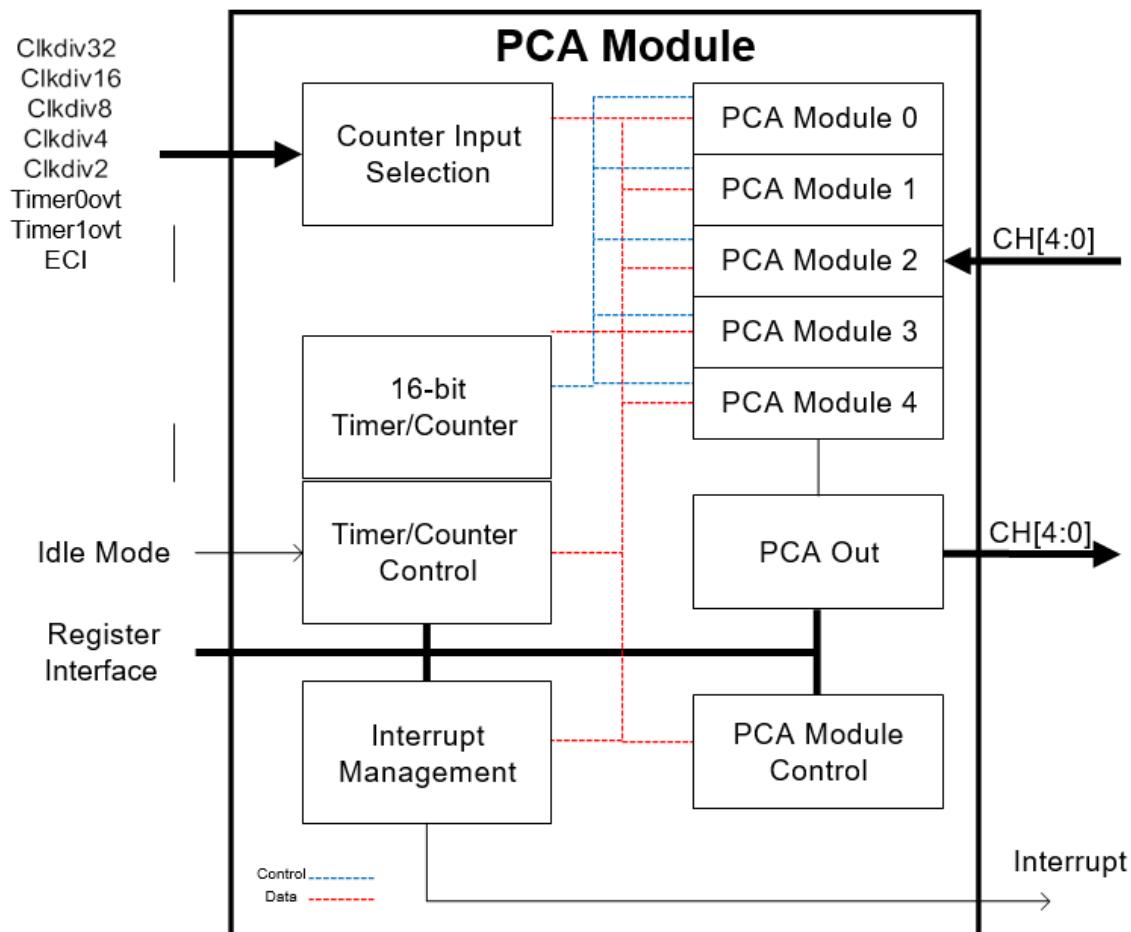


Figure 15- 1 PCA 整体框图

## 15.2PCA 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。

### 15.2-1PCA 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。

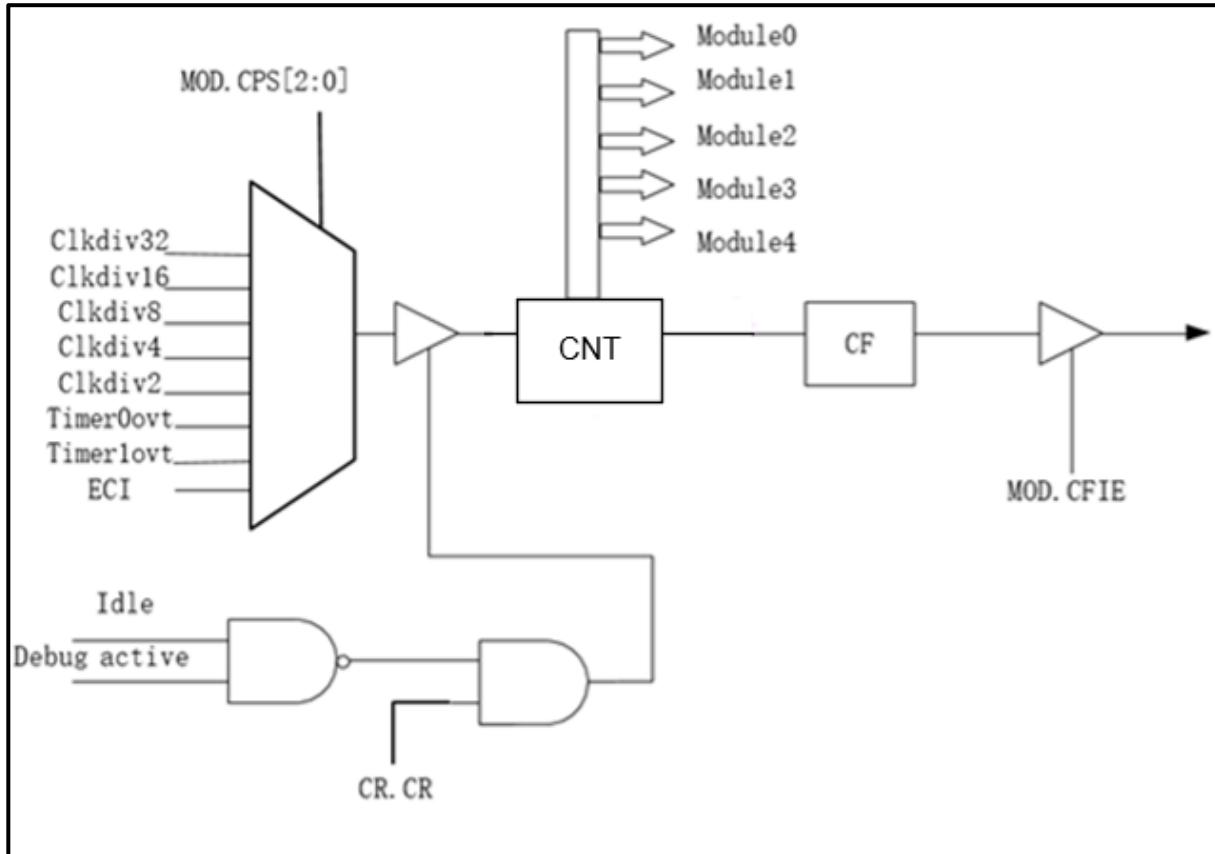


Figure 15- 2 PCA 计数器框图

### 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：

- 当需要在外部上升沿进行捕获, CCPMx.CAPP = "1" 以及 CCAPMx.CAPN = "0"
  - 当需要在外部下降沿进行捕获, CCPMx.CAPP = "0" 以及 CCAPMx.CAPN = "1"
  - 当需要在外部上升、下降沿进行捕获, CCPMx.CAPP = "1" 以及 CCAPMx.CAPN = "1"

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。

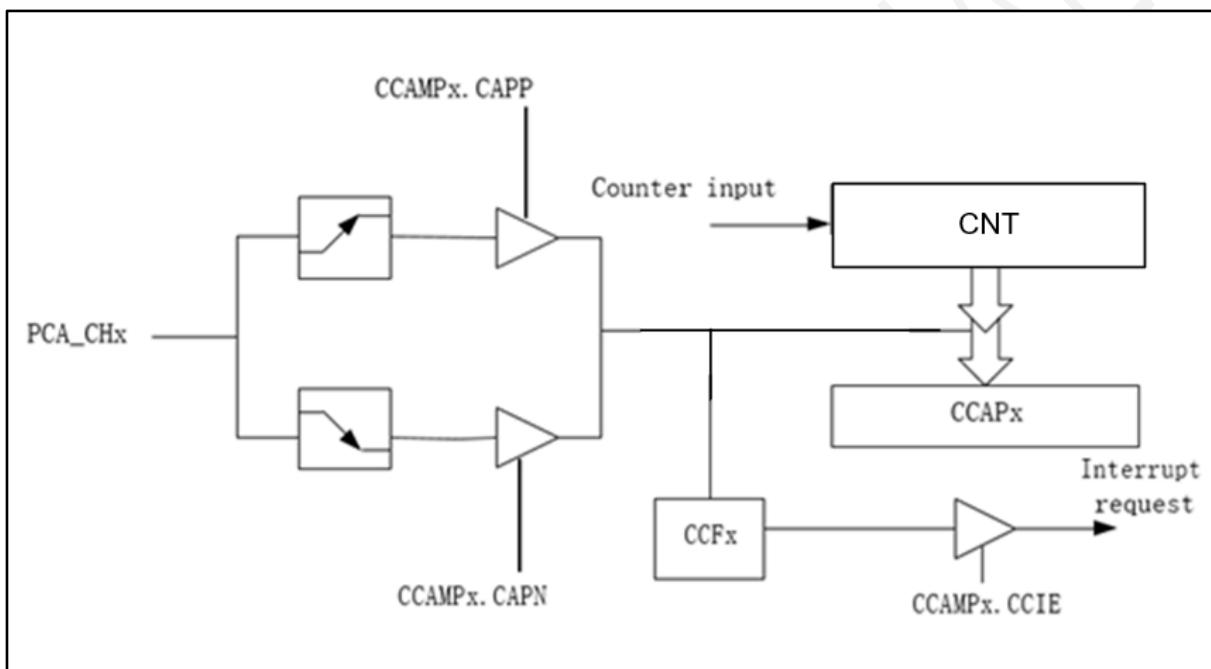


Figure 15- 3 PCA 捕获功能框图

## 15.2-3PCA 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

### 15.2-3.116 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。

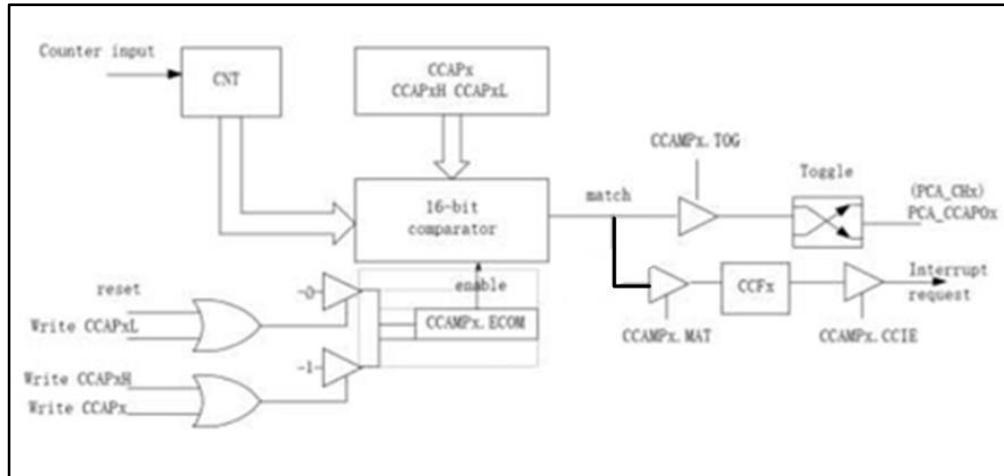


Figure 15-4 PCA 比较功能框图

### 15.2-3.38 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。

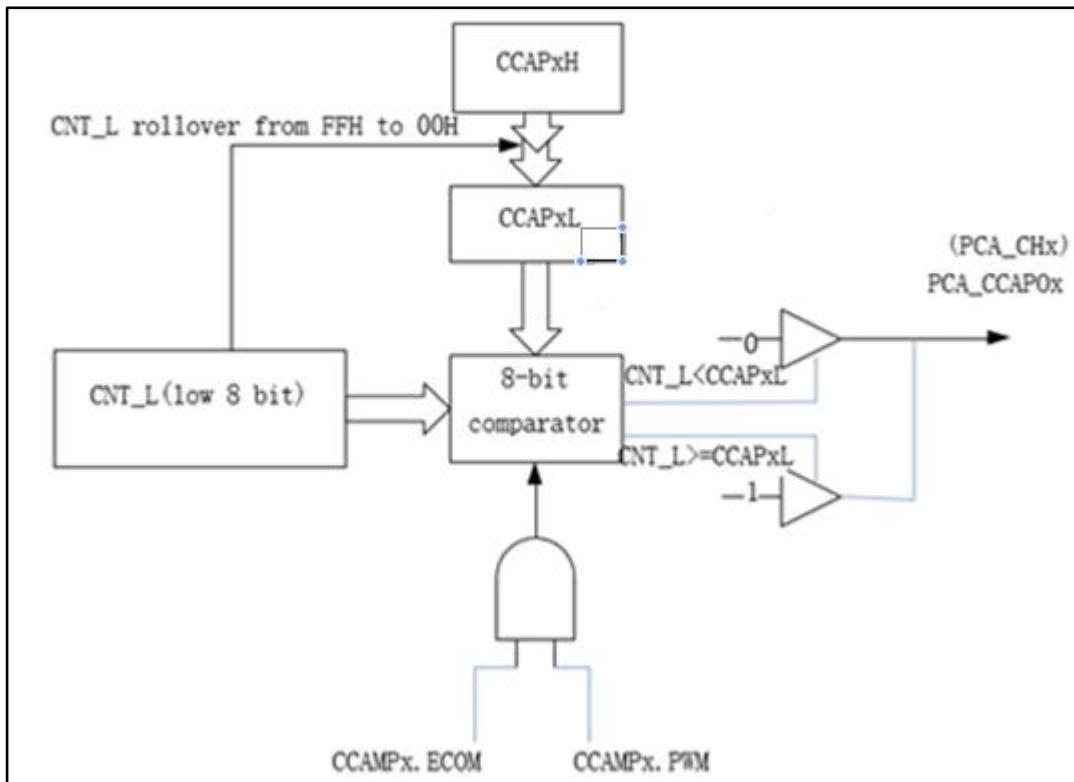


Figure 15- 5 PCA PWM 功能框图

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。

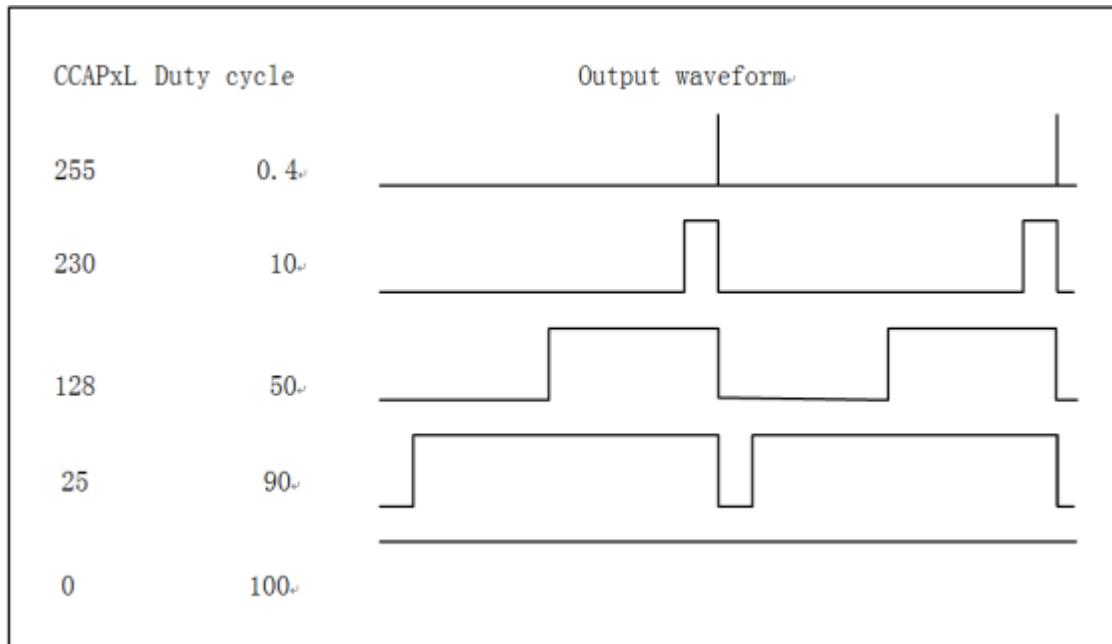


Figure 15- 6 PCA PWM 输出波形

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

### 15.3PCA 模块与其他模块互连及控制

#### 15.3-1ECI 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。

#### 15.3-2PCACAP0

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。

#### 15.3-3PCACAP1/2/3/4

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

基址址 0x4000 1400

地址偏移	名称	描述	复位值
0x00	PCA_CR	PCA 控制寄存器	0x0000 0000
0x04	PCA_MOD	PCA 模式寄存器	0x0000 0000
0x08	PCA_CNT	PCA 计数寄存器	0x0000 0000
0x0C	PCA_INTCLR	PCA 中断清除寄存器	0x0000 009F
0x10	PCA_CCAPM0	PCA 比较/捕获模块 0 模式寄存器	0x0000 0000
0x14	PCA_CCAPM1	PCA 比较/捕获模块 1 模式寄存器	0x0000 0000
0x18	PCA_CCAPM2	PCA 比较/捕获模块 2 模式寄存器	0x0000 0000
0x1C	PCA_CCAPM3	PCA 比较/捕获模块 3 模式寄存器	0x0000 0000
0x20	PCA_CCAPM4	PCA 比较/捕获模块 4 模式寄存器	0x0000 0000
0x30	PCA_CCAP0L	PCA 比较/捕获模块 0 低 8 位寄存器	0x0000 0000
0x34	PCA_CCAP0H	PCA 比较/捕获模块 0 高 8 位寄存器	0x0000 0000
0x38	PCA_CCAP1L	PCA 比较/捕获模块 1 低 8 位寄存器	0x0000 0000
0x3C	PCA_CCAP1H	PCA 比较/捕获模块 1 高 8 位寄存器	0x0000 0000
0x40	PCA_CCAP2L	PCA 比较/捕获模块 2 低 8 位寄存器	0x0000 0000
0x44	PCA_CCAP2H	PCA 比较/捕获模块 2 高 8 位寄存器	0x0000 0000
0x48	PCA_CCAP3L	PCA 比较/捕获模块 3 低 8 位寄存器	0x0000 0000
0x4C	PCA_CCAP3H	PCA 比较/捕获模块 3 高 8 位寄存器	0x0000 0000
0x50	PCA_CCAP4L	PCA 比较/捕获模块 4 低 8 位寄存器	0x0000 0000
0x54	PCA_CCAP4H	PCA 比较/捕获模块 4 高 8 位寄存器	0x0000 0000
0x58	PCA_CCAPO	PCA PWM 与高速输出标志寄存器	0x0000 0000
0x5C	PCA_POCR	PCA 端子输出控制寄存器	0x0000 0000
0x60	PCA_CCAP0	PCA 比较/捕获模块 0 的 16 位寄存器	0x0000 0000
0x64	PCA_CCAP1	PCA 比较/捕获模块 1 的 16 位寄存器	0x0000 0000
0x68	PCA_CCAP2	PCA 比较/捕获模块 2 的 16 位寄存器	0x0000 0000
0x6C	PCA_CCAP3	PCA 比较/捕获模块 3 的 16 位寄存器	0x0000 0000
0x70	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: 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								CF	CR	保留	CCF4	CCF3	CCF2	CCF1	CCF0
--								RO	R/W	--	RO				

位	标记	功能描述	复位值	读写
31:8	保留	--	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	--	0x00	--
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CCAPM4.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

位	标记	功能描述	复位值	读写
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

### 15.5-2模式寄存器(PCA\_MOD)

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								CIDL	保留				CPS	CFIE	
--								R/W	--				R/W		

位	标记	功能描述	复位值	读写
31:8	保留	--	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	--	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

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															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	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: 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
保留								CF	保留		CCF4	CCF3	CCF2	CCF1	CCF0
--								WO	--		WO				

位	标记	功能描述	复位值	读写
31:8	保留	--	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	--	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:

CCAPM0: 0x10

CCAPM1: 0x14

CCAPM2: 0x18

CCAPM3: 0x1C

CCAPM4: 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								ECOM	CAPP	CAPN	MAT	TOG	PWM	CCIE	
--								R/W							

位	标记	功能描述	复位值	读写
31:7	保留	--	0x00	--
6	ECOM	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 ECOM; 当写CCAMPHx 或CCAMPx 寄存器会自动置位ECOM; 当写CCAMPLx 寄存器会自动清除ECOM 位;	0	R/W
5	CAPP	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	CAPN	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	MAT	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

位	标记	功能描述	复位值	读写
2	TOG	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:

CCAP0L: 0x30

CCAP1L: 0x38

CCAP2L: 0x40

CCAP3L: 0x48

CCAP4L: 0x50

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								CCAPx[7:0]							
--								R/W							

位	标记	功能描述	复位值	读写
31:8	保留	--	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

### 15.5-7 比较捕获数据寄存器高 8 位(PCA\_CCAP0~4H)

地址偏移:

CCAP0H: 0x34

CCAP1H: 0x3C

CCAP2H: 0x44

CCAP3H: 0x4C

CCAP4H: 0x54

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								CCAPx[15:8]							
--								R/W							

位	标记	功能描述	复位值	读写
31:8	保留	--	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: 0x58      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															
--															

位	标记	功能描述	复位值	读写
31:5	保留	--	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: 0x5C 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	POINV4	POINV3	POINV2	POINV1	POINV0	保留	POE4	POE3	POE2	POE1	POE0	--	--	--	--
--	R/W	--	--	--	--	--	R/W	--	--	--	--	--	--	--	--

位	标记	功能描述	复位值	读写
31:13	保留	--	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	--	0x00	--
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

位	标记	功能描述	复位值	读写
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

### 15.5-10 比较捕获 16 位寄存器(PCA\_CCAP0~4)

地址偏移:

CCAP0: 0x60

CCAP1: 0x64

CCAP2: 0x68

CCAP3: 0x6C

CCAP4: 0x70

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
CCAPx															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	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。

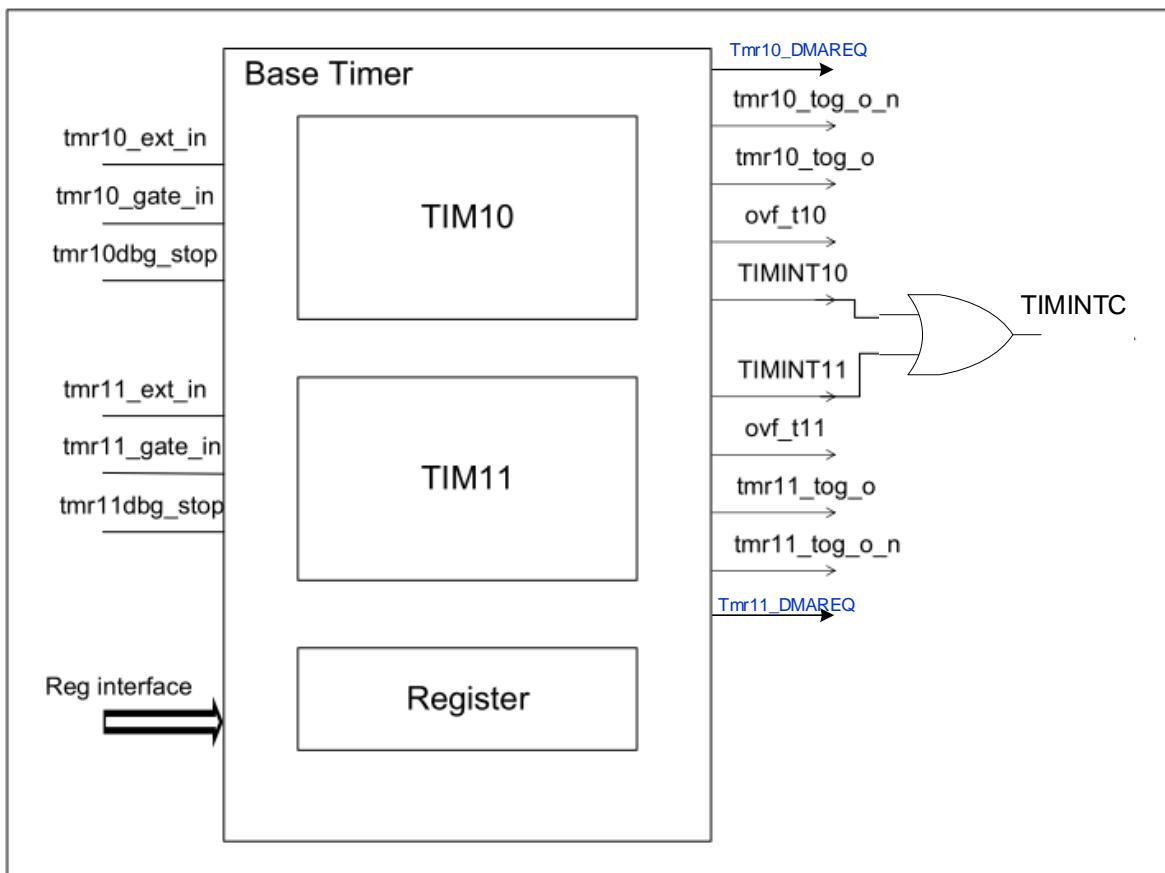


Figure 16- 1 Base Timer 框图

## 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。

<code>TMR_PRE[2:0]</code>	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  $\text{Max}=0xFFFF$ , 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。

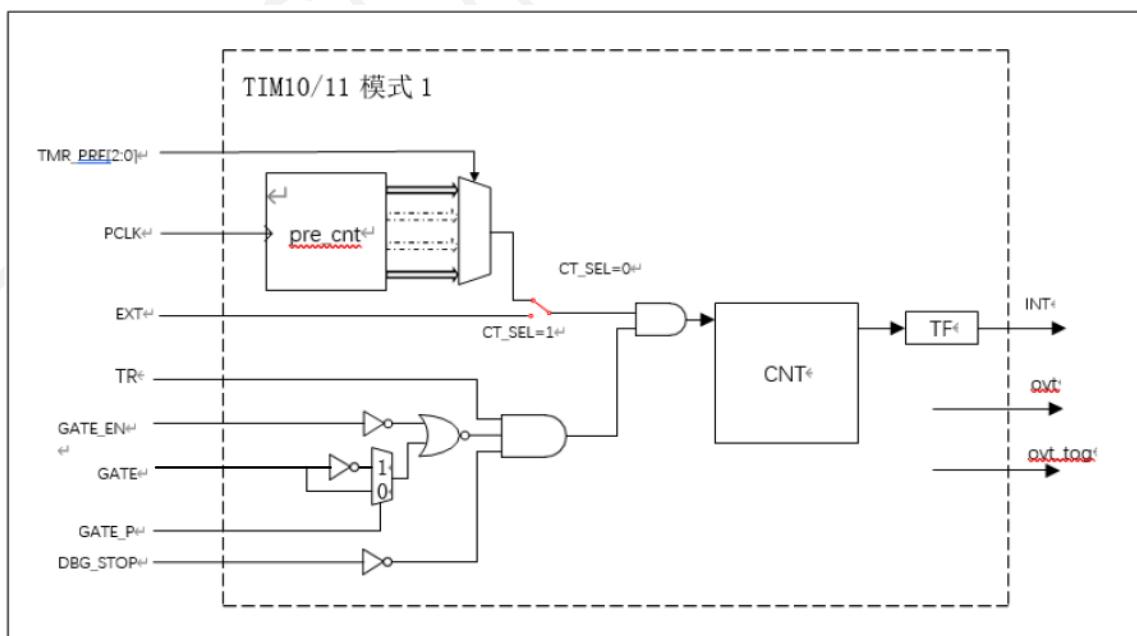


Figure 16- 2 Timer 模式 1 框图

## ● 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。

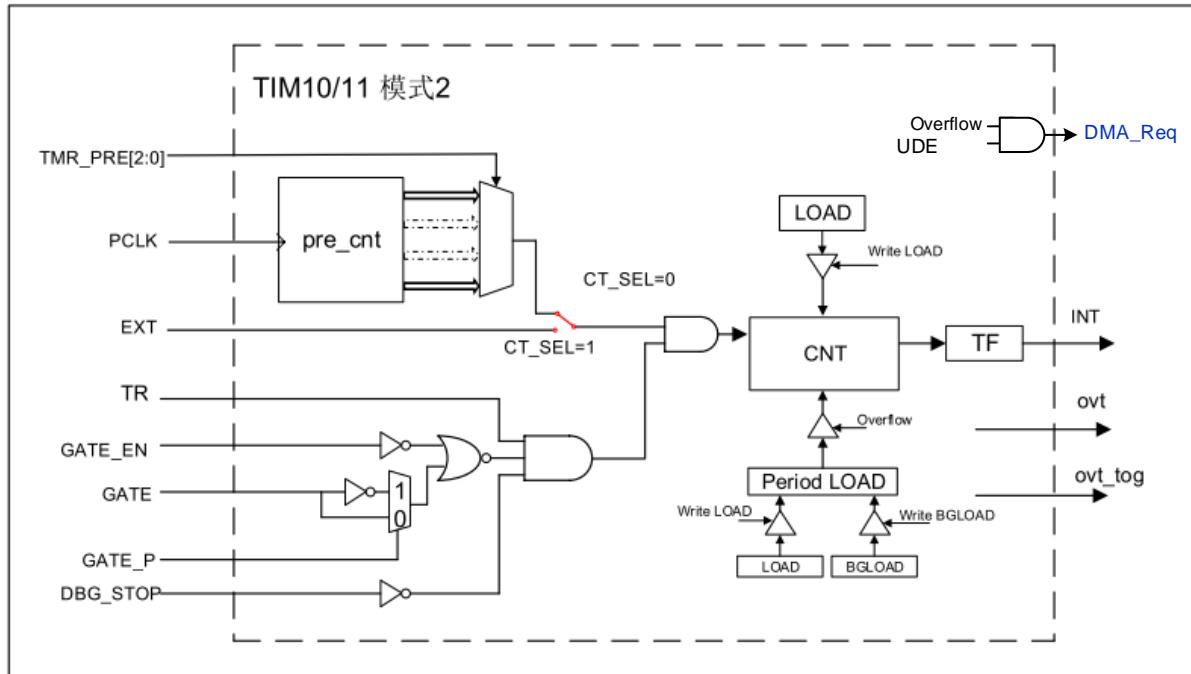


Figure 16-3 Timer 模式 2 框图

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。

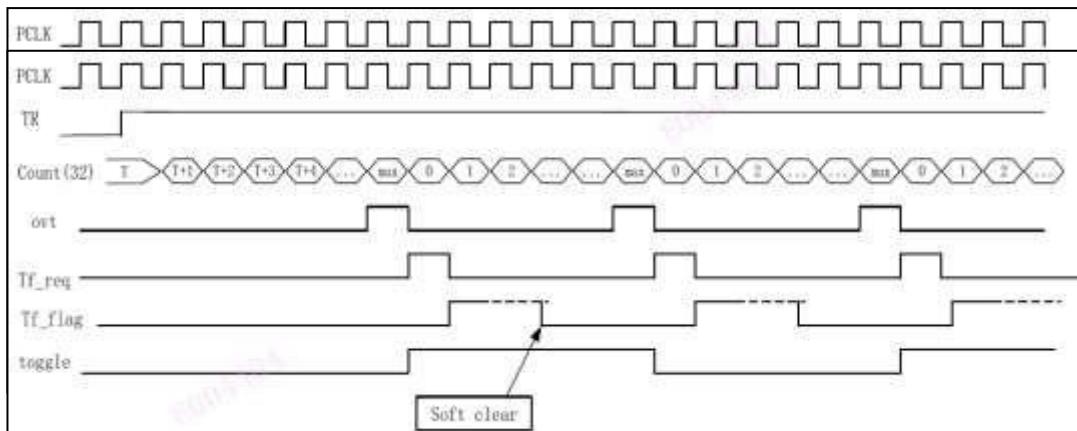


Figure 16- 4 32 位模式 1 时序图(max=0xFFFF FFFF)

### 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。

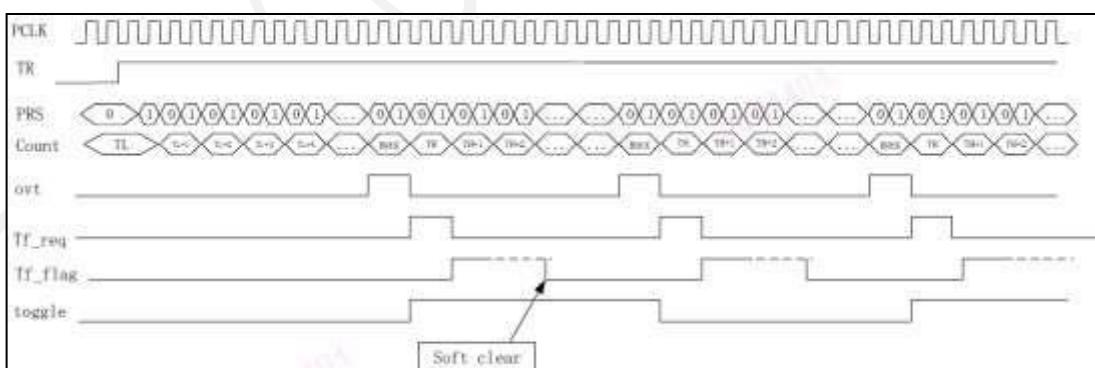


Figure 16- 5 32 位模式 2 时序图(PCLK 二分频, max=0xFFFF FFFF)

### 16.2-3Buzzer 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

### 16.3Base Timer 互连

#### 16.3-1GATE 互联

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。

#### 16.3-2Toggle 输出互联

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 寄存器列表

x=10 或 11;

Base Timer 基地址 0x 4000 1800

	地址偏移	描述
TIM10	0x00	TIM10 地址偏移
TIM11	0x100	TIM11 地址偏移

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_RAWINTSR	读原始中断寄存器,	0x0000 0000
0x10	TIMx_MSKINTSR	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 Base Timer 寄存器列表和复位值

### 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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		UDE	GATE_P	GATE_EN	TOG_EN	CT_SEL	TR	MODE	INTEN	TMR_SIZE	ONE_SHOT		TMR_PRSC[2:0]		
--												R/W			

位	符号	描述	复位值	读写
31:13	保留	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。外部输入由 PCLK 采样, 外部输入时钟频率要低于 1/2 采样时钟。	0	R/W
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

位	符号	描述	复位值	读写
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

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

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

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

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

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

位	符号	描述	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

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															
--															

位	符号	描述	复位值	读写
31:1	保留	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

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															
--															

位	符号	描述	复位值	读写
31:1	保留	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															
INTCLR															
--															

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。

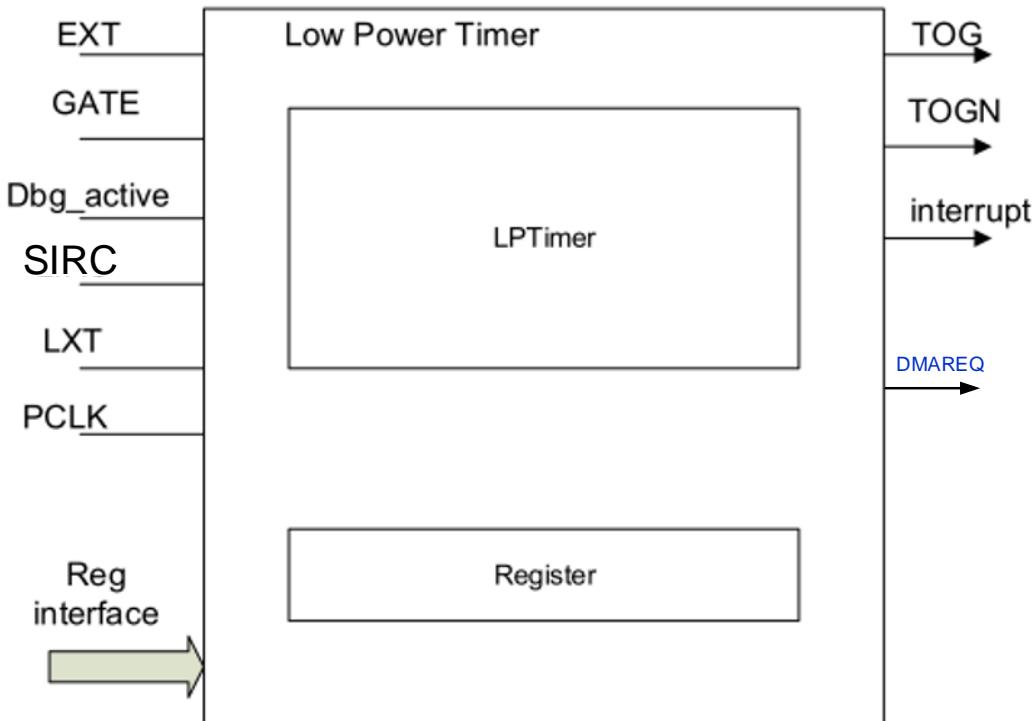


Figure 17- 1 LPTIMER 结构框图

### 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更新 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。

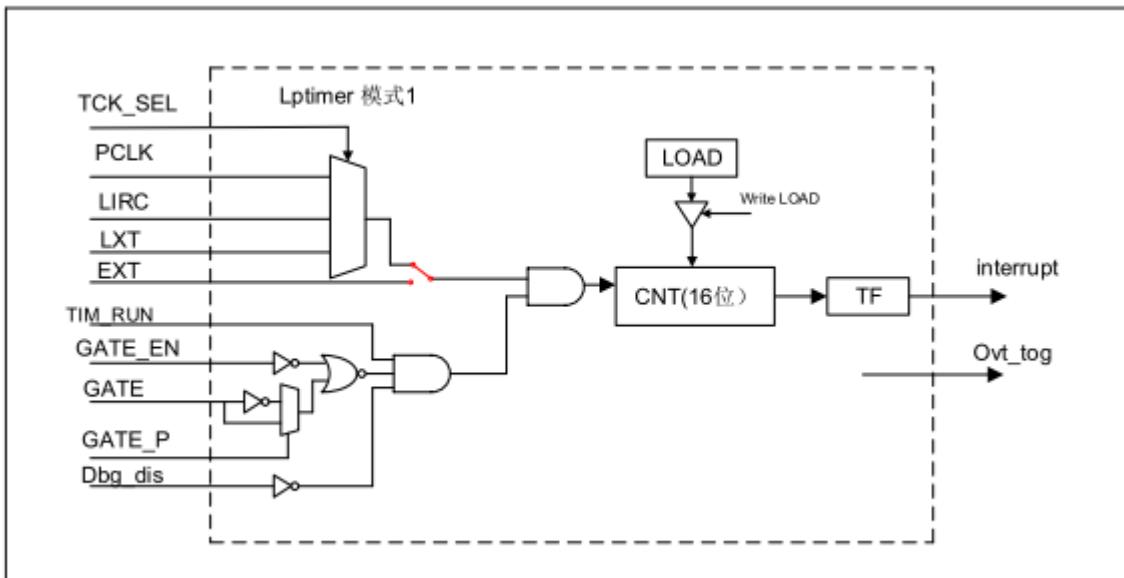


Figure 17- 2 LPTIMER 模式 1

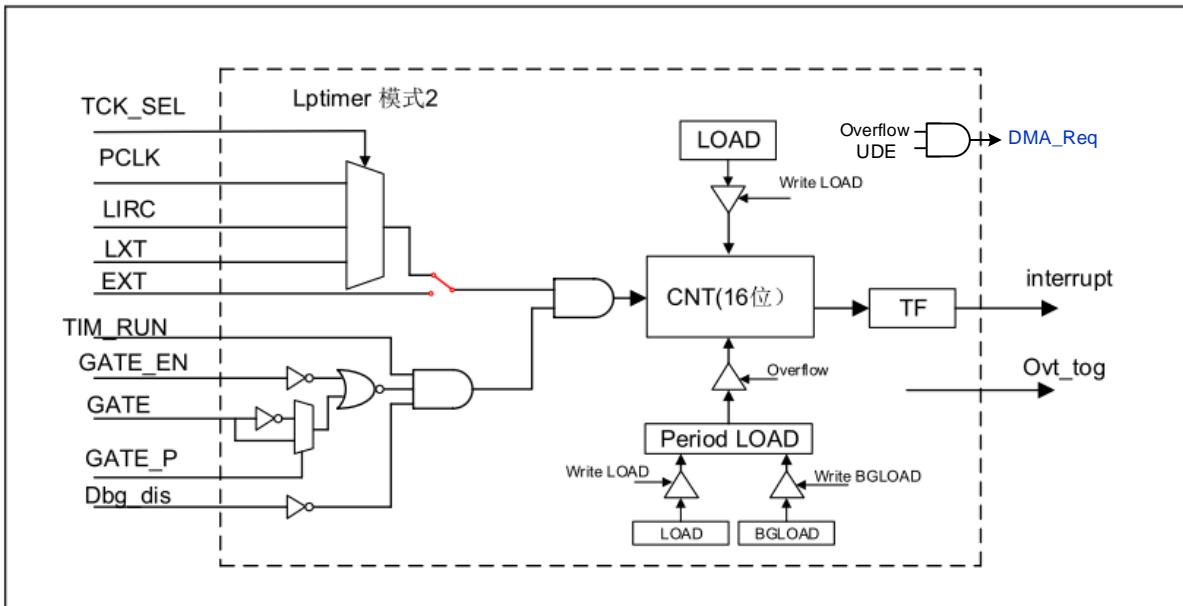


Figure 17- 3 LPTIMER 模式 2

### 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

基址址: 0x4000 4400

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

## 17.4LPTIM 寄存器说明

### 17.4-1LPTIM 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															
--															

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

### 17.4-2LPTIM 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	
--														RO	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					UDE	TCK_EN	INT_EN	GATE_P	GATE_EN	TCK_SEL[1:0]	TOG_EN	CT_SEL	MODE	TIER_RUN	
--															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
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

位	标记	功能描述	复位值	读写
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

### 17.4-3LPTIM 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[15:0]															
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

### 17.4-4LPTIM 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															
--															

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

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

### 17.4-5LPTIM 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															
--															

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

### 17.4-6LPTIM 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)

ZB32L032 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。

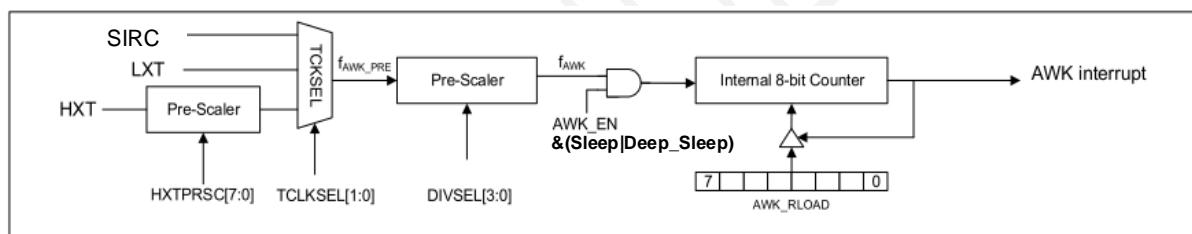


Figure 18- 1 自唤醒定时器结构图

### 18.1 寄存器列表

基址地址: 0x4000 2800

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

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	
XTLPRSC[7:0]								保留	TCLKSEL[1:0]	AWKEN	DIVSEL[3:0]					
R/W								--	R/W							

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: fHXT/(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

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								RLDVAL[7:0]							
--								R/W							

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															
--															

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

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

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															
--															

位	标记	功能描述	复位值	读写
31:1	保留	--	0x0	--
0	INTCLR	自动唤醒中断清除 0: 无作用 1: 清除自动唤醒中断	0x0	WO

## 【19】蜂鸣器(BEEP)

### 19.1简介

选择 SIRC 时钟、HXT 时钟或者 PCLK，可通过分频设定来产生各种频率的蜂鸣信号。

BEEP\_CSR.CLKSEL[1:0]位来选择得到  $f_{BEEP\_PRE}$  时钟，通过设置 BEEPDIV[11:0]将蜂鸣器时钟  $f_{BEEP\_PRE}$  分频得到  $f_{BEEP}$ 。通过设置 BEEPSEL[1:0]，得到  $f_{BEEP\_O}$  蜂鸣信号。

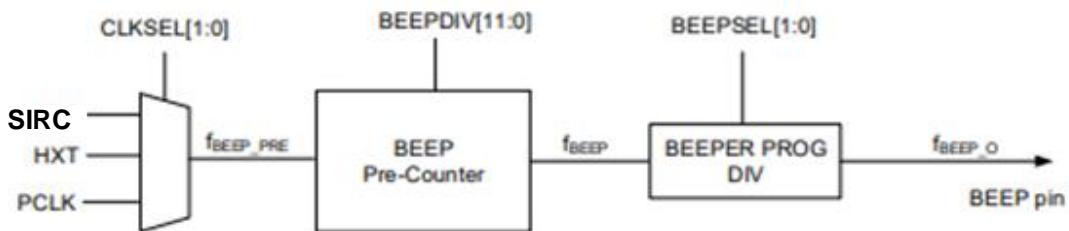


Figure 19- 1 蜂鸣器功能图

### 19.2功能描述

#### 19.2-1蜂鸣器操作

为了使用蜂鸣功能，按顺序执行如下的步骤：

1. 根据需要的蜂鸣器频率输出值确定 BEEPDIV[11:0]的值；
2. 通过写 BEEP\_CSR 的 BEEPSEL[1:0]位来选择  $f_{BEEP\_O}$  的输出频率；
3. 置位 BEEP\_CSR 的 BEEPEN 位来使能时钟源；

注意：

预分频计算器仅仅在当 BEEPDIV[11:0]的值不同于复位值 0xFFFF 时才开始运行。

在蜂鸣器运行过程中应该保持 BEEPDIV[11:0]的值不变。

### 19.2-2蜂鸣器校准

该步骤可以用来校准 SIRC 时钟以便达到更标准的  $f_{BEEP\_O}$ (1KHz、2KHz 或 4KHz)频率输出。

采用如下的步骤：

1. TIM1、TIM2 或者 CLKTRIM 模块来测量内部低速时钟(SIRC)的时钟频率
2. 采用如下方法计算 BEEPDIV 的值，这里 A 和 x 是  $f_{BEEP\_PRE}/f_{BEEP}$  的整数和小数部分值：当 x 小于或者等于  $A/(1+2*A)$ 时， $BEEPDIV = A-1$ ；否则  $BEEPDIV = A$
3. 将 BEEPDIV 值写入到 BEEP\_CSR 的 BEEPDIV[11:0]位

### 19.3 寄存器列表

BEEP 基地址: 0x4000 4800

偏移地址	名称	描述	复位值
0x000	BEEP_CSR	蜂鸣器控制寄存器	0x0000 0FFF

Table 19- 1 BEEP 寄存器列表和复位值

## 19.4寄存器说明

### 19.4-1蜂鸣器控制/状态寄存器(BEEP\_CSR)

地址偏移: 0x00 复位值: 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CLKSEL	保留		BEEPEN	BEEPSEL			
--								R/W	--		R/W	R/W			

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

位	标记	功能描述	复位值	读写
31:22	保留	--	0x0	--
21:20	CLKSEL	时钟选择 00: 停止 01: SIRC 10: HXT 11: PCLK	0x0	R/W
19	保留	--	0x0	--
18	BEEPEN	使能蜂鸣	0	R/W
17:16	BEEPSEL	蜂鸣器输出BEEP_O 频率选择位 00: fBEEP_PRE/8 01: fBEEP_PRE/4 1x: fBEEP_PRE/2	0x0	R/W
15:12	保留	--	0x0	--
11:0	BEEPDIV [11:0]	蜂鸣器预分频器将fBEEP_PRE 分频得到 8KHz 周期的信号分频因子为BEEPDIV + 1 $f_{BEEP} = f_{BEEP\_PRE}/(BEEPDIV + 1)$	0xffff	R/W

## 【20】独立看门狗(IWDG)

### 20.1 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。

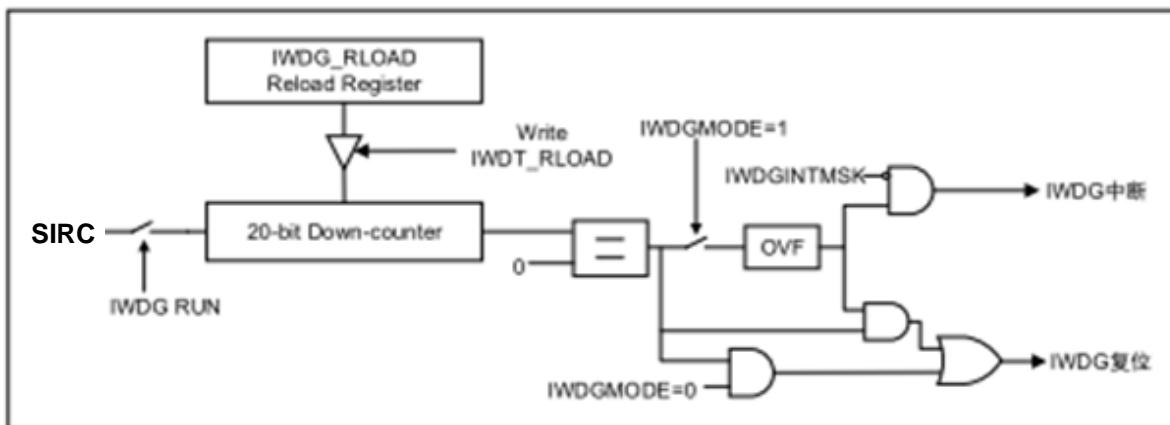


Figure 20- 1 IWDG 整体框图

### 20.2 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。

## 20.2-1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
0x00000	26us
...	...
0x003FF	26.6ms
...	...
0xFFFFF	27s

## 20.2-2IWDG 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。

## 20.2-3IWDG 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。

### 20.3寄存器列表

基址地址: 0x4000 2400

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 20- 1 IWDG 寄存器列表和复位值

## 20.4寄存器说明

### 20.4-1IWDG 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]							
--								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

### 20.4-2IWDG 配置寄存器(IWDG\_CFR)

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															
--															

位	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

#### 20.4-3IWDG 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]			
--												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	--	0x0	--
19:0	IWDGRLOAD[19:0]	IWDG 重装载寄存器	0xFFFF	R/W

注意: 受 IWDG\_UNLOCK 保护

#### 20.4-4IWDG 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]			
--												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	--	0x0	--
19:0	IWDGCNT[19:0]	IWDG 计数值寄存器。	0xFFFF	RO

## 20.4-5IWDG 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
--															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</p> <p>注意:</p> <p>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

## 20.4-6IWDG 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															
--															

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

## 20.4-7IWDG 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															
--															

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\_CFGR、IWDG\_RLOAD、IWDG\_INTCLR)。

## 20.5 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。

## 【21】系统窗口看门狗(WWDG)

### 21.1 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。

### 21.2 特征

- (1) 一个 8 位向下数计数器(WWDG\_CNT)和一个 8 位比较值(WINCM)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

### 21.3 结构框图

窗口看门狗定时器框图如下：

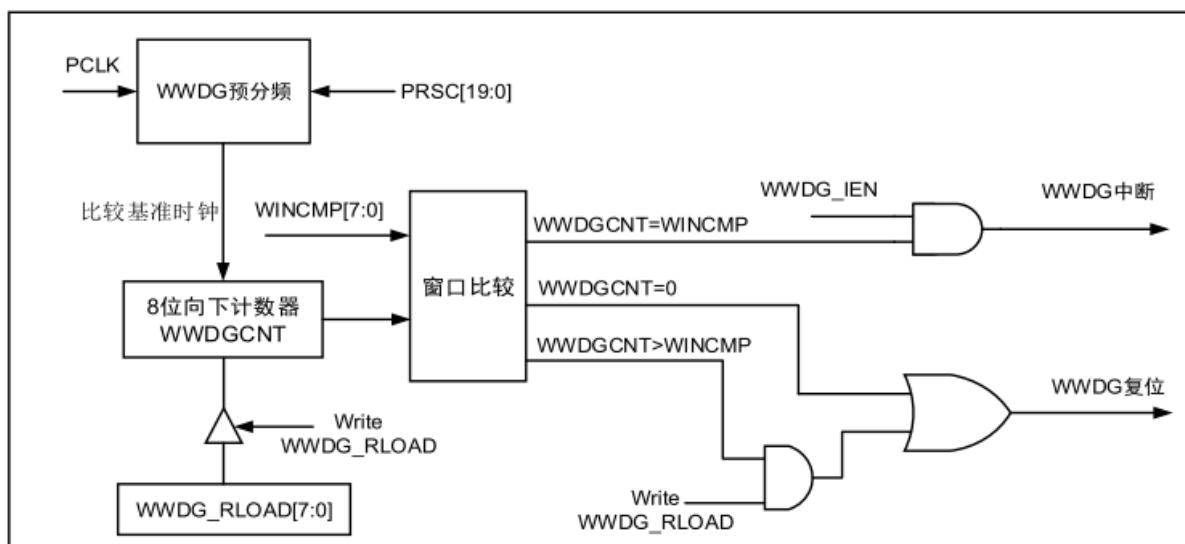


Figure 21- 1 WWDG 结构框图

### 21.4 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.WINCM[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

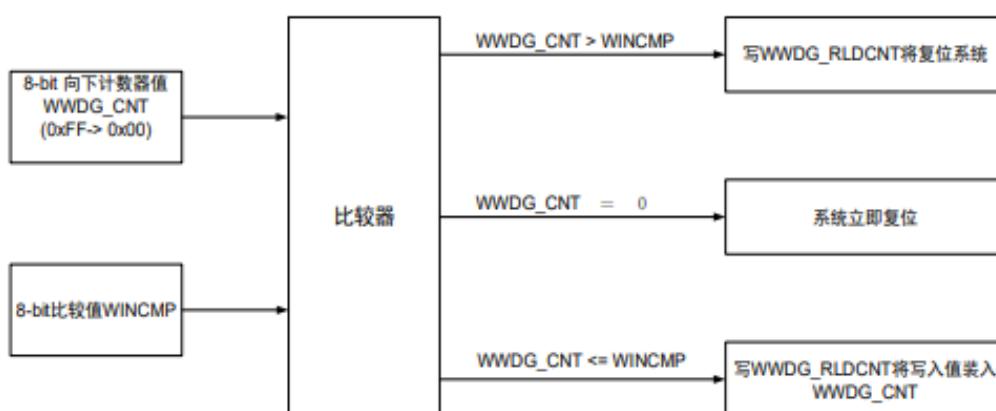
## 21.5 功能描述

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	TPCLK * 1	41.7ns
0x00001	2	TPCLK * 2	83.4ns
0x00002	3	TPCLK * 3	125.1ns
0x00003	4	TPCLK * 4	166.8ns
0x00004	5	TPCLK * 5	208.5ns
...	...	...	...
0x80000	524289	TPCLK * 524289	21.9ms
...	...	...	...
0xFFFFF	1048576	TPCLK * 1048576	43.8ms

### 21.5-1 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.WINCMP[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。



### 21.5-2 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。

### 21.5-3 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[7:0]` 时进行。如果 `WWDG` 计数器当前值大于 `WINCM[7:0]`，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。

### 21.5-4 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，

$$T_{PCLK} = T_{HCLK} * (2 * RCC_PCLKDIV.APBCKDIV[7:0])$$

设定时间间隔：

$$T = T_{PCLK} * (WWDG_CR.WWDG_PRSC[19:0] + 1) * (WWDG_RLOAD.WWDG_RLOAD[7:0] + 1)$$

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。

## 21.6 与独立看门狗定时器(IWDG)比较

### 21.6-1 复位条件和复位延时

IWDG 和 WWDG 通常是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[7:0]` 时往 `WWDG_RLOAD` 写入。

一旦 `WWDGRST` 被置 1，WWDG 将立即复位系统。

### 21.6-2 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。

## 21.7 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 21- 1 WWDG 寄存器列表和复位值

## 21.8寄存器说明

### 21.8-1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

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								WWDG_RLOAD[7:0]							
--								WO							

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

### 21.8-2 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															

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)	0x80000	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。

### 21.8-3 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

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														WWDGIEN	
--														R/W	

位	标记	功能描述	复位值	读写
31:1	保留	--	0x0	--
0	WWDGIEN	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

### 21.8-4 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

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														WWDGIF	
--														RO	

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

### 21.8-5 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

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															
--															

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

### 21.8-6 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

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								WWDGcnt[7:0]							
--								RO							

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

## 【22】通用异步/同步收发器(UART0/1,USART2/3)

### 22.1 概述

- UART0/1(No DMA request)

本产品带有 2 个通用 UART 模块(UART0/1)，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；UART0 的波特率可以由 TIM10 产生或者自动波特率发生器产生，UART1 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(UART0/1)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), 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
- 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

## 22.2 结构框图

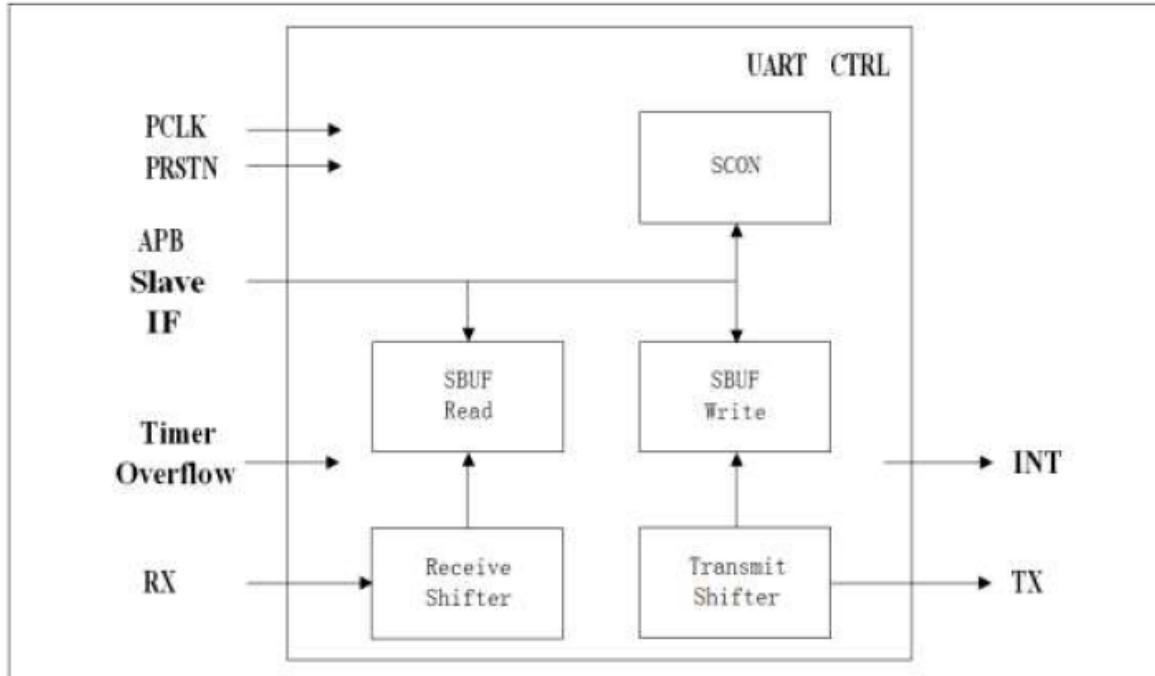


Figure 22- 1 UART 结构图

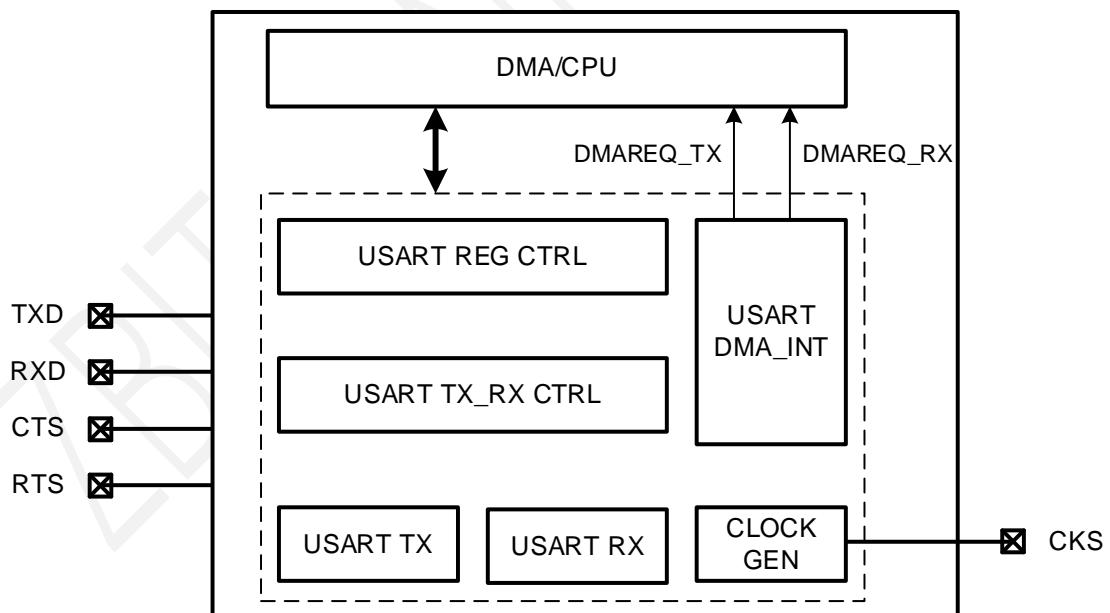


Figure 22- 2 USART 结构图

**RX:** 接收数据。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 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

## 22.3 UART 工作模式

### 22.3-1 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.SMO 和 UARTx\_SCON.SM1 清零, 可进入 Mode0 工作模式。

#### 22.3-1.1 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。

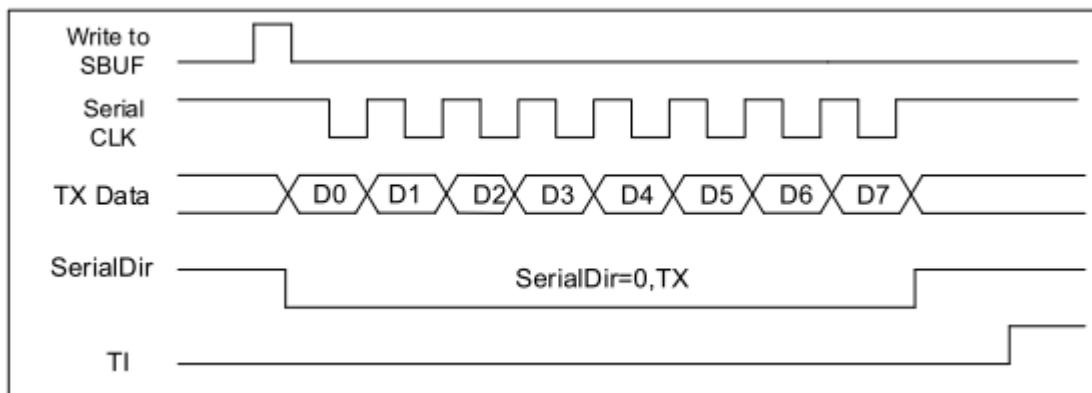


Figure 22- 3 Mode0 发送数据

#### 22.3-1.2 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。

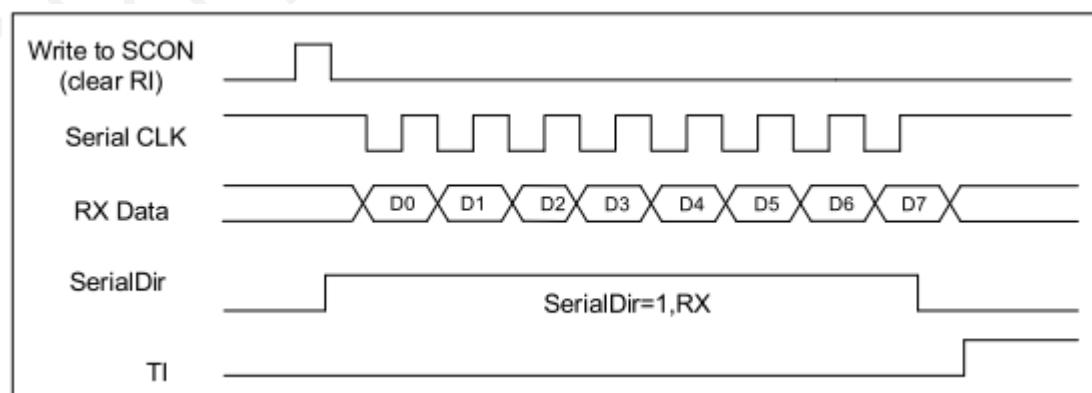


Figure 22- 4 Mode0 接收数据

### 22.3-2 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。

#### 22.3-2.1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)。

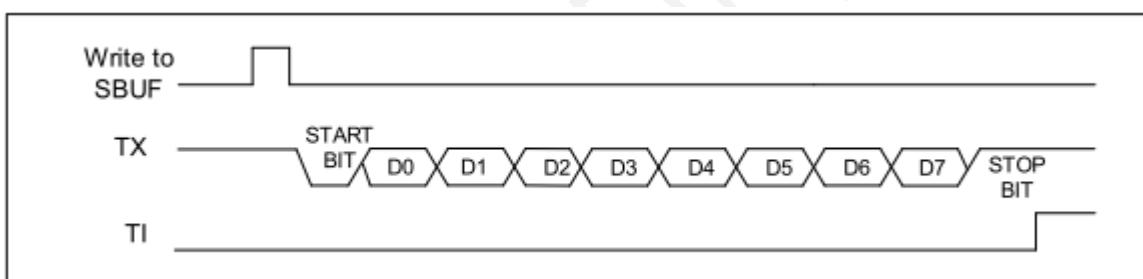


Figure 22- 5 Mode1 发送数据

#### 22.3-2.2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。

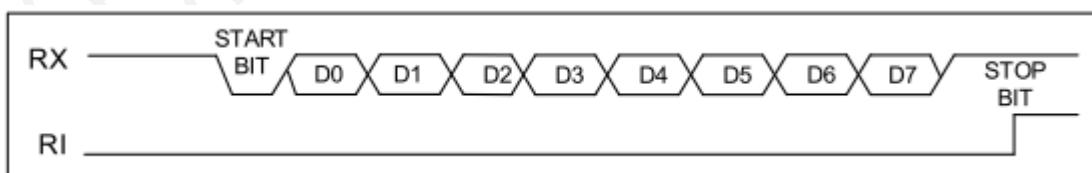


Figure 22- 6 Mode1 接收数据

### 22.3-3 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。

#### 22.3-3.1 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)。

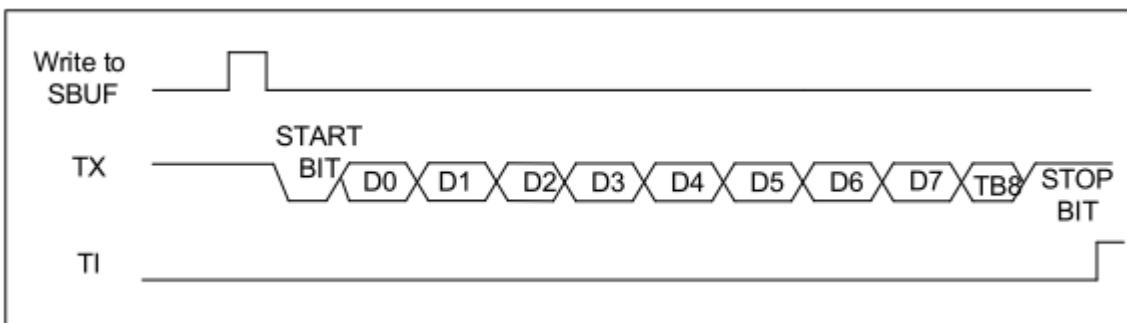


Figure 22- 7 Mode2 发送数据

#### 22.3-3.2 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。

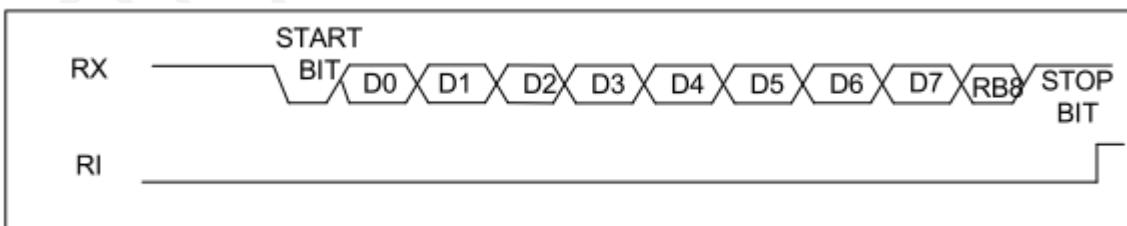


Figure 22- 8 Mode2 接收数据

### 22.3-4Mode 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。

### 22.3-5波特率编程

#### 22.3-5.1Mode 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。

#### 22.3-5.2Mode 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.SEFL_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.SEFL_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。

#### 22.3-5.3Mode 2

当工作在 Mode 2 时，波特率被固定在如下公式所得值：

$$\text{BaudRate} = \frac{(\text{UARTX\_SCON.DBAUD} + 1) * F_{\text{PCLK}}}{64}$$

其中，`UARTX_SCON.DBAUD` 表示双倍波特率，`FPCLK` 为 PCLK 时钟频率。

### 22.3-6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。

### 22.3-7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。

### 22.3-8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。

### 22.3-9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`。

**GivenAddr = `UARTx_SADDR & UARTx_SADEN`**

### 22.3-10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。

**BoardCastAddr = UARTx\_SADDR | 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)

### 22.3-11收发端缓存

#### 22.3-11.1接收缓存

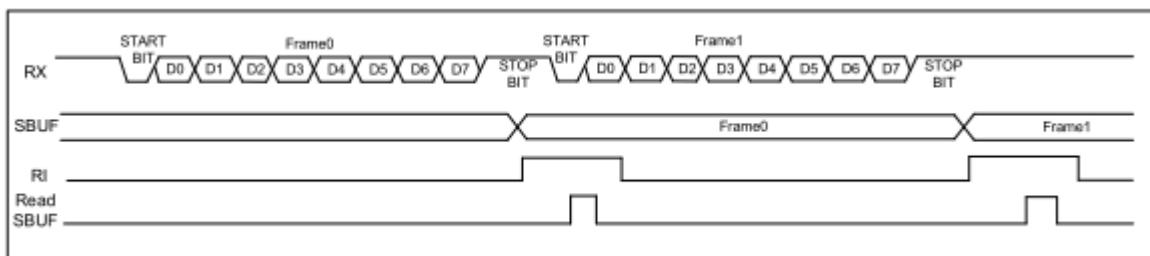


Figure 22- 9 接收缓存

#### 22.3-11.2发送缓存

通用 UART(UART0/1)发送端不支持发送缓存。如果在发送数据过程中，填写 UARTx\_SBUF 寄存器，将会破坏当前正在发送数据。软件应该避免这种操作。

### 22.3-12IrDA 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。

### 22.3-12.1IrDA 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  $1.42\text{MHz}$ 。通常这个值是  $1.8432\text{MHz}$  ( $1.42\text{MHz} < \text{PSC} < 2.12\text{MHz}$ )

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  $10\text{ms}$  的延时(IrDA 是一个半双工协议)。

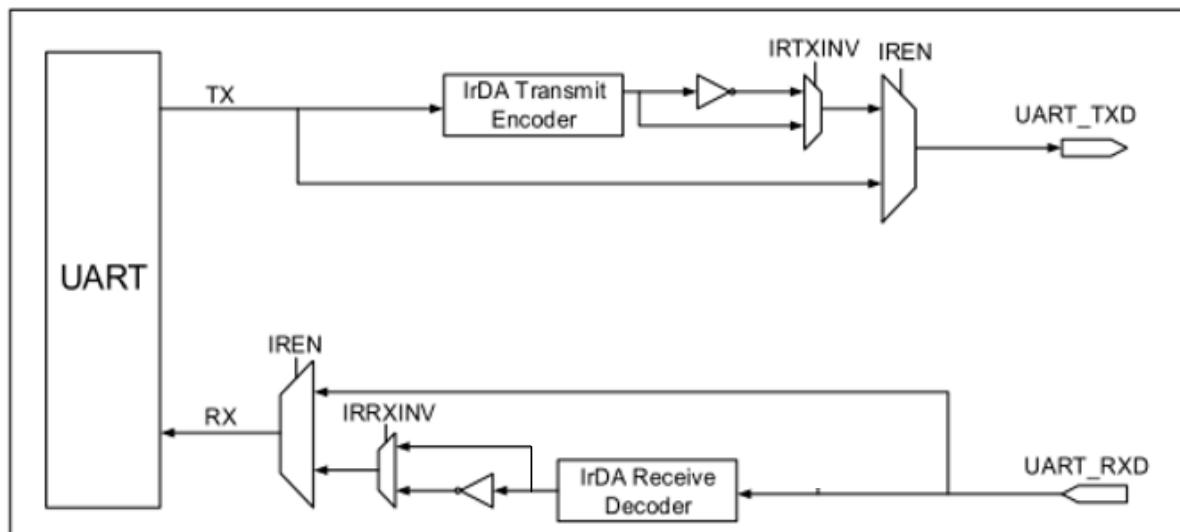


Figure 22- 10 IrDA 结构框图

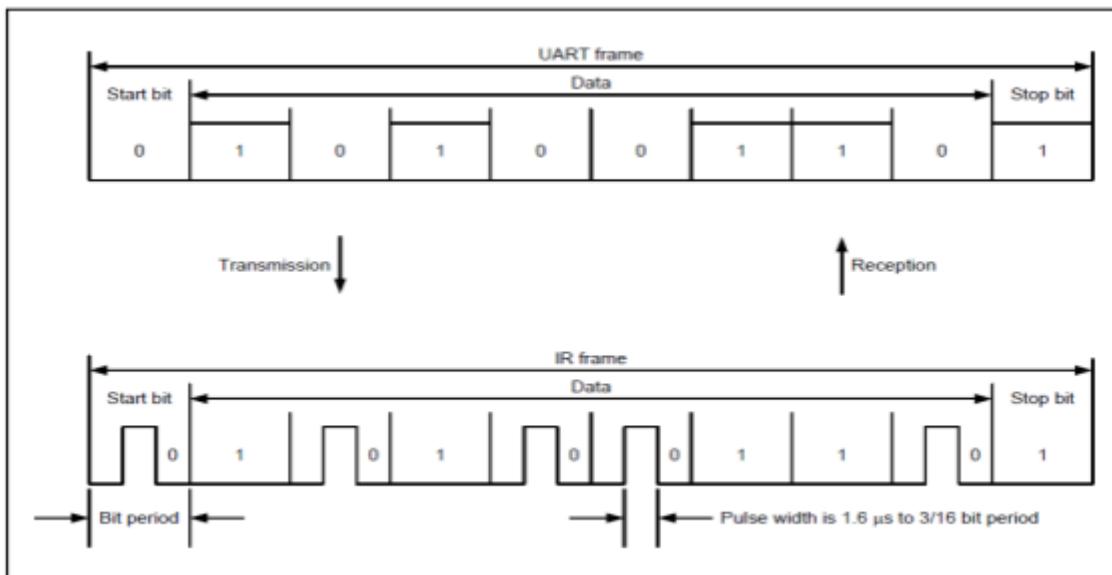


Figure 22-11 IrDA 收发脉冲

## 22.4 USART 工作模式

USART 工作模式 选择 如下表:

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	0	0	0	1	00	红外线数据传输(IrDA)

Figure 22- 12 USART 工作模式选择表

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
IrDA	USART_CR1.M (M=0:8 位; M=1:9 位)	1	PSC 配置 (low-power)	10 or 11 "0"	None

Figure 22- 13 USART 工作模式控制表

## 22.4-1 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 22-14)。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。

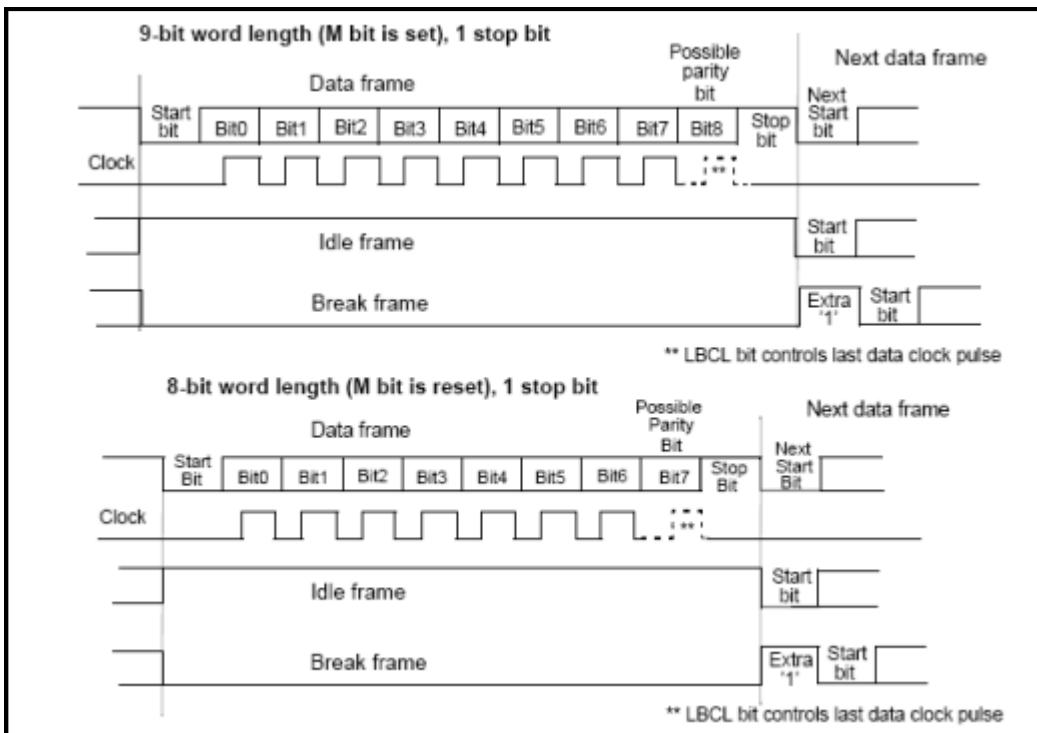


Figure 22- 14 字长设置

## 22.4-2 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。

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)

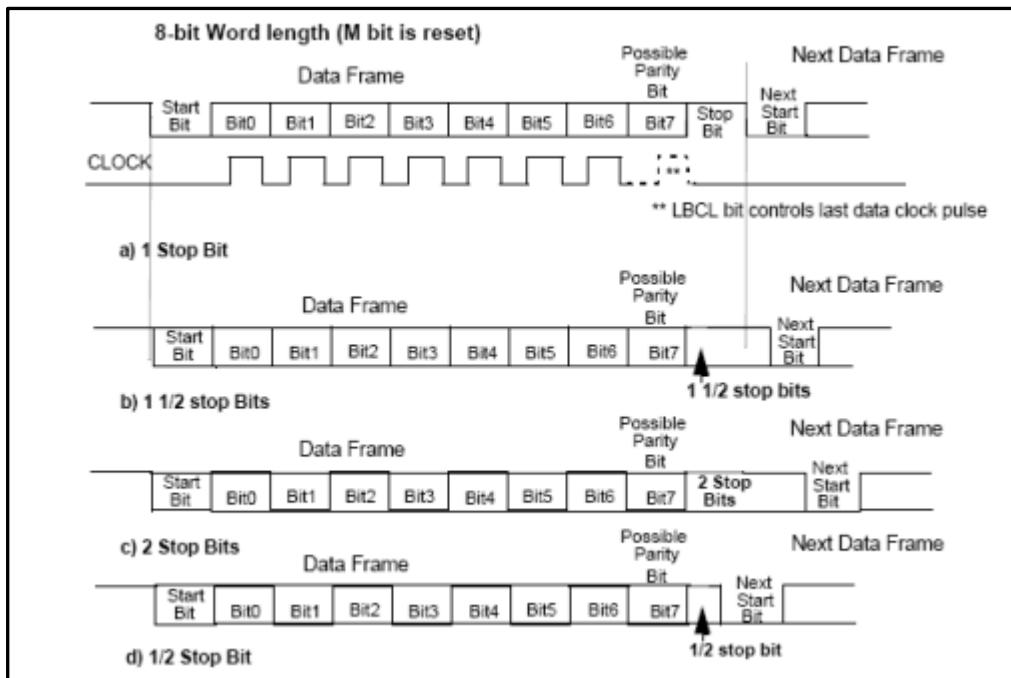


Figure 22- 15 配置停止位

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。

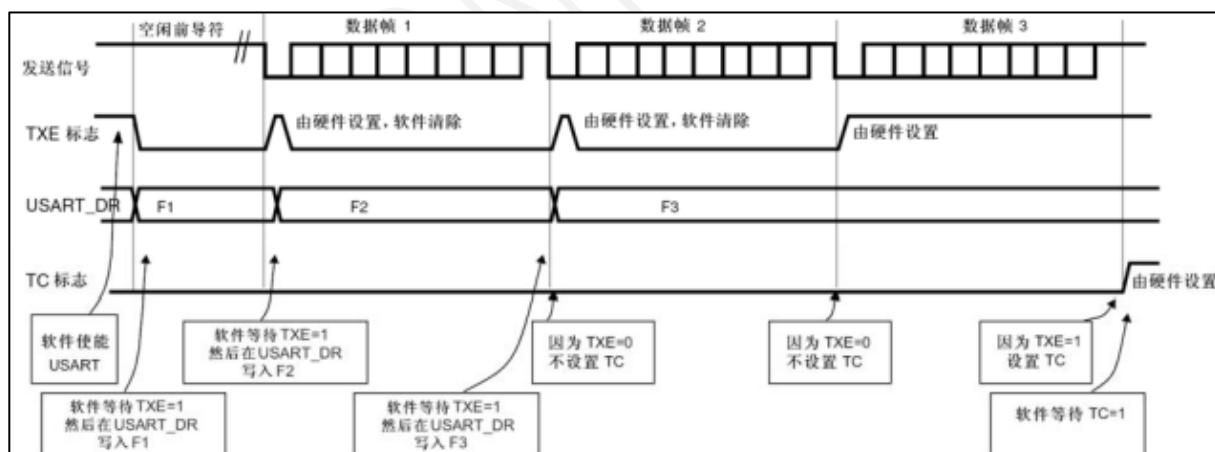


Figure 22- 16 配置停止位

- 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。

### 22.4-3 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。
- 6) 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。有两种可能性：

- 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。

#### 22.4-4 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

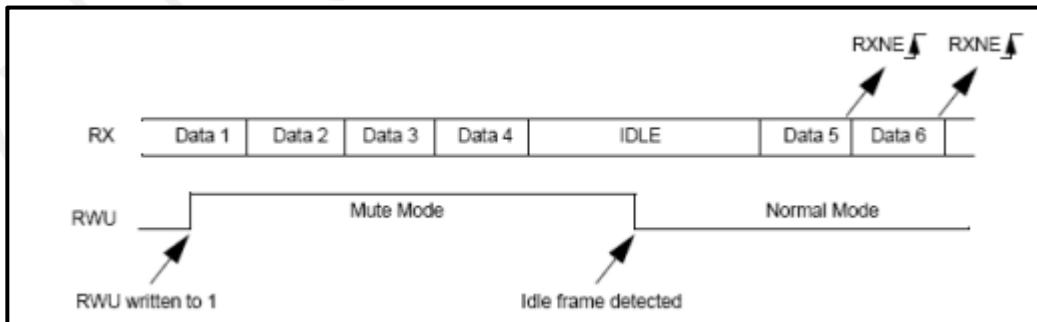


Figure 22- 17 利用空闲总线检测的静默模式

- 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。

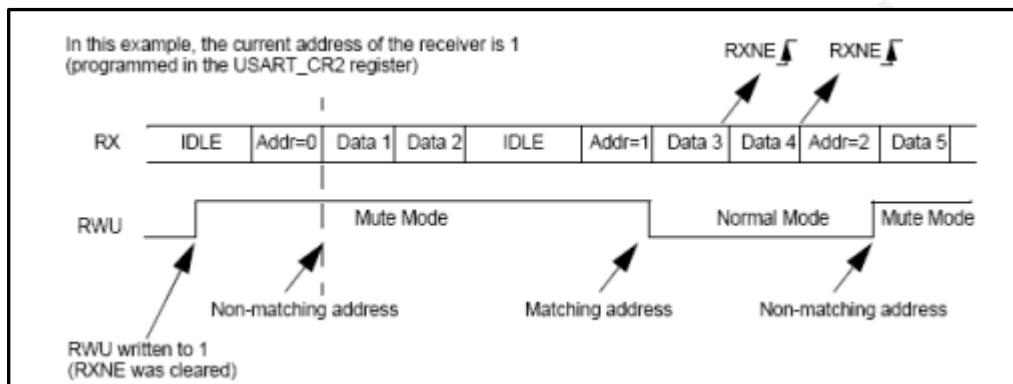


Figure 22- 18 利用地址标记检测的静默模式

#### 22.4-5 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 22-1 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 22- 1 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。

## 22.4-6LIN（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

### 22.4-6.1LIN 发送

22.4-2 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。

### 22.4-6.2LIN 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)。

Fig 22-19 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。

Fig 22-20 给了一个断开帧的例子。

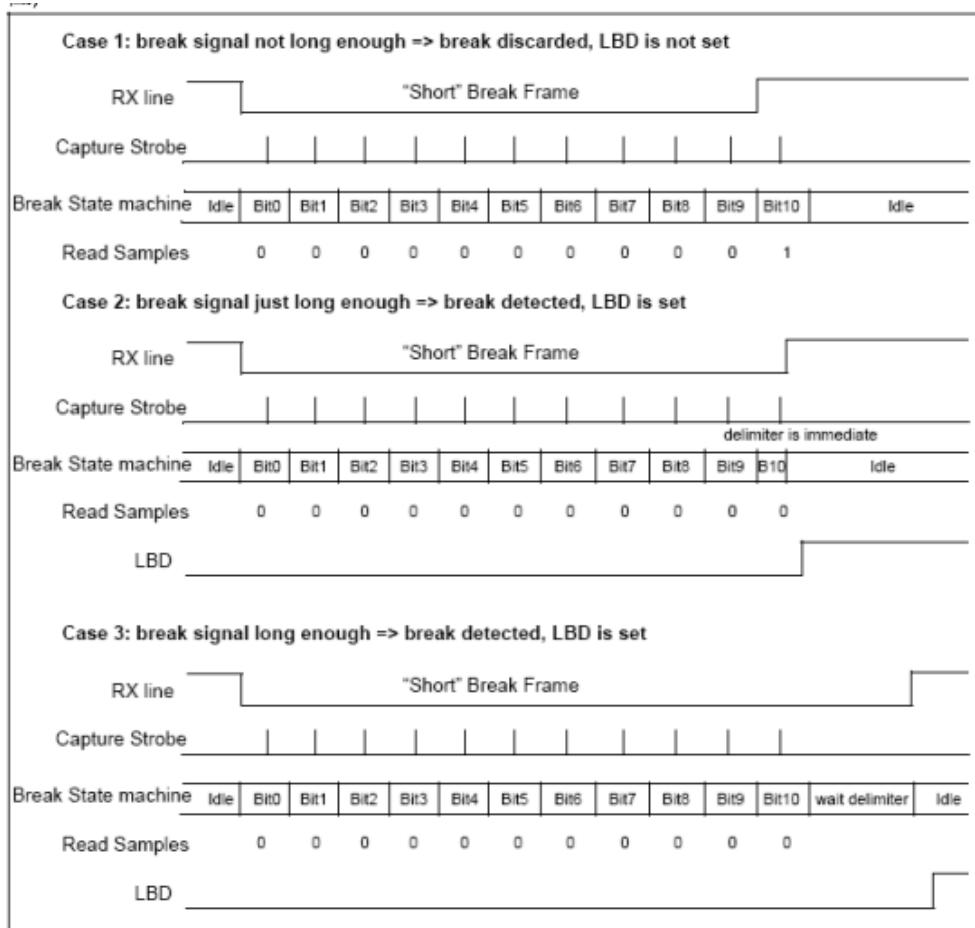
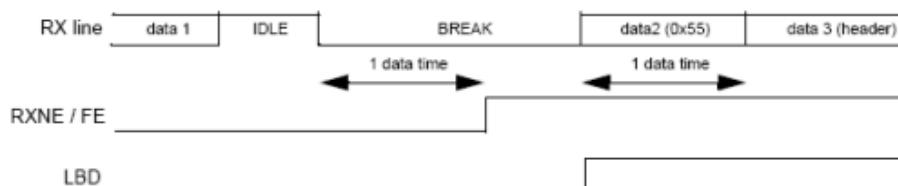


Figure 22- 19 LIN 模式下的断开检测(11 位断开长度 – 设置了 LBDL 位)

In these examples, we suppose that LBDL=1 (11-bit break length), M=0 (8-bit data)

**Case 1: break occurring after an Idle**



**Case 1: break occurring while a data is being received**

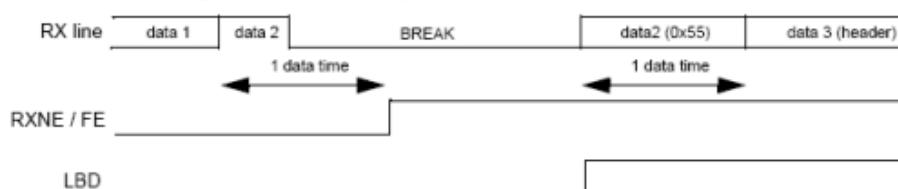


Figure 22- 20 LIN 模式下的断开检测与帧错误的检测

## 22.4-7USART 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)。

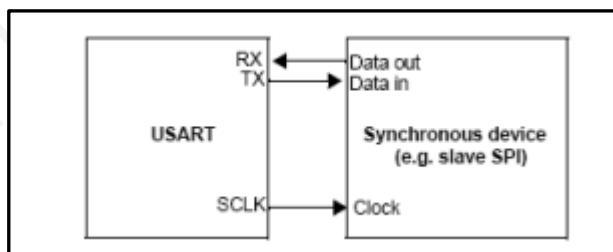
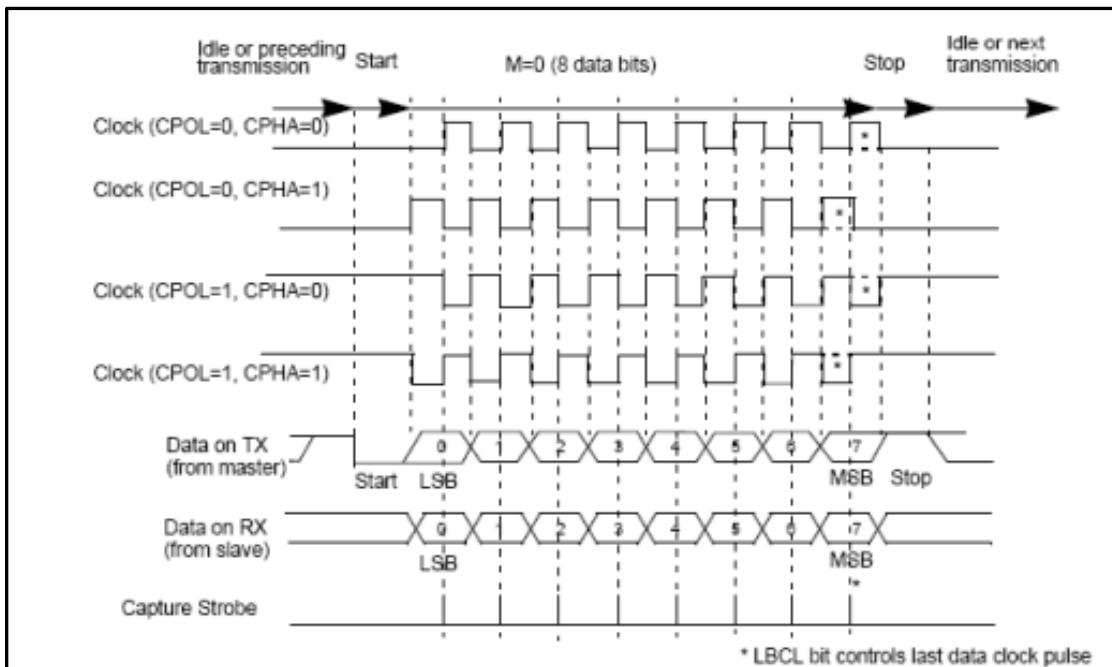
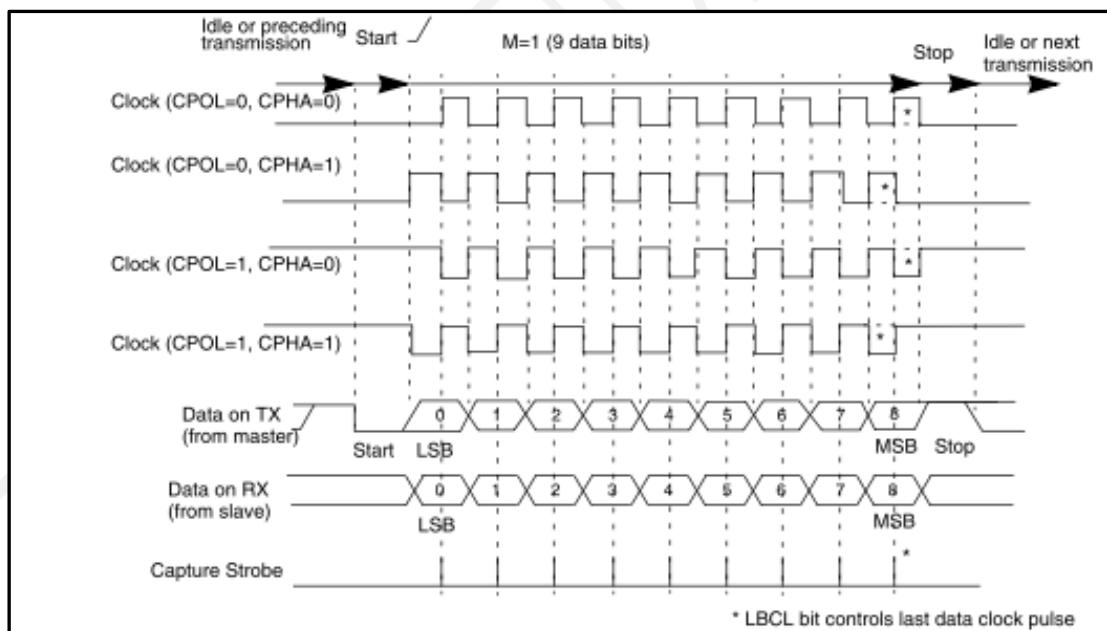


Figure 22- 21 USART 同步传输的例子



**Figure 22- 22 USART 数据时钟时序示例(M=0)**



**Figure 22- 23 USART 数据时钟时序示例(M=1)**

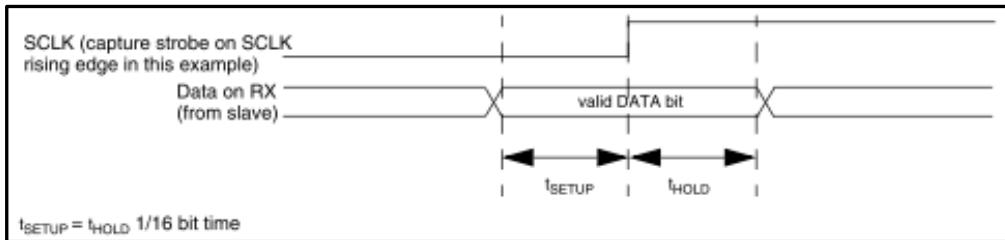


Figure 22- 24 RX 数据采样/保持时间

#### 22.4-8 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。

## 22.4-9 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。

## 22.4-9.1IrDA 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。

(2) 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）。

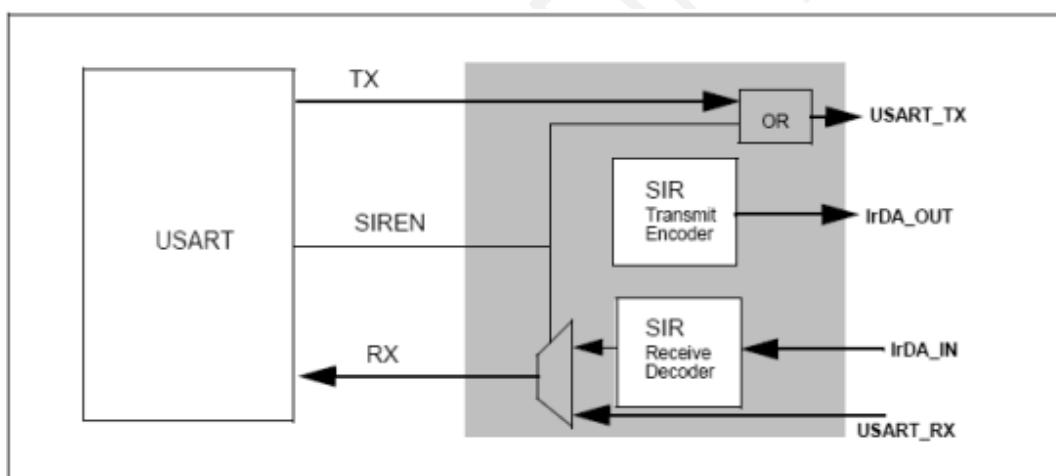


Figure 22- 25 IrDA 框图

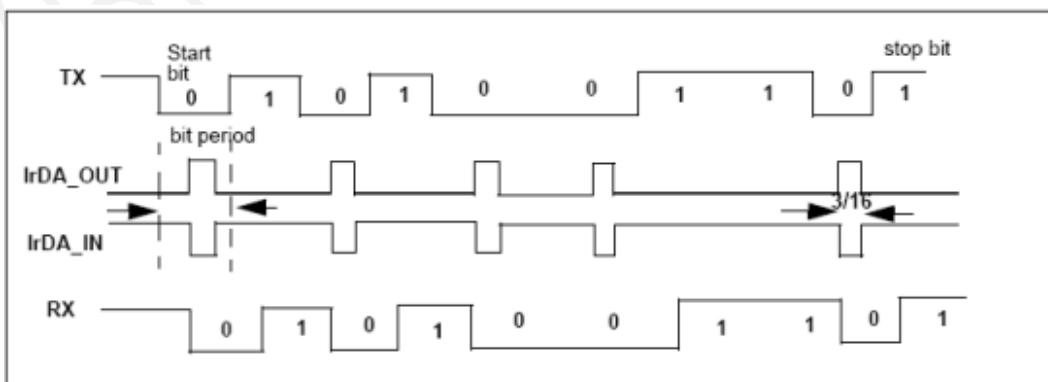
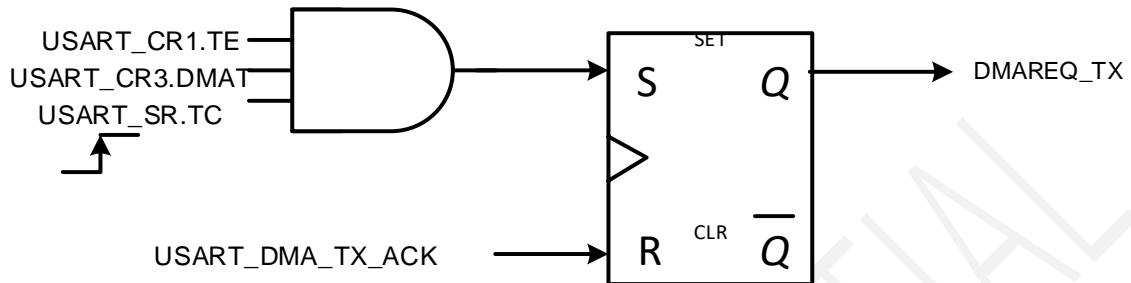


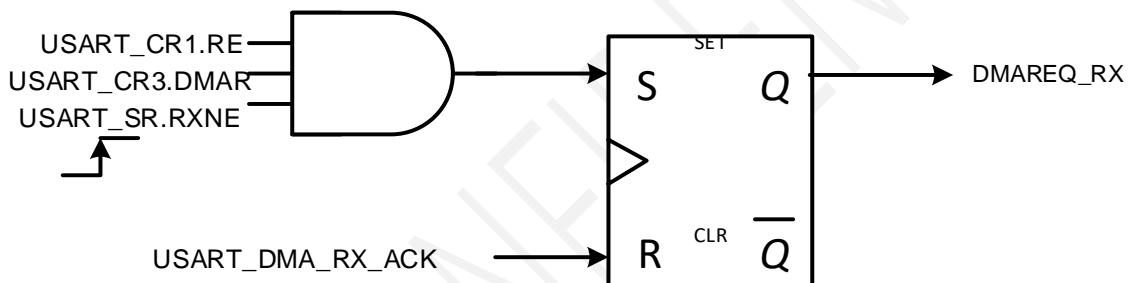
Figure 22- 26 IrDA 数据调制

#### 22.4-10 使用 DMA 通信

- DMA 传送请求条件



- DMA 接收请求条件



#### 22.4-11 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。

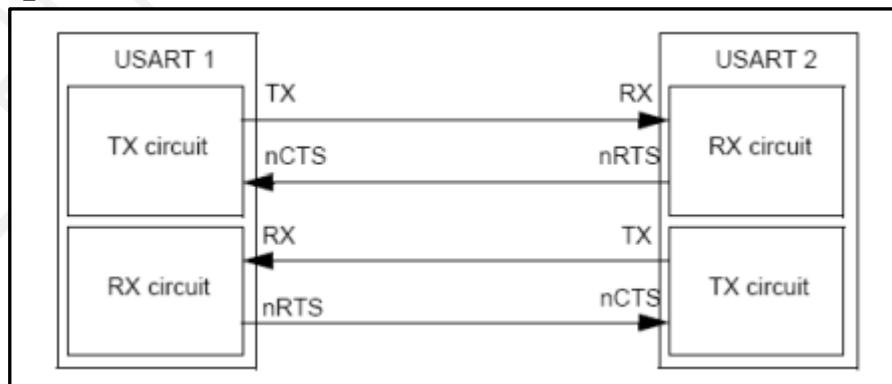


Figure 22- 27 两个 USART 间的硬件流控制

### 22.4-11.1 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。

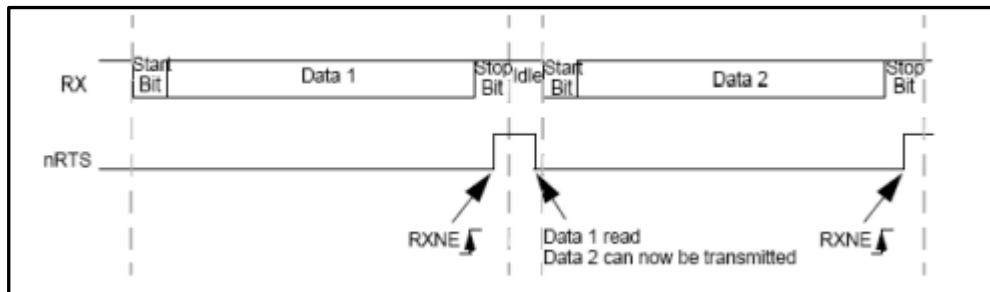


Figure 22- 28 RTS 流控制

### 22.4-11.2 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。

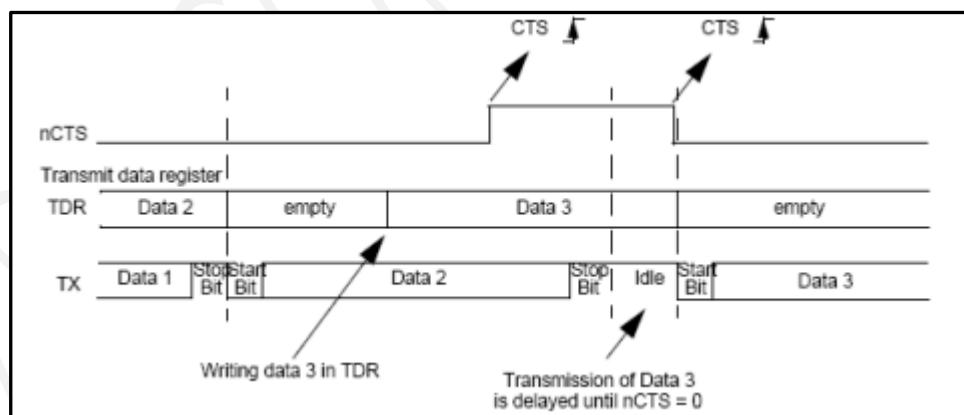


Figure 22- 29 CTS 流控制

### 22.4-12 中断请求

中断请求设定如下表：

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

Table 22- 2 USART 中断请求

USART 的各种中断事件被连接到同一个中断向量(见 图 22-30)，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。

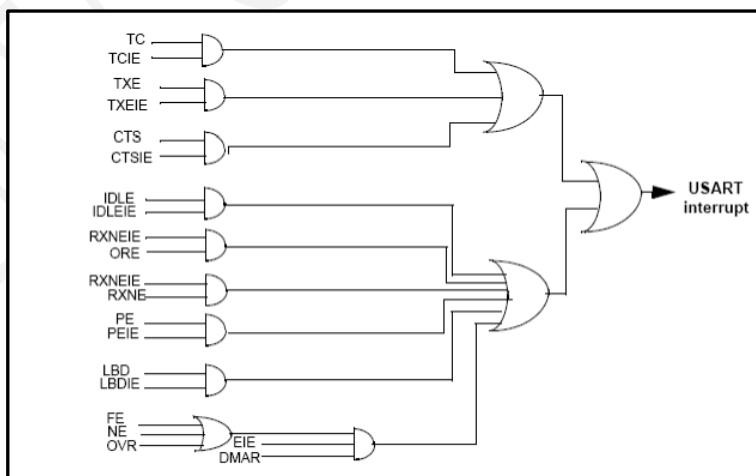


Figure 22- 30 USART 中断映像图

## 22.5 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	<b>PCLK = 14 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	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	<b>PCLK = 20 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	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	<b>PCLK = 24 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	625	2400.00	0.00%	313	2396.17	-0.16%
4800	313	4792.33	-0.16%	156	4807.69	0.16%
9600	156	9615.38	0.16%	78	9615.38	0.16%
19200	78	19230.77	0.16%	39	19230.77	0.16%
38400	39	38461.54	0.16%	20	37500.00	-2.34%
57600	26	57692.31	0.16%	13	57692.31	0.16%
76800	20	75000.00	-2.34%	10	75000.00	-2.34%
115200	13	115384.62	0.16%	7	107142.86	-6.99%

波特率	PCLK = 2 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	52	2403.85	0.16%	26	2403.85	0.16%
4800	26	4807.69	0.16%	13	4807.69	0.16%
9600	13	9615.38	0.16%	7	8928.57	-6.99%
19200	7	17857.14	-6.99%	3	20833.33	8.51%
38400	3	41666.67	8.51%	2	31250.00	-18.62%
57600	2	62500.00	8.51%	1	62500.00	8.51%
76800	2	62500.00	-18.62%	1	62500.00	-18.62%
115200	1	125000.00	8.51%	1	62500.00	-45.75%

波特率	PCLK = 8 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	208	2403.85	0.16%	104	2403.85	0.16%
4800	104	4807.69	0.16%	52	4807.69	0.16%
9600	52	9615.38	0.16%	26	9615.38	0.16%
19200	26	19230.77	0.16%	13	19230.77	0.16%
38400	13	38461.54	0.16%	7	35714.29	-6.99%
57600	9	55555.56	-3.55%	4	62500.00	8.51%
76800	7	71428.57	-6.99%	3	83333.33	8.51%
115200	4	125000.00	8.51%	2	125000.00	8.51%

波特率	PCLK = 11.0592 MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	288	2400.00	0.00%	144	2400.00	0.00%
4800	144	4800.00	0.00%	72	4800.00	0.00%
9600	72	9600.00	0.00%	36	9600.00	0.00%
19200	36	19200.00	0.00%	18	19200.00	0.00%
38400	18	38400.00	0.00%	9	38400.00	0.00%
57600	12	57600.00	0.00%	6	57600.00	0.00%
76800	9	76800.00	0.00%	5	69120.00	-10.00%
115200	6	115200.00	0.00%	3	115200.00	0.00%

波特率	<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	<b>PCLK = 22.12 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	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%

## 22.6UART/USART 寄存器列表

UART0 基址: 0x4000 0000

UART1 基址: 0x4000 0400

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 22- 3 USART 寄存器列表

USART2 基址: 0x4000 0200

USART3 基址: 0x4000 0600

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	USARTx_LP_CR	低功耗控制寄存器	0x0000 0000
0x20	USARTx_LP_BRR	低功耗波特比率寄存器寄存器	0x0000 0000

Table 22- 4 USART 寄存器列表

## 22.7UART0,1 寄存器说明

### 22.7-1UART 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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						DBAUD	FEEN	SM0:SM1	SM2	REN	TB8	RB8	TIEN	RIEN	--
--						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="margin-left: 20px;"> <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																									
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: (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	0	R/W																									

位	标记	功能描述	复位值	读写
		RI=0。 (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SUBF，置位RI，RB8在这种模式下为校验位。		
4	REN	接收使能 Mode 0: 0: 发送 1: 接收  其他： 0: 发送 1: 接收/发送	0	R/W
3	TB8	发送TB8位	0	R/W
2	RB8	接收RB8位	0	R/W
1	TIEN	发送完成中断使能 0: Disable 1: Enable	0	R/W
0	RIEN	接收完成中断使能 0: Disable 1: Enable	0	R/W

### 22.7-2UART 数据寄存器(UARTx\_SBUF)

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								SBUF[7:0]							
--								R/W							

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

### 22.7-3UART 地址寄存器(UARTx\_SADDR)

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
保留								SADDR[7:0]							
--								R/W							

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

### 22.7-4UART 地址掩码寄存器(UARTx\_SADEN)

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
保留								SADEN[7:0]							
--								R/W							

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

### 22.7-5UART 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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								FE TI RI							
--								RO							

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

## 22.7-6UART 标志位清除寄存器(UARTx\_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															
--															

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

## 22.7-7UART 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	16
保留															SELF_BRG
--															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

## 22.7-8IrDA 控制寄存器(UARTx\_IRDACR)

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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IRLPMOD	IrRXD	IrTXD	IrDA	PSC[7:0]										
--					R/W										

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	--
11	IRLPMODE	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[7:0]	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

## 22.8 USART2,3 寄存器说明

### 22.8-1 状态寄存器(USARTx\_SR)

地址偏移: 0x00 复位值: 0x0000 00C0

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				CTS	LBD	TXE	TC	RXNE	IDLE	ORE	保留	FE	PE		
--				RC	RO		RC	RO		--		RO			

位	标记	功能描述	复位值	读写
31:10	保留	--	0x0	--
9	CTS	CTS: CTS 标志 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 1, 产生中断 0: nCTS 状态线上没有变化 1: nCTS 状态线上发生变化	0x0	RC
8	LBD	LBD: LIN break 检测标志 (状态标志) 0: 没有检测到 LIN break 1: 检测到 LIN break 注意: 若 LBDIE=1, 当 LBD 为 1 时要产生中断	0x0	RC
7	TXE	TXE:发送数据寄存器空 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。 0: 数据还没有被转移到移位寄存器 1: 数据已经被转移到移位寄存器 注意: 单缓冲器传输中使用该位	0x1	RO
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

位	标记	功能描述	复位值	读写
5	RXNE	<p><b>RXNE:</b>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。</p> <p>0: 数据没有收到 1: 收到数据，可以读出</p>	0x0	RC
4	IDLE	<p><b>IDLE:</b>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）</p> <p>0: 没有检测到空闲总线 1: 检测到空闲总线</p> <p>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)</p>	0x0	RO
3	ORE	<p><b>ORE:</b>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）</p> <p>0: 没有过载错误 1: 检测到过载错误</p> <p>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，在多缓冲区通信模式下，ORE 标志置位会产生中断的。</p>	0x0	RO
2	保留	--	0x0	--
1	FE	<p><b>FE:</b>帧错误 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	RO
0	PE	<p><b>PE:</b>校验错误 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

## 22.8-2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
保留															
--															

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

位	标记	功能描述	复位值	读写
31:9	保留	--	0x0	--
8:0	DR[8:0]	<p><b>DR[8:0]:数据值</b>            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

### 22.8-3 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
保留															
--															

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

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15:4	DIV_MANTISSA [11:0]	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	R/W
3:0	DIV_FRACITON [3:0]	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	R/W

注意：如果 TE 或 RE 被分别禁止，波特计数器停止计数

### 22.8-4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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SYNSPI	UE	M	WAKE	保留	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK	
--		R/W		--											R/W

位	标记	功能描述	复位值	读写
31:15	保留	--	0x0	--
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	R/W
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	R/W
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: 保留(禁止写 1)  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	R/W
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	R/W
10	保留	-- (禁止写 1)	0x0	--
9	保留	-- (禁止写 1)	0x0	--
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	R/W

位	标记	功能描述	复位值	读写
7	TXEIE	<p><b>TXEIE:</b> 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</p>	0x1	R/W
6	TCIE	<p><b>TCIE:</b> 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</p>	0x1	R/W
5	RXNEIE	<p><b>RXNEIE:</b> 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</p>	0x0	R/W
4	IDLEIE	<p><b>IDLEIE:</b> 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</p>	0x0	R/W
3	TE	<p><b>TE:</b> 发送使能 该位使能发送器。软件对该位置位或者清零 0: 发送被禁止 1: 发送被使能 注意: 在数据传输过程中, 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>	0x0	R/W
2	RE	<p><b>RE:</b> 接收使能 软件对该位置位或者清零 0: 接收被禁止 1: 接收被使能, 开始搜寻 RX 引脚上的起始位。</p>	0x0	R/W
1	RWU	<p><b>RWU:</b> 接收唤醒 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。 0: 接收器处于正常工作模式 1: 接收器处于静默模式 注意: 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	R/W
0	SBK	<p><b>SBK:</b> 发送断开帧 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。 0: 没有发送断开字符 1: 将要发送断开字符</p>	0x0	R/W

## 22.8-5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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LINEN	STOP[1:0]	CLKEN	CPOL	CPHA	LBCL	保留	LBDIE	LBDL	保留	保留	ADD[3:0]			
--		R/W					--	R/W	--	R/W					

位	标记	功能描述	复位值	读写
31:15	保留	--	0x0	--
14	LINEN	<p><b>LINEN:</b> 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	R/W
13:12	STOP	<p><b>STOP:</b> 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	R/W
11	CLKEN	<p><b>CLKEN:</b> 时钟使能 该位用来使能 SCLK 引脚 0: SCLK 引脚被禁止 1: SCLK 引脚被使能</p>	0x0	R/W
10	CPOL	<p><b>CPOL:</b> 时钟极性 用户可以用该位来选择同步模式下 SLCK 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</p>	0x1	R/W
9	CPHA	<p><b>CPHA:</b> 时钟相位 用户可以用该位来选择同步模式下 SLCK 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 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</p>	0x1	R/W

位	标记	功能描述	复位值	读写
8	LBCL	<p><b>LBCL:</b> 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(<b>MSB</b>)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) .</p>	0x0	R/W
7	保留	--	0	--
6	LBDIE	<p><b>LBDIE:</b> 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</p>	0x0	R/W
5	LBDL	<p><b>LBDL:</b> 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</p>	0x0	R/W
4	保留	--	0	--
3:0	ADD	<p><b>ADD[3:0]:</b> 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</p>	0x0	R/W

Note : 在发送被使能后不能写这三个位(CPOL、CPHA、LBCL)

Note\_22.1.1: 禁用 CPOL=1 and CPHA=0

## 22.8-6控制寄存器 3(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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
--					RW										

位	标记	功能描述	复位值	读写
31:11	保留	--	--	--
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	R/W
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	R/W
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	R/W
7	DMAT	DMAT: DMA 使能发送 由软件对该位清零或者置位 1: 发送时的 DMA 模式使能 0: 发送时的 DMA 模式被禁止	0x0	R/W
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	R/W
5	保留	禁用此位(程序不可写入)	0x0	--

位	标记	功能描述	复位值	读写
4	FEREN	FEREN 使能 0: Frame 错误侦测禁用 1: Frame 错误侦测使能	0x0	R/W
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	R/W
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	R/W
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	R/W
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 的 FE=1, 或者 ORE=1, 或者 NE=1), 产生中断。 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	0x0	R/W

## 22.8-7保护时间和预分频寄存器(USART\_GTPR)

地址偏移: 0x18 复位值: 0x0000 0000

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
GT[7:0]								PSC[7:0]							
RW															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15:8	GT[7:0]	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。 当保护时间过去后，发送完成标志才被置起。	0x0	R/W
7:0	PSC[7:0]	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 ..... - 在红外的通常模式下： PSC 只能设置为 00000001	0x0	R/W

## 22.8-8低功耗波特比率寄存器(USARTx\_LP\_CR)

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															
--															

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

位	标记	功能描述	复位值	读写
31:2	保留	--	--	--
1	LP_WAKUP_FLAG	在 sleep mode 下的功能: 0 : 写 0 清除 1 : 唤醒标志被设置.	0x0	R/W
0	LP_WAKUP_EN	在 sleep mode 下的功能: 0 : 关闭低功耗唤醒 1 : 使能低功耗唤醒	0x0	R/W

## 22.8-9低功耗波特比率寄存器(USARTx\_LP\_BRR)

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															
--															

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

位	标记	功能描述	复位值	读写
31:16	保留	--	--	--
15:4	LP_DIV_MANTISSA [11:0]	LP_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	R/W
3:0	LP_DIV_FRACITON [3:0]	LP_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	R/W

注意: 如果 TE 或 RE 被分别禁止, 波特计数器停止计数

## 【23】低功耗通用异步收发器(LPUART)

### 23.1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。

### 23.2结构框图

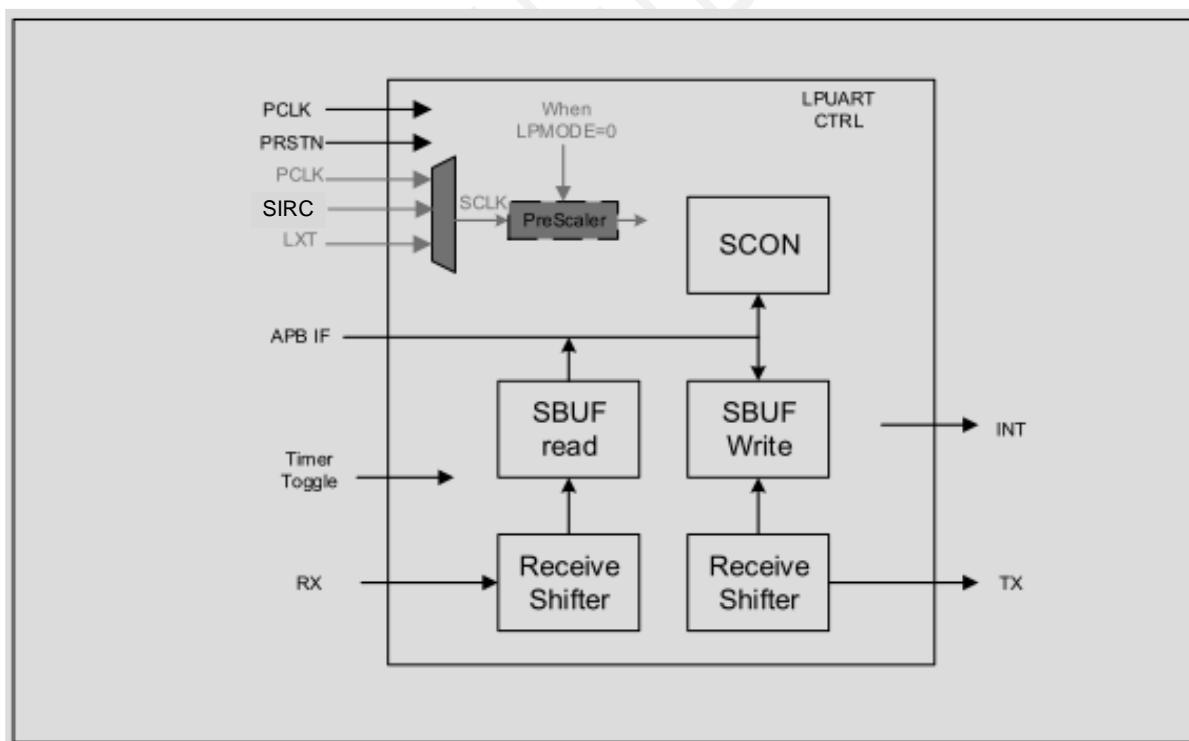


Figure 23- 1 LPUART 结构框图

### 23.3 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。

#### 23.3-1 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.SMO 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。

##### 23.3-1.1 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。

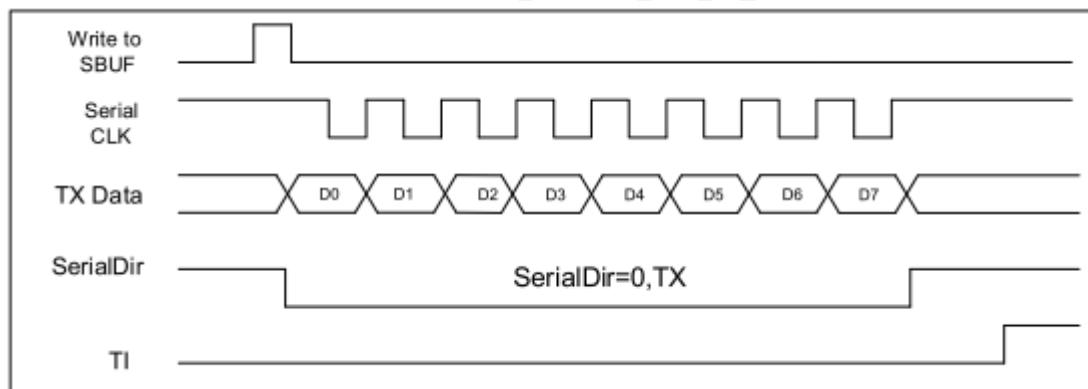


Figure 23- 2 Mode 0 发送数据

### 23.3-1.2 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。

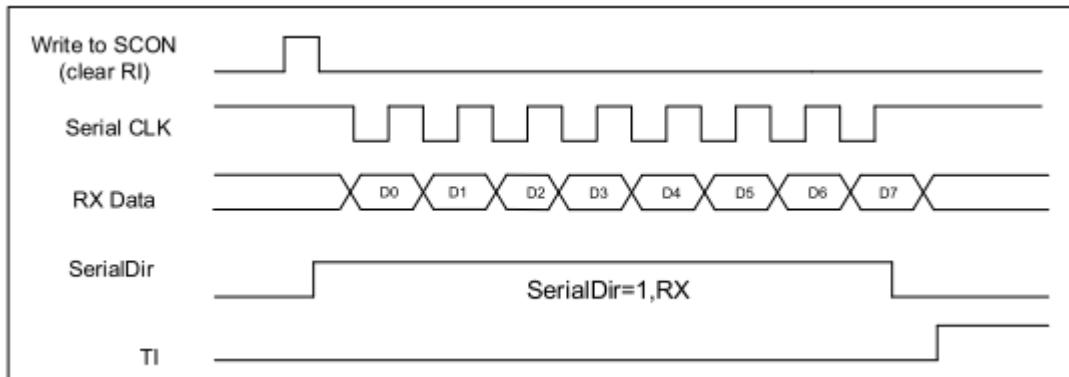


Figure 23- 3 Mode 0 接收数据

### 23.3-2 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。

### 23.3-2.1 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)。

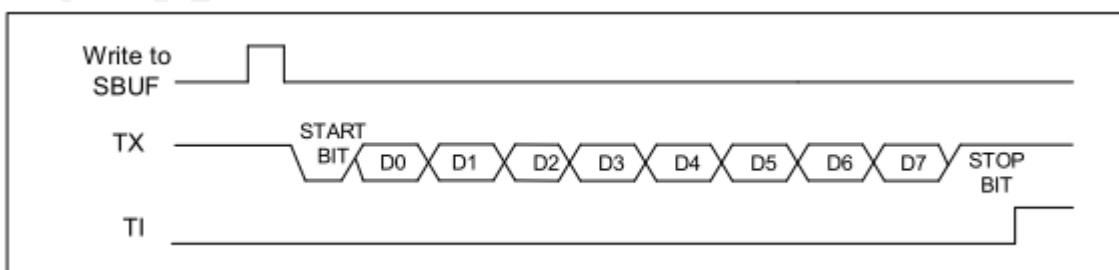


Figure 23- 4 Mode 1 发送数据

### 23.3-2.2 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。

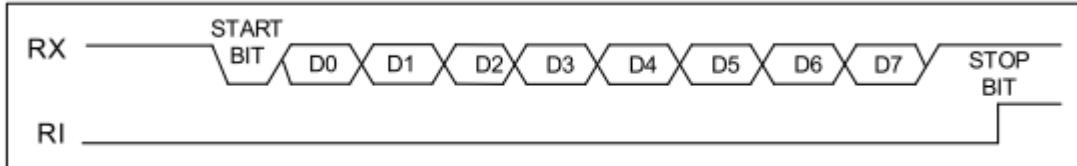


Figure 23- 5 Mode 1 接收数据

### 23.3-3 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。

### 23.3-3.1 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)。

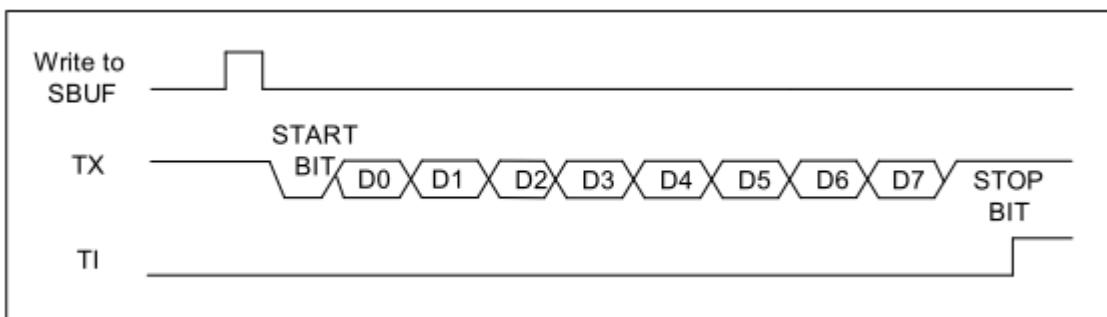


Figure 23- 6 Mode 2 发送数据

### 23.3-3.2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。

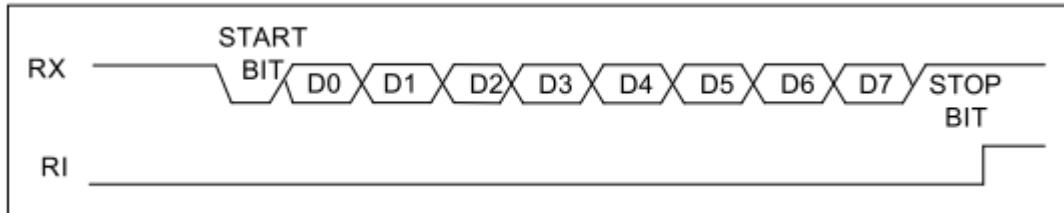


Figure 23- 7 Mode 2 接收数据

### 23.3-4Mode 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。

## 23.4 波特率编程

### 23.4-1 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。

### 23.4-2Mode 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。

### 23.4-3Mode 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。

### 23.5 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。

### 23.6 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 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 LP UART\_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。

### 23.7 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。

### 23.8 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。

**GIVENADDR = SADDR & SADEN**

### 23.9 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：

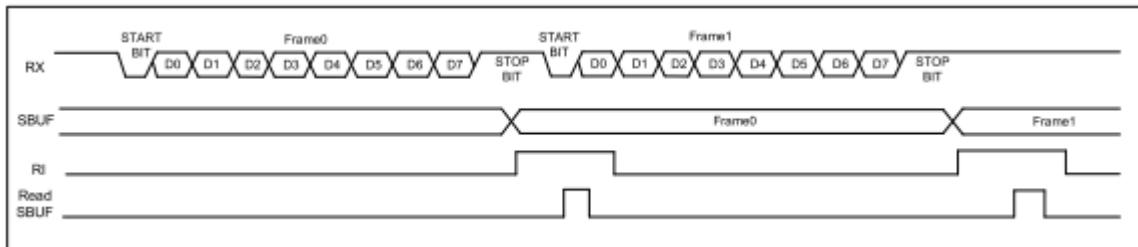
0b01101001 和 0b01101101(given address)

0b11111011 和 0b11111111(broadcast address)。

### 23.10 收发端缓存

#### 23.10-1 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。



#### 23.10-2 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。

### 23.11 寄存器列表

LPUART 基址地址: 0x4000 5000

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 23- 1 LPUART 寄存器列表

### 23.12 寄存器说明

#### 23.12-1 LPUART 数据寄存器(LPUART\_SBUF)

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								SBUF[7:0]							
--								R/W							

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。	0x0	R/W

### 23.12-2LPUART 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														DMATEN	DMAREN	EN
保留														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	RIEN				
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: DIV16; 100: DIV8; 101: DIV4; 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
12:11	SCLKSEL [1:0]	传输时钟SCLK 选择 00 / 01: PCLK 10: LXT 11: SIRC	0x0	R/W

位	标记	功能描述	复位值	读写																									
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" data-bbox="420 797 1167 999"> <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 SUBF, 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: 接收 其他: 0: 发送, 1: 接收/发送	0	R/W																									
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																									

### 23.12-3LPUART 地址寄存器(LPUART\_SADDR)

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
保留								SADDR[7:0]							
--								R/W							

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

### 23.12-4LPUART 地址掩码寄存器(LPUART\_SADEN)

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
保留								SADEN[7:0]							
--								R/W							

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

### 23.12-5LPUART 标志位寄存器(LPUART\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													FE	TI	RI
--													RO		

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

### 23.12-6LPUART 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												FECLR	TICLR	RICLR	
--												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

### 23.12-7LPUART 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															SELF_BRG
--															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	LPUART 波特率选择位: 0: LPUART 的波特率由timer 产生 1: LPUART 的波特率由(DBAUD+1)*FPCLK/(32*(BRG[15:0]+1))生成	0	R/W
15:0	BRG[15:0]	LPUART 自动波特率生成配置位: 波特率=(DBAUD+1)*FPCLK/(32*(BRG[15:0]+1))	0x0	R/W

## 【24】I2C 接口(I2C0/1)

### 24.1 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。

### 24.2 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) I2C0 支持 DMA 发送/接收

### 24.3 I2C 协议描述

通常标准 I2C 传输协议包含四个部分：

- 起始信号或重复起始信号
- 从机地址传输和 R/W 位传输
- 数据传输
- 停止信号

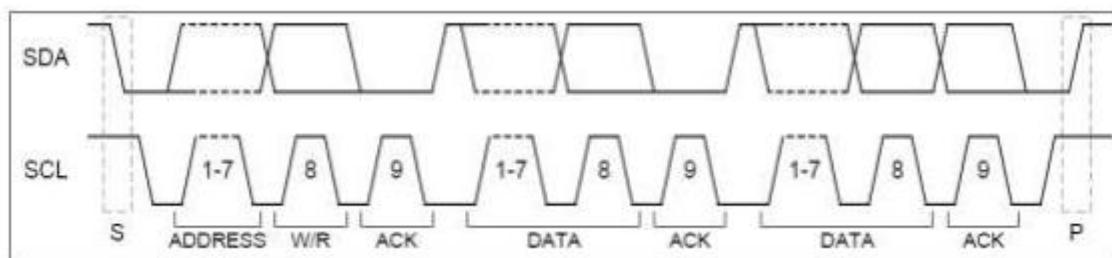


Figure 24- 1 I2C 传输协议

### 24.3-1 I<sup>2</sup>C 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。

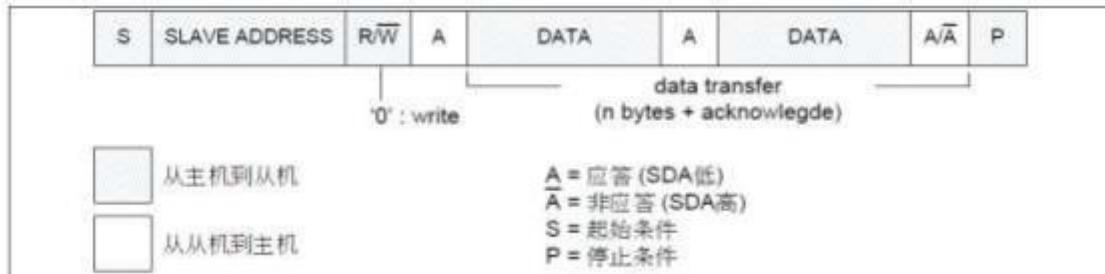


Figure 24- 2 主机向从机传输数据

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。

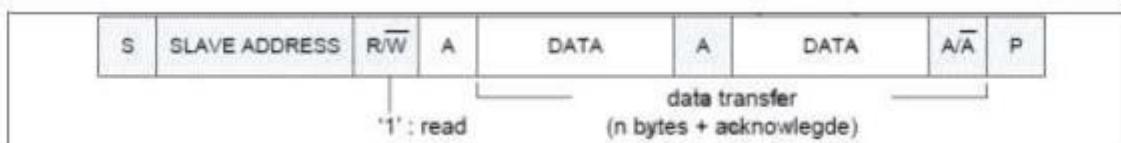


Figure 24- 3 主机向从机传输数据

### 24.3-2 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。

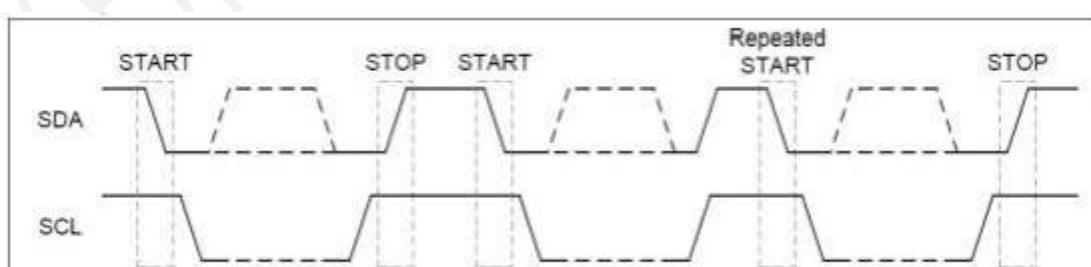


Figure 24- 4 START 和 STOP 条件

### 24.3-3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。

### 24.3-4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。

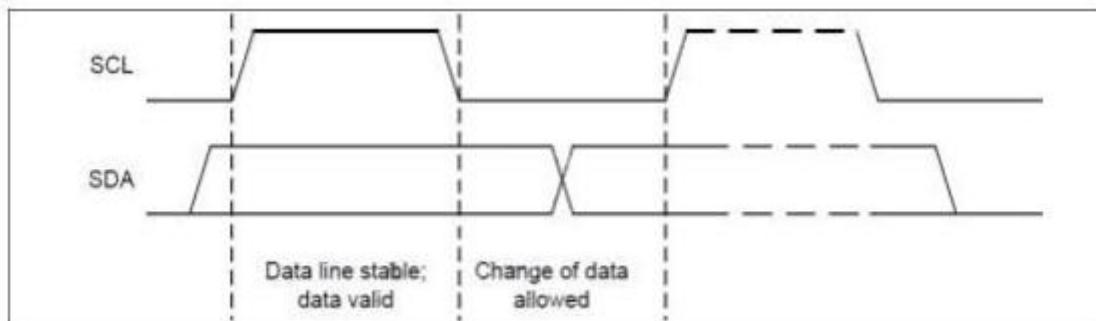


Figure 24- 5 I2C 总线上位传输

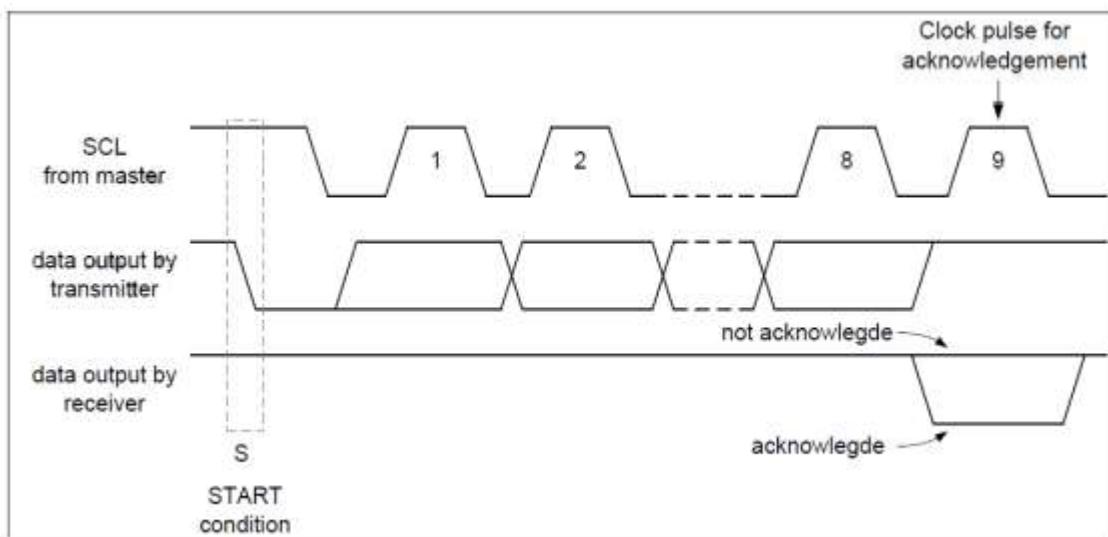


Figure 24- 6 I2C 总线上应答信号

## 24.4 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。

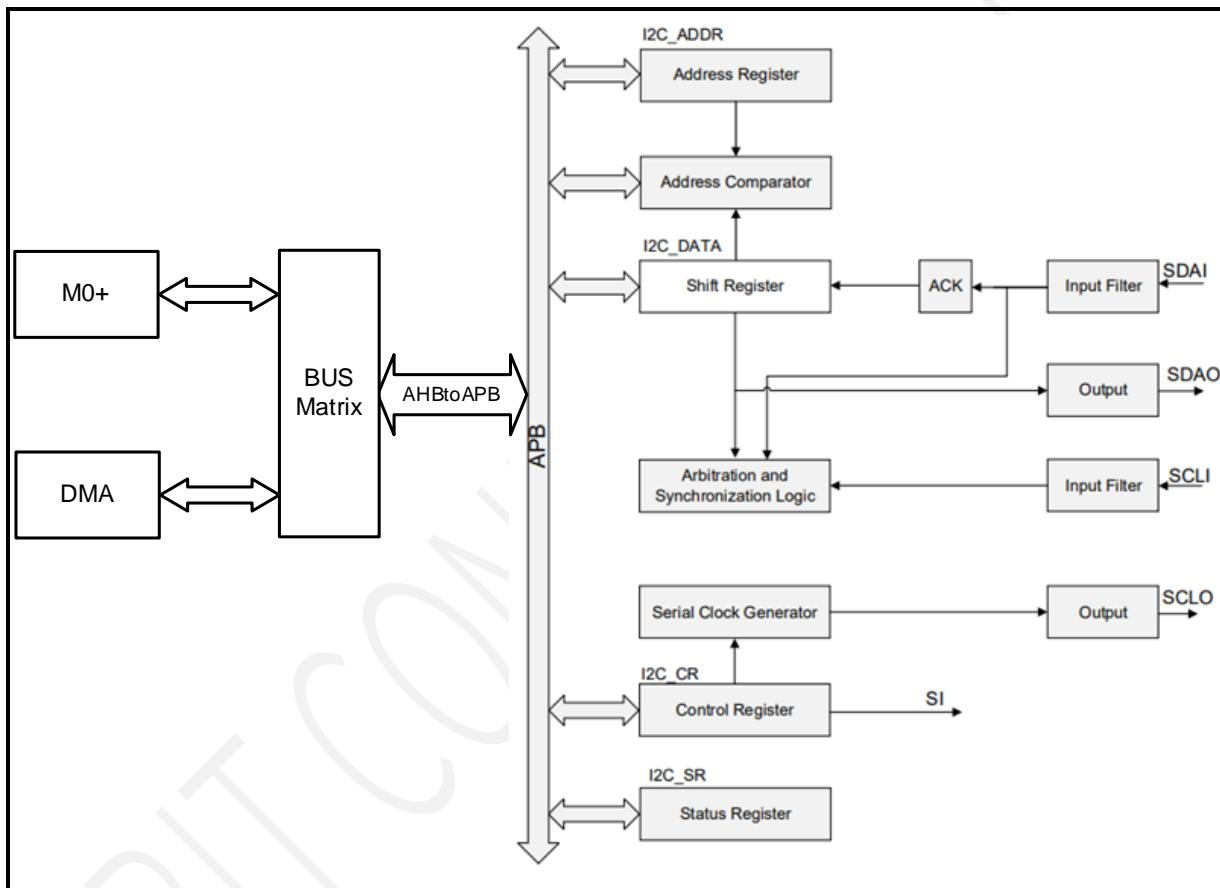


Figure 24- 7 I2C 功能模块图

## 24.5 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

### 24.5-1 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。

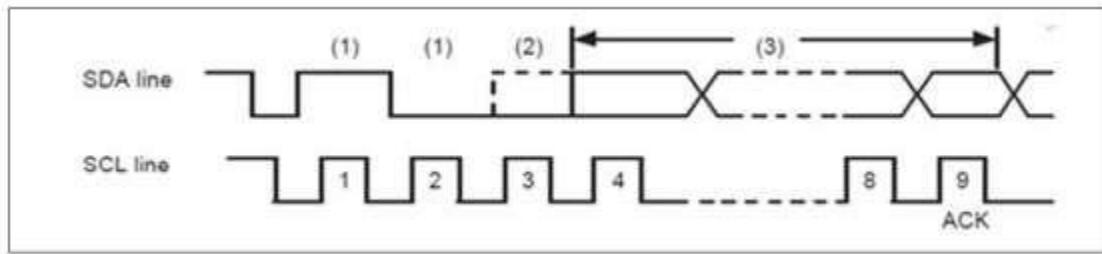


Figure 24- 8 总线上的仲裁

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。

#### 24.5-2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

$$F_{SCL} = F_{PCLK}/8*(N+1)$$

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

#### 24.5-3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。

#### 24.5-4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。

#### 24.5-5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。

## 24.5-6I2C 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				
		I2C_DATA	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
30H	已发送 I2C_DATA 中 的数据	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
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

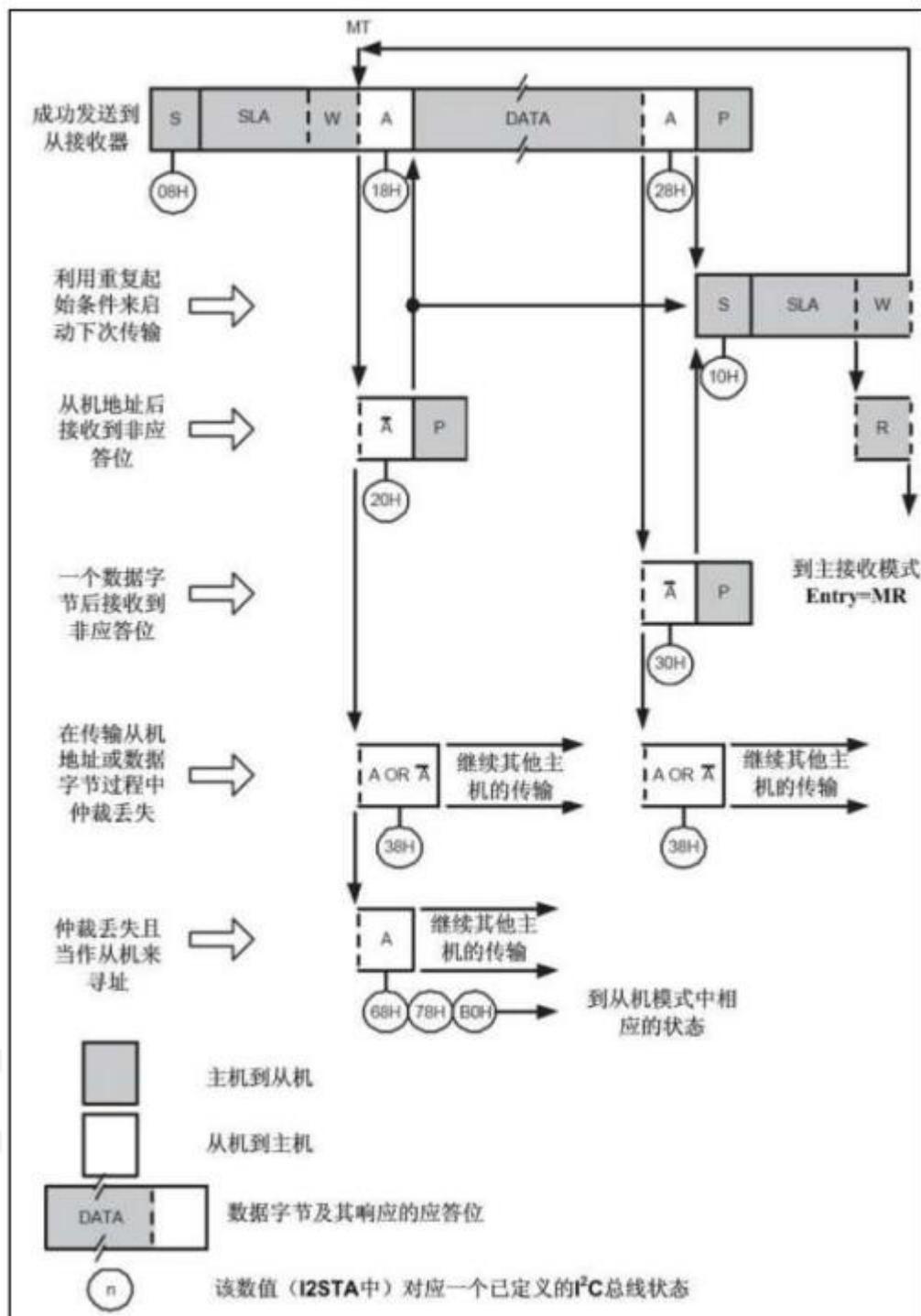


Figure 24- 9 I<sup>2</sup>C 主机发送状态图

## 24.5-7I2C 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 I2C_DATA	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	
50H	已接收数据字 节，ACK 已 返回	读取数据字节	0	0	0	0	将接收数据字节，将返回非ACK	
		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	

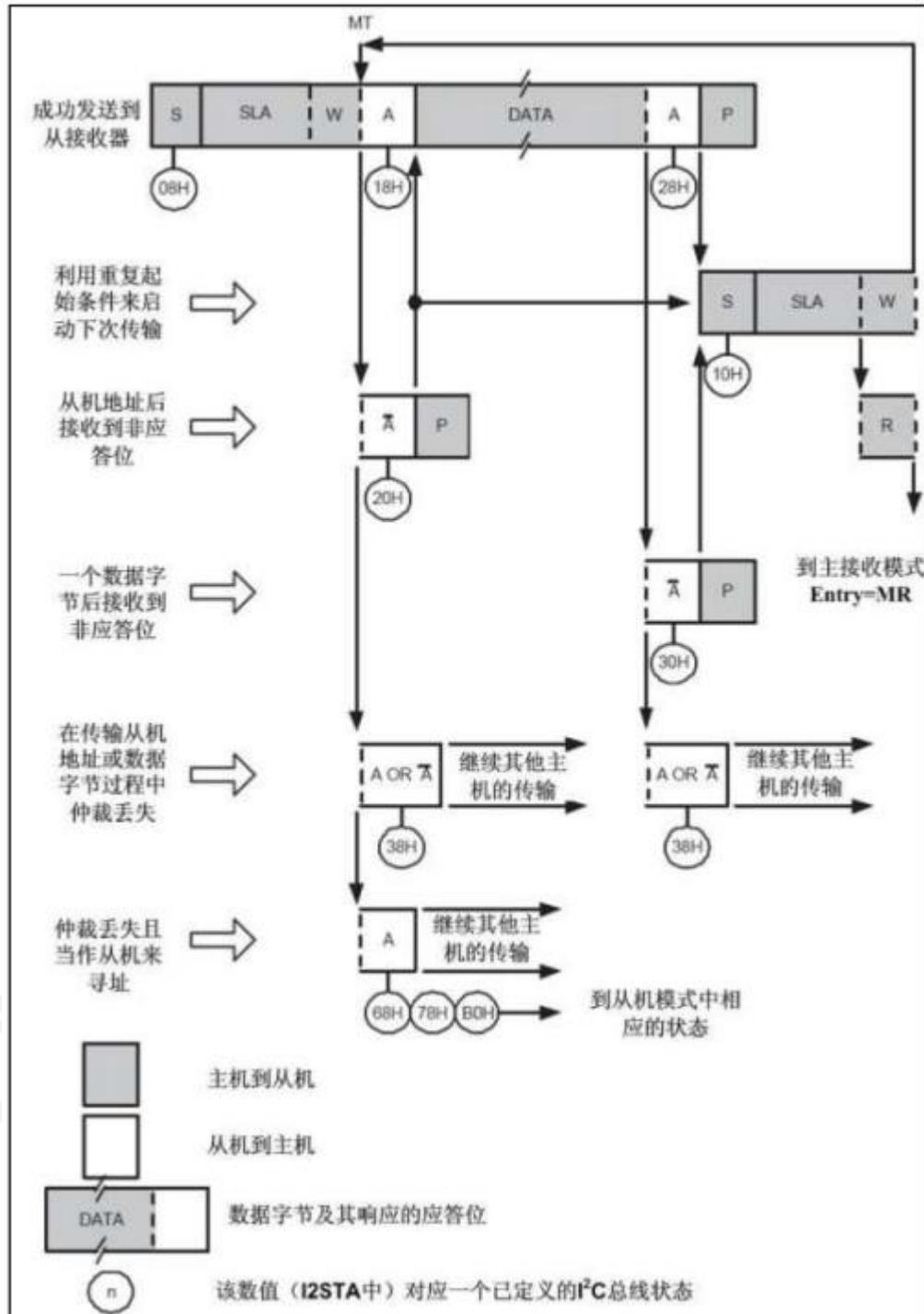


Figure 24- 10 I<sup>2</sup>C 主机接收状态图

## 24.5-8I2C 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	
		读/写 I2C_DATA	写I2C_CR				
			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
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

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		
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；	
A0H	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；	

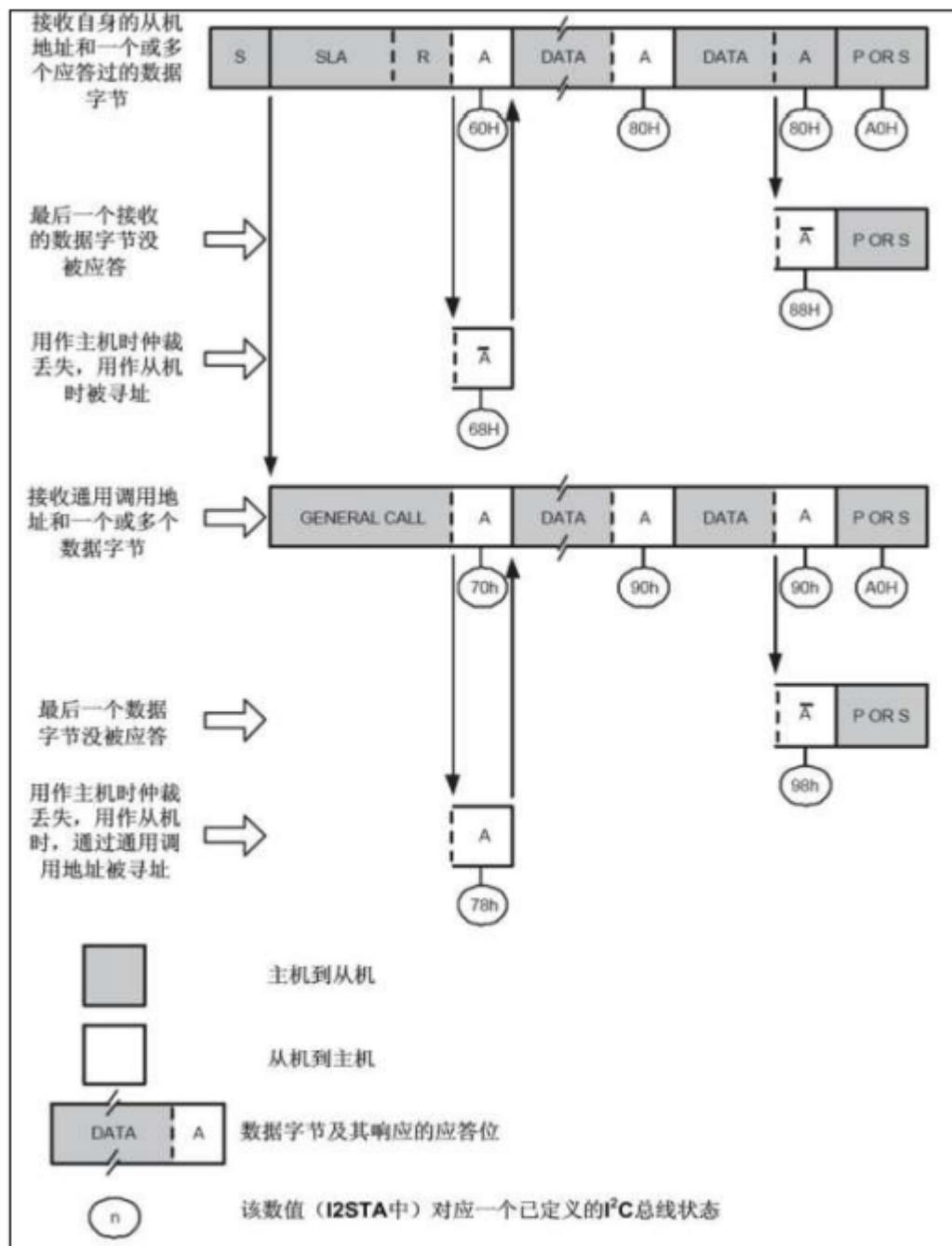


Figure 24- 11 I<sup>2</sup>C 从机接收状态图

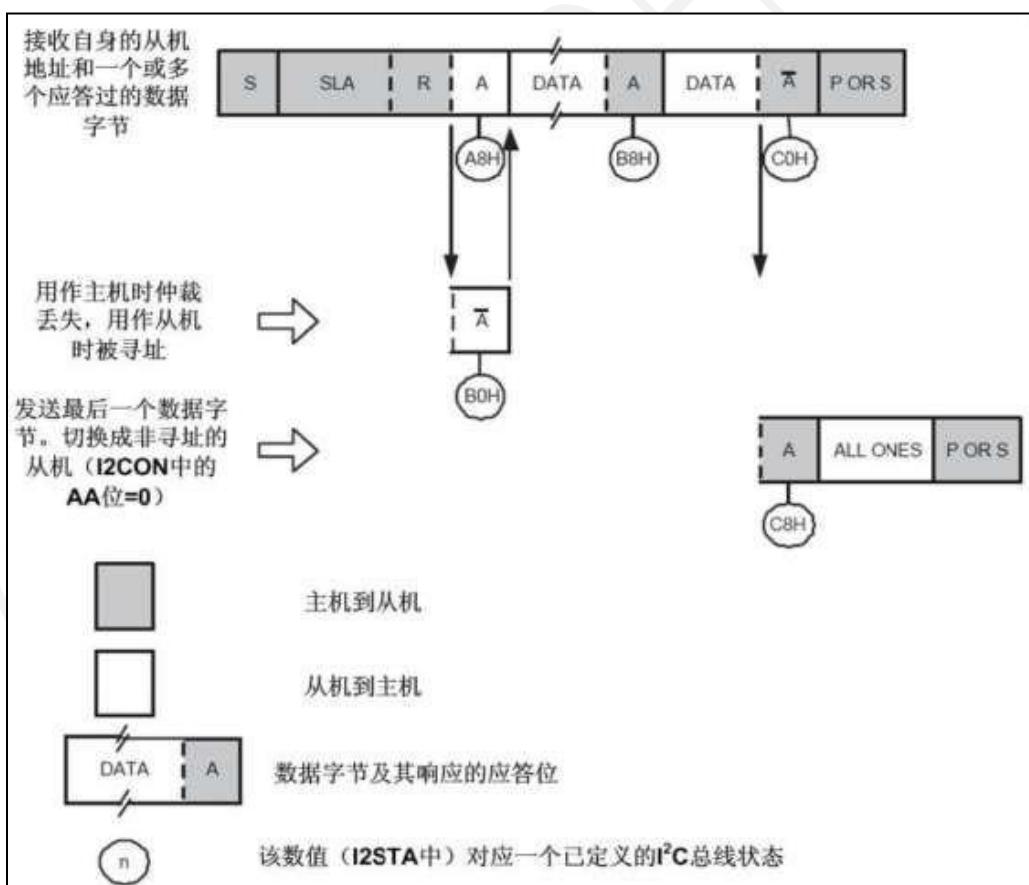
## 24.5-9I2C 从机发送模式

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	写 I2C_CR					
			I2C_DATA	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;	
C0H	已发送数据字节; 已接收非 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;	

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				
			I2C_DATA	STA	STO	SI	AA
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 24- 12 I2C 从机发送状态图**

## 24.5-10I2C 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 I2C_DATA	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。	

## 24.6 I2C 操作模式

### 24.6-1 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。

### 24.6-2 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

## 24.6-3启动主机发送功能

### 24.6-3.1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

### 24.6-3.2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.

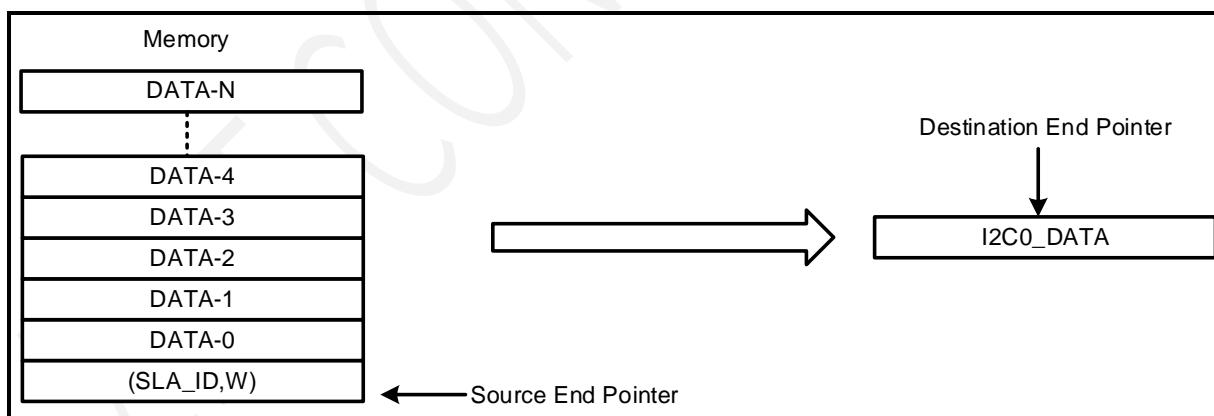


Figure 24- 13 I2C0 Master DMA TX

Note: 如为从机发送, 就不需 SLA\_ID 数据.

## 24.6-4启动主机接收功能

### 24.6-4.1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。

### 24.6-4.2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

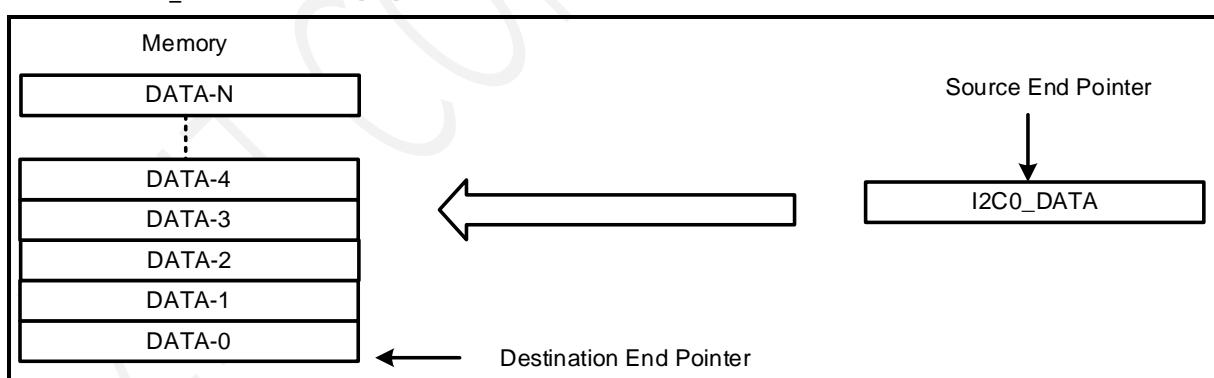


Figure 24- 14 I2C0 Master DMA RX

## 24.6-5 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。

## 24.6-6 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。

## 24.6-7 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。

## 24.6-8 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) 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 中读取数据和发送停止条件。

- 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。

## 24.6-9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。

## 24.6-10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。

## 24.7 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 24- 1 I2CX\* 寄存器列表

Note : I2CX\* : I2C0,I2C1

## 24.8 I<sup>2</sup>C 寄存器说明

### 24.8-1 I<sup>2</sup>C 配置寄存器(I<sup>2</sup>CX\*\_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								ENS	STA	STO	SI	AA	INTE	H1	
--								R/W							

位	标记	功能描述	复位值	读写
31:7	保留	--	0x0	--
6	ENS	I <sup>2</sup> C 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	I <sup>2</sup> C 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	I <sup>2</sup> CX 中断使能 0: 使能 I <sup>2</sup> CX 中断发生 1: 禁止 I <sup>2</sup> CX 中断发生	0	R/W
0	H1M	I <sup>2</sup> C 高速 1Mbps 模式使能。 0: 禁止 1: 使能	0	R/W

#### 24.8-2I2C 数据寄存器(I2CX\*\_DATA)

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								I2CDAT[7:0]							
--								R/W							

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

#### 24.8-3I2C 地址寄存器(I2CX\*\_ADDR)

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								I2CADR[6:0]							
--								R/W							

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

#### 24.8-4I2C 状态寄存器(I2C\_SR)

偏移地址: 0x0C 复位值: 0x0000 00F8

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								I2CSTA[7:0]							
--								RO							

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

### 24.8-5I2C 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															
--															

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

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	--
7	DMATEN	I2C0 DMA TX enable (Only I2C0) 1 : Enable 0 : Disable	0	R/W
6	DMAREN	I2C0 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

### 24.8-6I2C 波特率计数器配置寄存器(I2CX\*\_BAUDCR)

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								TM[7:0]							
--								R/W							

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

## 【25】串行外设接口(SPI/I2S)

### 25.1 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)。

在大容量产品和互连型产品上，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。

### 25.2 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 >= 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

**Note\_25. 1 :** 目前 SPI 从模式的非连续传送仅支持硬件 DMA 更新发送缓冲器，不支持软件更新发送缓冲器

**Note\_25. 2 :** 目前 SPI 从模式非连续接收仅支持字节与字节的空隙需小于 4\*SPI clock 的时间

### 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 256xFs(Fs 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

### 25.3 功能描述

#### 25.3-1 概述

SPI 的方框图见下图。

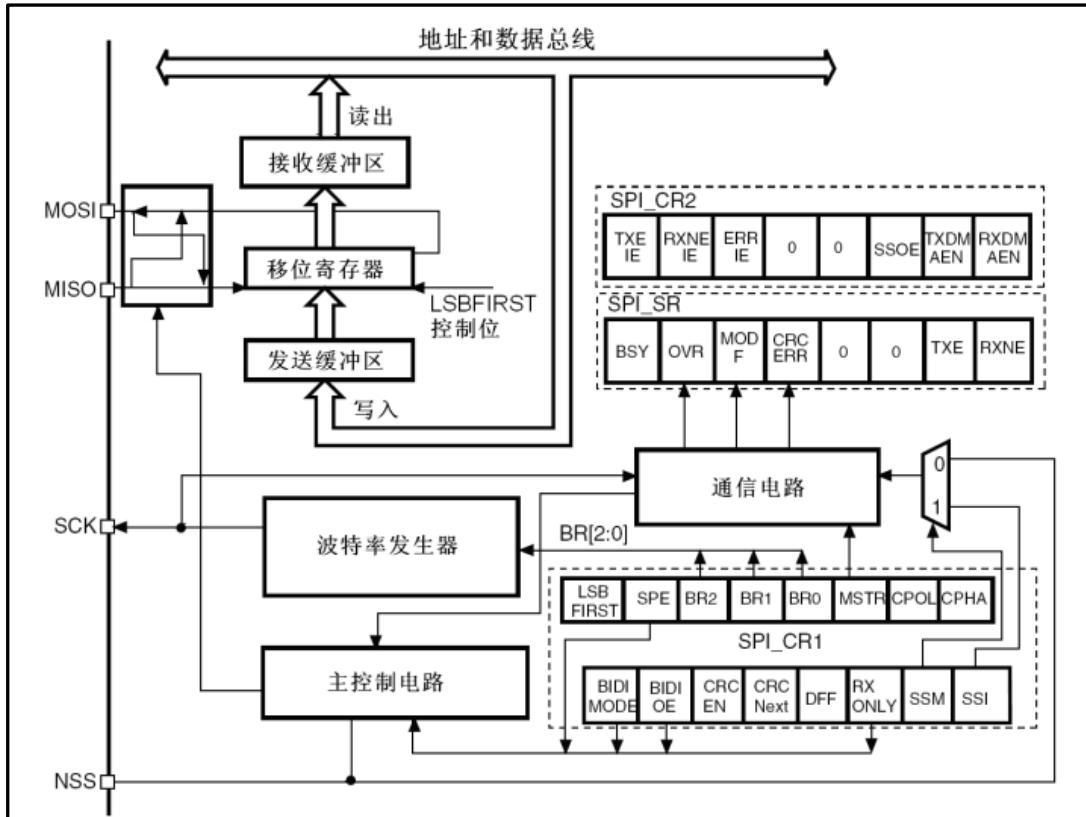
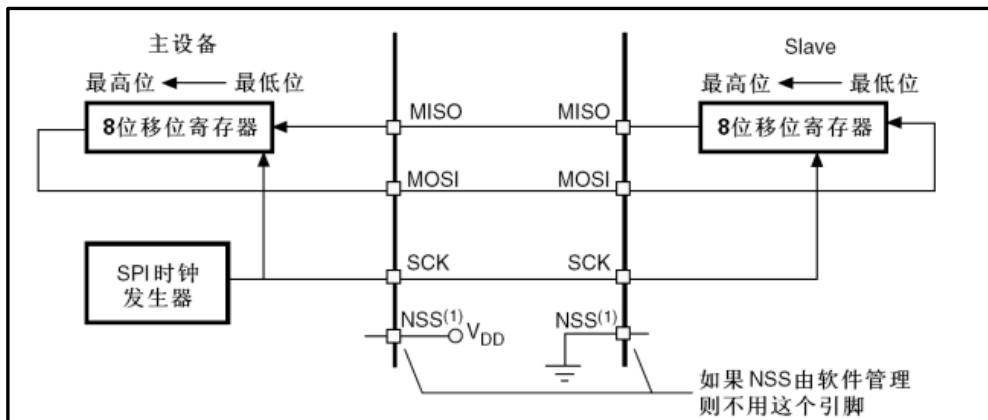


Figure 25- 1 SPI 的方框图

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（参考）

下图是一个单主和单从设备互连的例子。



**Figure 25- 2 单主和单从应用**

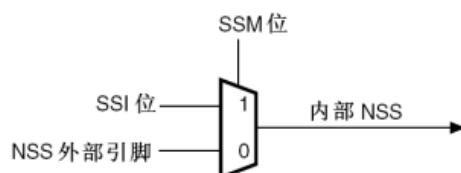
### 1. 这里 NSS 引脚设置为输入

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(Fig 25-3)。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 25- 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 25-4 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。

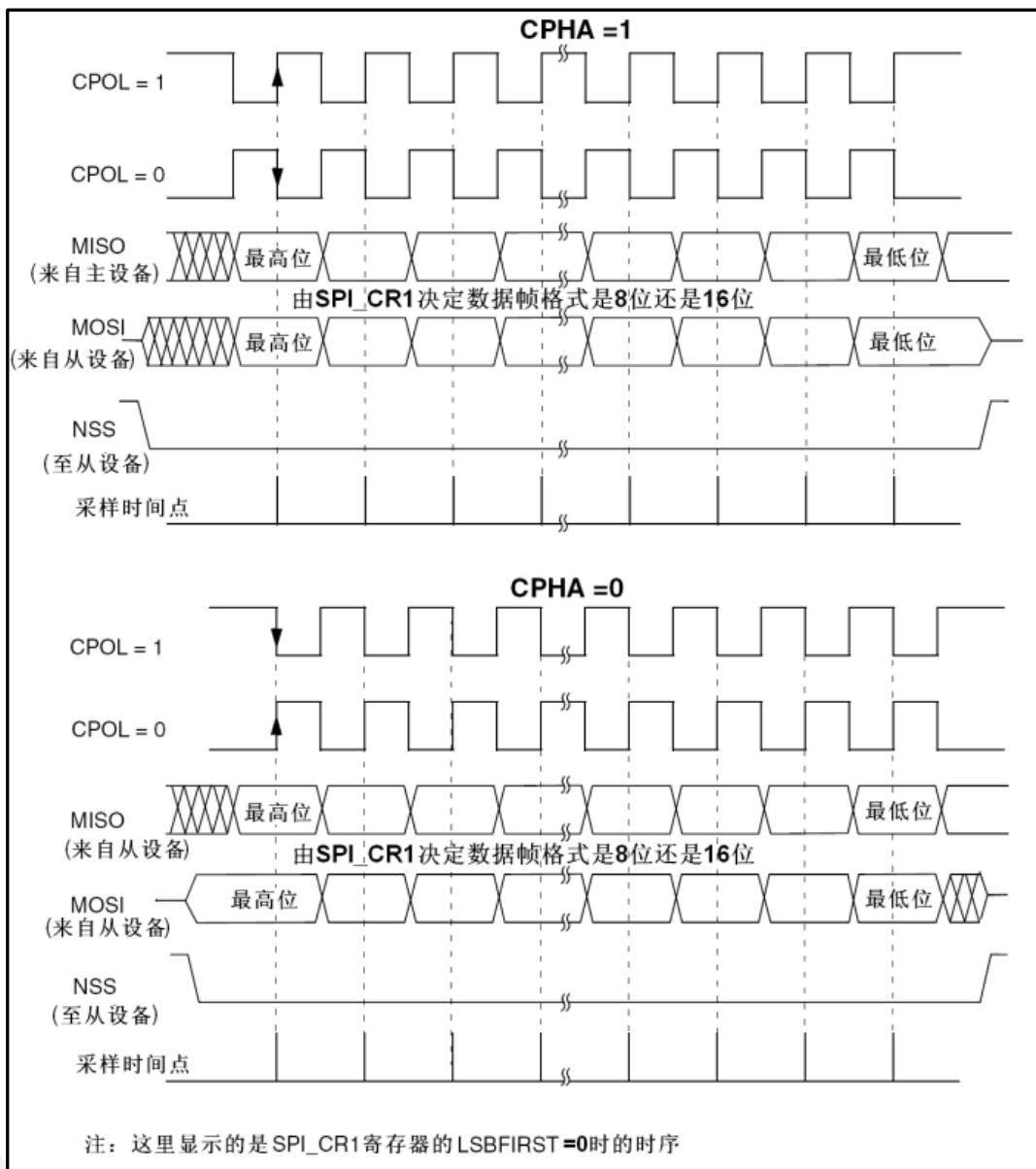


Figure 25- 4 数据时钟时序图

### 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接收都有效。

### 25.3-2 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  $\geq$  fPCLK/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 25-4)。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。

### 25.3-3 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 25-4)。
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。

### 25.3-4 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。

### 25.3.5 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；
  - 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。

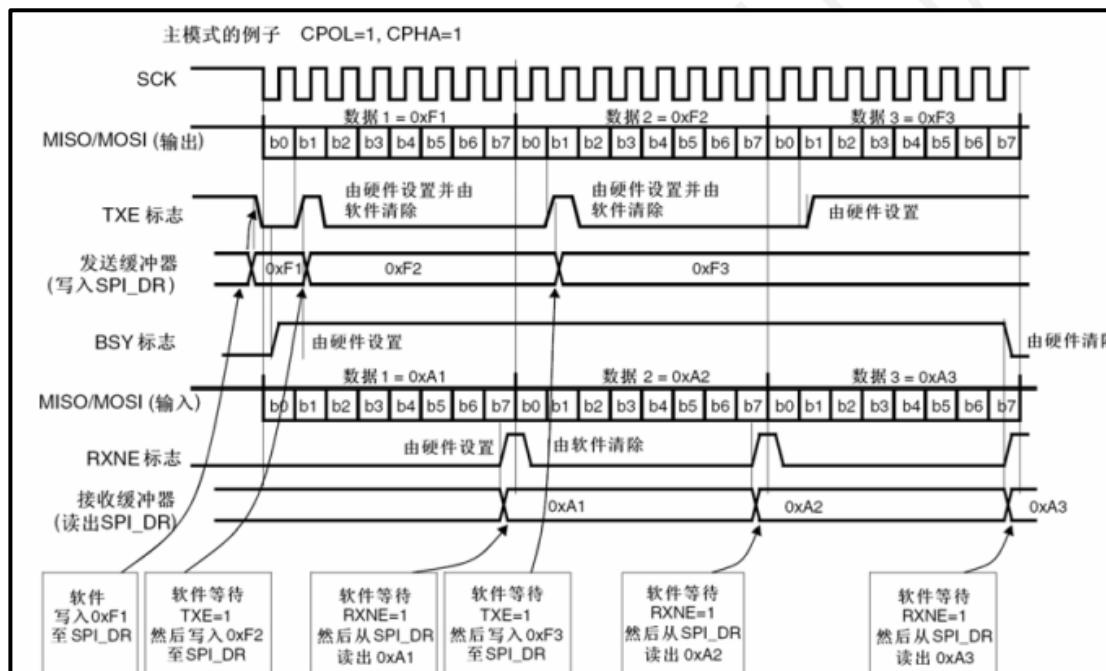
在采样时钟的最后一个边沿，当数据被从移位寄存器传送到接收缓冲器时，设置 RXNE 标志(接收缓冲器非空)；它表示数据已经就绪，可以从 SPI\_DR 寄存器读出；如果在 SPI\_CR2 寄存器中设置了 RXNEIE 位，则此时会产生一个中断；读出 SPI\_DR 寄存器即可清除 RXNIE 标志位。

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 25- 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**

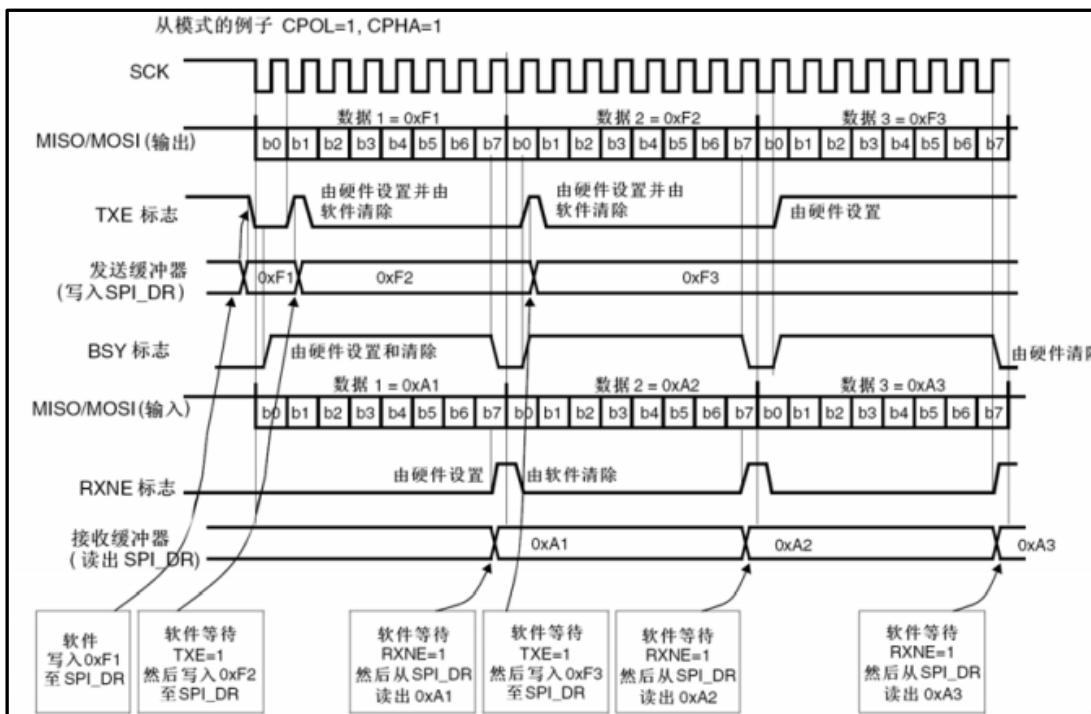


Figure 25- 6 从模式、全双工模式下(BIDIMODE=0 并且 RXONLY=0)连续传输时 TXE/RXNE/BSY 的变化示意图

### 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)

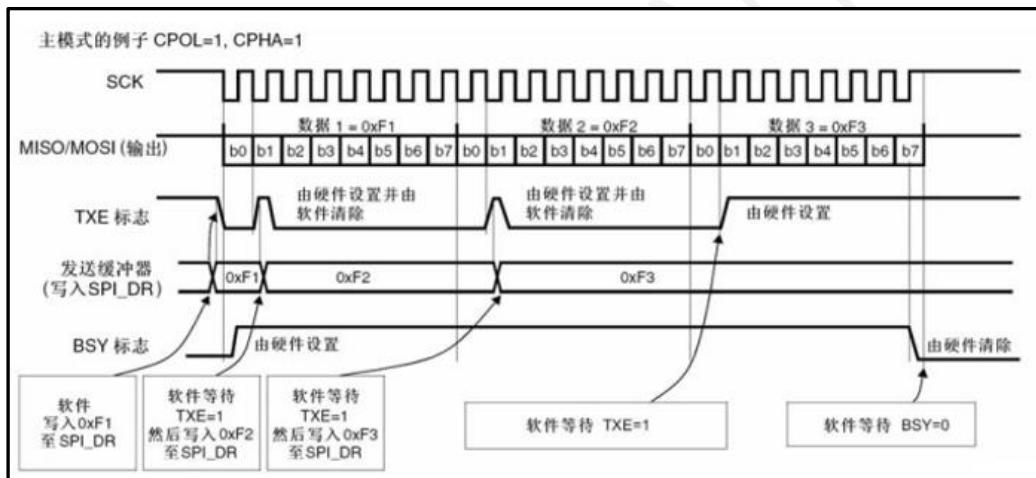


Figure 25- 7 主设备只发送模式(BIDIMODE=0 并且 RXONLY=0)下连续传输时， TXE/BSY 变化示意图

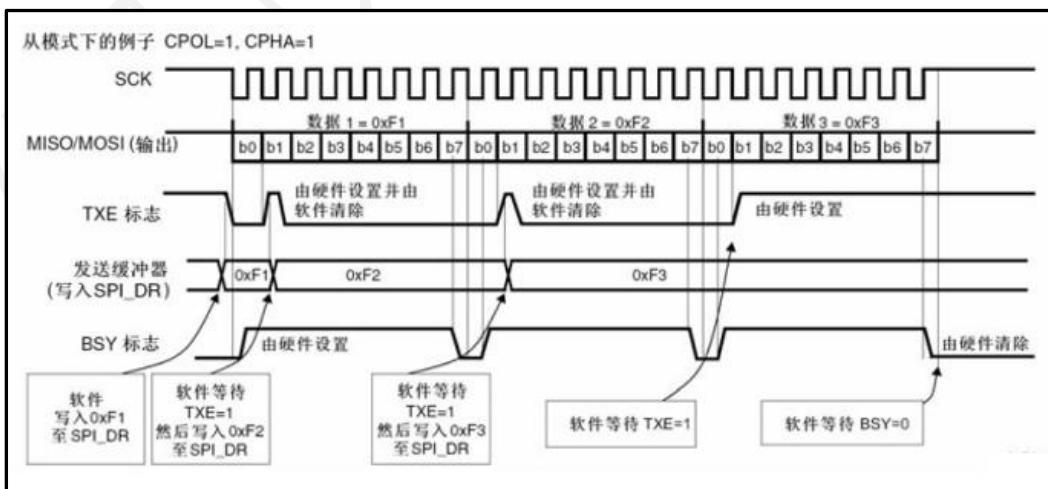


Figure 25- 8 从设备只发送模式(BIDIMODE=0 并且 RXONLY=0)下连续传输时， TXE/BSY 变化示意  
双向发送过程(BIDIMODE=1 并且 BIDIOE=1)

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。

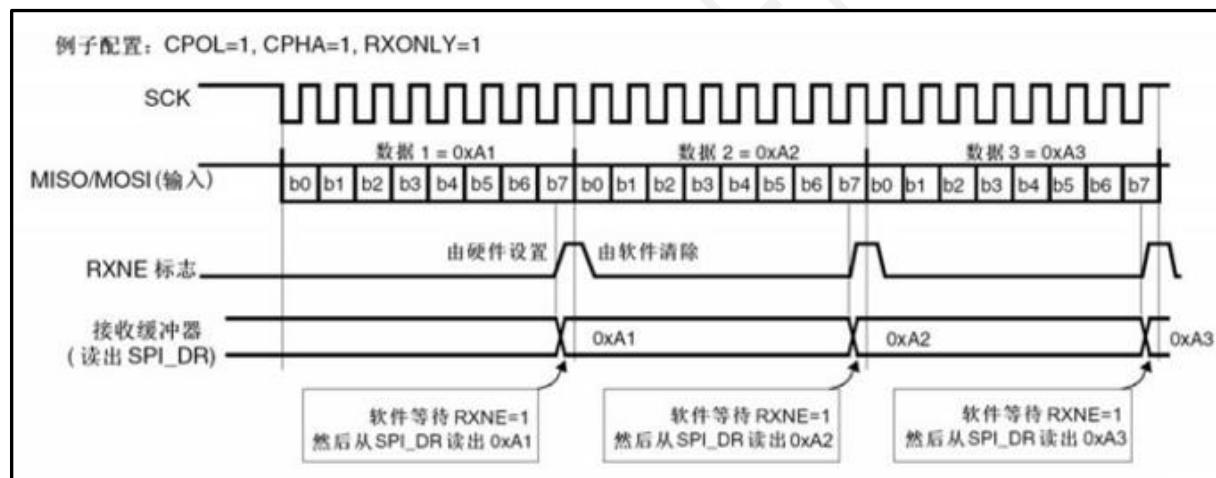


Figure 25- 9 只接收模式(BIDIMODE=0 并且 RXONLY=1)下连续传输时，RXNE 变化示意图

### 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’ 。

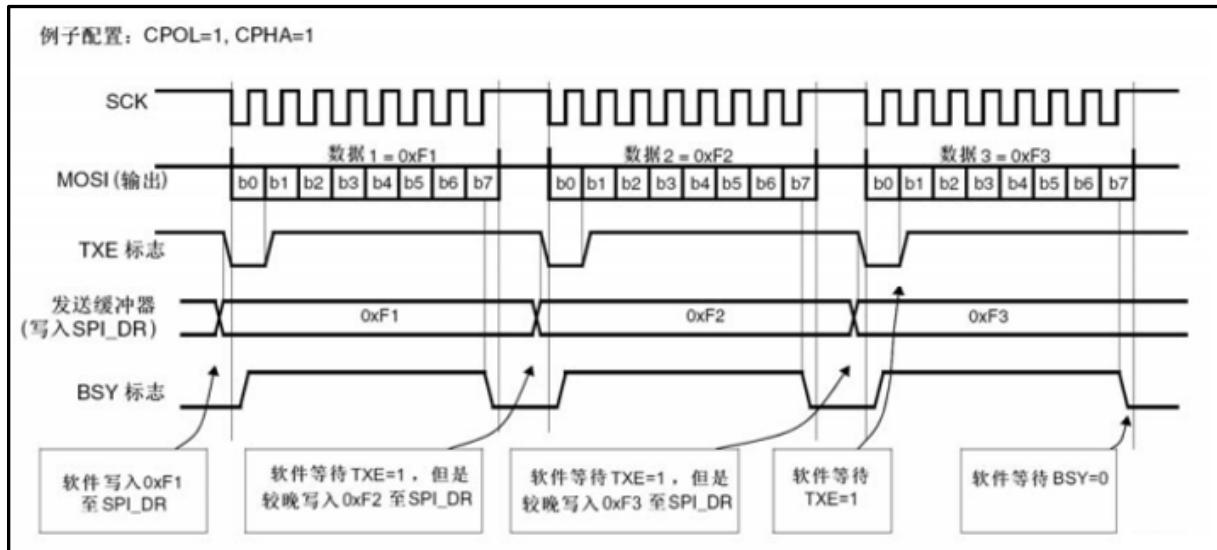
#### 连续和非连续传输

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 25- 10 非连续传输发送(BIDIMODE=0 并且 RXONLY=0)时，TXE/BSY 变化示意图**

**Note 25. 3 :** 目 **RXNE** 在发送时，**RXNE flag** 是无用的，请忽略此标志位

### 25.3-6CRC 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)。

#### **Note 25. 4 : 使用 SPI\_EN 开关方式重置 CRC 计算资料**

### **25.3-7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。

### 25.3-8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。

### 25.3-9 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。

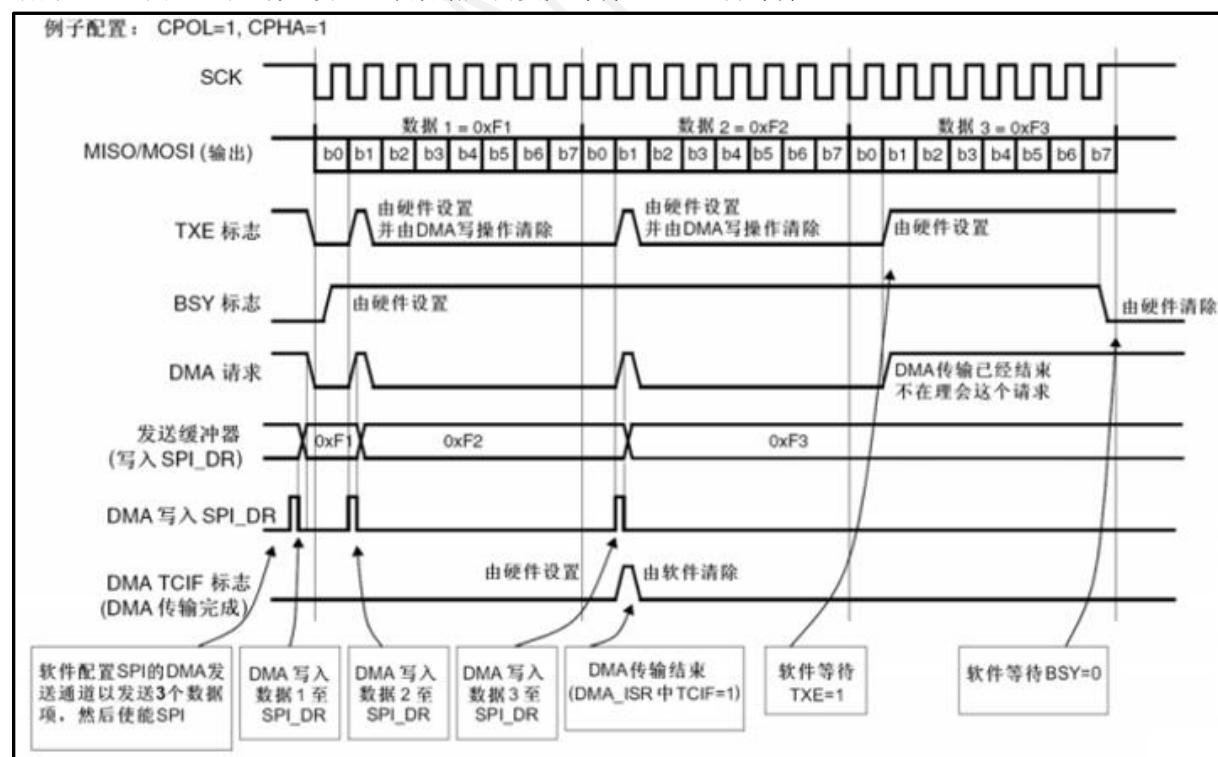


Figure 25- 11 使用 DMA 发送

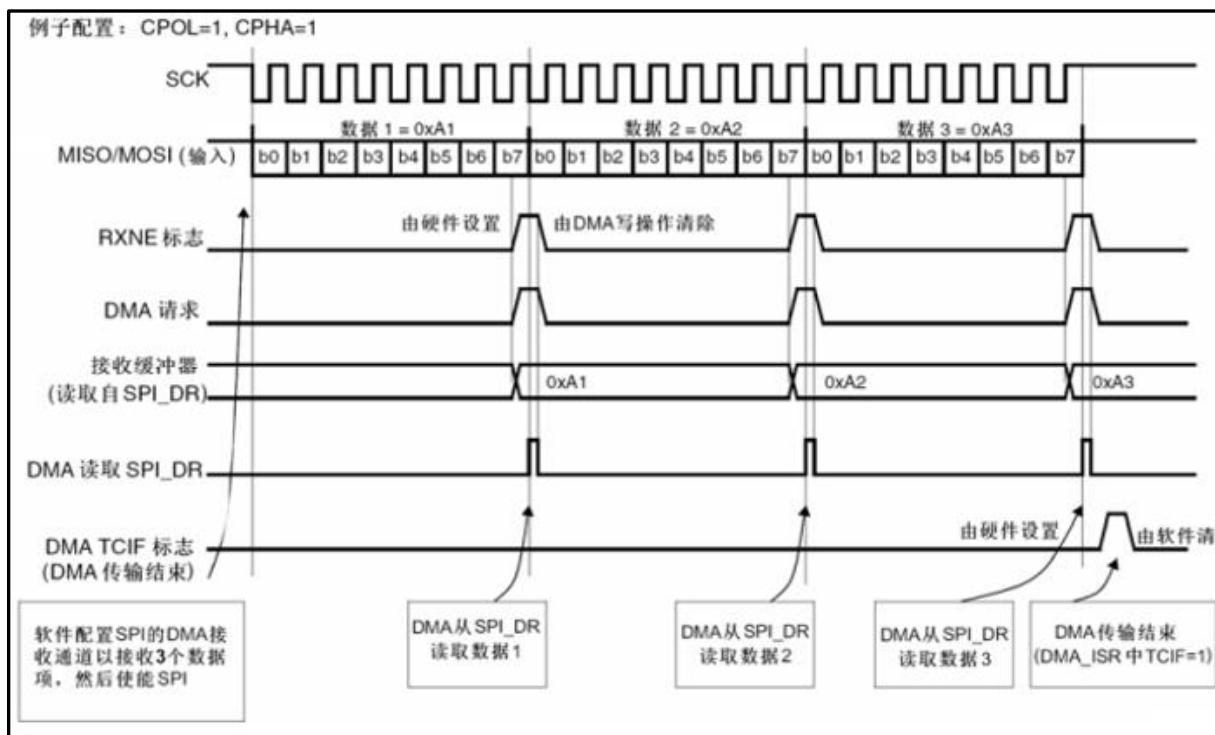


Figure 25- 12 使用 DMA 接收

### 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。

### 25.3-10 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’。

### 25.3-11 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	ERIE
溢出错误	OVR	
CRC 错误标志	CRCERR	

Table 25- 1 SPI 中断请求

## 25.4 I2S 功能描述

### 25.4-1 I2S 功能描述

I2S 的框图如下图所示：

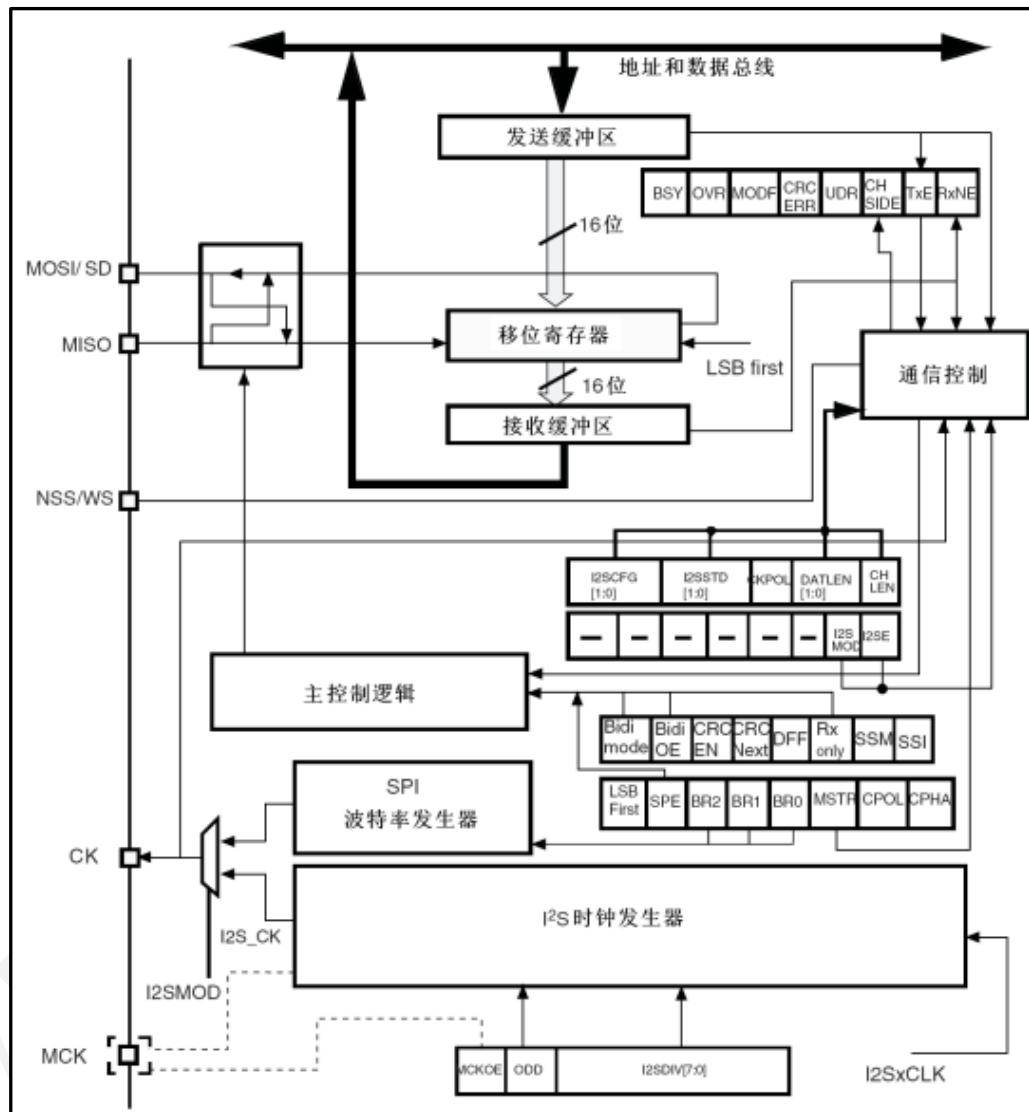


Figure 25- 13 I2S 框图

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 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。

#### 25.4-2 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。

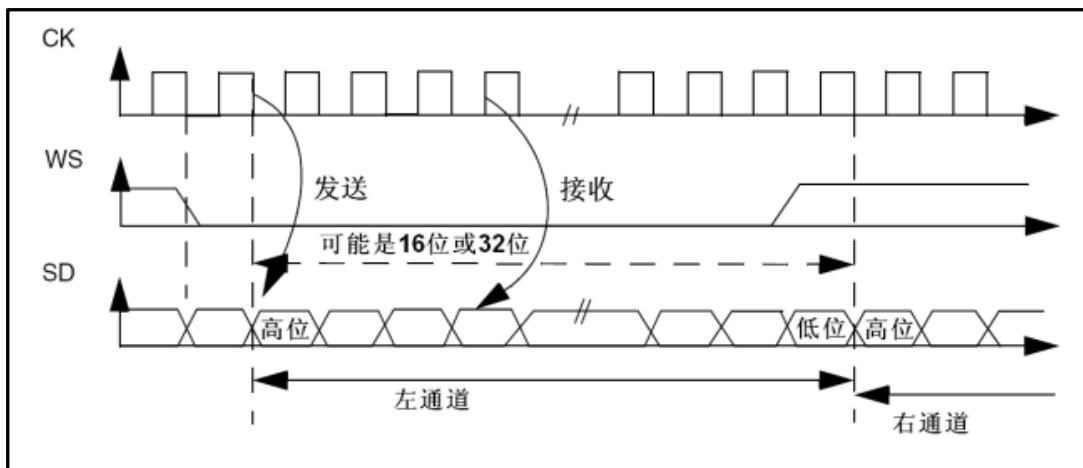


Figure 25- 14 I2S 飞利浦协议波形(16/32 位全精度, CPOL = 0)

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。

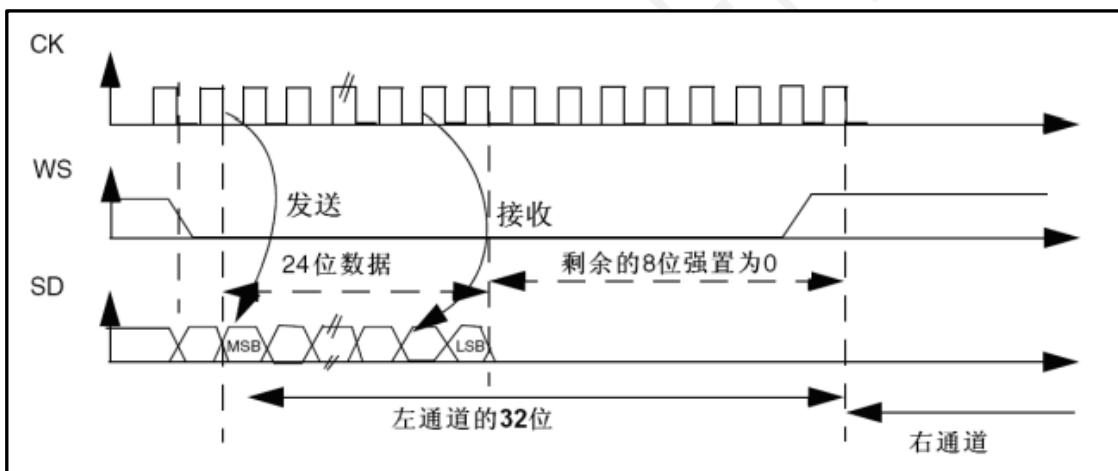


Figure 25- 15 I2S 飞利浦协议标准波形(24 位帧, CPOL = 0)

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)：

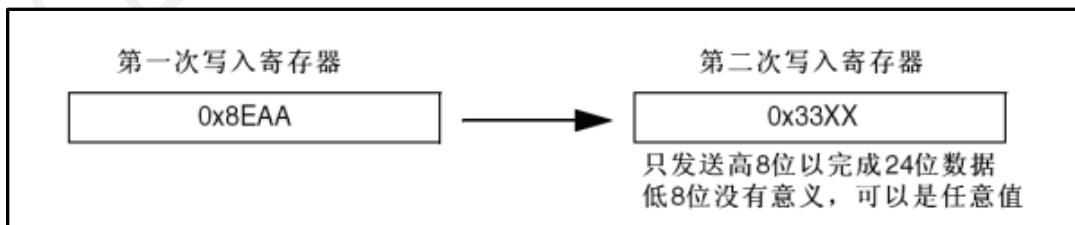


Figure 25- 16 发送 0x8EAA33

- 在接收模式下：如果接收 0x8EAA33：

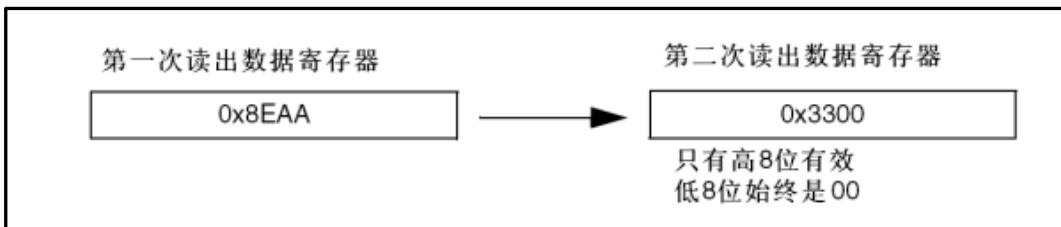


Figure 25- 17 接收 0x8EAA33

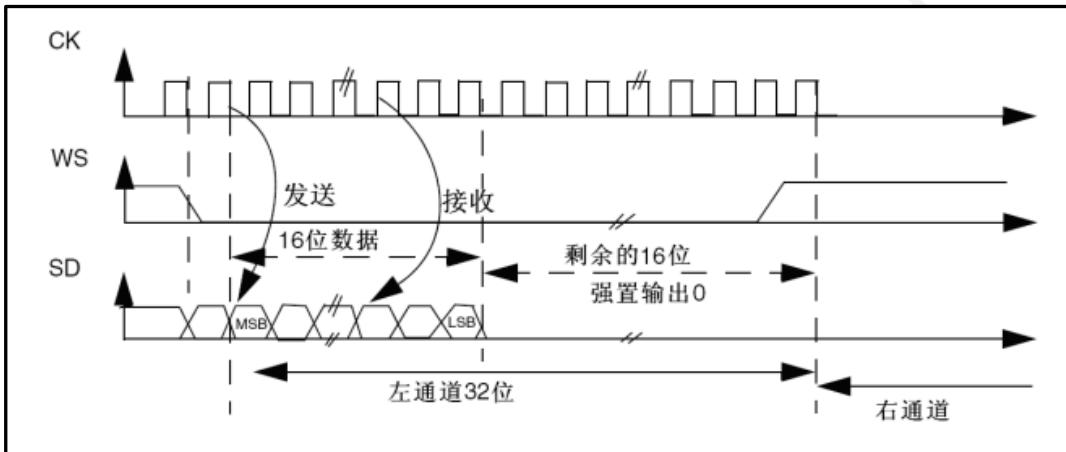
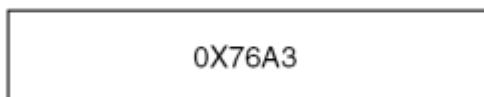


Figure 25- 18 I2S 飞利浦协议标准波形(16 位扩展至 32 位包帧, **CPOL = 0**)

在 I2S 配置阶段, 如果选择将 16 位数据扩展到 32 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。

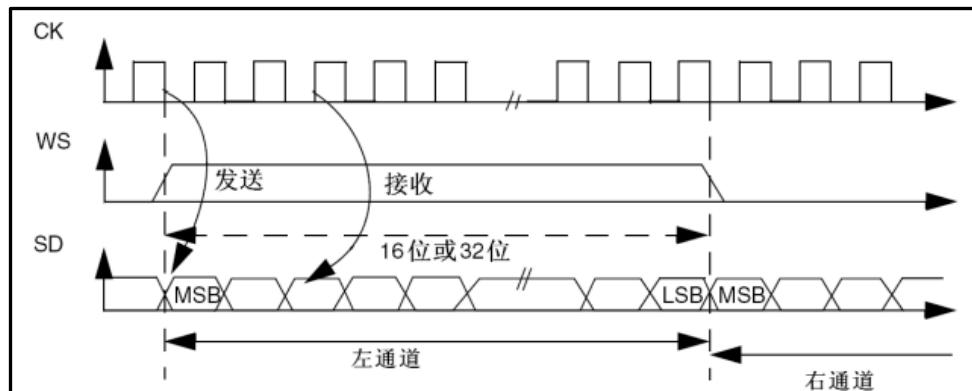
只需要操作一次 SPI\_DR



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。

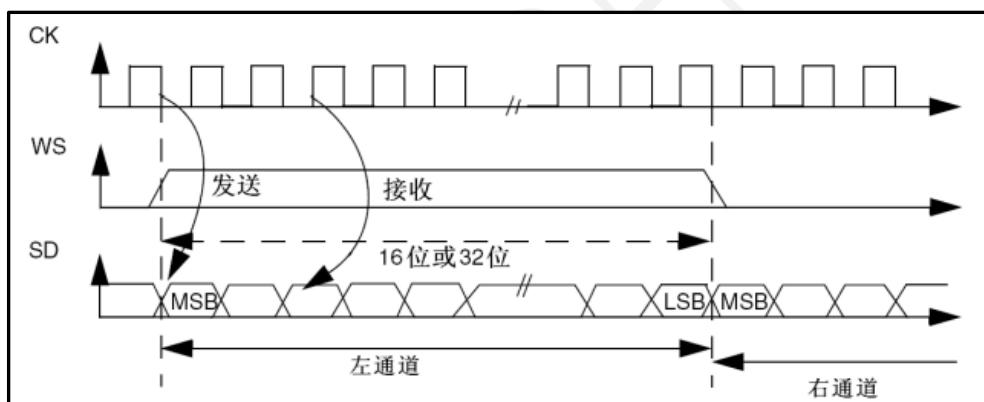
### MSB 对齐标准

在此标准下，WS 信号和第一个数据位，即最高位(MSB)同时产生。

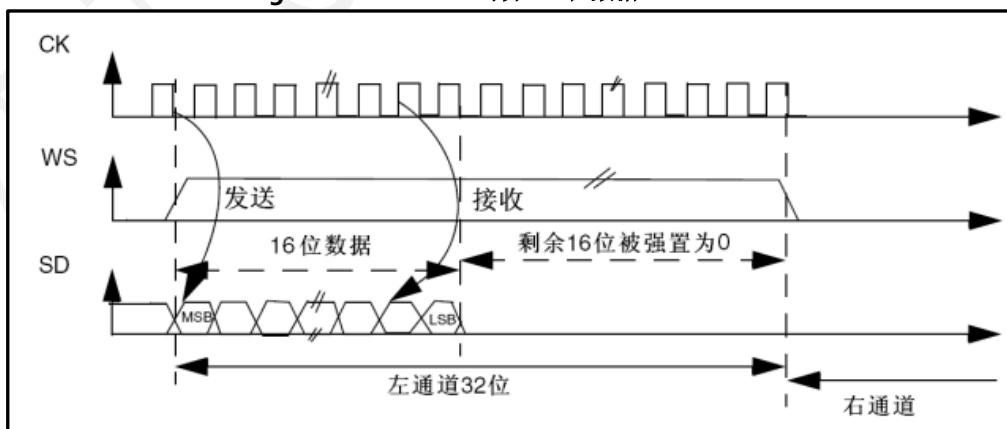


**Figure 25- 19 MSB 对齐 16 位或 32 位全精度, CPOL = 0**

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 25- 20 MSB 对齐 24 位数据, CPOL = 0**



**Figure 25- 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)。

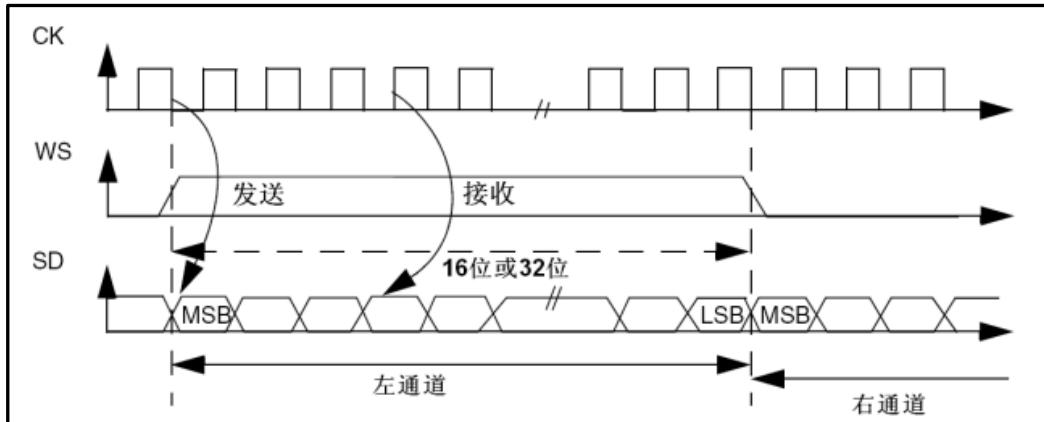


Figure 25- 22 LSB 对齐 16 位或 32 位全精度, CPOL = 0

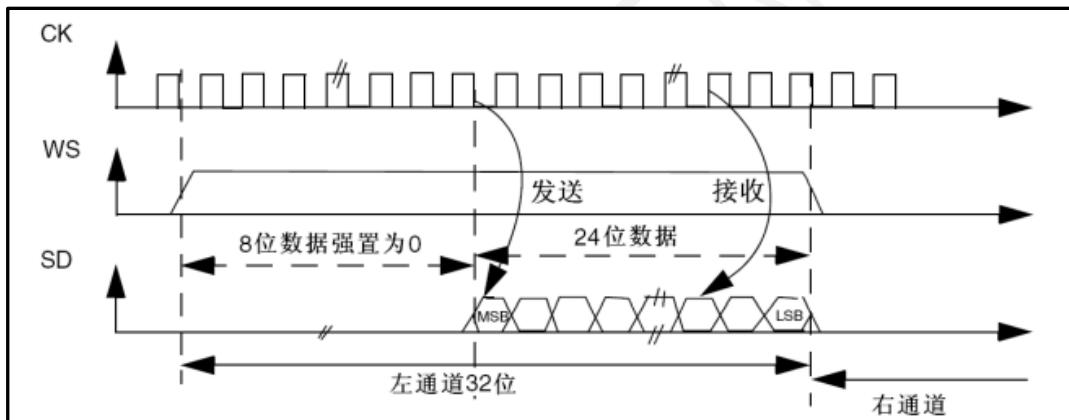


Figure 25- 23 LSB 对齐 24 位数据, CPOL = 0

- 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。

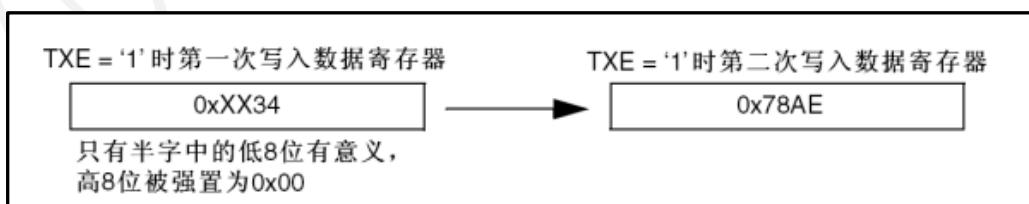


Figure 25- 24 要求发送 0x3478AE 的操作

- 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。

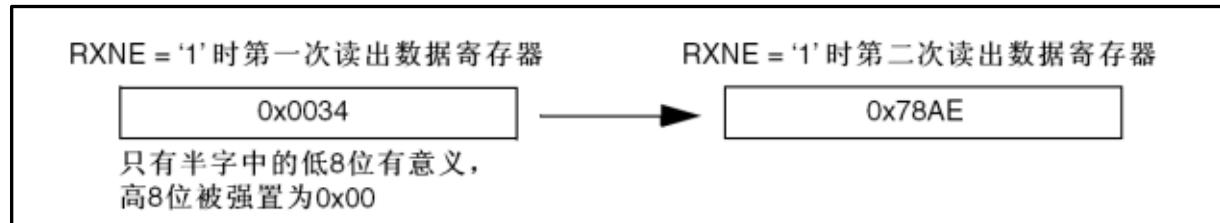


Figure 25-25 要求接收 0x3478AE 的操作

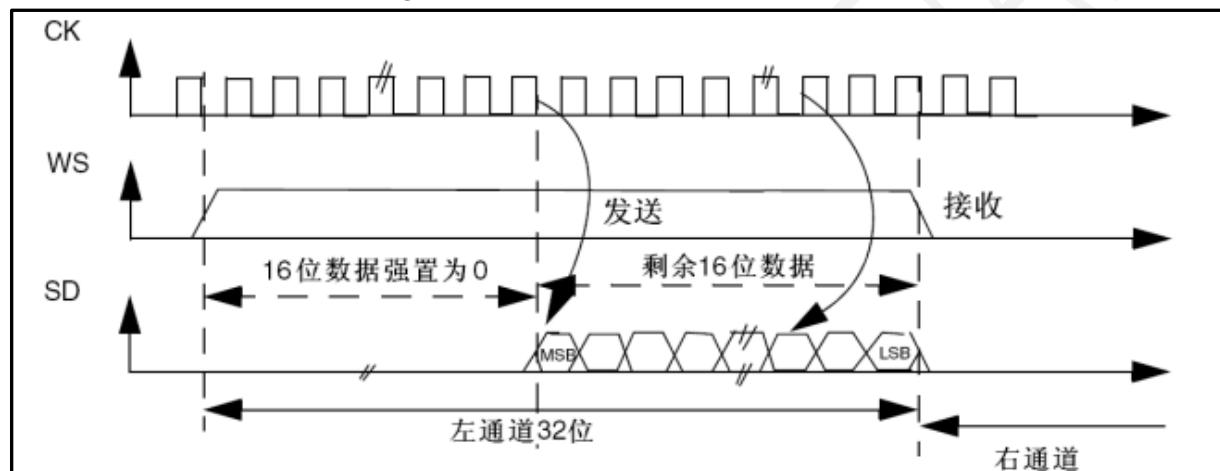


Figure 25-26 LSB 对齐 16 位数据扩展到 32 位包帧，CPOL =0

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。

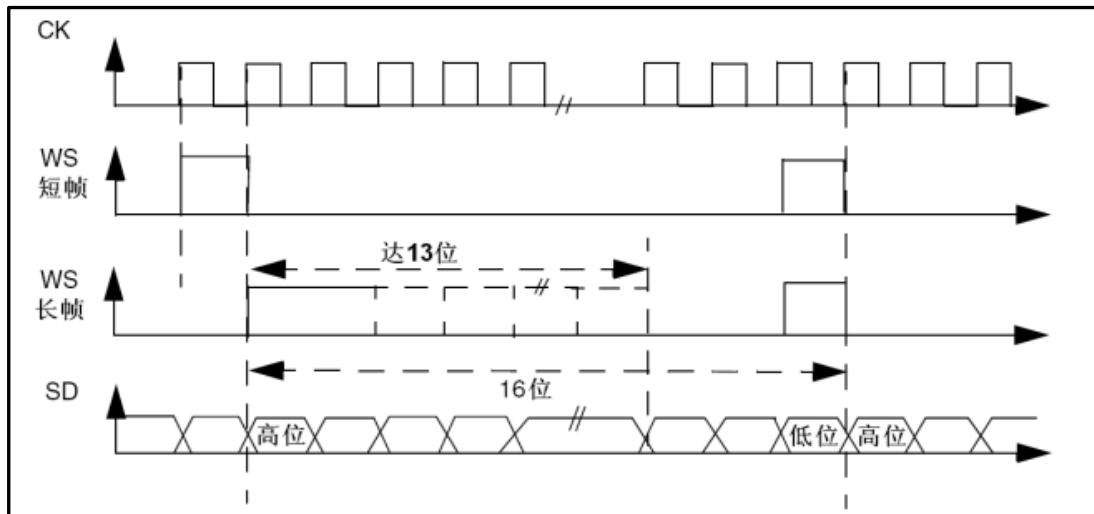


Figure 25- 27 PCM 标准波形(16 位)

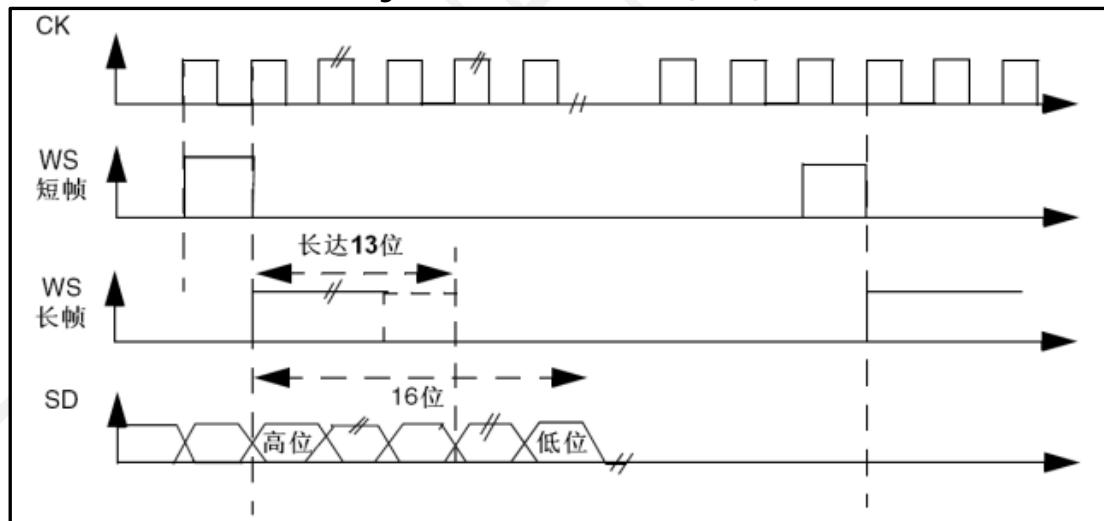


Figure 25- 28 PCM 标准波形(16 位扩展到 32 位包帧)

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。

### 25.4-3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：

$$\text{I2S 比特率} = 16 \times 2 \times F_s$$

如果包长为 32 位，则有：I2S 比特率 =  $32 \times 2 \times F_s$

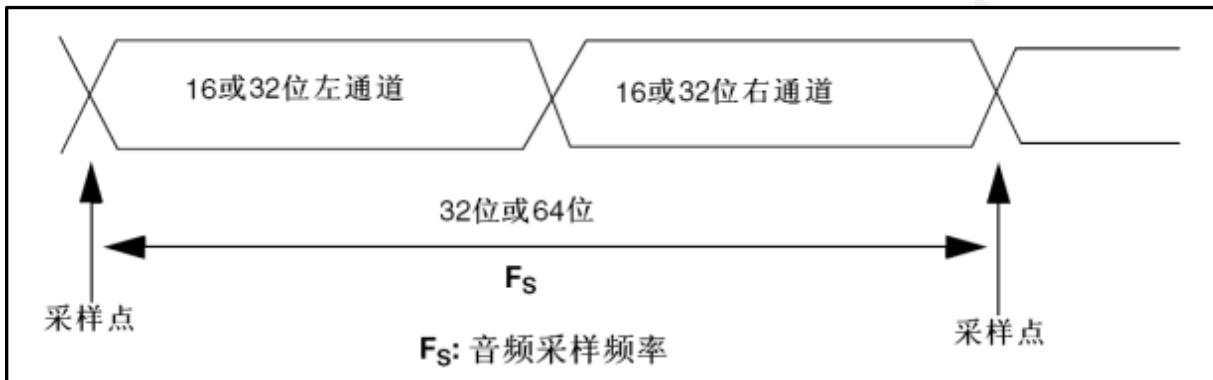


Figure 25- 29 音频采样频率定义

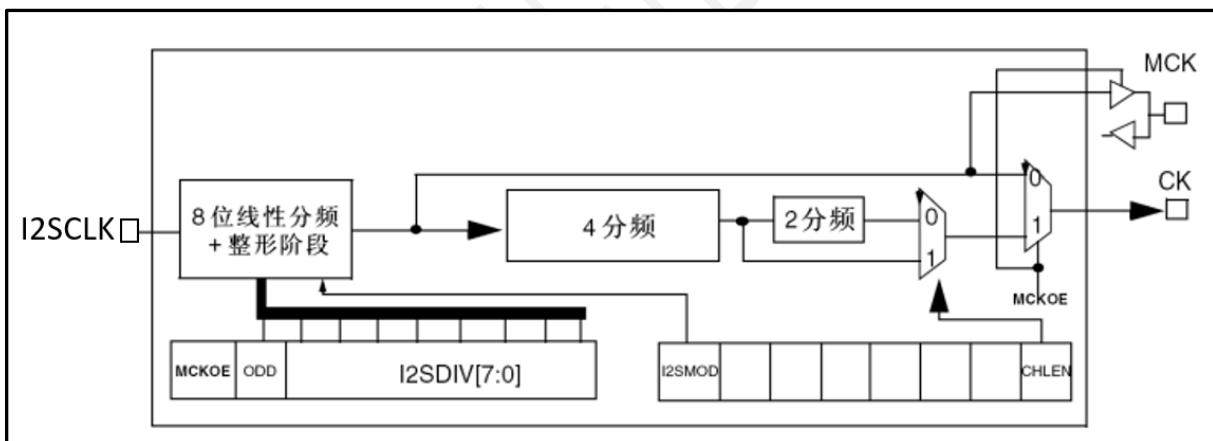


Figure 25- 30 I2S 时钟发生器结构

上图中 I2SxCLK 的时钟源是系统时钟(即驱动 AHB 时钟的 HSI、HSE 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' )：

$$\text{声道的帧长为 16 位时, } F_s = \text{I2SxCLK} / [(16*2) * ((2*I2SDIV) + ODD)*8]$$

$$\text{声道的帧长为 32 位时, } F_s = \text{I2SxCLK} / [(32*2) * ((2*I2SDIV) + ODD)*4]$$

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 $f_s$ (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 25- 2 使用标准的 8MHz HSE 时钟得到精确的音频频率

## 25.4-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。

## 25.4-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。

## 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。

## 25.4-6状态标志位

有 3 个状态标志位供用户监控 I<sub>S</sub>S 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 I<sub>S</sub>S 通信层的状态。

该位为' 1' 时表明 I<sub>S</sub>S 通讯正在进行中，但有一个例外：主接收模式(I<sub>S</sub>SCFG=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 I<sub>S</sub>S 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 I<sub>S</sub>S 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 I<sub>S</sub>S 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 I<sub>S</sub>S 被关闭时(I<sub>S</sub>SE 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 I<sub>S</sub>S 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 I<sub>S</sub>S 关闭再打开(同时，如果必要修改 I<sub>S</sub>S 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。

## 25.4-7 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。

## 25.4-8 I2S 中断

下表列举了全部 I2S 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
下溢标志位	OVR	ERRIE
上溢标志位	UDR	

Table 25- 3 I2S 中断请求

## 25.4-9 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。

## 25.5 SPI/I2S 寄存器列表

SPI/I2S 基址: 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	SPIX*_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 25- 4 SPI/I2S 寄存器列表和复位值

## 25.6 SPI/I2S 寄存器说明

### 25.6-1 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

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
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15	BIDIMODE	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	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。 该位只能在全双工模式下使用。 注: I2S模式下不使用。	0	R/W
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

位	标记	功能描述	复位值	读写
11	DFF	<p>DFF: 数据帧格式 (Data frame format)</p> <p>0: 使用8位数据帧格式进行发送/接收；</p> <p>1: 使用16位数据帧格式进行发送/接收。</p> <p>注: 只有当SPI禁止(SPE=0)时, 才能写该位, 否则出错。</p> <p>注: I2S模式下不使用。</p>	0	R/W
10	RXONLY	<p>RXONLY: 只接收 (Receive only)</p> <p>该位和BIDIMODE位一起决定在“双线双向”模式下的传输方向。</p> <p>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。</p> <p>0: 全双工(发送和接收);</p> <p>1: 禁止输出(只接收模式)。</p> <p>注: I2S模式下不使用。</p>	0	R/W
9	SSM	<p>SSM: 软件从设备管理 (Software slave management)</p> <p>当SSM被置位时, NSS引脚上的电平由SSI位的值决定。</p> <p>0: 禁止软件从设备管理;</p> <p>1: 启用软件从设备管理。</p> <p>注: I2S模式下不使用。</p>	0	R/W
8	SSI	<p>SSI: 内部从设备选择 (Internal slave select)</p> <p>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。</p> <p>注: I2S模式下不使用。</p>	0	R/W
7	LSBFIRST	<p>LSBFIRST: 帧格式 (Frame format)</p> <p>0: 先发送MSB;</p> <p>1: 先发送LSB。</p> <p>注: 当通信正在进行时不能改变该位的值。</p> <p>注: I2S模式下不使用。</p>	0	R/W
6	SPE	<p>SPE: SPI使能 (SPI enable)</p> <p>0: 禁止SPI设备;</p> <p>1: 开启SPI设备。</p> <p>注: I2S模式下不使用。</p> <p>注: 当关闭SPI设备时, 请按照第23.3.8节的过程操作。</p>	0	R/W
5:3	BR[2:0]	<p>BR[2:0]: 波特率控制 (Baud rate control)</p> <p>000/001: 保留</p> <p>010: fPCLK/8</p> <p>011: fPCLK/16</p> <p>100: fPCLK/32</p> <p>101: fPCLK/64</p> <p>110: fPCLK/128</p> <p>111: fPCLK/256</p> <p>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。</p>	0	R/W

位	标记	功能描述	复位值	读写
2	MSTR	<b>MSTR:</b> 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	<b>CPOL:</b> 时钟极性 (Clock polarity) 0: 空闲状态时, SCK保持低电平; 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。	0	R/W
0	CPHA	<b>CPHA:</b> 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。	0	R/W

## 25.6-2SPI 控制寄存器 2(SPI\_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				TXEIE	RXNEIE	ERRIE	保留				SSOE	TXDMAEN	RxDmaen		
--				R/W				--	R/W				R/W		

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	保留	--	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

## 25.6-3SPI 状态寄存器(SPI\_SR)

偏移地址: 0x08 复位值: 0x0000 0002

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				BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE				
--				R/W		RC W0	R/W								

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	--
7	BSY	<p>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。</p> <p>注: 使用这个标志时需要特别注意, 详见第 25.7,8 节</p>	0	R/W
6	OVR	<p>OVR: 溢出标志 (Overrun flag)  0: 没有出现溢出错误;  1: 出现溢出错误。</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25.4.7节。</p>	0	R/W
5	MODF	<p>MODF: 模式错误 (Mode fault)  0: 没有出现模式错误;  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)  0: 收到的CRC值和SPI_RXCRCR寄存器中的值匹配;  1: 收到的CRC值和SPI_RXCRCR寄存器中的值不匹配。  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	RC W0

位	标记	功能描述	复位值	读写
3	UDR	UDR: 下溢标志位 (Underrun flag) 0: 未发生下溢; 1: 发生下溢。 该标志位由硬件置'1'，由一个软件序列清'0'，详见25.4.7节。 注：在SPI模式下不使用。。	0x0	R/W
2	CHSIDE	CHSIDE: 声道 (Channel side) 0: 需要传输或者接收左声道; 1: 需要传输或者接收右声道。  注：在SPI模式下不使用。在PCM模式下无意义。	0	R/W
1	TXE	TXE: 发送缓冲为空 (Transmit buffer empty) 0: 发送缓冲非空; 1: 发送缓冲为空。	1	R/W
0	RXNE	RXNE: 接收缓冲非空 (Receive buffer not empty) 0: 接收缓冲为空; 1: 接收缓冲非空。	0	R/W

## 25.6-4SPI 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															
--															

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	--	0x0	--
15:0	DR[15:0]	<p>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。</p> <p>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]。</p>	0x0	R/W

## 25.6-5SPI 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															
--															

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	--	0x0	--
15:0	CRCPOLY[15:0]	CRC多项式寄存器 (CRC polynomial register) 该寄存器包含了CRC计算时用到的多项式。 其复位值为0x0007，根据应用可以设置其他数值。 注：在I2S模式下不使用。	0x0007	R/W

## 25.6-6SPI 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															
--															

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

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15:0	RXCRC [15:0]	<p>RXCRC[15:0]: 接收CRC寄存器位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>	0x0000	RO

### 25.6-7SPI 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															
--															

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

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15:0	TXCRC [15:0]	<p><b>TXCRC[15:0]:</b> 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。 注：在I2S模式下不使用。</p>	0x0000	RO

### 25.6-8 SPI Tx CRC 寄存器(SPI\_TXCRCR)

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
保留				I2SMOD	I2SE	I2SCFG	PCMSYNC	保留	I2SSTD	CKPOL	DATLEN	CHLEN			
--				R/W				--	R/W						

位	标记	功能描述	复位值	读写
31:12	保留	--	0x0	--
11	I2SMOD	<p>I2SMOD: I2S模式选择 (I2S mode selection)  0: 选择SPI模式;  1: 选择I2S模式。</p> <p>注: 该位只有在关闭了SPI或者I2S时才能设置。。</p>	0	R/W
10	I2SE	<p>I2SE: I2S使能 (I2S enable)  0: 关闭I2S;  1: I2S使能。</p> <p>注: 在SPI模式下不使用。。</p>	0	R/W
9:8	I2SCFG	<p>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。</p> <p>注: 该位只有在关闭了I2S时才能设置。  在SPI模式下不使用。</p>	0	R/W
7	PCMSYNC	<p>PCMSYNC: PCM帧同步 (PCM frame synchronization)  0: 短帧同步;  1: 长帧同步。</p> <p>注: 该位只在I2SSTD = 11 (使用PCM标准)时有意义。  在SPI模式下不使用。</p>	0	R/W

位	标记	功能描述	复位值	读写
6:5	保留	--	0x0	--
4:5	I2SSTD	<p>I2SSTD: I2S标准选择 (I2S standard selection)</p> <p>00: I2S飞利浦标准；</p> <p>01: 高字节对齐标准 (左对齐)；</p> <p>10: 低字节对齐标准(右对齐)；</p> <p>11: PCM 标准。</p> <p>关于I2S标准的细节，详见23.4.2节。</p> <p>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。。</p>	0	R/W
3	CKPOL	<p>CKPOL: 静止态时钟极性 (Steady state clock polarity)</p> <p>0: I2S时钟静止态为低电平；</p> <p>1: I2S时钟静止态为高电平。</p> <p>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。</p>	0	R/W
2:1	DATLEN	<p>DATLEN: 待传输数据长度 (Data length to be transferred)</p> <p>00: 16位数据长度；</p> <p>01: 24位数据长度；</p> <p>10: 32位数据长度；</p> <p>11: 不允许。</p> <p>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。</p>	0	R/W
0	CHLEN	<p>CHLEN: 声道长度 (每个音频声道的数据位数) (Channel length (number of bits per audio channel))</p> <p>0: 16位宽；</p> <p>1: 32位宽。</p> <p>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。</p> <p>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。</p>	0	R/W

### 25.6-9SPI\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						MCKOE	ODD	I2SDIV[7:0]							
--						R/W	R/W	R/W							

位	标记	功能描述	复位值	读写
31:10	保留	--	0x0	--
9	MCKOE	<p>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。</p> <p>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。</p>	0	R/W
8	ODD	<p>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。</p> <p>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。</p>	0	R/W
7:0	I2SDIV[7:0]	<p>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。</p> <p>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。</p>	0x02	R/W

## 【26】One-Wire 接口(OWIRE)

### 26.1 单总线协议(One-Wire)

主机和从机通过 1 根线进行通信，在一条总线上可挂接的从器件数量几乎不受限制。

#### 26.1-1 特点

它采用单根信号线，既可传输时钟，又能传输数据，而且数据传输是双向的。

#### 26.1-2 优点

单总线技术具有线路简单，硬件开销少，成本低廉，便于总线扩展和维护等。

### 26.2 单总线通信过程

#### 26.2-1 初始化

初始化过程 = 复位脉冲 + 从机应答脉冲。

主机通过拉低单总线 480~960 us 产生复位脉冲，然后释放总线，进入接收模式。主机释放总线时，会产生低电平跳变为高电平的上升沿，单总线器件检测到上升沿之后，延时 15~60 us，单总线器件拉低总线 60~240 us 来产生应答脉冲。主机接收到从机的应答脉冲说明单总线器件就绪，初始化过程完成。

初始化时序图如下图初始化过程中的复位与应答脉冲所示：

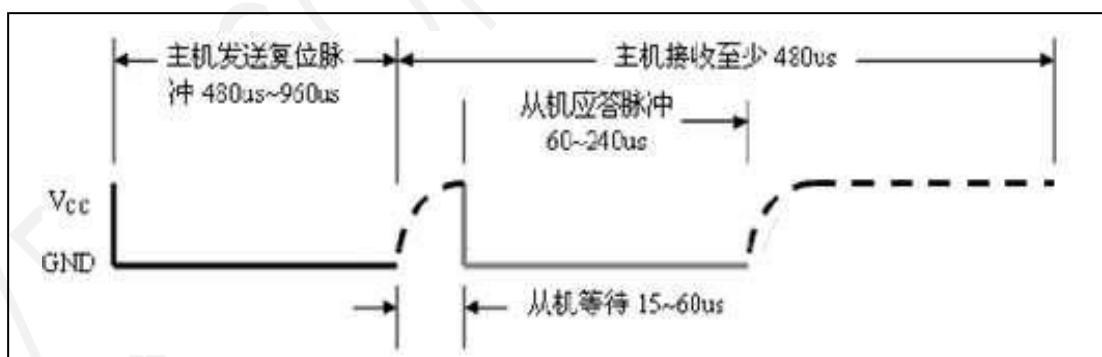


Figure 26- 1 初始过程中的复位与应答脉冲

#### 26.2-2 写时间间隙

写时间隙有两种，包括写 0 的时间隙和写 1 的时间隙。

当数据线拉低后，在 15~60us 的时间窗口内对数据线进行采样。如果数据线为低电平，就是写 0，如果数据线为高电平，就是写 1。主机要产生一个写 1 时间隙，就必须把数据线拉低，在写时间隙开始后的 15us 内允许数据线拉高。主机要产生一个写 0 时间隙，就必须把数据线拉低并保持 60us。

写时间隙时序图如下图单总线通信协议中写时间隙时序图所示：

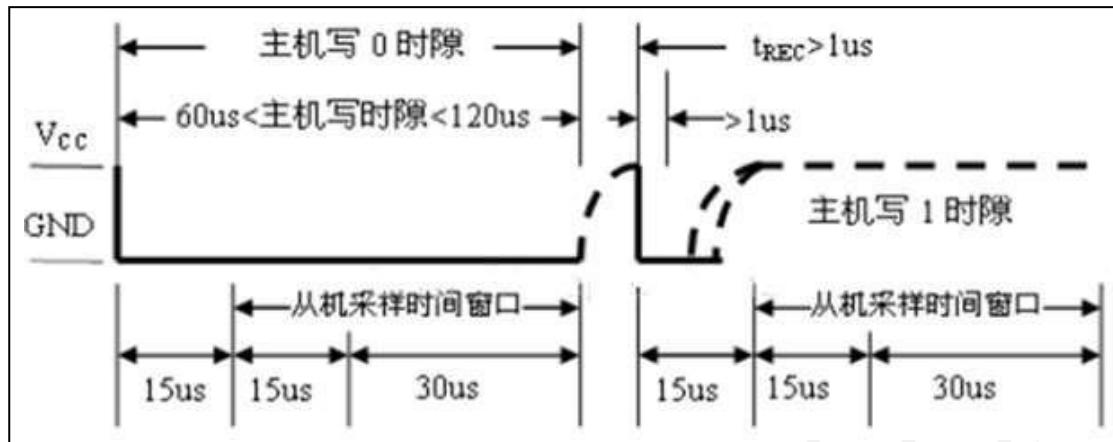


Figure 26- 2 单总线通信协议中写时间隙时序图

### 26.2-3 读时间隙

当主机把总线拉低时，并保持至少 1us 后释放总线，必须在 15us 内读取数据。

读时间隙时序图如下图单总线通信协议中读时间隙时序图所示：

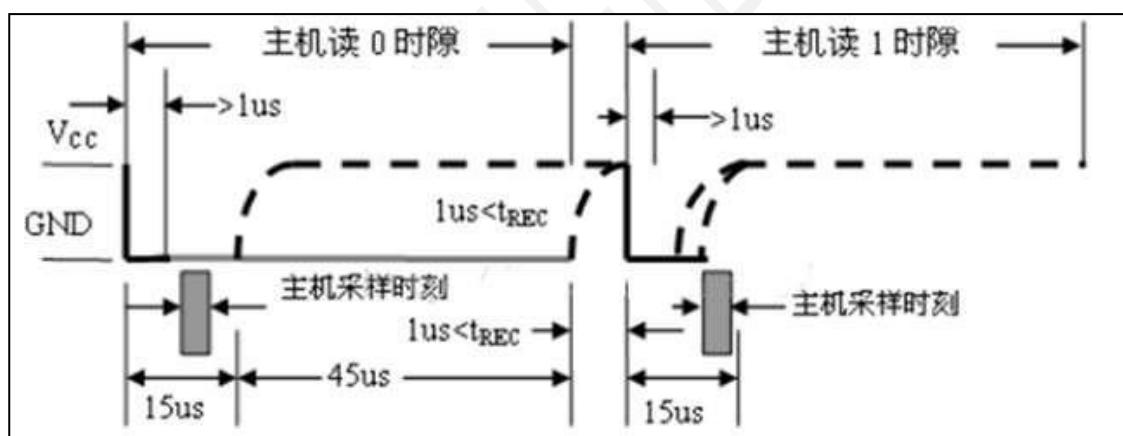


Figure 26- 3 单总线通信协议中读时间隙时序图

### 26.3 配置说明

#### 26.3-1 初始化配置说明

1. 配置 GPIO 相应的管脚复用 One-Wire 管脚;
2. 配置 OWIRE\_CR.CLKDIV 寄存器, 设置 One-Wire 模式时钟选择;
3. 配置 One-Wire Reset 宽度控制寄存器 OWIRE\_RSTCNT, 设置 One-Wire 主发送复位时间 (480us~960us);
4. 配置 One-Wire 应答宽度计数寄存器 OWIRE\_PRESCNT, 设置 One-Wire 从应答设定计数值 (60us~240us);
5. 配置 One-Wire 中断使能寄存器 OWIRE\_INTEN.INITEN 使能初始化完成中断;
6. 配置 OWIRE\_CR.EN 寄存器, 使能 One-Wire 模块;
7. 配置 One-Wire 总线操作命令寄存器 OWIRE\_CMD, 设置 Initial 指令;
8. 系统进入中断子程序, 在中断子程序中配置状态清除寄存器 OWIRE\_INTCLR, 清除相应的中断标志;

#### 26.3-2 读数据配置说明

1. 配置 One-Wire Bit rate 计数器 OWIRE\_BITRATECNT, 设置 1 Bit 数据宽度(15us~60us);
2. 配置 One-Wire 主器件读/写 PULL0 驱动时间 OWIRE\_DRVCNT, 设置驱动时间宽度 (0us~15us);
3. 配置 One-Wire 主器件读采样时间设定 OWIRE\_RDSMPCNT, 设置读采样时间(1us~15us);
4. 配置 1-Wire Recover Time 计数区间值 OWIRE\_RECCNT, 设置 RECOVER 时间为 (TREC>1us);
5. 配置 One-Wire 中断使能寄存器 OWIRE\_INTEN.RXDONEEN 使能接收完成中断;
6. 配置 One-Wire 总线操作命令寄存器 OWIRE\_CMD, 设置 RX 指令;
7. 系统进入中断子程序, 中断子程序中配置状态清除寄存器 OWIRE\_INTCLR, 清除接收完成中断 FLAG, 并读 One-Wire 数据寄存器 OWIRE\_DATA;

#### 26.3-3 写数据配置说明

1. 配置 One-Wire Bit rate 计数器 OWIRE\_BITRATECNT, 设置 1 Bit 数据宽度(15us~60us);
2. 配置 One-Wire 主器件读/写 PULL0 驱动时间 OWIRE\_DRVCNT, 设置驱动时间宽度 (0us~15us);
3. 配置 1-Wire Recover Time 计数区间值 OWIRE\_RECCNT, 设置 RECOVER 时间为 (TREC>1us);
4. 配置 One-Wire 中断使能寄存器 OWIRE\_INTEN.TXDONEEN 使能发送完成中断;
5. 写数据到 One-Wire 数据寄存器 OWIRE\_DATA;
6. 配置 One-Wire 总线操作命令寄存器 OWIRE\_CMD, 设置 TX 指令;
7. 数据发送完后系统进入中断子程序, 中断子程序中配置状态清除寄存器 OWIRE\_INTCLR, 清除 TX 完成中断 FLAG;

## 26.4 One-Wire 接口寄存器列表

地址: 0x4000 3800

地址偏移	名称	描述	默认值
0x00	OWIRE_CR	1-wire 模块控制寄存器	0x0000 0000
0x04	OWIRE_NFCR	1-Wire 输入端子滤波控制寄存器	0x0000 0000
0x08	OWIRE_RSTCNT	1-Wire Master Reset pulse 宽度计数寄存器	0x0000 0000
0x0C	OWIRE_PRESCNT	1-Wire Device Presence Pulse 宽度计数寄存器	0x0000 0000
0x10	OWIRE_BITRATECNT	1-Wire Bit rate 设计计数器	0x0000 0000
0x14	OWIRE_DRVCNT	1-Wire 主器件读/写PULL0 驱动时间	0x0000 0000
0x18	OWIRE_RDSMPCNT	1-Wire 主器件读的采样时间设定	0x0000 0000
0x1C	OWIRE_RECCNT	1-Wire Recover Time 计数区间值	0x0000 0000
0x20	OWIRE_DATA	1-Wire 数据寄存器	0x0000 0000
0x24	OWIRE_CMD	1-Wire 总线操作命令寄存器	0x0000 0000
0x28	OWIRE_INTEN	1-wire 中断使能寄存器	0x0000 0000
0x2C	OWIRE_SR	1-wire 状态寄存器	0x0000 0000
0x30	OWIRE_INTCLR	1-wire 中断状态清除寄存器	0x0000 0000

Table 26- 1 One-Wire 接口寄存器列表和复位值

## 26.5寄存器说明

### 26.5-11-Wire 模块控制寄存器(OWIRE\_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								RDMODE	MSBFIRST	EN	SIZE	保留		CLKDIV[1:0]	
--								R/W				--		R/W	

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	--
7	RDMODE	0: 普通模式 1: 写 0/读 0 间隙相等	0	R/W
6	MSBFIRST	字节发送的位模式设定 0: LSB(bit0) send/receive first 1: MSB(bit7) send/receive first 当OWIRE_CR.SIZE=0 时, 该位要设定为 0	0	R/W
5	EN	1-wire 模块使能控制位 0: 1-wire 模块停止 1: 1-wire 模块使能	0	R/W
4	SIZE	数据处理位数控制位 0: 单次处理 1 bit(Bit mode) 1: 单次处理 8 bit(Byte mode)	0	R/W
3:2	保留	--	0x0	--
1:0	CLKDIV[1:0]	计数器用时钟源选择位 00: FPCLK 01: FPCLK/2 10: FPCLK/4 11: FPCLK/16	0	R/W

### 26.5-21-Wire 输入端子滤波控制寄存器(OWIRE\_NFCR)

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										NFEN	保留		NFDIV[1:0]		
--										R/W	--		R/W		

位	标记	功能描述	复位值	读写
31:5	保留	--	0x0	--
4	NFEN	输入端子滤波使能控制位 0: 滤波功能无效 1: 滤波功能有效	0	R/W
3:2	保留	--	0x0	--
1:0	NFDIV[1:0]	输入端子滤波时钟源选择位 00: FPCLK 01: FPCLK/2 10: FPCLK/4 11: FPCLK/8	0x0	R/W

### 26.5-31-Wire RESET 宽度控制寄存器(OWIRE\_RSTCNT)

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
RSTCNT[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0	--
15:0	RSTCNT[15:0]	主发送复位时间设定计数值	0x0	R/W

### 26.5-41-Wire Presence Pulse 宽度计数寄存器(OWIRE\_PRESCNT)

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
保留															
--															

位	标记	功能描述	复位值	读写
31:13	保留	--	0x0	--
12:0	PRESCTN[12:0]	从应答时间设定计数值	0	R/W

### 26.5-51-Wire Bit rate 设计计数器(OWIRE\_BITRATECNT)

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				BITRATECNT[11:0]											
--				R/W											

位	标记	功能描述	复位值	读写
31:12	保留	--	0x0	--
11:0	BITRATECNT[11:0]	Bit Rate 时间设定计数值	0	R/W

### 26.5-61-Wire 主器件读/写 PULL0 驱动时间(OWIRE\_DRVCNT)

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							DRVCNT[8:0]								
--							R/W								

位	标记	功能描述	复位值	读写
31:9	保留	--	0x0	--
8:0	DRVCNT[8:0]	主器件读/写PULL0 驱动时间设定计数值	0x0	R/W

### 26.5-71-Wire 主器件读采样时间设定(OWIRE\_RDSMPCNT)

地址偏移: 0x018 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								RDSMPCNT[8:0]							
--								R/W							

位	标记	功能描述	复位值	读写
31:9	保留	--	0x0	--
8:0	RDSMPCNT[8:0]	主器件读采样时间设定计数值	0x0	R/W

### 26.5-81-Wire Recover Time 计数区间值(OWIRE\_RECCNT)

地址偏移: 0x01C 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								RECCNT[10:0]							
--								R/W							

位	标记	功能描述	复位值	读写
31:11	保留	--	0x0	--
10:0	RECCNT[10:0]	Recover Time 计数区间值	0x0	R/W

### 26.5-91-Wire 数据寄存器(OWIRE\_DATA)

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															
--															

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

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	--
7:0	DATA[7:0]	1bit 模式(Bit mode): 只能发送和接收bit0 8bit 模式(Byte mode): 能发送和接收全部8个bit	0x0	R/W

### 26.5-101-Wire 总线操作命令寄存器(OWIRE\_CMD)

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															
--															

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

位	标记	功能描述	复位值	读写
31:2	保留	--	0x0	--
1:0	CMD[1:0]	00 : 保留 01 : Initial 10 : TX 11 : RX	0x0	R/W

### 26.5-111-wire 中断使能寄存器(OWIRE\_INTEN)

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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												RX DONE EN	TX DONE EN	INIT DONE EN	ACK ERR EN
--												R/W			

位	标记	功能描述	复位值	读写
31:4	保留	--	0x0	--
3	RXDONEEN	接收完成中断使能 0: 禁止 1: 使能	0x0	R/W
2	TXDONEEN	发送完成中断使能 0: 禁止 1: 使能	0x0	R/W
1	INITDONEEN	初始化完成中断使能 0: 禁止 1: 使能	0x0	R/W
0	ACKERREN	从机应答错误中断使能 0: 禁止 1: 使能	0x0	R/W

### 26.5-121-wire 状态寄存器(OWIRE\_SR)

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															
--															

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

位	标记	功能描述	复位值	读写
31:4	保留	--	0x0	--
3	RXDONE	接收完成中断标志 0 : 未完成接收 1 : 接收完成	0x0	RO
2	TXDONE	发送完成中断标志 0 : 未完成发送 1 : 发送完成	0x0	RO
1	INITDONE	初始化完成中断标志 0 : 未完成初始化 1 : 初始化完成	0x0	RO
0	ACKERR	从机应答错误中断标志 0 : 未发生从机应答错误 1 : 发生从机应答错误	0x0	RO

### 26.5-131-wire 状态清除寄存器(OWIRE\_INTCLR)

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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												RX DONE CLR	TX DONE CLR	INIT DONE CLR	ACK ERR CLR
--												WO			

位	标记	功能描述	复位值	读写
31:4	保留	--	0x0	--
3	RXDONECLR	接收完成中断标志清除 0: 无作用 1: 清除接收完成中断	0x0	RO
2	TXDONECLR	发送完成中断标志清除 0: 无作用 1: 清除发送完成中断	0x0	RO
1	INITDONECLR	初始化完成中断标志清除 0: 无作用 1: 清除初始化完成中断	0x0	RO
0	ACKERRCLR	从机应答错误中断标志清除 0: 无作用 1: 清除从机应答错误中断	0x0	RO

## 【27】时钟校准/监测模块(CLKTRIM)

### 27.1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)。

### 27.2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

## 27.3 CLKTRIM 功能描述

### 27.3-1 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。

#### 27.3-1.1 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。

### 27.3-2CLKTRIM 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。

#### 27.3-2.1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。

## 27.4 CLKTRIM 寄存器列表

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 27- 1 CLKTRIM 寄存器列表和复位值

## 27.5CLKTRIM 寄存器说明

### 27.5-1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				CLKEN	IE	MON_EN	CALCLK_SEL[1:0]	REFCLK_SEL[2:0]	REFCLK_SEL[2:0]	TRIM_START					
--				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

### 27.5-2参考计数器初值配置寄存器(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 [31:0]	参考计数器初始值	0x0	R/W

### 27.5-3参考计数器值寄存器(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 [31:0]	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

### 27.5-4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 [31:0]	校准计数器值	0x0	RO

### 27.5-5 中断标志位寄存器(CLKTRIM\_IFR)

偏移地址: 0x10 复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															
HXT_FAULT LXT_FAULT CALCNT_OVF STOP															
-- RO															

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

### 27.5-6 中断标志位清除寄存器(CLKTRIM\_ICLR)

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										HXT_FAULT_CLR	LXT_FAULT_CLR	保留			
--										WO		--			

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	--

### 27.5-7 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
CALOVCNT[15:0]															
R/W															

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

## 【28】实时时钟(RTC)

### 28.1 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、月和年的信息。

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的任何误差，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。

### 28.2 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

## 28.3 RTC 功能描述

### 28.3-1 RTC 结构框图

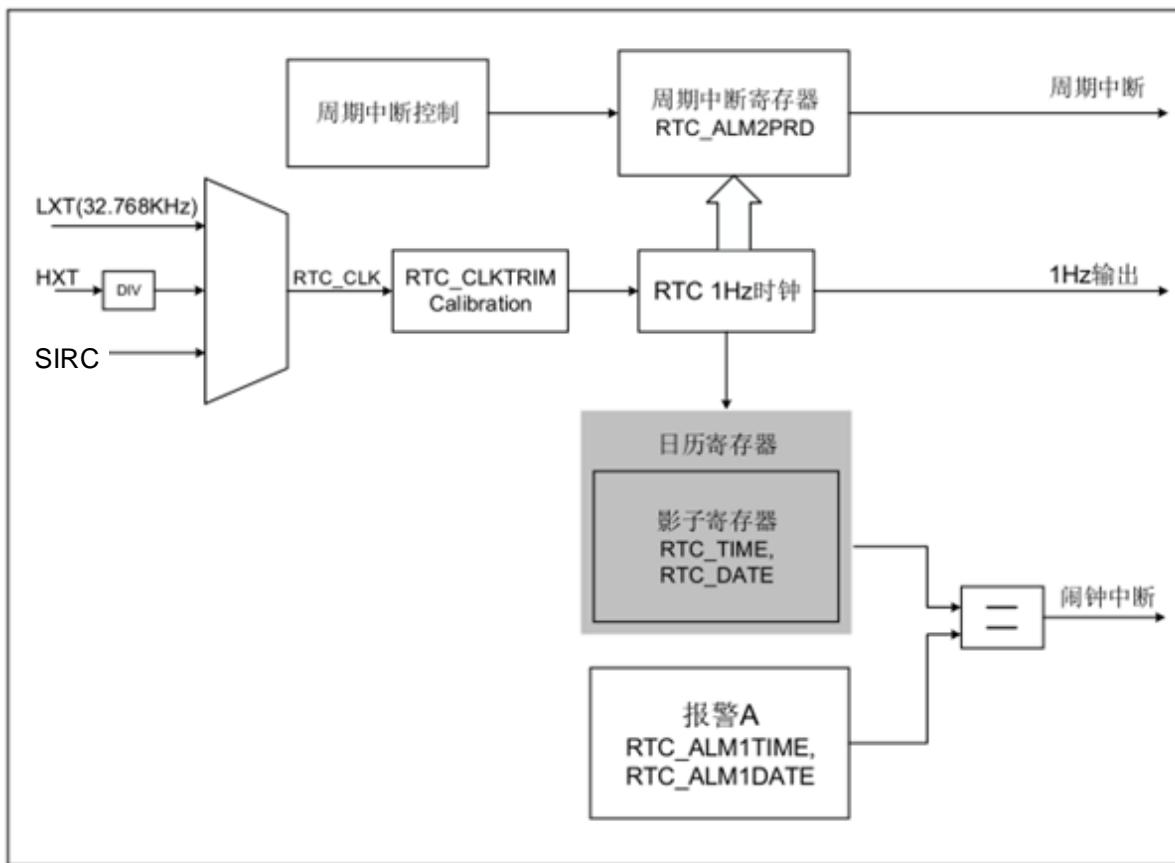


Figure 28- 1 RTC 框图

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

### 28.3-2RTC 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)

### 28.3-3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。

### 28.3-4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。

### 28.3.5 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。

### 28.3-6 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。

### 28.3-7 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**。

### 28.3-8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。

### 28.3-9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。

### 28.3-10RTC 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。

## 28.4 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。

### 28.4-1 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；

### 28.4-2 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；

## 28.5 RTC 寄存器列表

基址址: 0x4000 3000

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 28- 1 RTC 寄存器列表和复位值

## 28.6 RTC 寄存器说明

### 28.6-1 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						START	保留	ALM1 EN	ALM2_ INTEN	ALM1_ INTEN	保留	FMT	RTC 1HZOE	BYPSHAD	
--				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 1: 使能	0	R/W
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

## 28.6-2RTC 时钟控制寄存器(RTC\_CLKCR)

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										RTCCKEN	保留	RTCKSEL[1:0]			
--										R/W	--	R/W			

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

位	标记	功能描述	复位值	读写
31:21	保留	--	0x0	--
20	RTCCKEN	<p>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</p> <p>读: 0: RTC 时钟关闭 1: RTC 时钟开启</p>	0	R/W
19:18	保留	--	0x0	--
17:16	RTCKSEL [1:0]	<p>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。</p> <p>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</p>	0x0	R/W
15:10	保留	--	0x0	--
9:0	HXTDIV[9:0]	<p>外部高速晶振时钟分频 0: 停止 其它值: F=FHXT/(HXTDIV[9:0])</p>	0x0	R/W

## 28.6-3RTC 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									

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](12AM)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	--	0	--
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

## 28.6-4RTC 日期寄存器(RTC\_DATE)

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								YEAR[7:0]							
--								R/W							

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

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

### 28.6-5RTC 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									

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

### 28.6-6RTC 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	ALM YEAR EN	ALM MON EN	ALM DAY EN	ALM WEEK EN	ALM HOUR EN	ALM MIN EN	ALM SEC EN	ALYEAR[7:0]							
	--							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					

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

## 28.6-7RTC 周期闹钟寄存器( RTC\_ALM2PRD )

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										ALM2PR_CNT[3:0]					
--										R/W					

位	标记	功能描述	复位值	读写
31:4	保留	--	0x0	--
3:0	ALM2PR_CNT [3:0]	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

## 28.6-8RTC 时钟调校寄存器(RTC\_CLKTRIM)

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
保留						TRIM_MODE[1:0]		TRIM[7:0]							
--						R/W		R/W							

位	标记	功能描述	复位值	读写
31:10	保留	--	0x0	--
9:8	TRIM_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

## 28.6-9RTC 初始化和状态寄存器(RTC\_ISR)

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										ALM2_F	ALM1_F	保留	RSF	WAITF	WAIT
--										RO		--		R/W	

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

## 28.6-10RTC 状态清除寄存器(RTC\_INTCLR)

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
保留								ALM2_CLR	ALM1_CLR	保留					
--								WO				--			

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	--

### 28.6-11RTC 写保护寄存器(RTC\_WPR)

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
保留								WPR[7:0]							
--								WO							

位	标记	功能描述	复位值	读写
31:8	保留	--	0x0	--
7:0	WPR[7:0]	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

## 【29】模拟/数字转换器(ADC)

### 29.1 模块简介

ZB32L032 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
- 22 路转换通道：
  - 17 个引脚通道、1 个 VCAP, 1 个 V12 校准通道, 1 个 DACO 通道, 2 个 OPA 输出通道.
- 四个参考电压(Reference Voltage VREF):
  - (1) 电源电压(AVDD), (2)VCAP(2.5V), (3)PB01(外部参考电压), (4)Vref(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

### 29.2 ADC 框图

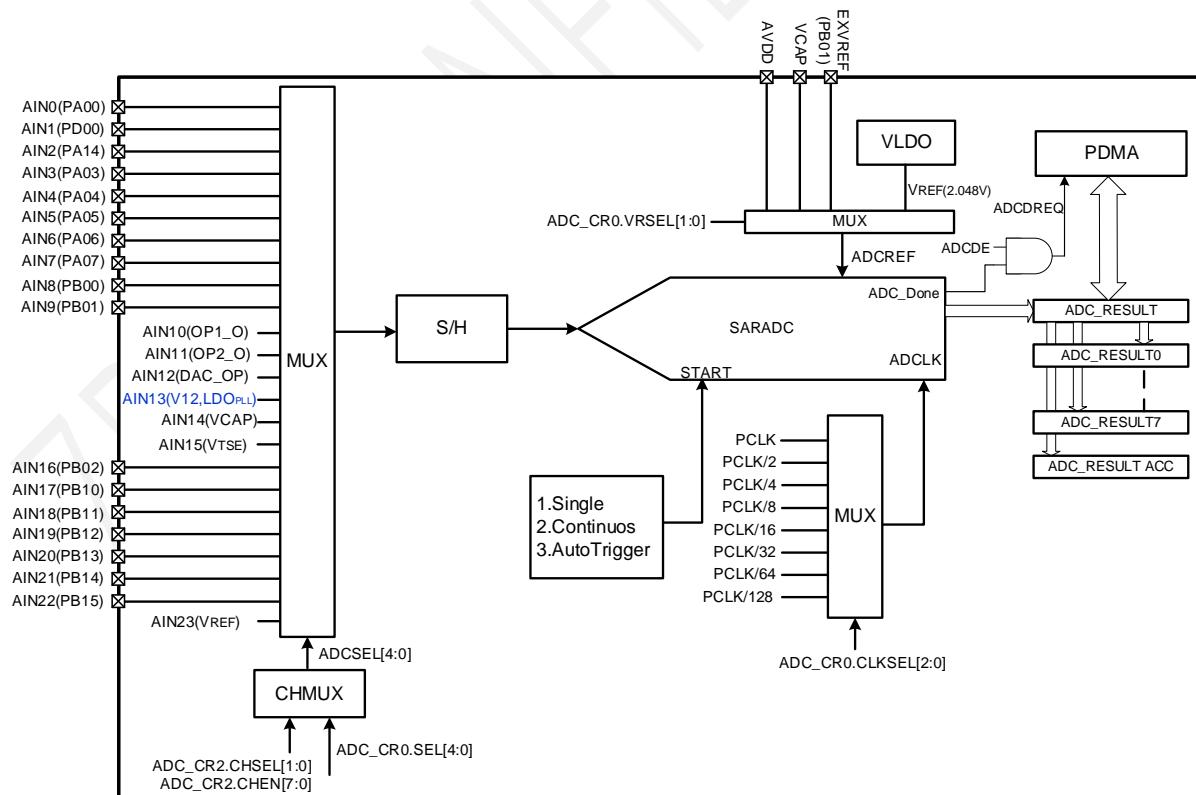
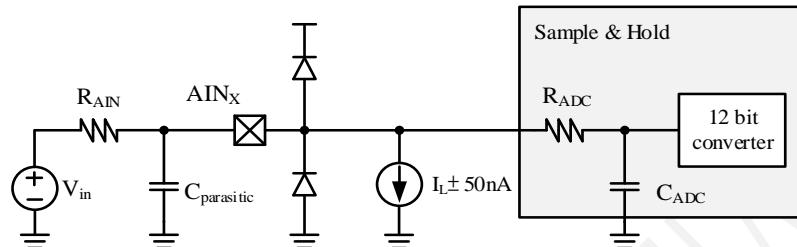


Figure 29- 1 ADC 架构方块图

## 29.2-1ADC 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.  $C_{\text{parasitic}}$  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(B)中最大  $R_{\text{AIN}}$  值为参考 ADC 规格表(见 Datasheet.ADC 特性)中  $C_{\text{ADC}}$  与  $R_{\text{ADC}}$  和图 A 所得。

表(B).  $R_{\text{AIN}}$  对应  $f_{\text{ADCCLK}}$

$t_s(\mu\text{s})$	$f_{\text{ADCCLK}}(\text{Hz})$	SAM	$R_{\text{AIN}}(\text{k}\Omega)$
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

### 29.3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 ADC\_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。

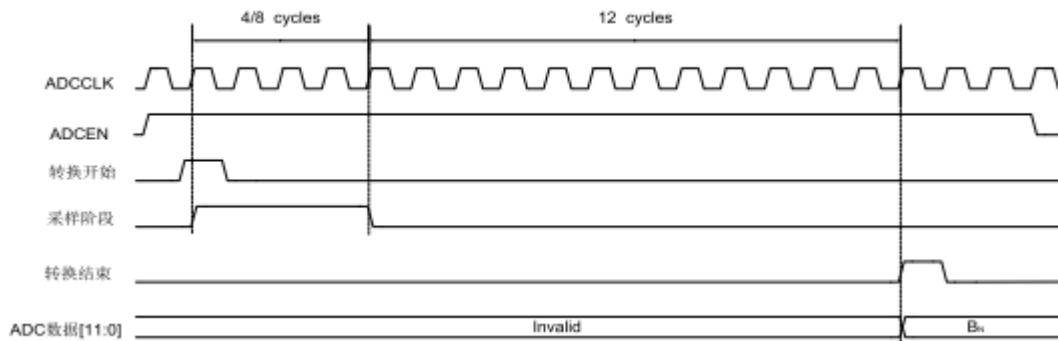


Figure 29- 2 ADC 转换时序图

### 29.4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 ADC\_CR0.START 位启动也可通过设置 ADC\_CR1[9:0]的外部触发启动。一旦选定通道的 ADC 转换完成，ADC\_CR0.START 位自动清零，转换结果保存在 ADC\_RESULT 寄存器中。

通过 START 位启动 ADC 单次转换操作流程：

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 ADC\_CR2.CIRCLE\_MODE 为 0，选择非循环模式。

配置 ADC\_CR1.CT 为 0，选择单次转换模式。

配置 ADC\_CR0.SAM 及 ADC\_CR0.CLKSEL，设置 ADC 的转换速度。

配置 ADC\_CR0.SEL，选择待转换的通道(注意需要和步骤 1 保持一致)。

配置 ADC\_CR0.ADCEN 为 1，使能 ADC 模块。

配置 ADC\_CR0.START 为 1，启动 ADC 单次转换。

等待 ADC\_CR0.START 为 0，读取 ADC\_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 ADC\_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。

## 29.5 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 ADC\_CR2.ADCCNT[7:0]进行配置；待进行转换的通道由 ADC\_CR2.CHEN[7:0], ADC\_CR2.CHSEL[1:0] 进行配置,如下表所示。

CHSEL[1:0] CHEN[7:0]	2' b00	2' b01	2' b10	ADC Convert Result
8' b0000_0001	AIN0	AIN8	AIN16	ADC_Result0
8' b0000_0010	AIN1	AIN9	AIN17	ADC_Result1
8' b0000_0100	AIN2	AIN10	AIN18	ADC_Result2
8' b0000_1000	AIN3	AIN11	AIN19	ADC_Result3
8' b0001_0000	AIN4	AIN12	AIN20	ADC_Result4
8' b0010_0000	AIN5	AIN13	AIN21	ADC_Result5
8' b0100_0000	AIN6	AIN14	AIN22	ADC_Result6
8' b1000_0000	AIN7	AIN15	AIN20	ADC_Result7

该模式既可通过设置 ADC\_CR0.START 位启动，也可通过设置 ADC\_CR1[9:0]的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC\_RAWINTSR.CONT\_INTF 位会自动置 1，转换结果保存在转换通道所对应的 ADC\_RESULT0~ADC\_RESULT7 寄存器中。如果总转换次数大于待转换的 ADC 通道的数量，则 ADC\_RESULT0~ADC\_RESULT7 寄存器中只保存最后一次的转换结果。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换的过程。通过寄存器将 START 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。

例如：ADC\_CR2.CHSEL[1:0] = 2' b00 :

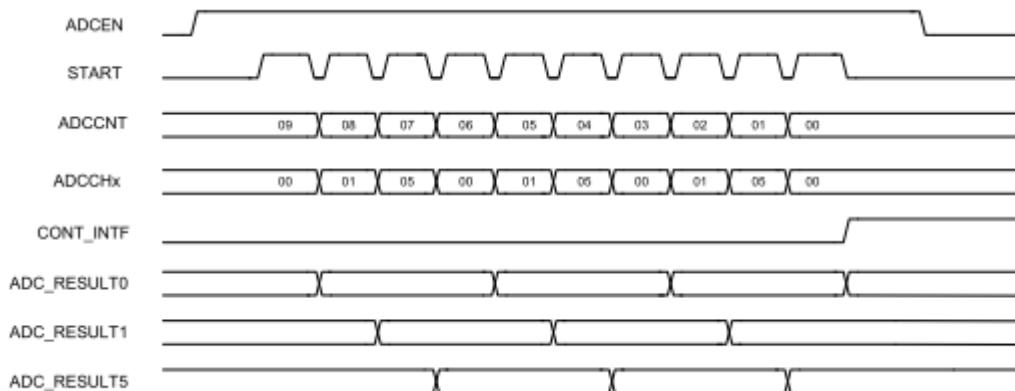


Figure 29- 3 ADC 连续转换过程示例

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 ADC\_CR2.circle\_mode 为 0，选择非循环模式。
3. 配置 ADC\_CR1.ct 为 1，选择连续转换模式。
4. 配置 ADC\_CR2.adccnt[7:0]，选择连续转换的总转换次数。
5. 配置 ADC\_CR0.sam 及 ADC\_CR0.clksel，设置 ADC 的转换速度。
6. 配置 ADC\_CR2.chen[7:0]，使能待转换的通道。
7. 配置 ADC\_INTCLR.cont\_intc 为 1，清除 ADC\_RAWINTSR.cont\_intf 标志。
8. 配置 ADC\_INTEN.CONT\_IEN 为 1，打开连续转换完成中断掩码使能。
9. 配置 ADC\_CR0.staterst 为 1，复位连续转换状态。
10. 配置 ADC\_CR0.adcen 为 1，使能 ADC 模块。
11. 配置 ADC\_CR0.start 为 1，启动 ADC 连续转换。
12. 等待 ADC\_RAWINTSR.cont\_intf 变为 1，读取 ADC\_result0~ADC\_result7 寄存器以获取相应通道的转换结果。
13. 如需对其他通道进行转换，重复执行步骤 6~11。
14. 配置 ADC\_CR0.adcen，关闭 ADC 模块。

## 29.6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 ADC\_CR2.ADCCNT[7:0]进行配置；待进行转换的通道由 ADC\_CR2.CHEN[7:0]进行配置。该模式既可通过设置 ADC\_CR0.START 位启动，也可通过设置 ADC\_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，ADC\_RAWINTSR.CONT\_INTF 位会自动置 1，转换结果的累加值保存在 ADC\_RESULT\_ACC 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。通过寄存器将 START 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。每次转换完成时，ADC\_RESULT\_ACC 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

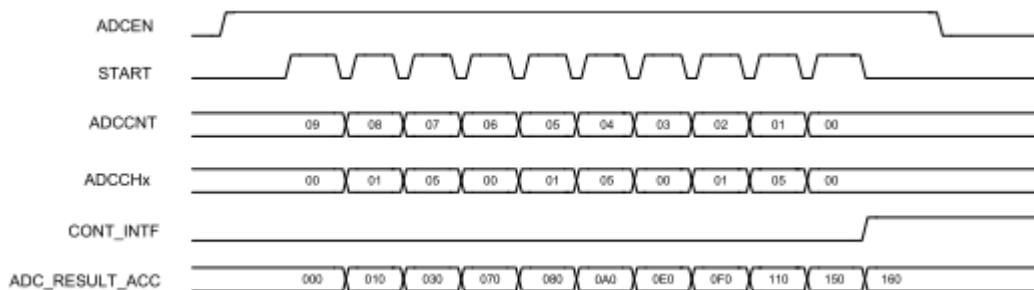


Figure 29- 4 连续转换累加过程示例

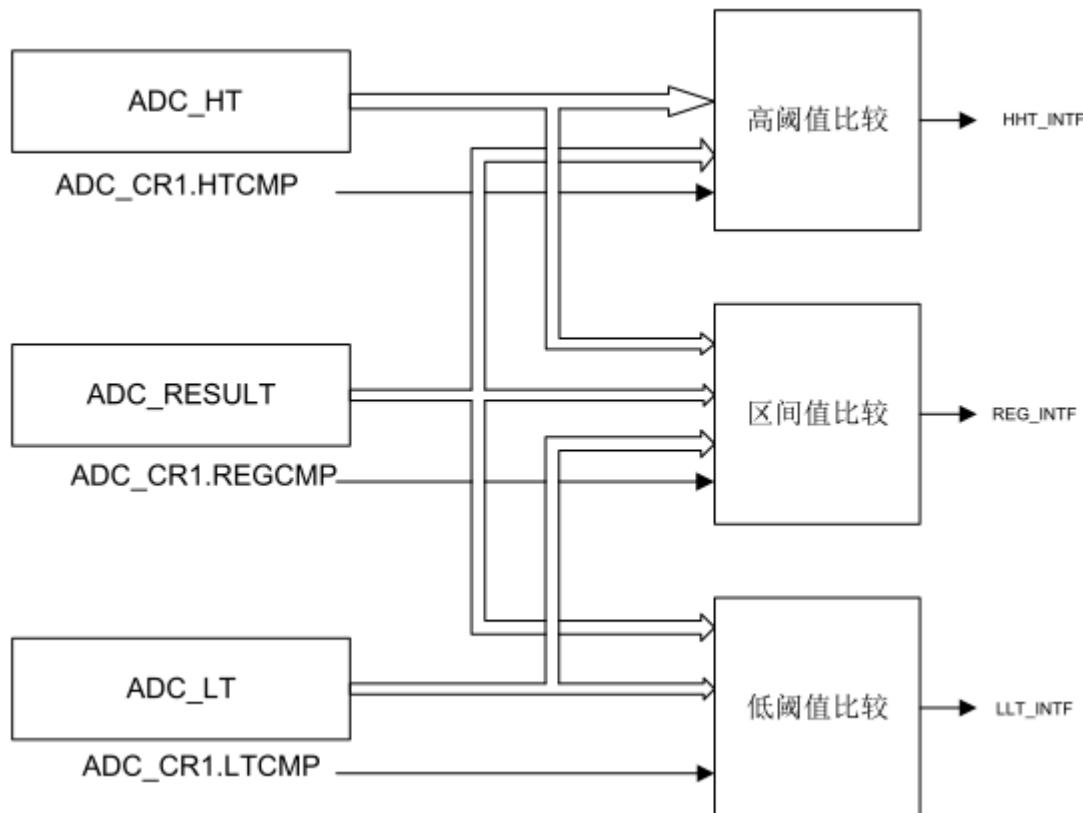
配置步骤：

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 ADC\_CR2.circle\_mode 为 0，选择非循环模式。
3. 配置 ADC\_CR1.ct 为 1，选择连续转换模式。
4. 配置 ADC\_CR1.racc\_en 为 1，选择 ADC 转换自动累加功能。
5. 配置 ADC\_CR2.adccnt[7:0]，选择连续转换的总转换次数。
6. 配置 ADC\_CR0.sam 及 ADC\_CR0.clksel，设置 ADC 的转换速度。
7. 配置 ADC\_CR2.chen[7:0]，选择待转换的通道。
8. 配置 ADC\_INTCLR.cont\_intc 为 1，清除 ADC\_RAWINTSR.cont\_intf 标志。
9. 配置 ADC\_INTEN.CONT\_IEN 为 1，打开连续转换完成中断掩码使能。
10. 设置 ADC\_CR1.racc\_clr 为 1，清除 ADC\_result\_acc 寄存器。
11. 配置 ADC\_CR0.staterst 为 1，复位连续转换状态。
12. 配置 ADC\_CR0.adcen 为 1，使能 ADC 模块。
13. 配置 ADC\_CR0.start 为 1，启动 ADC 连续转换。
14. 等待 ADC\_RAWINTSR.cont\_intf 变为 1，读取 ADC\_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 ADC\_CR0.adcen，关闭 ADC 模块。

## 29.7 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 ADC\_CR1.HTCMP、ADC\_CR1.LTCMP、ADC\_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介入。

- 上阈值比较：当 ADC 转换结果位于[ADC\_LT, ADC\_HT]区间内，则 ADC\_RAWINTSR.HHT\_INTF 置 1；向 ADC\_INTCLR.HHT\_INTC 写入 1 则清零 ADC\_RAWINTSR.HHT\_INTF。
- 下阈值比较：当 ADC 转换结果位于[0, ADC\_LT)区间内，则 ADC\_RAWINTSR.LLT\_INTF 置 1；向 ADC\_INTCLR.LLT\_INTC 写入 1 则清零 ADC\_RAWINTSR.LLT\_INTF。
- 区间值比较：当 ADC 转换结果位于[ADC\_LT, ADC\_HT]区间内，则 ADC\_RAWINTSR.REG\_INTF 置 1；向 ADC\_INTCLR.REG\_INTC 写入 1 则清零 ADC\_RAWINTSR.REG\_INTF。



## 29.8 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	ADC_MSKINTSR.CONT_MIF	ADC_INTEN.CONT_IEN
ADC 转换结果位于区间值区域	ADC_MSKINTSR.REG_MIF	ADC_INTEN.REG_IEN
ADC 转换结果位于上阈值区域	ADC_MSKINTSR.HHT_MIF	ADC_INTEN.HHT_IEN
ADC 转换结果比较下阈值区域	ADC_MSKINTSR.LLT_MIF	ADC_INTEN.LLT_IEN

## 29.9 ADC 寄存器列表

基址地址：0x4000 2C00

偏移地址	名称	描述	复位值
0x00	ADC_CR0	ADC 配置寄存器 0	0x0000 0000
0x04	ADC_CR1	ADC 配置寄存器 1	0x0000 7000
0x08	ADC_CR2	ADC 配置寄存器 2	0x0000 0000
0x0C	ADC_RESULT0	ADC 通道 0 转换结果	0x0000 0000
0x10	ADC_RESULT1	ADC 通道 1 转换结果	0x0000 0000
0x14	ADC_RESULT2	ADC 通道 2 转换结果	0x0000 0000
0x18	ADC_RESULT3	ADC 通道 3 转换结果	0x0000 0000
0x1C	ADC_RESULT4	ADC 通道 4 转换结果	0x0000 0000
0x20	ADC_RESULT5	ADC 通道 5 转换结果	0x0000 0000
0x24	ADC_RESULT6	ADC 通道 6 转换结果	0x0000 0000
0x28	ADC_RESULT7	ADC 通道 7 转换结果	0x0000 0000
0x2C	ADC_RESULT	ADC 转换结果	0x0000 0000
0x30	ADC_RESULT_ACC	ADC 转换结果累加值	0x0000 0000
0x34	ADC_HT	ADC 比较上阈值	0x0000 0FFF
0x38	ADC_LT	ADC 比较下阈值	0x0000 0000
0x44	ADC_INTEN	ADC 中断使能寄存器	0x0000 0000
0x48	ADC_INTCLR	ADC 中断清除寄存器	0x0000 0000
0x4C	ADC_RAWINTSR	ADC 掩码前中断状态寄存器	0x0000 0000
0x50	ADC_MSKINTSR	ADC 掩码后中断状态寄存器	0x0000 0000
0x54	ADCVTSETRIM	ADC/VTSE TRIM寄存器	0x0000 030F

Table 29- 1 ADC 寄存器列表和复位值

## 29.10 寄存器说明

### 29.10-1 ADC 配置寄存器 0(ADC\_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]		
--									R/W		RO	R/W			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATERST	SAM	VREFEN	CLKDIV3	SEL[4:0]			CLKSEL[2:0]			VRSEL[1:0]	START	ADCEN			
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 为 PLL LDO 1.2V 0: Ain13选择通道为V12(CPU) LDO 1.2V	0	R/W
21	VTSEN	VTSE使能(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
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

位	标记	功能描述	复位值	读写
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 01010: 选择通道 AIN10(OP1_O) 01011: 选择通道 AIN11(OP2_O) 01100: 选择通道 AIN12(DAC_OP) 01101: 选择通道 AIN 13(SEL12V - 0: V12; 1 : PLL LDO) 01110: 选择通道 AIN 14(VCAP) 01111: 选择通道 AIN 15(VTSE) 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(VREF)	0x0	R/W

位	标记	功能描述	复位值	读写
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(2.048V) 10 : EX_VREF PB1 11 : VCAP(2.5V)	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

### 29.10-2ADC 配置寄存器 1(ADC\_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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RACC_CLR	REGCMP	HTCMP	LTCMP	RACC_EN	CT	TRIGS1[4:0]							TRIGS0[4:0]		
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	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(ADC_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

位	标记	功能描述	复位值	读写
9:5	TRIGS1[4:0]	<p>ADC 转换自动触发选择 1:</p> <p>00000: 禁用自动触发 ADC 转换            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: USART0 中断, 自动触发ADC 转换            01001: USART1 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。如果不需要进入中断服务程序, 请不要使能 NVIC 的中断使能</p>	0x0	R/W

位	标记	功能描述	复位值	读写
4: 0	TRIGS0[4:0]	<p>ADC 转换自动触发选择 0:</p> <p>00000: 禁用自动触发 ADC 转换            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: USART0 中断, 自动触发ADC 转换            01001: USART1 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。如果不需要进入中断服务程序, 请不要使能 NVIC 的中断使能。</p>	0x0	R/W

### 29.10-3ADC 配置寄存器 2(ADC\_CR2)

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												CHSEL[1:0]	CIRCLE_MODE			
--												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

### 29.10-4ADC 通道 0 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT0															
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

### 29.10-5ADC 通道 1 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT1															
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]	ADC 通道 1/9/17 转换结果	0x0	RO

### 29.10-6ADC 通道 2 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT2															
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]	ADC 通道 2/10/18 转换结果	0x0	RO

### 29.10-7ADC 通道 3 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT3															
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]	ADC 通道 3/11/19 转换结果	0x0	RO

### 29.10-8ADC 通道 4 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT4															
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]	ADC 通道 4/12/20 转换结果	0x0	RO

### 29.10-9ADC 通道 5 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT5															
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]	ADC 通道 5/13/21 转换结果	0x0	RO

### 29.10-10ADC 通道 6 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT6															
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]	ADC 通道 6/14/22 转换结果	0x0	RO

### 29.10-11ADC 通道 7 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT7															
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]	ADC 通道 7/15/23 转换结果	0x0	RO

### 29.10-12ADC 转换结果(ADC\_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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESULT															
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]	ADC 转换结果	0x0	RO

### 29.10-13ADC 转换结果累加值(ADC\_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]			
--												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]	ADC 转换结果累加值	0x0	RO

### 29.10-14ADC 比较上阈值(ADC\_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															
--															

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]	ADC 转换结果比较上阈值	0xFFFF	R/W

### 29.10-15ADC 比较下阈值(ADC\_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															
--															

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]	ADC 转换结果比较下阈值	0x0	R/W

## 29.10-16ADC 中断使能寄存器(ADC\_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
ADCXIEN[23:16]								ADCXIEN[15:8]							
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	ADCXIEN[7:0]							
R/W															

位	标记	功能描述	复位值	读写
31:24	ADCXIEN [23:16]	ADC 通道 23~16 中断掩码配置(AIN23~AIN16) 0: 禁止 1: 使能	0x0	R/W
23:16	ADCXIEN [15:8]	ADC 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	--
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	ADC 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	ADC 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	ADC 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	ADCXIEN [7:0]	ADC 通道 7~0 中断掩码配置(AIN7~AIN0) 0: 禁止 1: 使能	0x0	R/W

### 29.10-17ADC 中断清除寄存器(ADC\_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
ADCICLR[23:16]								ADCICLR[15:8]							
WO															

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	ADCICLR[7:0]							
--		WO													

位	标记	功能描述	复位值	读写
31:24	ADCICLR[23:16]	写 1 清除ADC 通道 23~16 中断状态 写 0 无作用	0x0	WO
23:16	ADCICLR[15:8]	写 1 清除ADC 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ADC 转换结果比较区间标志 写 0 无作用	0x0	WO
9	HHT_INTC	写 1 清除ADC 转换结果比较上阈值 写 0 无作用	0x0	WO
8	LLT_INTC	写 1 清除ADC 转换结果比较下阈值标志 写 0 无作用	0x0	WO
7:0	ADCICLR[7:0]	写 1 清除ADC 通道 7~0 中断状态 写 0 无作用	0x0	WO

### 29.10-18ADC 掩码前中断状态寄存器(ADC\_RAWINTSR)

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	16
ADCRIS[23:16]								ADCRIS[15:8]							
RO															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CONT_INTF	REG_INTF	HHT_INTF	LLT_INTF	ADCRIS[7:0]							
RO															

位	标记	功能描述	复位值	读写
31:24	ADCRIS[23:16]	ADC 通道 23~16转换完成中断状态（掩码前）	0x0	RO
23:16	ADCRIS[15:8]	ADC 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: ADC 连续转换未完成 1: ADC 连续转换完成	0x0	RO
10	REG_INTF	ADC 转换结果比较区间标志 0: ADC 转换结果位于[ADC_LT, ADC_HT]区间外 1: ADC 转换结果位于[ADC_LT, ADC_HT]区间内	0x0	RO
9	HHT_INTF	ADC 转换结果比较上阈值标志 0: ADC 转换结果位于[ADC_HT, 4095]区间外 1: ADC 转换结果位于[ADC_HT, 4095]区间内	0x0	RO
8	LLT_INTF	ADC 转换结果比较下阈值标志 0: ADC 转换结果位于[0, ADC_LT]区间外 1: ADC 转换结果位于[0, ADC_LT]区间内	0x0	RO
7:0	ADCRIS[7:0]	ADC 通道 7~0 转换完成中断状态（掩码前）	0x0	RO

### 29.10-19ADC 掩码后中断状态寄存器(ADC\_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
ADCMIS[23:16]								ADCMIS[15:8]							
RO															

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	ADCMIS[7:0]							
--		RO													

位	标记	功能描述	复位值	读写
31:24	ADCMIS[23:16]	ADC 通道 23~16转换完成中断状态（掩码后）	0x0	RO
23:16	ADCMIS[15:8]	ADC 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: ADC 连续转换未完成 1: ADC 连续转换完成	0x0	RO
10	REG_MIF	ADC 转换结果比较区间掩码后中断 0: ADC 转换结果位于[ADC_LT, ADC_HT]区间外 1: ADC 转换结果位于[ADC_LT, ADC_HT]区间内	0x0	RO
9	HHT_MIF	ADC 转换结果比较上阈值掩码后中断 0: ADC 转换结果位于[ADC_HT, 4095]区间外 1: ADC 转换结果位于[ADC_HT, 4095]区间内	0x0	RO
8	LLT_MIF	ADC 转换结果比较下阈值掩码后中断 0: ADC 转换结果位于[0, ADC_LT]区间外 1: ADC 转换结果位于[0, ADC_LT]区间内	0x0	RO
7:0	ADCMIS[7:0]	ADC 通道 7~0 转换完成中断状态（掩码后）	0x0	RO

### 29.10-20ADC/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					VTSETRIM[2:0]			保留			ADCTRIM[4:0]				
--					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
15:11	保留	--	0x0	--
10:8	VTSETRIM [2:0]	内部TSE 调整: ● VTSE参考电压 校准值地址: 0x1800_008E 对于裸片(KGD): ● VTSE参考电压 校准值地址: 0x1800_008C	0b011	R/W
7:5	保留	--	0x0	--
4:0	ADCTRIM [4:0]	内部ADC Vref=2.048V 调整: ● ADC参考电压 校准值地址: 0x1800_0096 对于裸片(KGD): ● ADC参考电压 校准值地址: 0x1800_0094	0b01111	R/W

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

## 【30】低电压检测器(LVD)

### 30.1 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。

### 30.2 LVD 框图

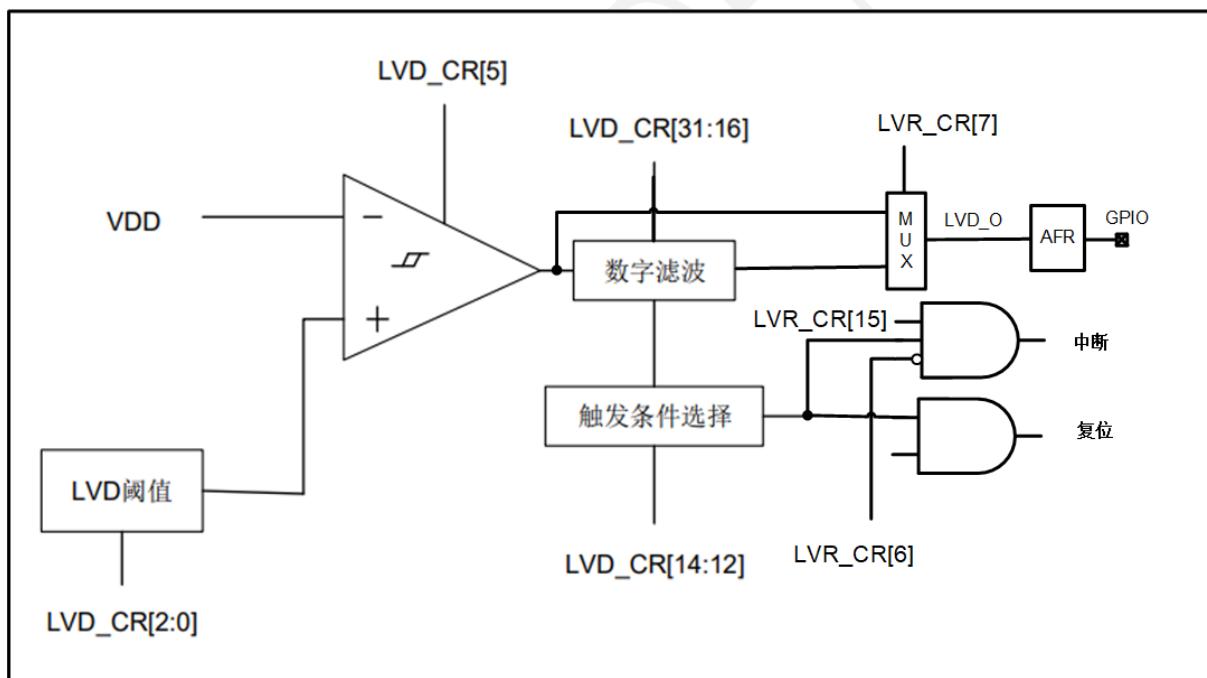


Figure 30 - 1 LVD 结构框图

### 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 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：

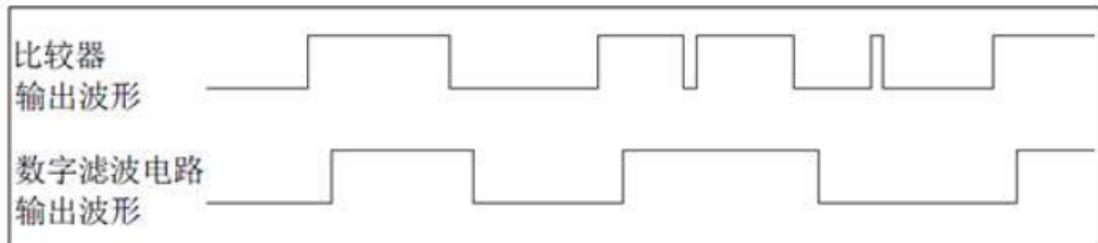


Figure 30 - 2 LVD 滤波输出

### 30.4 配置示例

#### 30.4-1 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。

#### 30.4-2 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。

### 30.5 LVD 寄存器列表

基址址: 0x4000 4000

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 30- 1 LVD 寄存器列表

### 30.6 寄存器说明

#### 30.6-1 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	HIGH	RISE	FALL	保留	FLTCLK_SEL[1:0]	FLTEN	ACT	LVDEN	保留	DIV_SEL[2:0]						
	INTEN	INTEN	INTEN			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	--
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

位	标记	功能描述	复位值	读写
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: 4.4V 001: 4.0V 010: 3.6V 011: 3.3V 100: 3.1V 101: 2.9V 110: 2.7V 111: 2.5V	0x7	R/W

### 30.6-2LVD 状态寄存器(LVD\_SR)

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															
--															

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	R/W

## 【31】电压比较器(VC)

### 31.1 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。

### 31.2 VC 框图

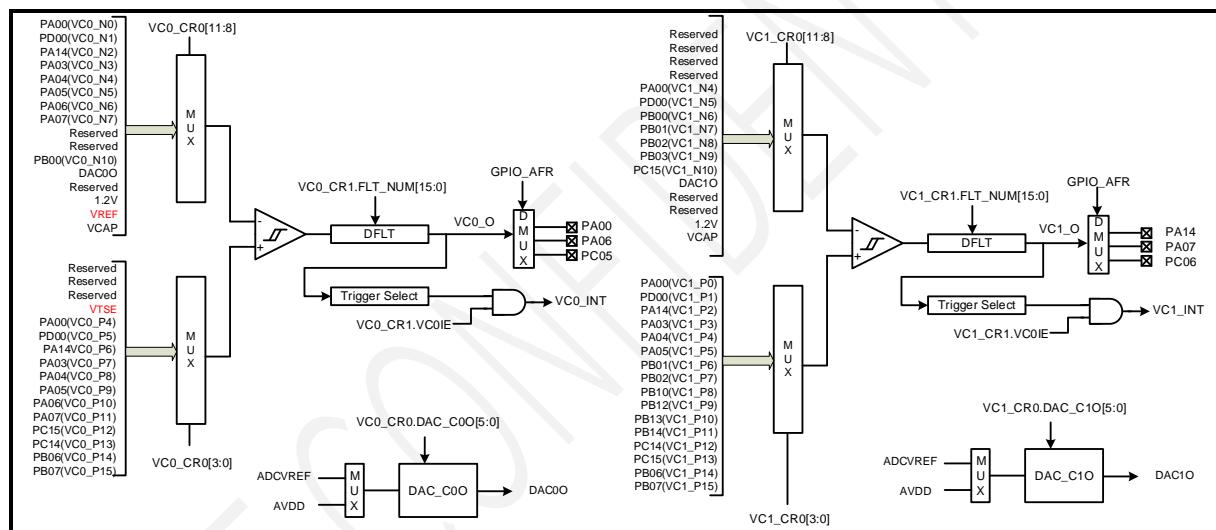


Figure 31 - 1 VC0,VC1 结构框图

### 31.3 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：

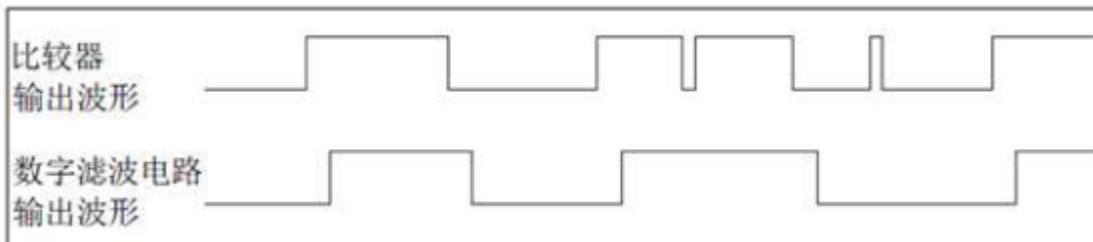


Figure 31 - 2 VC 滤波输出

### 31.4 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。

### 31.5VC 寄存器列表

基址址: 0x4000 4000

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 31- 1 VC 寄存器列表和复位值

### 31.6VC 寄存器说明

#### 31.6-1VC0 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										VCDAC_EN	DAC_C0O[5:0]				
--										R/W	R/W				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DACVREF	VN0INSEL[3:0]				保留				VPOINSEL[3:0]				
--		R/W	R/W				--				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 输出电压选择: 6' h0 : DAC_C0O = 0 6' h1 : DAC_C0O = (AVDD/VCAP)*1/64 6' h2 : DAC_C0O = (AVDD/VCAP)*2/64 ... ... ... 6' h63 : DAC_C0O = (AVDD/VCAP)*63/64	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

位	标记	功能描述	复位值	读写																																		
11:8	VN0INSEL[3:0]	VC0 “-” 输入端电压电压选择: <table border="1"> <thead> <tr> <th>VN0INSEL[3:0]</th><th>VN0 input</th></tr> </thead> <tbody> <tr><td>4' b0000</td><td>PA00(VC0_N0)</td></tr> <tr><td>4' b0001</td><td>PD00(VC0_N1)</td></tr> <tr><td>4' b0010</td><td>PA14(VC0_N2)</td></tr> <tr><td>4' b0011</td><td>PA03(VC0_N3)</td></tr> <tr><td>4' b0100</td><td>PA04(VC0_N4)</td></tr> <tr><td>4' b0101</td><td>PA05(VC0_N5)</td></tr> <tr><td>4' b0110</td><td>PA06(VC0_N6)</td></tr> <tr><td>4' b0111</td><td>PA07(VC0_N7)</td></tr> <tr><td>4' b1000</td><td>保留</td></tr> <tr><td>4' b1001</td><td>保留</td></tr> <tr><td>4' b1010</td><td>PB00(VC0_N10)</td></tr> <tr><td>4' b1011</td><td>DAC0O</td></tr> <tr><td>4' b1100</td><td>保留</td></tr> <tr><td>4' b1101</td><td>1.2V</td></tr> <tr><td>4' b1110</td><td>ADCVREF</td></tr> <tr><td>4' b1111</td><td>VCAP</td></tr> </tbody> </table>	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	保留	4' b1001	保留	4' b1010	PB00(VC0_N10)	4' b1011	DAC0O	4' b1100	保留	4' b1101	1.2V	4' b1110	ADCVREF	4' b1111	VCAP	0x0	R/W
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	保留																																					
4' b1001	保留																																					
4' b1010	PB00(VC0_N10)																																					
4' b1011	DAC0O																																					
4' b1100	保留																																					
4' b1101	1.2V																																					
4' b1110	ADCVREF																																					
4' b1111	VCAP																																					
7:4	--	保留	0x0	--																																		
3:0	VPOINSEL[3:0]	VC0 “+” 输入端电压电压选择: <table border="1"> <thead> <tr> <th>VPOINSEL[3:0]</th><th>VPOinput</th></tr> </thead> <tbody> <tr><td>4' b0000</td><td>保留</td></tr> <tr><td>4' b0001</td><td>保留</td></tr> <tr><td>4' b0010</td><td>保留</td></tr> <tr><td>4' b0011</td><td>保留</td></tr> <tr><td>4' b0100</td><td>PA00(VC0_P4)</td></tr> <tr><td>4' b0101</td><td>PD00(VC0_P5)</td></tr> <tr><td>4' b0110</td><td>PA14(VC0_P6)</td></tr> <tr><td>4' b0111</td><td>PA03(VC0_P7)</td></tr> <tr><td>4' b1000</td><td>PA04(VC0_P8)</td></tr> <tr><td>4' b1001</td><td>PA05(VC0_P9)</td></tr> <tr><td>4' b1010</td><td>PA06(VC0_P10)</td></tr> <tr><td>4' b1011</td><td>PA07(VC0_P11)</td></tr> <tr><td>4' b1100</td><td>PC15(VC0_P12)</td></tr> <tr><td>4' b1101</td><td>PC14(VC0_P13)</td></tr> <tr><td>4' b1110</td><td>PB06(VC0_P14)</td></tr> <tr><td>4' b1111</td><td>PB07(VC0_P15)</td></tr> </tbody> </table>	VPOINSEL[3:0]	VPOinput	4' b0000	保留	4' b0001	保留	4' b0010	保留	4' b0011	保留	4' b0100	PA00(VC0_P4)	4' b0101	PD00(VC0_P5)	4' b0110	PA14(VC0_P6)	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)	0x0	R/W
VPOINSEL[3:0]	VPOinput																																					
4' b0000	保留																																					
4' b0001	保留																																					
4' b0010	保留																																					
4' b0011	保留																																					
4' b0100	PA00(VC0_P4)																																					
4' b0101	PD00(VC0_P5)																																					
4' b0110	PA14(VC0_P6)																																					
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)																																					

### 31.6-2VC0 控制寄存器(VC0\_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
VC0FLT_NUM[15:0]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC0 INT _EN	VC0 HIGH INTEN	VC0 RISE INTEN	VC0 FALL INTEN	保留		VC0 FLTEN	保留		VC0_FLTCLK _SEL[1:0]	VC0 HY_EN		VC0 EN			
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	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	VC0FLTEN	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	R/W
0	VC0EN	VC0 Enable 0: 电压比较功能禁止 1: 电压比较功能使能	0	R/W

### 31.6-3VC0 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
保留												INV_PAD	TM1BKE	TM1CH4_EN	
--												R/W	R/W	R/W	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INV_TM1CH4_EN	TM1CH3	INV_TM1CH3	TM1CH2	INV_TM1CH2	TM1CH1	INV_TM1CH1	PCA_ECI	PCA_CAP0	INV_PCA	LPTIM_EXT	LPTIM_EN	保留	TIM1_EN	TIM0_EN	INV_TIMX
R/W														--	R/W

位	标记	功能描述	复位值	读写
31:19	保留	--	0x0	--
18	INV_PAD	VC0 filter 结果输出到PAD 反向使能 0: 禁止 1: 使能	0	R/W
17	TM1BKE	VC0 中断作为TIM1 刹车控制 0: 禁止 1: 使能	0	R/W
16	TM1CH4_EN	VC0 filter 结果输出到TIM1 CH4 门控使能 0: 禁止 1: 使能	0	R/W
15	INV_TM1CH4	VC0 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	VC0 filter 结果输出到TIM1 CH3 门控使能 0: 禁止 1: 使能	0	R/W
13	INV_TM1CH3	VC0 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	VC0 filter 结果输出到TIM1 CH2 门控使能 0: 禁止 1: 使能	0	R/W

位	标记	功能描述	复位值	读写
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
10	TM1CH1_EN	VCO filter 结果输出到TIM1 CH1 门控使能 0: 禁止 1: 使能	0	R/W
9	INV_TM1CH1	VCO filter 结果输出到TIM1 CH1 门控反向使能 0: 禁止 1: 使能	0	R/W
8	PCAECL_EN	VCO 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	VCO filter 结果输出到PCA 捕获 0 使能 0: 禁止 1: 使能	0	R/W
6	INV_PCA	VCO filter 结果输出到PCA 反向使能 0: 禁止 1: 使能	0	R/W
5	LPTIMEXT_EN	VCO 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	VCO 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	VCO filter 结果输出到TIM11 门控使能 0: 禁止 1: 使能	0	R/W
1	TIM0_EN	VCO filter 结果输出到TIM10 门控使能 0: 禁止 1: 使能	0	R/W
0	INV_TIMX	VCO filter 结果输出到TIM10, TIM11, LPTIM 门控反向使能 0: 禁止 1: 使能	0	R/W

### 31.6-4VC0 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															
--															

位	标记	功能描述	复位值	读写
31:2	保留	--	0x0	--
1	VC0_FLOUT	VC0 滤波后状态 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 VC0 中断 1: 发生 VC0 中断 写 0 清除中断标志, 写 1 无效	0	WOC

### 31.6-5VC1 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								DAC_C1O[5:0]							
--								R/W							

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

位	标记	功能描述	复位值	读写																																		
31:22	保留	--	0x0	--																																		
21:16	DAC_C1O[5:0]	DAC_C1O 输出电压选择: 6' h0 : DAC_C1O = 0 6' h1 : DAC_C1O = (AVDD/VCAP)*1/64 6' h2 : DAC_C1O = (AVDD/VCAP)*2/64 ... ... ... 6' h63 : DAC_C1O = (AVDD/VCAP)*63/64	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> <th>VN1INSEL[3:0]</th> <th>VN1 input</th> </tr> <tr> <td>4' b0000</td> <td>保留</td> </tr> <tr> <td>4' b0001</td> <td>保留</td> </tr> <tr> <td>4' b0010</td> <td>保留</td> </tr> <tr> <td>4' b0011</td> <td>保留</td> </tr> <tr> <td>4' b0100</td> <td>PA00(VC1_N4)</td> </tr> <tr> <td>4' b0101</td> <td>PD00(VC1_N5)</td> </tr> <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>保留</td> </tr> <tr> <td>4' b1101</td> <td>保留</td> </tr> <tr> <td>4' b1110</td> <td>1.2V</td> </tr> <tr> <td>4' b1111</td> <td>VCAP</td> </tr> </table>	VN1INSEL[3:0]	VN1 input	4' b0000	保留	4' b0001	保留	4' b0010	保留	4' b0011	保留	4' b0100	PA00(VC1_N4)	4' b0101	PD00(VC1_N5)	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	保留	4' b1101	保留	4' b1110	1.2V	4' b1111	VCAP	0x0	R/W
VN1INSEL[3:0]	VN1 input																																					
4' b0000	保留																																					
4' b0001	保留																																					
4' b0010	保留																																					
4' b0011	保留																																					
4' b0100	PA00(VC1_N4)																																					
4' b0101	PD00(VC1_N5)																																					
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	保留																																					
4' b1101	保留																																					
4' b1110	1.2V																																					
4' b1111	VCAP																																					

位	标记	功能描述	复位值	读写																																		
7:4	保留	--	0x0	--																																		
3:0	VP1INSEL[3:0]	VC1" +" 输入端电压电压选择: <table border="1"> <thead> <tr> <th>VP1INSEL[3:0]</th><th>VP1 input</th></tr> </thead> <tbody> <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> <tr><td>4' b1111</td><td>PB07(VC1_P15)</td></tr> </tbody> </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)	4' b1111	PB07(VC1_P15)	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)																																					
4' b1111	PB07(VC1_P15)																																					

### 31.6-6VC1 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
VC1FLT_NUM[15:0]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC1 INT _EN	VC1 HIGH INTEN	VC1 RISEI NTEN	VC1 FALL INTEN	保留		VC1 FLTEN	保留				VC1_FLTCLK _SEL[1:0]	VC1 HY_EN	VC1 EN		
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	R/W
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

### 31.6-7VC1 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
保留										INV_PAD	TM1ABKE	TM1ACH4_EN			
--										R/W					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INV_TM1A_CH4_EN	TM1A_CH3	INV_TM1A_CH3	TM1A_CH2	INV_TM1A_CH2	TM1A_CH1	INV_TM1A_CH1	保留								
R/W														--	

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

位	标记	功能描述	复位值	读写
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	--	0x0	--

### 31.6-8VC1 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															
--															

位	标记	功能描述	复位值	读写
31:2	保留	--	0x0	--
1	VC1_FLOUT	VC1 滤波后状态 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	W0C

## 【32】运算放大器 (OPA)

### 32.1 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。

### 32.2 OPA 特性

- (1) 两个独立配置运放
- (2) 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
- (3) OPA 通过外部电阻连接可配置为仪表放大器
- (4) 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 的参考电压, 作 OPA offset Cancel)

### 32.3OPA 框图

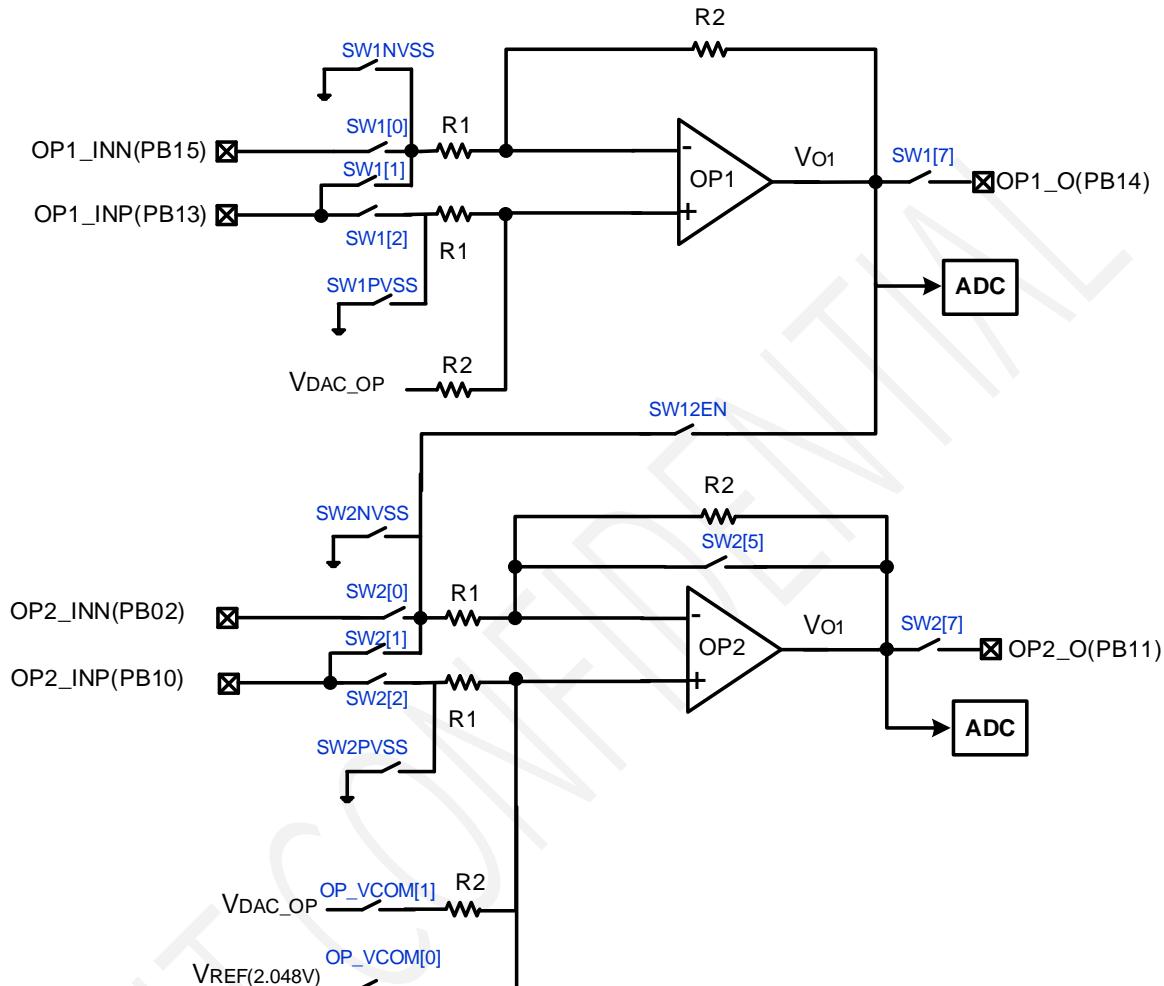
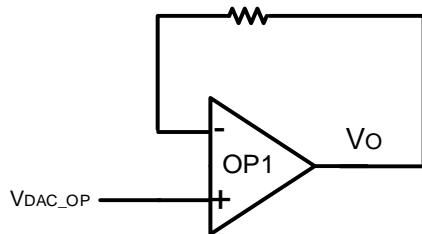


Figure 32 - 1 OPA1,OPA2 结构框图

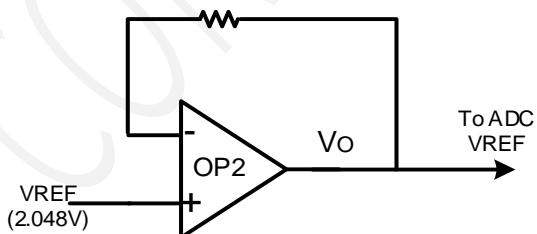
### 32.4PGA 增益

OPA 放大倍率 PGA 的增益  $R_2/R_1=16$

#### 32.4-1单位增益 PGA



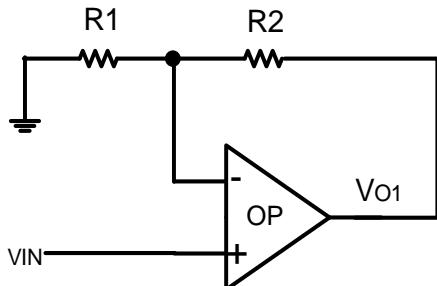
OP2 Switch	设定值
SW1[0]	0
SW1[1]	0
SW1[2]	0
SW1[7]	0
SW1PVSS, SW1NVSS	0
SW2EN	0



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

### 32.4-2 正向输入 PGA



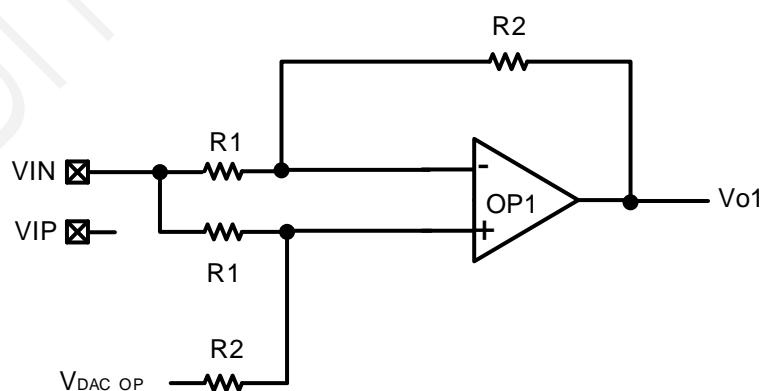
$$V_{O1} = V_{IN} \times (1 + GAIN) \rightarrow GAIN = R_2 / R_1 = 16$$

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

### 32.4-3 互补输入 PGA

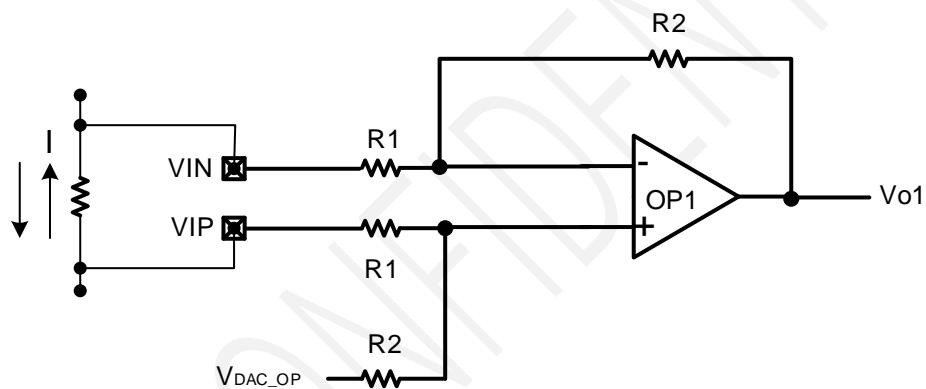
Step 1 : OP1 Vos Cancsel



OP1 Switch	设定值
SW1[0]	1
SW1[1]	1
SW1[2]	0
SW1[7]	0
SW1VSS	0
SW12EN	0

-> ADC 采样 Vo1 的电压值 : Vcom\_ADC

Step 1 : 量测(VIP-VIN)



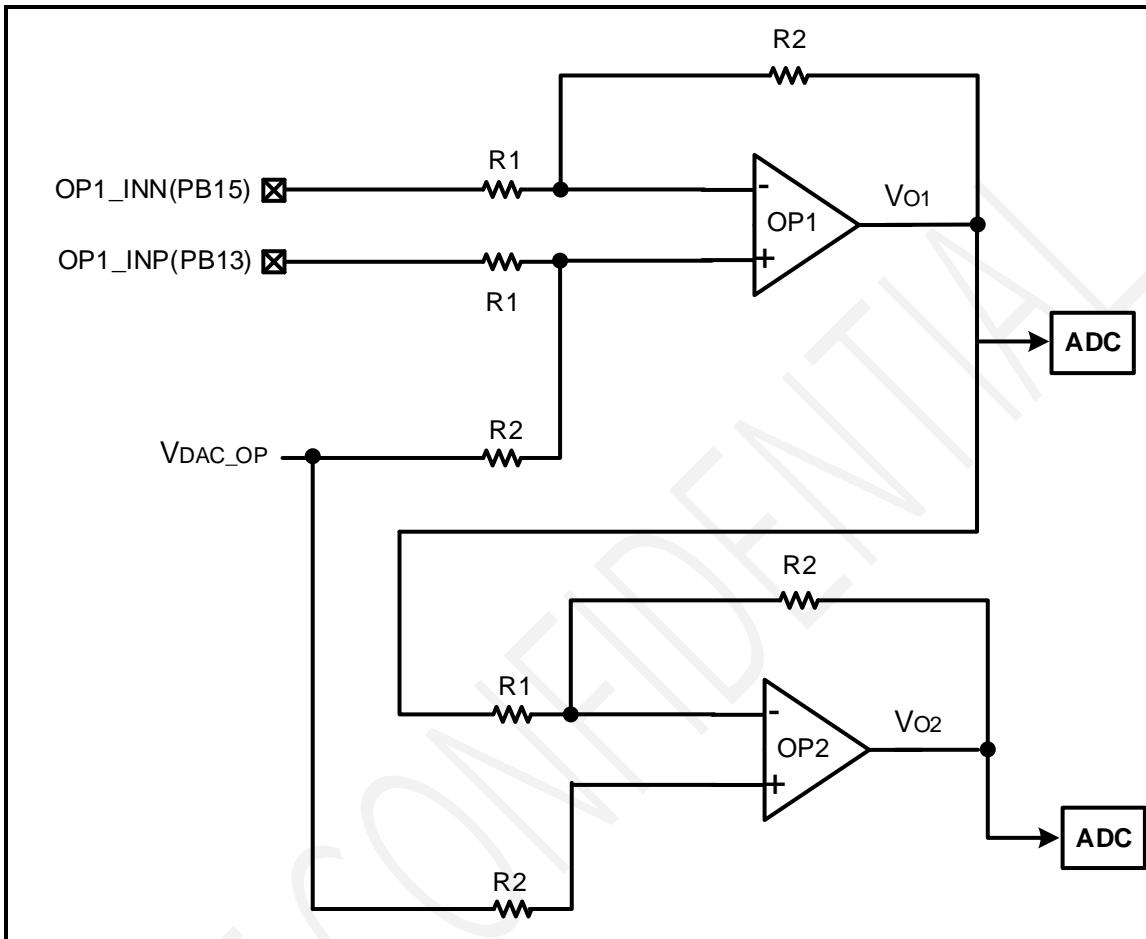
$$VO1 = V_{DAC\_OP} + GAIN2 * (VIP - VIN) \rightarrow GAIN2 = R2 / R1 = 16$$

OP1 Switch	设定值
SW1[0]	1
SW1[1]	0
SW1[2]	1
SW1[7]	0
SW1VSS	0
SW12EN	0

-> ADC 采样 Vo1 的电压值 : Vo\_ADC

-> Vo1\_ADC = Vo\_ADC - Vcom\_ADC (正值 : VIP > VIN, 负值 : VIP < VIN)

### 32.4-4串級 PGA



$$VO_2 = V_{DAC\_OP} + GAIN2 * (OP1\_INP - OP1\_INN) \rightarrow GAIN2 = 256$$

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

### 32.5OPA 寄存器列表

基址址: 0x4000 5400

偏移地址	名称	描述	默认值
0x00	OP1_CR0	OP1 控制寄存器 0	0x00000000
0x04	OP2_CR0	OP2 控制寄存器 0	0x00000000
0x08	OPDAC_CR	OP1,OP2 的 DAC 控制寄存器	0x00000000

Table 32- 1 OPA 寄存器列表和复位值

### 32.6寄存器说明

#### 32.6-1OP1 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	保留								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												SW1 NVSS	SW1 PVSS	OP1 EN	
--												R/W	R/W	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	保留	--	0x0	--
23:16	SW1[7:0]	OP1 选择配置 0: Open 1: Short	8' h00	R/W
15:3	保留	--	0x0	--
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

### 32.6-2OP2 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]							
--								R/W							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								OP2_VCOMS	保留				SW2_NVSS	SW2_PVSS	OP2_EN
--								R/W	--				R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:24	保留	--	0	--
23:16	SW2[7:0]	OP2 选择配置 0: Open 1: Short	8' h00	R/W
15:7	保留	--	0	--
6	OP2_VCOMS	OP2 VCOM selection 0: 选择 ADC VREF(2.048V) 电压 1: 选择 DAC output	0	R/W
5:3	保留	--	0	--
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

### 32.6-3OPDAC 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															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					OPDAC_VREF	OPDAC_BUF_EN	OPDAC_EN	保留		OPDAC[5:0]					
--					R/W	R/W	R/W	--		R/W					

位	标记	功能描述	复位值	读写
31:11	保留	--	0	--
10	OPDAC_VREF	OPA DAC 参考电压选择 1: AVDD 0: VCAP(2.5V)	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	保留	--	0	--
5:0	OPDAC[5:0]	OPA DAC ( $V_{DAC\_OP}$ ) 输出电压选择: $V_{DAC\_OP}$ 可以使用 ADC 的 AIN12 读出电压值。 6' h0 : $V_{DAC\_OP} = 0$ 6' h1 : $V_{DAC\_OP} = VDD * 1/64$ 6' h2 : $V_{DAC\_OP} = VDD * 2/64$ ... ... ... 6' h63 : $V_{DAC\_OP} = VDD * 63/64$	0	R/W

## 【33】PDMA 控制器

### 33.1 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。

### 33.2 PDMA 主要特性

- PDMA 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。
  - **乒乓模式(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
  - 定时器/计数器: TIM1,1A,1B; TIM2,2A,2B; TIM10,TIM11,LPTIM
  - 模拟模块: ADC

### 33.3 功能框图

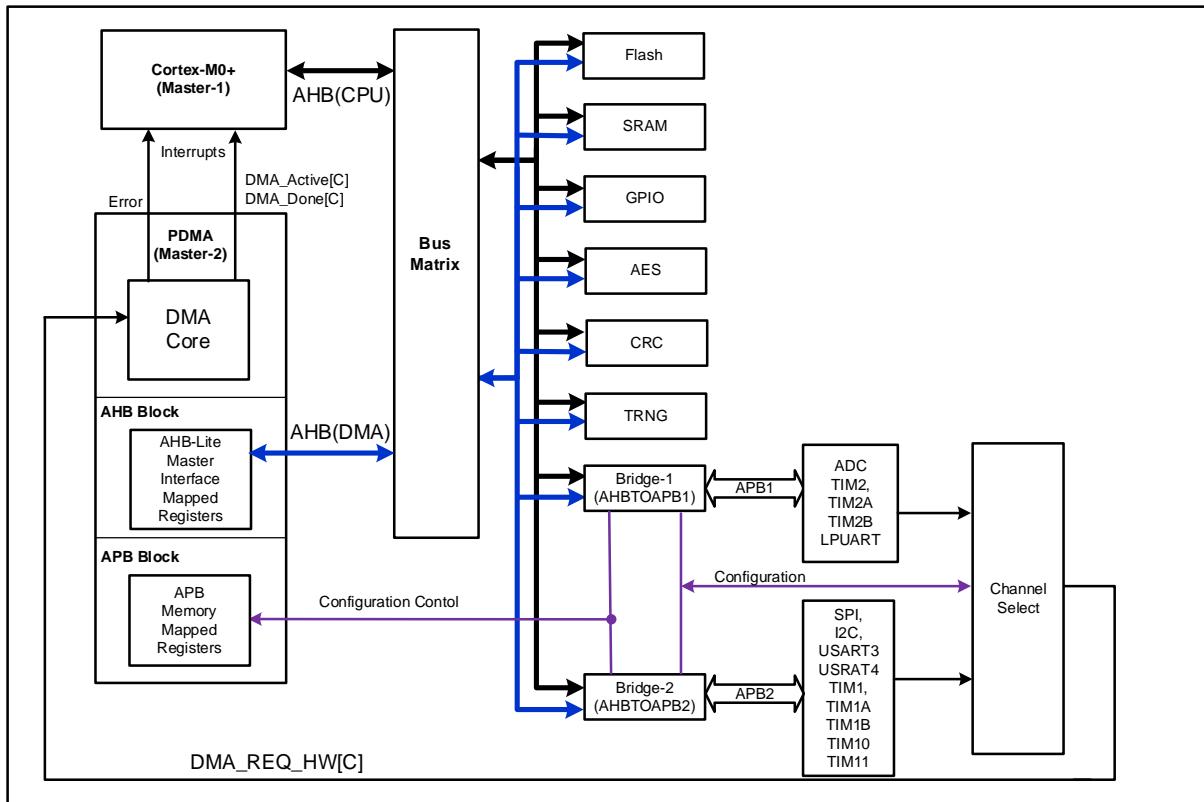


Figure 33- 1 PDMA 框图

Note : DMA\_Active[C]/DMA\_Done[C] , 其中 C 为通道数=0~15

- PDMA : L032 PDMA 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。

### 33.4 功能描述

PDMA 控制器和 Cortex®-M0 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 33.4.1 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](#)

### 33.4-1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 Table33-1

外设请求		通道(主要) [C=0, 1, 2]	通道(备用) [C=0, 1, 2]	DMA Request	
CHSELREQ[C] =0	CHSELREQ[C] =1			HWREQ	SWREQ
ADC		0	0	V	V
TIM2		1	1	V	V
TIM2A	AES-RX	2	2	V	V
QSPI-RX/TX		3	3	V	V
TIM2B	AES-TX	4	4	V	V
SPI/I2S-RX		5	5	V	V
SPI/I2S-TX		6	6	V	V
USART3-RX		7	7	V	V
USART3-TX		8	8	V	V
USART4-RX	LPUART-RX	9	9	V	V
USART4-TX	LPUART-TX	10	10	V	V
I2C0-RX	TIM10	11	11	V	V
I2C0-TX	TIM11	12	12	V	V
TIM1	LPTIM	13	13	V	V
TIM1A	TIM10	14	14	V	V
TIM1B	TIM11	15	15	V	V

Table 33- 1 DMA 使用的通道与请求列表

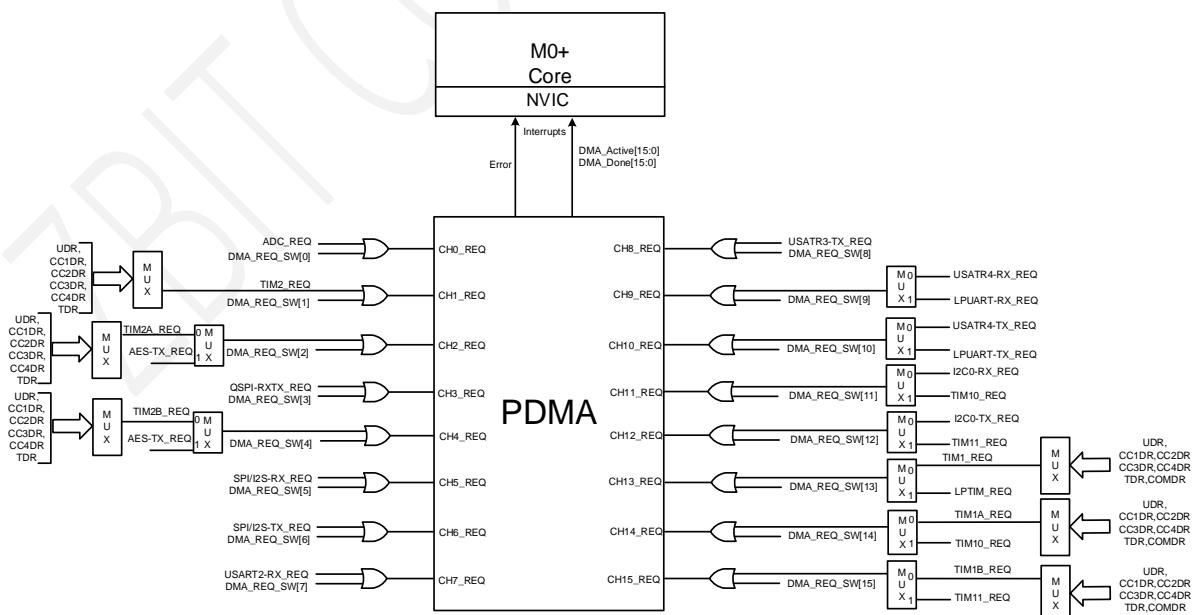


Figure 33- 2 通道请求框图

### 33.4-2PDMA 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。

<b>R_power</b>	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 33- 2 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。

### 33.4-3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 DMAPRIS 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	
14	High	-
15	High	-
0	Default	-
1	Default	-
.	Default	
14	Default	-
15	Default	最低等级的 DMA 信道

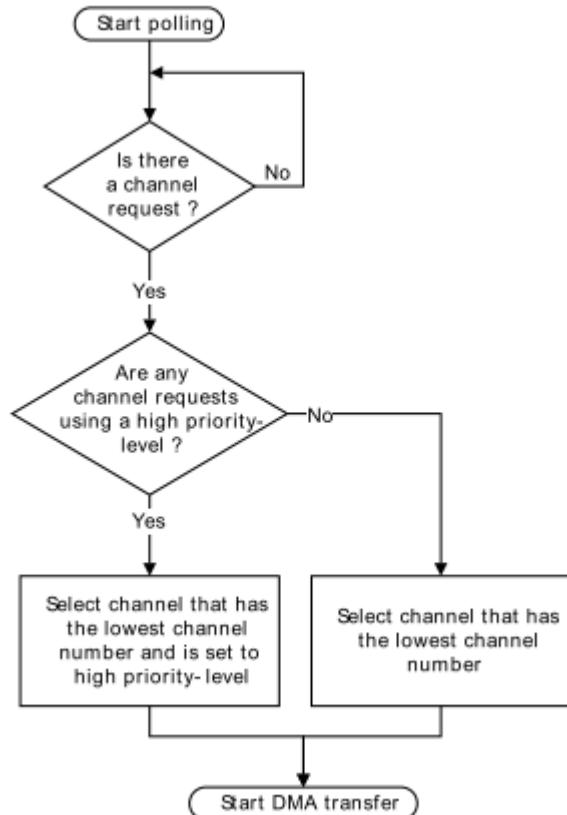


Figure 33- 3 Polling flowchart

### 33.4-4DMA 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 33- 3 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 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

主要通道-15	Unused	0x0FC+DMAPDBPTR	0x1FC+DMAPDBPTR
	Control	0x0F8+DMAPDBPTR	0x1F8+DMAPDBPTR
主要通道-14	Destination End Pointer	0x0F4+DMAPDBPTR	0X1F4+DMAPDBPTR
	Source End Pointer	0x0F0+DMAPDBPTR	0X1F0+DMAPDBPTR
主要通道-13	Unused	0x0E8+DMAPDBPTR	0X1E8+DMAPDBPTR
	Control	0x0E4+DMAPDBPTR	0X1E4+DMAPDBPTR
主要通道-12	Destination End Pointer	0x0E0+DMAPDBPTR	0X1E0+DMAPDBPTR
	Source End Pointer	0x0DC+DMAPDBPTR	0X1DC+DMAPDBPTR
主要通道-11	Unused	0x0D4+DMAPDBPTR	0X1D4+DMAPDBPTR
	Control	0x0D0+DMAPDBPTR	0X1D0+DMAPDBPTR
主要通道-10	Destination End Pointer	0x0D8+DMAPDBPTR	0X1D8+DMAPDBPTR
	Source End Pointer	0x0D4+DMAPDBPTR	0X1D4+DMAPDBPTR
主要通道-9	Unused	0x0CC+DMAPDBPTR	0X1C8+DMAPDBPTR
	Control	0x0C8+DMAPDBPTR	0X1C4+DMAPDBPTR
主要通道-8	Destination End Pointer	0x0C4+DMAPDBPTR	0X1C0+DMAPDBPTR
	Source End Pointer	0x0C0+DMAPDBPTR	0X1BC+DMAPDBPTR
主要通道-7	Unused	0x0BC+DMAPDBPTR	0X1B8+DMAPDBPTR
	Control	0x0B8+DMAPDBPTR	0X1B4+DMAPDBPTR
主要通道-6	Destination End Pointer	0x0B4+DMAPDBPTR	0X1B0+DMAPDBPTR
	Source End Pointer	0x0B0+DMAPDBPTR	0X1AC+DMAPDBPTR
主要通道-5	Unused	0x0AC+DMAPDBPTR	0X1A8+DMAPDBPTR
	Control	0x0A8+DMAPDBPTR	0X1A4+DMAPDBPTR
主要通道-4	Destination End Pointer	0x0A4+DMAPDBPTR	0X1A0+DMAPDBPTR
	Source End Pointer	0x0A0+DMAPDBPTR	0X19C+DMAPDBPTR
主要通道-3	Unused	0x09C+DMAPDBPTR	0X198+DMAPDBPTR
	Control	0x098+DMAPDBPTR	0X194+DMAPDBPTR
主要通道-2	Destination End Pointer	0x094+DMAPDBPTR	0X190+DMAPDBPTR
	Source End Pointer	0x090+DMAPDBPTR	0X18C+DMAPDBPTR
主要通道-1	Unused	0x08C+DMAPDBPTR	0X188+DMAPDBPTR
	Control	0x088+DMAPDBPTR	0X184+DMAPDBPTR
主要通道-0	Destination End Pointer	0x084+DMAPDBPTR	0X180+DMAPDBPTR
	Source End Pointer	0x080+DMAPDBPTR	0X17C+DMAPDBPTR
备用通道-15	Unused	0x07C+DMAPDBPTR	0X178+DMAPDBPTR
	Control	0x078+DMAPDBPTR	0X174+DMAPDBPTR
备用通道-14	Destination End Pointer	0x074+DMAPDBPTR	0X170+DMAPDBPTR
	Source End Pointer	0x070+DMAPDBPTR	0X16C+DMAPDBPTR
备用通道-13	Unused	0x06C+DMAPDBPTR	0X168+DMAPDBPTR
	Control	0x068+DMAPDBPTR	0X164+DMAPDBPTR
备用通道-12	Destination End Pointer	0x064+DMAPDBPTR	0X160+DMAPDBPTR
	Source End Pointer	0x060+DMAPDBPTR	0X15C+DMAPDBPTR
备用通道-11	Unused	0x05C+DMAPDBPTR	0X148+DMAPDBPTR
	Control	0x048+DMAPDBPTR	0X144+DMAPDBPTR
备用通道-10	Destination End Pointer	0x044+DMAPDBPTR	0X140+DMAPDBPTR
	Source End Pointer	0x040+DMAPDBPTR	0X13C+DMAPDBPTR
备用通道-9	Unused	0x03C+DMAPDBPTR	0X138+DMAPDBPTR
	Control	0x038+DMAPDBPTR	0X134+DMAPDBPTR
备用通道-8	Destination End Pointer	0x034+DMAPDBPTR	0X130+DMAPDBPTR
	Source End Pointer	0x030+DMAPDBPTR	0X12C+DMAPDBPTR
备用通道-7	Unused	0x02C+DMAPDBPTR	0X128+DMAPDBPTR
	Control	0x028+DMAPDBPTR	0X124+DMAPDBPTR
备用通道-6	Destination End Pointer	0x024+DMAPDBPTR	0X120+DMAPDBPTR
	Source End Pointer	0x020+DMAPDBPTR	0X11C+DMAPDBPTR
备用通道-5	Unused	0x01C+DMAPDBPTR	0X118+DMAPDBPTR
	Control	0x018+DMAPDBPTR	0X114+DMAPDBPTR
备用通道-4	Destination End Pointer	0x014+DMAPDBPTR	0X110+DMAPDBPTR
	Source End Pointer	0x010+DMAPDBPTR	0X10C+DMAPDBPTR
备用通道-3	Unused	0x00C+DMAPDBPTR	0X108+DMAPDBPTR
	Control	0x008+DMAPDBPTR	0X104+DMAPDBPTR
备用通道-0	Destination End Pointer	0x004+DMAPDBPTR	0X100+DMAPDBPTR
	Source End Pointer	0x000+DMAPDBPTR	

DMA\_ADBPTR=0X100+DMAPDBPTR

**Table 33- 4 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 Table33-4 规定)加 DMAPDBPTR 寄存器值。

Note :

- (1) 0x2000\_000 ≤ DMAPDBPTR≤0x2000\_4000 → 储存器映像
- (2) 0x100+ DMAPDBPTR → DMA\_ADBPTR

### 33.4-4.1 控制数据配置(CHNL\_CFG)

Table 33- 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： <table border="1"> <thead> <tr> <th>来源数据宽度</th><th>DST_INC</th><th>目标地址增量</th></tr> </thead> <tbody> <tr> <td rowspan="4">字节</td><td>00</td><td>字节</td></tr> <tr> <td>01</td><td>半字</td></tr> <tr> <td>10</td><td>字</td></tr> <tr> <td>11</td><td>不递增。地址仍为 DST_END_PTR 存储器位置包含的值</td></tr> <tr> <td rowspan="4">半字</td><td>00</td><td>保留</td></tr> <tr> <td>01</td><td>半字</td></tr> <tr> <td>10</td><td>字</td></tr> <tr> <td>11</td><td>不递增。地址仍为 DST_END_PTR 存储器位置包含的值</td></tr> <tr> <td rowspan="5">字</td><td>00</td><td>保留</td></tr> <tr> <td>01</td><td>保留</td></tr> <tr> <td>10</td><td>字</td></tr> <tr> <td>11</td><td>不递增。地址仍为 DST_END_PTR 存储器位置包含的值</td></tr> </tbody> </table>	来源数据宽度	DST_INC	目标地址增量	字节	00	字节	01	半字	10	字	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值	半字	00	保留	01	半字	10	字	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值	字	00	保留	01	保留	10	字	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值
来源数据宽度	DST_INC	目标地址增量																														
字节	00	字节																														
	01	半字																														
	10	字																														
	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值																														
半字	00	保留																														
	01	半字																														
	10	字																														
	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值																														
字	00	保留																														
	01	保留																														
	10	字																														
	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值																														
	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： <table border="1"> <thead> <tr> <th>来源数据宽度</th><th>DST_INC</th><th>目标地址增量</th></tr> </thead> <tbody> <tr> <td rowspan="4">字节</td><td>00</td><td>字节</td></tr> <tr> <td>01</td><td>半字</td></tr> <tr> <td>10</td><td>字</td></tr> <tr> <td>11</td><td>不递增。地址仍为 DST_END_PTR 存储器位置包含的值</td></tr> <tr> <td rowspan="4">半字</td><td>00</td><td>保留</td></tr> <tr> <td>01</td><td>半字</td></tr> <tr> <td>10</td><td>字</td></tr> <tr> <td>11</td><td>不递增。地址仍为 DST_END_PTR 存储器位置包含的值</td></tr> <tr> <td rowspan="4">字</td><td>00</td><td>保留</td></tr> <tr> <td>01</td><td>保留</td></tr> <tr> <td>10</td><td>字</td></tr> <tr> <td>11</td><td>不递增。地址仍为 DST_END_PTR 存储器位置包含的值</td></tr> </tbody> </table>	来源数据宽度	DST_INC	目标地址增量	字节	00	字节	01	半字	10	字	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值	半字	00	保留	01	半字	10	字	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值	字	00	保留	01	保留	10	字	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值
来源数据宽度	DST_INC	目标地址增量																														
字节	00	字节																														
	01	半字																														
	10	字																														
	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值																														
半字	00	保留																														
	01	半字																														
	10	字																														
	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值																														
字	00	保留																														
	01	保留																														
	10	字																														
	11	不递增。地址仍为 DST_END_PTR 存储器位置包含的值																														

位	参数	说明
[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]	保留	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。 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。

### 33.4-5DMA 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。

### 33.4-5.1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。

### 33.4-5.2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  $2^R$

例如： $2^R=2$ ,  $N=6$ :

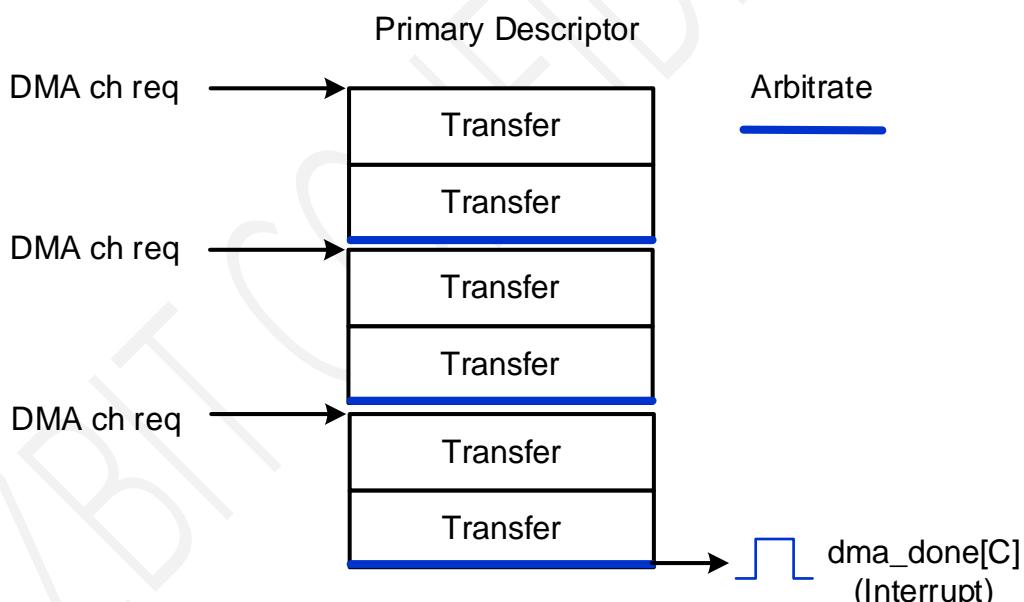


Figure 33- 4 基本模式图

### 33.4-5.3 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 会显着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

例如： $2^R=2$ ,  $N=6$ :

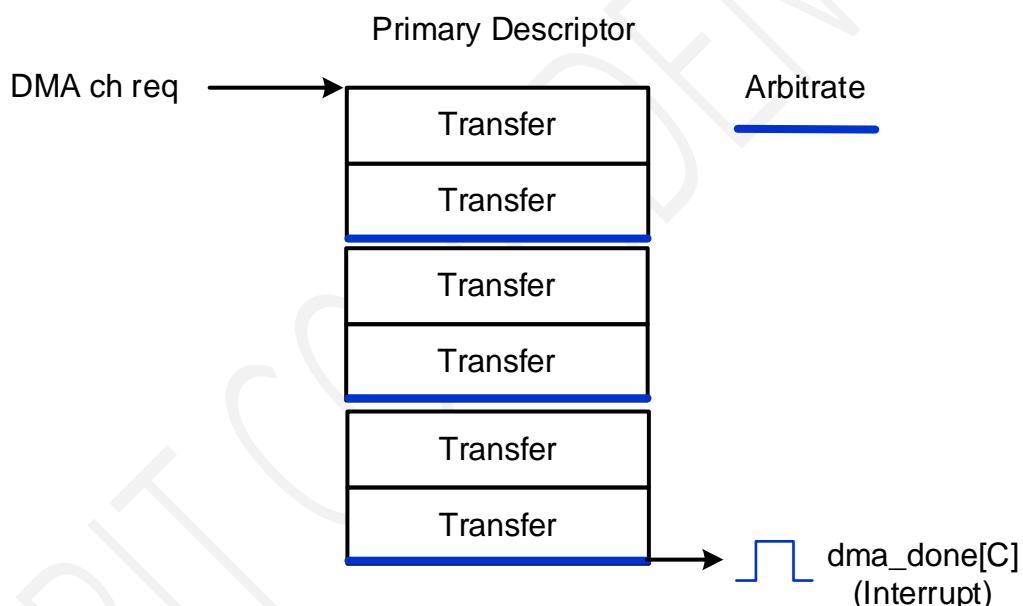


Figure 33- 5 自动请求模式范例图

### 33.4-5.4Ping-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)

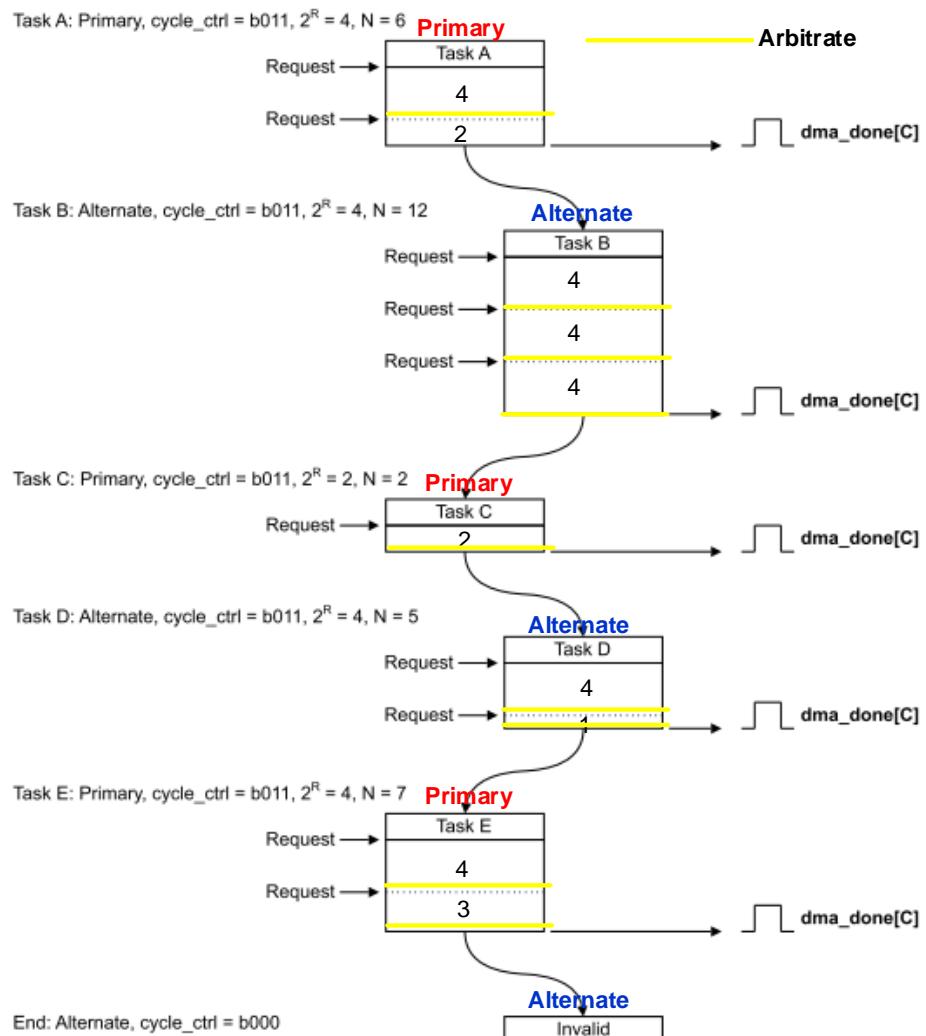


Figure 33- 6 Ping-pong 模式范例图

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。

### 33.4-5.5 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

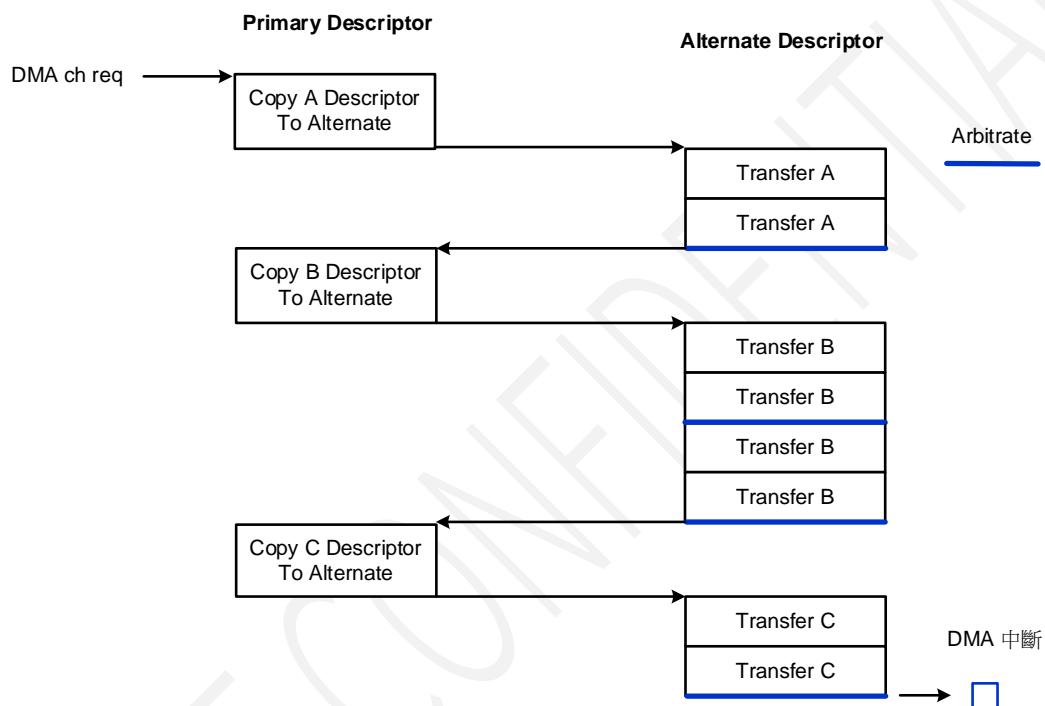
位	名称	值	描述
31:30	DST_INC	10	10: 配置控制器对地址使用字增量。
29:28	DST_SIZE	10	10: 配置控制器使用字转换。
27:26	SRC_INC	10	10: 配置控制器对地址使用字增量。
25:24	SRC_SIZE	10	10: 配置控制器使用字转换。
23:18	保留	--	未定义。写入 0.
17:14	R_POWER	0010	0010: 表示 DMA 控制器要执行 4 次传输。
13: 4	N_MINUS_1	-	配置控制器执行N次DMA传输，其中N为4的倍数。 The number of alternative task to be prepared ×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 33- 6 存储器分散/聚集式-信道配置表

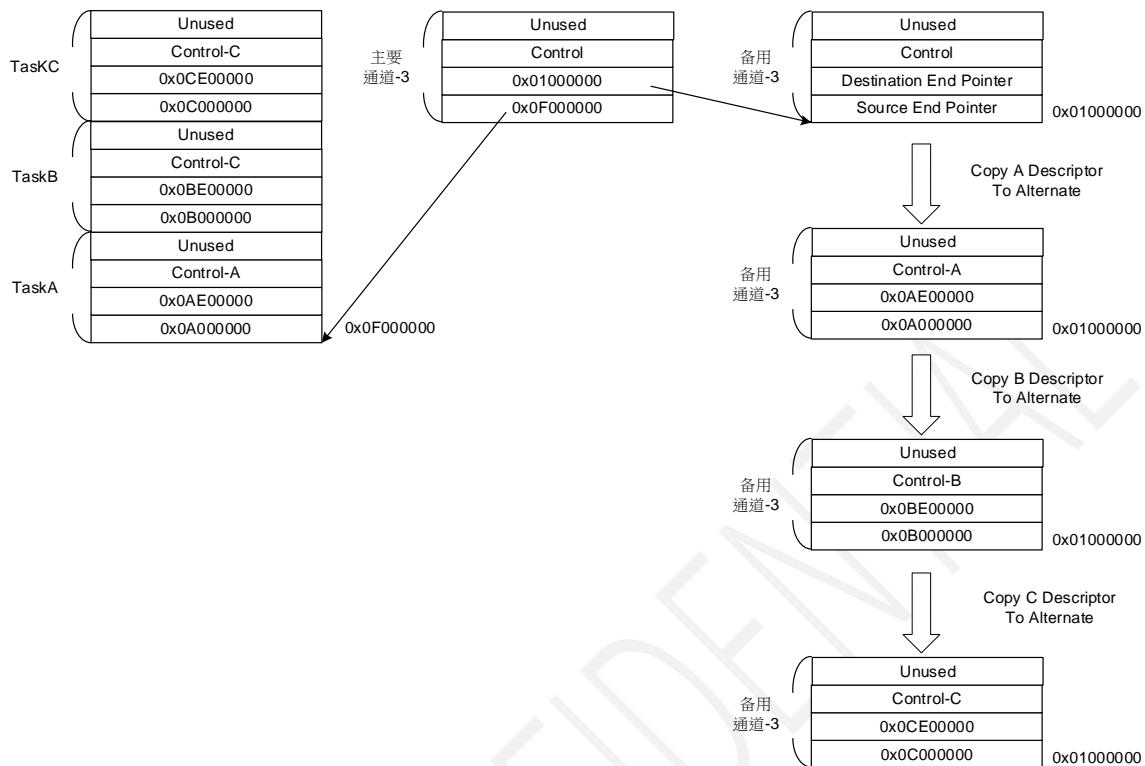
**范例说明：**

- 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 33- 7 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 33- 8 复制 Task A,B,C 描述元的配置数据图**

### 33.4-5.6 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

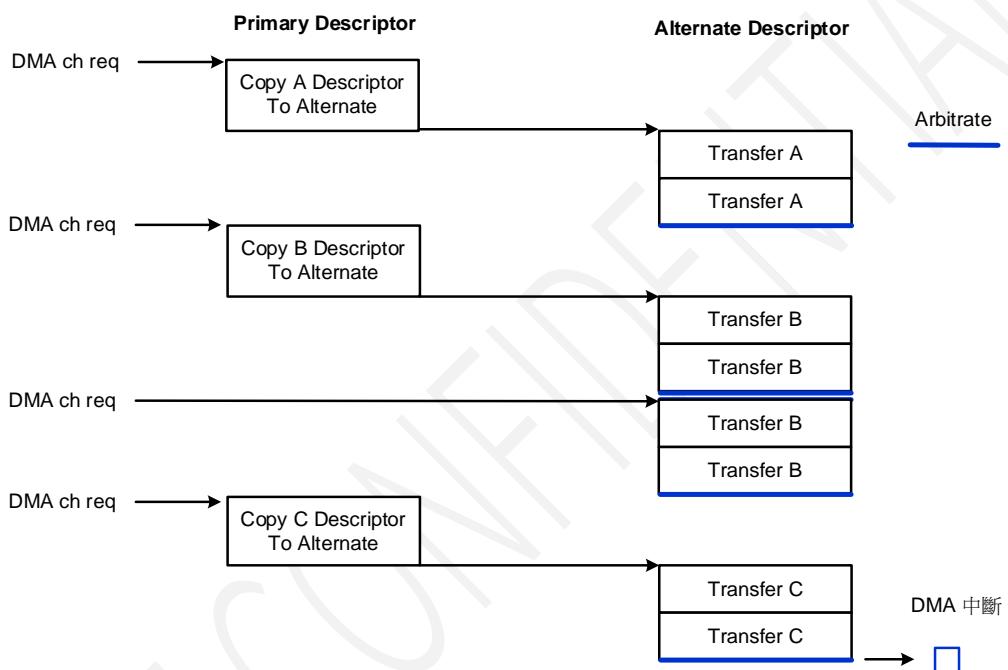
Table 33- 7 外设分散/聚集式 -信道配置表

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 33- 9 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

### 33.4-6 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旗标寄存器 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。

### 33.5 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]

### 33.6 DMA 寄存器列表

基址: 0x4000 6000

偏移地址	名称	描述	访问类型	默认值
0x00	DMA_STA	状态寄存器	RO	0x0000 0000
0x04	DMA_CFG	配置寄存器	WO	0x0000 0000
0x08	DMA_PDBPTR	主要控制数据地址指针寄存器	RW	0x0000 0000
0x0C	DMA_ADBPTR	备选控制数据地址指针寄存器	RO	0x0000 00nn <sup>(2*)</sup>
0x10	DMA_WTREQ_STA	DMA 等待请求状态	RO	0x0000 0000
0x14	DMA_SWREQ	软件请求寄存器	WO	0x0000 0000
0x20	DMA_RMSKSET	请求屏蔽设置寄存器	RW	0x0000 0000
0x24	DMA_RMSKCLR	请求屏蔽清零寄存器	WO	0x0000 0001
0x28	DMA_ENSET	使能设置寄存器	RW	0x0000 0001
0x2C	DMA_ENCLR	使能清零寄存器	WO	0x0000 0000
0x30	DMA_ALTSET	主要-备选设置寄存器	RW	0x0000 0000
0x34	DMA_ALTCLR	主要-备选清除寄存器	WO	0x0000 0000
0x38	DMA_PRISET	优先级设置寄存器	RW	0x0000 0000
0x3C	DMA_PRICLR	优先级清零寄存器	WO	0x0000 0000
0x4C	DMA_ERRCLR	总线错误清零寄存器	RW	0x0000 0000
0x60	DMA_CHINTSET	DMA 通道传输完成中断设置寄存器	RW	0x0000 0000
0x64	DMA_CHINTCLR	DMA 通道传输完成中断清除寄存器	RW	0x0000 0000
0x68	DMA_CHTCIF	DMA 通道传输完成中断旗标寄存器	RW	0x0000 0000
0x6C	DMA_CHWTTONREG	DMA 通道等待请求使能寄存器	RW	0x0000 0000
0x70	DMA_CHSELREQ	DMA 通道选择请求寄存器	RW	0x0000 0000
0xE8(1*)	DMA_STALL	停滞状态寄存器	RO	0x0000 0001
0xE10(1*)	DMA_CHREQ	DMA 请求状态寄存器	RO	0x0000 0001
0xE18(1*)	DMA_CHSREQ	DMA 通道单笔请求状态	RO	0x0000 0000
0xE20(1*)	DMA_CHDONESET	DMA 通道传输完成的状态设置寄存器	RW	0x0000 0000
0xE24(1*)	DMA_CHDONECLR	DMA 通道传输完成状态清 0 寄存器	WO	--
0xE28(1*)	DMA_CHACTSET	DMA 通道正在传输中的状态设置寄存器	RW	0x0000 0000
0xE2C(1*)	DMA_CHACTCLR	通道正在传输中的状态清 0 寄存器	WO	--
0xE48(1*)	DMA_ERRSET	DMA 总线错误清除寄存器寄存器	WO	--

**Table 33- 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

### 33.7 寄存器说明

#### 33.7-1 DMA 状态寄存器(DMA\_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								CHNLSMINUS1[4:0]							
--								RO							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								STATE[3:0]				保留		Master EN	
--								RO				--		RO	

位	标记	功能描述	复位值	读写
31:21	保留	--	0x0	--
20:16	CHNLSMINUS1 [4:0]	可用DMA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[3:0]	DMA控制器状态机的当前状态。读取此寄存器可了解DMA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 : WAITDMAREQCLR: 等待DMA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	--
0	MasterEN	DMA控制器使能状态。 0: 控制器已禁用。 1: 控制器已使能。	0	RO

### 33.7-2DMA 配置寄存器(DMA\_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															
--															

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

位	标记	功能描述	复位值	读写
31:1	保留	--	0x0000	--
0	MasterEN	控制器使能控制。 0: 禁用 DMA控制器。 1: 使能 DMA 控制器。	0	R/W

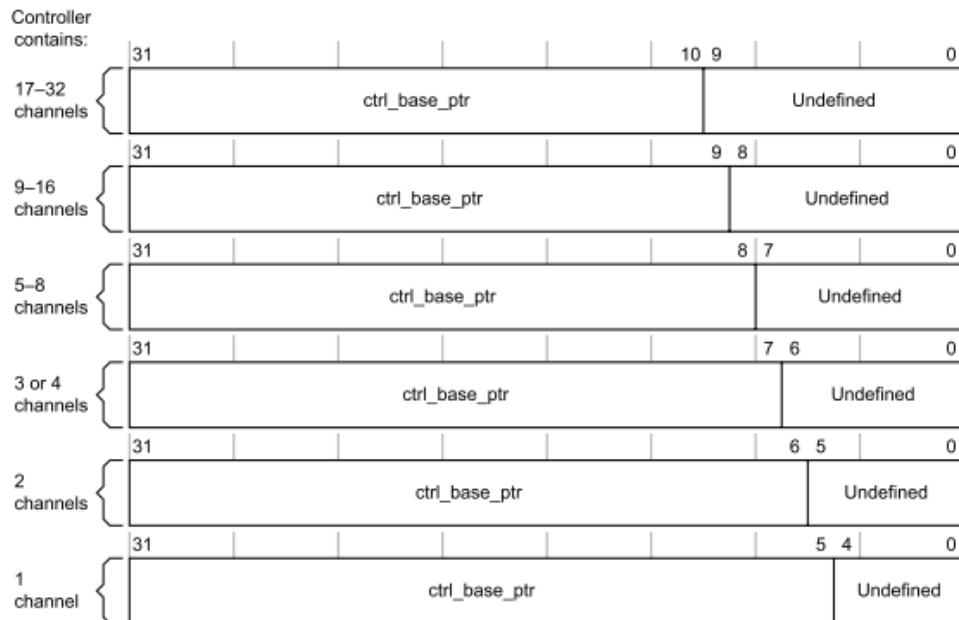
### 33.7-3DMA 主要控制数据基地址指针寄存器(DMA\_PDBPTR)

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
DMA_PDBPTR[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_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	--	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>CH_BITS</td><td>说明</td></tr> <tr> <td>0</td><td>使用 1 DMA 通道</td></tr> <tr> <td>1</td><td>使用 2 DMA 通道</td></tr> <tr> <td>2</td><td>使用 3, 4 DMA 通道</td></tr> <tr> <td>3</td><td>使用 5~8 DMA 通道</td></tr> <tr> <td>4</td><td>使用 9~16 DMA 通道</td></tr> <tr> <td>5</td><td>使用 17~32 DMA 通道</td></tr> </table>	CH_BITS	说明	0	使用 1 DMA 通道	1	使用 2 DMA 通道	2	使用 3, 4 DMA 通道	3	使用 5~8 DMA 通道	4	使用 9~16 DMA 通道	5	使用 17~32 DMA 通道	0x00	R/W
CH_BITS	说明																	
0	使用 1 DMA 通道																	
1	使用 2 DMA 通道																	
2	使用 3, 4 DMA 通道																	
3	使用 5~8 DMA 通道																	
4	使用 9~16 DMA 通道																	
5	使用 17~32 DMA 通道																	



### 33.7-4DMA 备用控制数据基地址指针寄存器(DMA\_ADBPTR)

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
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[31:0]	指向备用数据结构基址的指针	0x0000	R/W

### 33.7-5DMA 通道等待请求状态寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
15	CHWTREQ[C*]	1: 通道(15)在等待请求状态	0	RO
14	CHWTREQ[C*]	1: 通道(14)在等待请求状态	0	RO
13	CHWTREQ[C*]	1: 通道(13)在等待请求状态	0	RO
12	CHWTREQ[C*]	1: 通道(12)在等待请求状态	0	RO
11	CHWTREQ[C*]	1: 通道(11)在等待请求状态	0	RO
10	CHWTREQ[C*]	1: 通道(10)在等待请求状态	0	RO
9	CHWTREQ[C*]	1: 通道(9)在等待请求状态	0	RO
8	CHWTREQ[C*]	1: 通道(8)在等待请求状态	0	RO
7	CHWTREQ[C*]	1: 通道(7)在等待请求状态	0	RO
6	CHWTREQ[C*]	1: 通道(6)在等待请求状态	0	RO
5	CHWTREQ[C*]	1: 通道(5)在等待请求状态	0	RO
4	CHWTREQ[C*]	1: 通道(4)在等待请求状态	0	RO
3	CHWTREQ[C*]	1: 通道(3)在等待请求状态	0	RO
2	CHWTREQ[C*]	1: 通道(2)在等待请求状态	0	RO
1	CHWTREQ[C*]	1: 通道(1)在等待请求状态	0	RO
0	CHWTREQ[C*]	1: 通道(0)在等待请求状态	0	RO

Note : C\* : 15~0 通道

### 33.7-6DMA 通道软件请求寄存器(DMA\_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
保留															
--															

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

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	保留	--	0x0000	--
15:0	CHSWREQ[C*]	写入时： 0: 不产生通道(C*)的DMA 软件请求。 1: 产生通道(C*)的DMA软件请求。	0	WO

Note : C\* : 15~0 通道

### 33.7-7DMA 通道使用多笔传输设定寄存器 (DMA\_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
保留															
--															

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
31:16	保留	--	0x0000	--
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 <sup>R</sup> transfers) 传输。 写入时： 0: 不起作用。 1: 设置通道(C*)为 burst requests( 2 <sup>R</sup> transfers)。	0	R/W

Note : C\* : 15~0 通道

### 33.7-8DMA 通道使用多笔传输清除寄存器(DMA\_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
保留															
--															

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

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	保留	--	0x0000	--
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

### 33.7-9DMA 通道请求屏蔽设置寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
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 DMARMKCLR 寄存器使能 DMA 请求) 1：禁用与通道(C*)为相关的外设产生 DMA 请求。	0	R/W

Note : C\* : 15~0 通道

### 33.7-10DMA 通道请求屏蔽清除寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
15:0	CHRMKCLR[C*]	写入时: 0: 不起作用(使用 DMARMKCLR 寄存器使能 DMA 请求) 1: 启用与通道(C*)为相关的外设产生 DMA 请求。	0	R/W

Note : C\* : 15~0 通道

### 33.7-11DMA 通道使能设置寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
15:0	CHENSET[C*]	0: 不起作用(使用 DMAENCLR 寄存器禁用通道(C*)) 1: 使能通道(C*)	0	R/W

Note : C\* : 15~0 通道

### 33.7-12DMA 通道使能清除寄存器 (**DMA\_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
保留															
--															

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	保留	--	0x0000	--
15:1	CHENCLR[C*]	0: 不起作用(使用 CHENSET 寄存器使能通道(C*)) 1: 禁用通道(C*)	0	R/W

Note : C\* : 15~0 通道

### 33.7-13DMA 通道备用设置寄存器(DMA\_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
保留															
--															

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
31:16	保留	--	0x0000	--
15:0	CHALTSET[C*]	读取时: 0: 通道(C*)使用主要数据结构 1: 通道(C*)使用备用数据结构 写入时: 0: 不起作用 1: 设置通道(C*)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, DMA 控制器会视需要自动将这些位置 1/清 0。

### 33.7-14DMA 通道备用清除设置寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
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，DMA 控制器会视需要自动将这些位置 1/清 0。

### 33.7-15DMA 通道优先级设置寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
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

### 33.7-16DMA 通道优先级清除寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
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

### 33.7-17DMA 总线错误清除寄存器(DMA\_CHERRCLR)

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	16
保留															
--															

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

位	标记	功能描述	复位值	读写
31:1	保留	--	0x0000	--
0	ERRCLR	总线错误状态。 此寄存器用于读取和清除 DMA 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。  读取时： 0：未发生总线错误。 1：有一个总线错误待处理。 写入时： 0：不起作用。 1：位清 0。	0	R/W

### 33.7-18DMA 通道传输完成中断设置寄存器(**DMA\_CHINTSET**)

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
ERR INTEN	保留														
R/W	--														

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

位	标记	功能描述	复位值	读写
31	ERRINTEN	0: 不起作用 1: 设置 DMA 传输发生错误时会产生中断	0	R/W
30:16	保留	--	0x0000	--
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 写入时: 0: 不起作用 1: 设置通道(C*)传输完成产生中断	0x0000	R/W

Note : C\* : 15~0 通道

### 33.7-19DMA 通道中断使能清除寄存器(DMA\_CHINTCLR)

地址偏移: 0x64 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR INTCLR	保留														
R/W	--														

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 DMA ERROR 中断的使能	0x0000	R/W
30:16	保留	--	0	--
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

### 33.7-20DMA 通道传输完成中断旗标寄存器(DMA\_CHTCIF)

地址偏移: 0x68 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR IF	保留														
R/W	--														

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

位	标记	功能描述	复位值	读写
31	ERRIF	读取时: 1: DMA 传输中发生错误中断旗标 写入时: 1: 清除旗标为 0	0	R/W
30:16	保留	--	0x0000	--
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	R/W

Note : C\* : 15~0 通道

### 33.7-21DMA 通道等待请求使能寄存器(DMA\_CHWTONREG)

地址偏移: 0x6C 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
CHWTONREQ[C*]															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0000	--
15:0	CHWTONREQ[C*] (dma_waitonreg[C*])	当 Burst 传输时:[1*] 0: 通道(C*) 每读写一笔资料都须设定信道配置 1: 通道(C*),DMA 连续传输	0	R/W

Note : C\* : 15~0 通道

[1\*] :

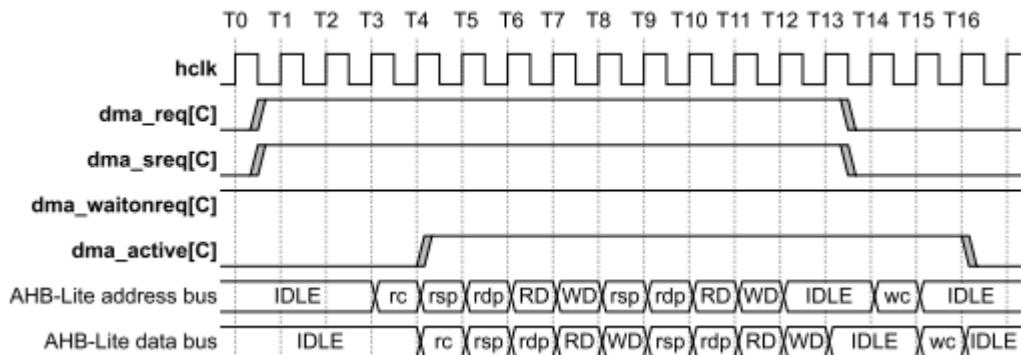


Figure 33- 10 CHWTONREQ[C\*]( dma\_waitonreg[C\*] )=1 的时序图

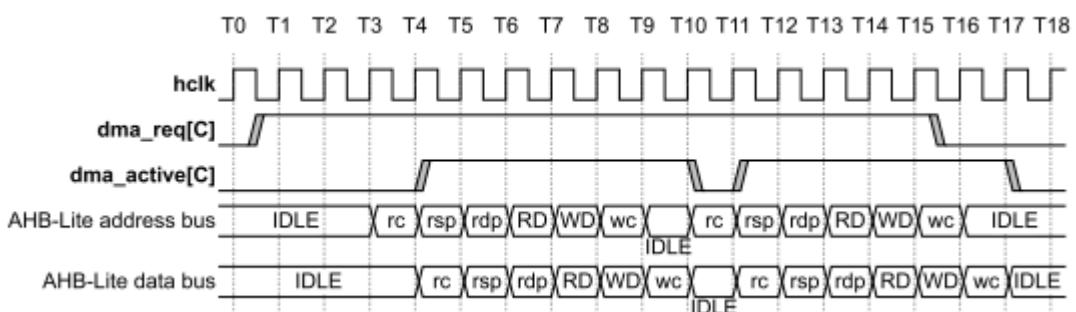


Figure 33- 11 CHx\_CHWTONREQ=0 的时序图

### 33.7-22DMA 通道选择请求寄存器(DMA\_CHSELREQ)

地址偏移: 0x70 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
CHSELREQ[C*]															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0000	--
15:0	CHSELREQ[C*]	DMA 传输时: 0 : 通道(C*)为原始的请求(Default) 1 : 通道(C*)为选项的请求	0x0000	R/W

Note : C\* : 15~0 通道

### 33.7-23DMA 停滞状态寄存器(DMA\_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
保留															
--															

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

位	标记	功能描述	复位值	读写
31:1	保留	--	0x0000	--
0	DMA_STALL	测试讯号状态 0: 正常工作 1: DMA 停滞状态	0	RO

### 33.7-24DMA 请求状态 (DMA\_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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHREQ[C*]															
RO															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0000	--
15:0	CHREQ[C*]	0: 通道(C*)为没有任何请求 1: 通道(C*)为请求状态	0	RO

Note : C\* : 15~0 通道

### 33.7-25DMA 通道单笔请求状态 (**DMA\_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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSREQ[C*]															
RO															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0000	--
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

### 33.7-26DMA 通道传输完成的状态设置寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
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 DMA_INTCFGEN = 1)	0	R/W

Note : C\* : 15~0 通道

### 33.7-27DMA 通道传输完成状态清 0 寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
15:0	DONECLR[C*]	写入时: 0: 不起作用 1: 清除通道(C*) 的传输完成讯号(DMA_Done[C*]) 为 0 (需 DMA_INTCFGEN = 1)	0	WO

Note : C\* : 15~0 通道

### 33.7-28DMA 通道正在传输中的状态设置寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
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 DMA_INTCFGEN = 1)	0	R/W

Note : C\* : 15~0 通道

### 33.7-29DMA 通道正在传输中的状态清 0 寄存器(DMA\_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
保留															
--															

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	保留	--	0x0000	--
15:0	CHACTCLR[C*]	写入时: 0: 不起作用 1: 清除通道(C*) 的传输中讯号(DMA_ACT[C*])为 0 (需 DMA_INTCFGEN = 1)	0	WO

Note : C\* : 15~0 通道

### 33.7-30DMA 总线错误清除寄存器(DMA\_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
保留															
--															

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

位	标记	功能描述	复位值	读写
31:1	保留	--	0x0000	--
0	ERRSET	设置总线错误状态。  写入时： 0: 不起作用。 1: 使能 ERROR	0	WO

## 【34】QUAD-SPI 接口(QSPI)

### 34.1 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：

- (1) 间接模式：所有操作都使用 QUADSPI 寄存器执行。
- (2) 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。
- (3) 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。

### 34.2 QSPI 功能框图

QSPI 功能框图，使用 BANK1 见图 (Dual Flash Disable set QUADSPI\_CR.DFM=0):

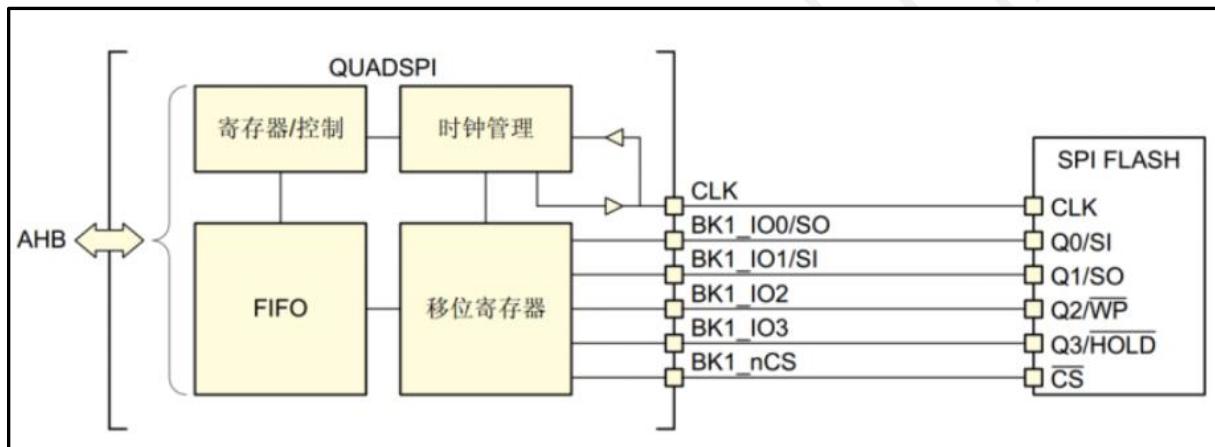


Figure 34- 1 QSPI 功能框图(BANK1)

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):

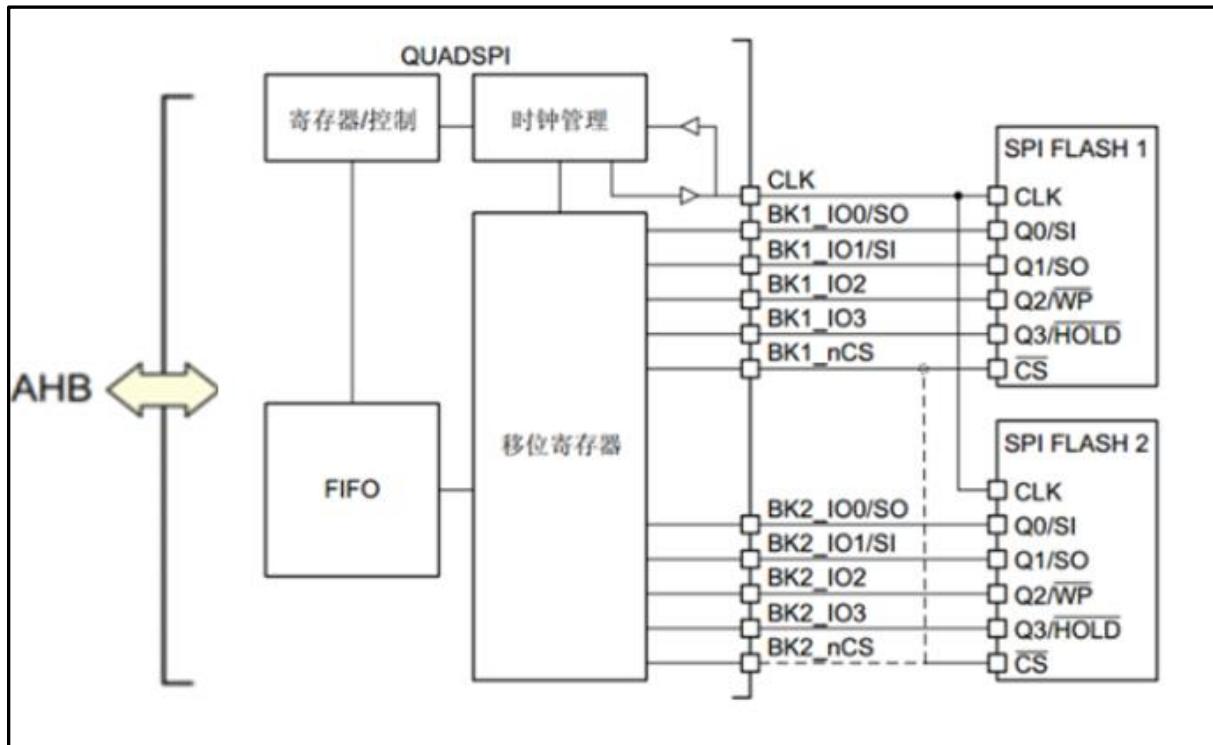


Figure 34-2 QSPI 功能框图(BANK1,BANK2)

- (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。

### 34.3QSPI 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 DMA 信道
- (8) 在 FIFO 阈值、超时、操作完成和访问错误时产生中断

### 34.4QSPI 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 34-3

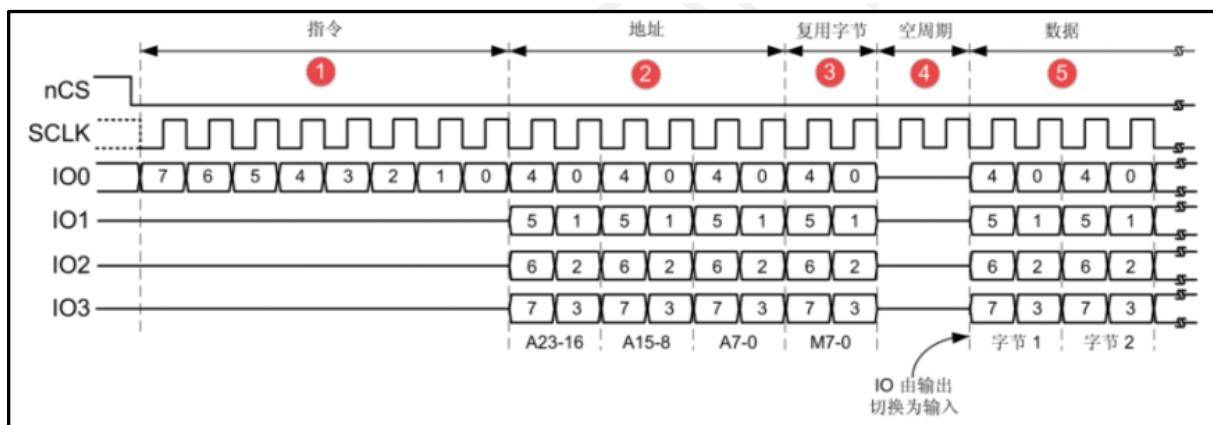


Figure 34-3 四线模式下的读命令时序

#### 34.4-1 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 DMA）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）。

#### 34.4-2四线交替字节阶段

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)。

#### 34.4-3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。

#### 34.4-4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 直接发回 Cortex 或 DMA。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。

### 34.5QUADSPI 信号接口协议模式

#### 34.5-1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。

#### 34.5-2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。

#### 34.5-3四线 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。

### 34.5-4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。

### 34.6QUADSPI 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。

### 34.6-1 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。

### 34.6-2 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 DMAEN = 1，则 FTF 置 1 时启动数据传送。如果阈值条件不再为“真”（CPU 或 DMA 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 ( $FLEVEL \leq 11$ ) 时才会重新开始读取 Flash。因此，若  $FTHRES \geq 13$ ，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  $11 < FLEVEL < FTHRES$ ，FTF 标志将保持为“0”。

### 34.7 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。

### 34.8QUADSPI 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 DMA，则会产生 DMA 传输错误，并自动禁用相应的 DMA 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。

### 34.9QUADSPI 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）。

### 34.10 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。

### 34.11 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。

### 34.12QUADSPI 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 DMAEN=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 DMA 写入 (如果 DMA 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 DMA 使能位(DMAEN)
- 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。

### 34.13 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。

### 34.14 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。

### 34.15 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。

### 34.16nCTS 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 34-4 所示

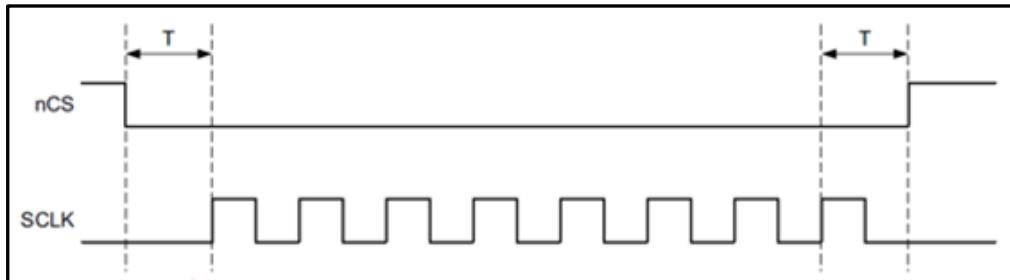


Figure 34- 4 nCS when CKMODE=0 ( $T = \text{CLK period}$ )

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 34-5 所示

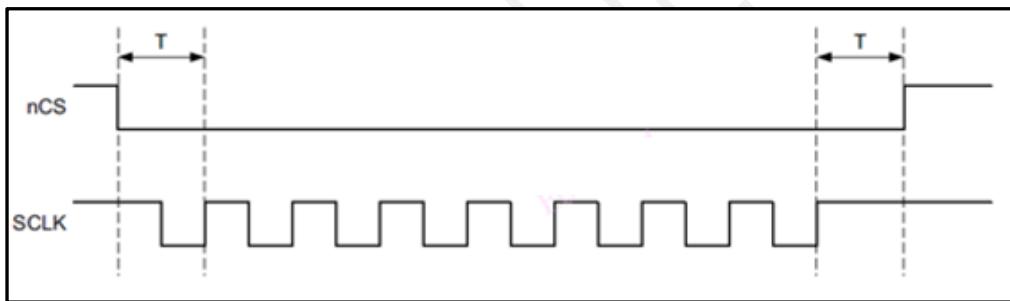


Figure 34- 5 nCS when CKMODE=1 in SDR mode ( $T=\text{CLK period}$ )

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 34-6 所示

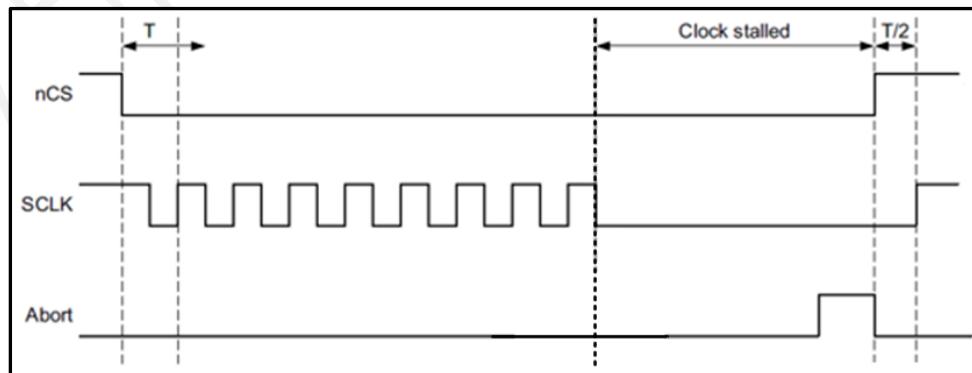


Figure 34- 6 nCS when CKMODE=1 with an abort ( $T=\text{CLK period}$ )

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。

### 34.17 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 34- 1 QSPI 中断表

### 34.18QSPI 寄存器列表

基址: 0x4002 0C00

偏移地址	名称	描述	访问类型	默认值
0x00	QUADSPI_CR	状态寄存器	RW	0x0000 0000
0x04	QUADSPI_DCR	装置配置寄存器	RW	0x0000 0000
0x08	QUADSPI_SR	状态寄存器	RW	0x0000 0000
0x0C	QUADSPI_FCR	标志清除寄存器	RW	0x0000 0000
0x10	QUADSPI_DLR	数据长度寄存器	RW	0x0000 0000
0x14	QUADSPI_CCR	通信配置寄存器	RW	0x0000 0000
0x18	QUADSPI_AR	地址寄存器	RW	0x0000 0000
0x1C	QUADSPI_ABR	选择字节寄存器	RW	0x0000 0001
0x20	QUADSPI_DR	数据寄存器	RW	0x0000 0001
0x24	QUADSPI_PSMKR	轮询屏蔽寄存器	RW	0x0000 0000
0x28	QUADSPI_PSMAR	轮询状态匹配寄存器	RW	0x0000 0000
0x2C	QUADSPI_PIR	轮询间隔寄存器	RW	0x0000 0000
0x30	QUADSPI_LPTR	低功号超时寄存器	RW	0x0000 0000

Table 34- 2 QSPI 寄存器列表和复位值

### 34.19QSPI 寄存器说明

#### 34.19-1QSPI 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[7:0]				PWM	APMS	保留	TOIE	SMIE	FTIE	TCIE	TEIE				
R/W								--	R/W						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FTHRES[4:0]				FSEL	DFM	保留	SSHIFT	TCEN	DMAEN	ABORT	EN		
--		R/W				--		--	R/W						

位	标记	功能描述	复位值	读写
31:24	PRESCALER [7:0]	QSPI 时钟预除器(基于AHB时钟): 0&1: QCLK = AHB/2 2: QCLK = AHB/3 3: QCLK = AHB/4 ... 240: QCLK = AHB/241 241~255: 保留 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。  该位只能在BUSY = 0 时修改。	0	R/W
21	保留	--	0	--
20	TOIE	超时中断使能 0: 中断禁止 1: 中断使能	0	R/W

位	标记	功能描述	复位值	读写
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	保留	--	0x0	--
12:8	FTHRES [4:0]	<p>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。</p> <p><b>在间接写入模式下 (FMODE = 00):</b></p> <p>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 FIFO 中有 2 个或更多可用字节可写入, 则设置 FTF  ...  31: 如果 FIFO 中有 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: 如果可以从 FIFO 中读取 1 个或多个有效字节, 则设置 FTF  1: 如果可以从 FIFO 中读取 2 个或更多有效字节, 则设置 FTF  ...  30: 如果可以从 FIFO 中读取 31 个有效字节, 则设置 FTF</p> <p>如果 DMAEN = 1, 那么必须禁用相应通道的 DMA 控制器在更改 FTHRES 值之前。</p>	0x0	R/W
7	FSEL	<p>Flash 存储器选择  该位选择要在单 Flash 模式下(当 DFM = 寻址的 Flash 存储器0)  0: 选择 FLASH 1  1: 选择 FLASH 2</p> <p>该位只有在 BUSY = 0 时才能修改。  当 DFM = 1 时, 该位被忽略。</p>	0x0	R/W

位	标记	功能描述	复位值	读写
6	DFM	<p>双闪存模式 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。 0: 禁用双闪模式 1: 启用双闪模式 该位只能在 BUSY = 0 时修改。</p>	0x0	R/W
5	保留	--	0x0	--
4	SSSHIFT	<p>采样移位 默认情况下，QUADSPI 在数据被驱动后采样 1/2 个 CLK 周期的数据 闪存。该位允许稍后对数据进行采样以解决外部信号延迟。 0: 无移位 1: 1/2 周期移位 固件必须确保在 DDR 模式下（当 DDRM = 1 时）SSSHIFT = 0。 只有当 BUSY = 0 时才能修改该字段</p>	0x0	R/W
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	DMAEN	<p>DMA 使能 在间接模式下，DMA 可用于通过 QUADSPI_DR 输入或输出数据寄存器。 DMA 传输在 FIFO 阈值标志 FTF 置位时启动。 0: DMA 在间接模式下禁用 1: DMA 在间接模式下启用</p>	0x0	R/W

位	标记	功能描述	复位值	读写
1	ABORT	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	0x0	R/W
0	EN	使能 QUADSPI。 0: 禁止QUADSPI 1: 使能QUADSPI	0x0	R/W

### 34.19-2QSPI 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
保留										FSIZE[4:0]					
--										R/W					

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

位	标记	功能描述	复位值	读写
31:21	保留	--	0	--
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	保留	--	0	--
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 1: Ncs 在闪存命令之间保持高电平至少 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。	0x0	R/W
7:1	保留	--	0	--
0	CKMODE	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	0	R/W

### 34.19-3QSPI 状态寄存器(QUADSPI\_SR)

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		FLEVEL[5:0]					保留	BUSY	TOF	SMF	FTF	TCF	TEF		
--		RO					--	RO							

位	标记	功能描述	复位值	读写
31:14	保留	--	0x0	--
13:8	FLEVEL [5:0]	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 31。 在内存映像模式和自动状态轮询模式下, FLEVEL为零。	0	RO
7:6	保留	--	0	--
5	BUSY	Busy 当操作正在进行时设置该位。 当Flash 存储器操作完成且 FIFO 为空, 该位自动清零。	0	RO
4	TOF	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。	0	RO
3	SMF	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。	0	RO
2	FTF	FIFO 阈值标志 在间接模式下, 当达到 FIFO 阈值时或从 Flash 存储器读取完成后, FIFO 中还有任何数据该位被设置。 一旦阈值条件不再为真, 它就会自动清除。 在自动轮询模式下, 每次读取状态寄存器时都会设置该位, 读取该位数据寄存器时会清除。	0	RO
1	TCF	传输完成标志 该位在间接模式下在编程数据的数量时, 传输或在传输中止时在任何模式下设置通过写入CTCF位1清除	0	RO
0	TEF	传输错误标志 当以间接模式访问无效地址时, 以间接模式设置此位。通过向 CTEF 写入 1 来清除。	0	RO

### 34.19-4QSPI 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
保留															
--															

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

位	标记	功能描述	复位值	读写
31:5	保留	--	0x0	--
4	CTOF	清除超时标志位 写 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 1 清除 QUADSPI_SR 寄存器中的 SMF 标志	--	R/W
2	保留	--	0	--
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	R/W
0	CTEF	清除传输错误标志 写 1 清除 QUADSPI_SR 寄存器中的 TEF 标志	0	R/W

### 34.19-5QSPI 数据长度寄存器(QUADSPI\_DLR)

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
DL[31:16]															
R/W															

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

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> <ul style="list-style-type: none"> <li>0x0000_0000: 要传输 1 个字节</li> <li>0x0000_0001: 要传输 2 个字节</li> <li>0x0000_0002: 要传输 3 个字节</li> <li>0x0000_0003: 要传输 4 个字节</li> <li>0xFFFF_FFFD: 要传输 4,294,976,294 (4G-2) 个字节</li> <li>0xFFFF_FFFE: 要传输 4,294,967,295 (4G-1) 个字节</li> <li>0xFFFF_FFFF: 未定义长度直到闪存结束(FSIZE 定义所有字节)</li> </ul> <p>如果 FSIZE = 0x1F，则无限期地继续读取</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”，因此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

### 34.19-6QSPI 通信配置寄存器(QUADSPI\_CCR)

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		SIO0	FMODE [1:0]		DMODE [1:0]		保留	DCYC [4:0]				ABSIZE [1:0]			
RO		R/W				--	R/W								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE [1:0]		ADSIZE [1:0]		ADMODE [1:0]		IMODE [1:0]		INSTRUCTION [7:0]							
R/W															

位	标记	功能描述	复位值	读写
31:29	保留	必须保持复位值。	0x0	RO
28	SIO0	仅发送指令一次模式 该位当 IMODE = 00 时，无效。 0: 每次事务发送指令 1: 仅发送第一条命令的指令 仅当 BUSY = 0 时才能写入该字段。	0x0	R/W
27:26	FMODE[1:0]	功能模式 该字段定义 QUADSPI 功能操作模式。 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 DMAEN = 1，则必须相应通道的 DMA 控制器 在更改 FMODE 值之前禁用。 该字段只能在 BUSY = 0 时写入	0x0	R/W
25:24	DMODE[1:0]	数据模式 该字段定义数据阶段的操作模式： 00: 无数据 01: 单行数据 10: 两行数据 11: 四行数据 该字段还决定了虚拟相位操作模式。 只有当 BUSY = 0 时才能写入该字段	0x0	R/W

位	标记	功能描述	复位值	读写
23	保留	--	0	--
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)。 该字段只能在 BUSY = 0 时写入</p>	0x00	R/W
17:16	ABSIZE[1:0]	<p>交替字节大小(Alternate bytes size) 该位定义交替字节大小： 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 BUSY = 0 时写入</p>	0x0	R/W
15:14	ABMODE[1:0]	<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： 00: No alternate bytes 01: Alternate bytes on a single line 10: Alternate bytes on two lines 11: Alternate bytes on four lines 该字段只能在 BUSY = 0 时写入</p>	0x0	R/W
13:12	ADSIZE[1:0]	<p>地址大小(Address size) 该字段定义地址大小： 00: 8-bit address 01: 16-bit address 10: 24-bit address 11: 32-bit address 该字段只能在 BUSY = 0 时写入</p>	0x0	R/W
11:10	ADMODE[1:0]	<p>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</p>	0x0	R/W

位	标记	功能描述	复位值	读写
9:8	IMODE[1:0]	<p>指令模式( Instruction mode)</p> <p>该字段定义了指令阶段的操作模式:</p> <p>00: No instruction</p> <p>01: Instruction on a single line</p> <p>10: Instruction on two lines</p> <p>11: Instruction on four lines</p> <p>该字段只能在 BUSY 时写入 = 0</p>	0x0	R/W
7:0	INSTRUCTION [7:0]	<p>Instruction</p> <p>要发送到外部 SPI 设备的指令</p> <p>仅当 BUSY = 0 时才能写入此字段</p>	0x00	R/W

### 34.19-7QSPI 地址寄存器(QUADSPI\_AR)

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
ADDRESS[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	ADDRESS[31:0]	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	R/W

### 34.19-8QSPI 备用字节寄存器(QUADSPI\_ABR)

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
ALTERNATE[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	ALTERNATE[31:0]	在地址之后立即发送到外部 SPI 设备的可选数据。 仅当 BUSY = 0 时才能写入此字段。	0x00000000	R/W

### 34.19-9QSPI 数据寄存器(QUADSPI\_DR)

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
DATA[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	Data [31:0]	<p>向外部 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）。支持对该寄存器的字、半字和字节访问。</p> <p>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。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	R/W

### 34.19-10QSPI 轮询屏蔽寄存器(QUADSPI\_PSMKR)

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
MASK[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	Mask [31:0]	<p>状态屏蔽 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	R/W

### 34.19-11QSPI 轮询状态匹配寄存器(QUADSPI\_PSMAR)

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
MATCH[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
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	R/W

### 34.19-12QSPI 轮询间隔寄存器(QUADSPI \_PIR)

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
INTERVAL[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0000	--
15:0	INTERVAL[15:0]	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	R/W

### 34.19-13QSPI 超时寄存器(QUADSPI \_LPTR)

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
TIMEOUT[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	保留	--	0x0000	--
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	R/W

## 【35】真随机数发生器 (TRNG)

### 35.1 概述

真随机数模块产生 32 位真随机数。

### 35.2 功能框图

TRNG 模块的数据流如下图示：

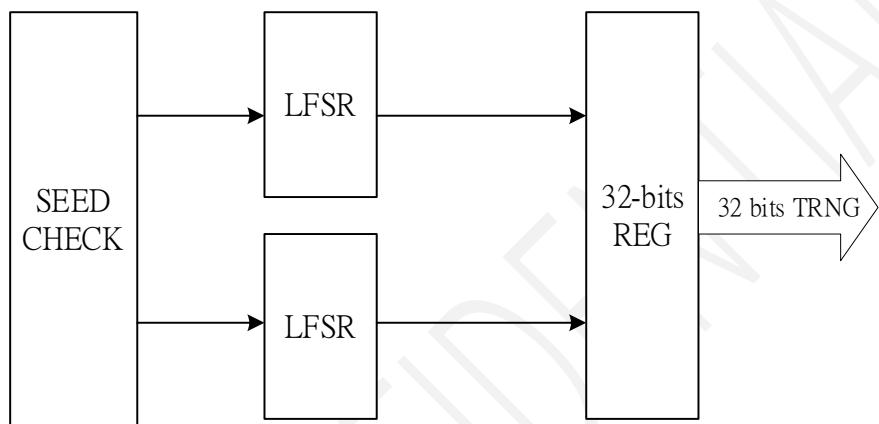


Figure 35- 1 真随机数框图

### 35.3 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 DMA 请求完成真随机数。

### 35.4TRNG 寄存器列表

基址: 0x4002 2400

偏移地址	名称	描述	默认值
0x00	TRNG_CR	TRNG 控制寄存器	0x00000000
0x04	TRNG_SR	TRNG 状态寄存器	0x00000000
0x08	TRNG_DR	TRNG 数据寄存器	0x00000000

Table 35- 1 寄存器列表和复位值

### 35.5 寄存器说明

#### 35.5-1TRNG 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
保留															
--															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										CED	保留		TRNGIE	TRNGEN	TRNG_CLK_DIV
--										R/W	--		R/W		

位	标记	功能描述	复位值	读写
31:6	保留	--	0x0	--
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	R/W
4	保留	--	0	--
3	TRNGIE	TRNG 中断始能 0: 无中断 1: TRNG 中断始能	0	R/W
2	TRNGEN	TRNG 始能 0: 关闭随机源 1: 打开随机源	0	R/W
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

### 35.5-2TRNG 状态寄存器(TRNG\_SR)

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								SEIS	CED	保留			SECS	CECS	DRDY
--								WO	--	RO					

位	标记	功能描述	复位值	读写
31:7	保留	--	0	--
6	SEIS	Seed 错误中断使能 0: 无中断 1: 使能中断, 至少有一次随机数产生时错误发生, 写 0 清除。	0	WO
5	CEIS	时钟错误中断使能 0: 无中断 1: 使能中断, 时钟错误发生, 写 0 清除。	0	WO
4:3	保留	--	0	--
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	RO
1	CECS	目前 TRNG 时钟的状态: 0: TRNG 时钟正常。 1: TRNG 时钟错误。	0	RO
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' 会发生中断。	0	RO

### 35.5-3TRNG 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[31:16]															
RO															

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

位	标记	功能描述	复位值	读写
31:0	RNDATA[31:0]	随机数数据 32 bits 当 DRDY = 1 时, RNDATA 有效 当 DRDY = 0 时, 数据全为 RNDATA = 0	0	RO

## 【36】高级加密标准模块（AES）

### 36.1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 DMA 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 DMA 传输（需要 2 个 DMA 通道）。

AES 硬件加速器可用于使用 AES 算法加密和解密数据。

### 36.2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 (DMA) 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。

### 36.3AES 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 36- 1 AES 中断表

### 36.4 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 DMAOUTEN = 1 时，标志 CCF 没有意义，因为读取 AES\_DOUTR 寄存器由 DMA 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 DMA 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）。

### 36.4-1 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。

### 36.4-2 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：

- (1) 电子密码本 (ECB)
- (2) 密码块链接 (CBC)
- (3) 计数器模式 (CTR)

### 36.4-2.1 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 Figure36-1,2 分别描述了电子密码本算法的加解密原理：

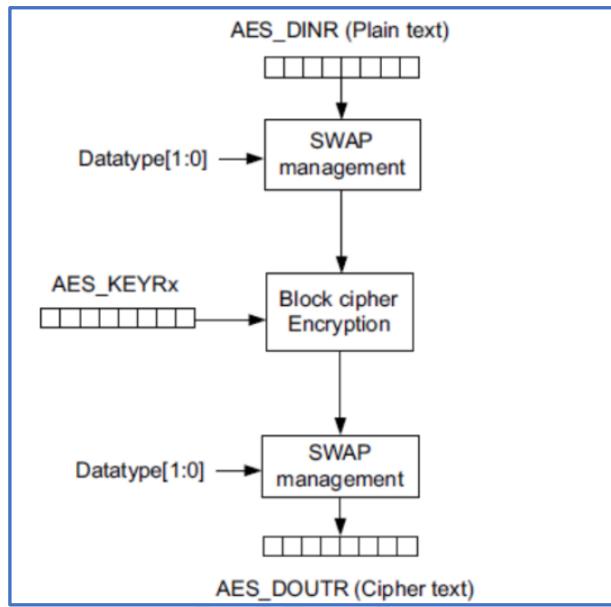


Figure 36- 1 ECB 加密模式.

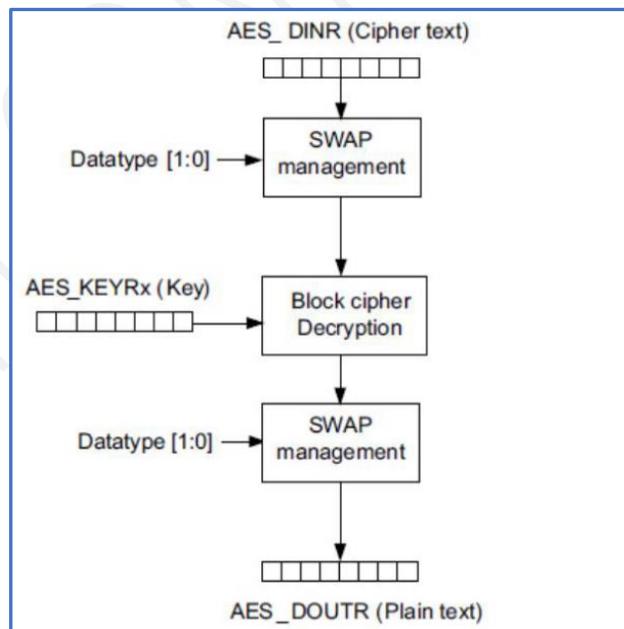


Figure 36- 2 ECB 解密模式

### 36.4-2.2 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）

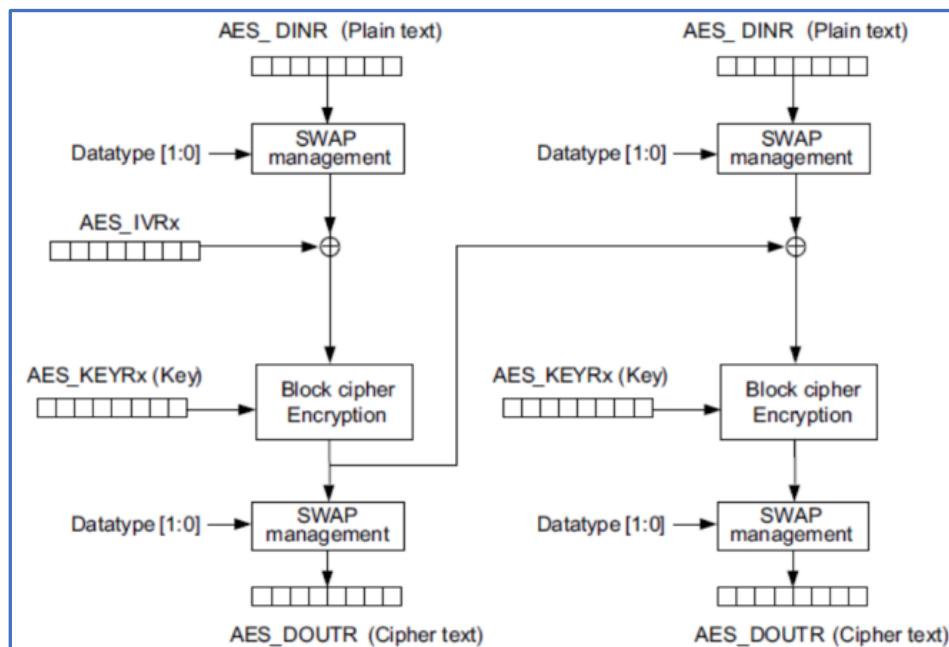


Figure 36- 3 CBC 加密模式.

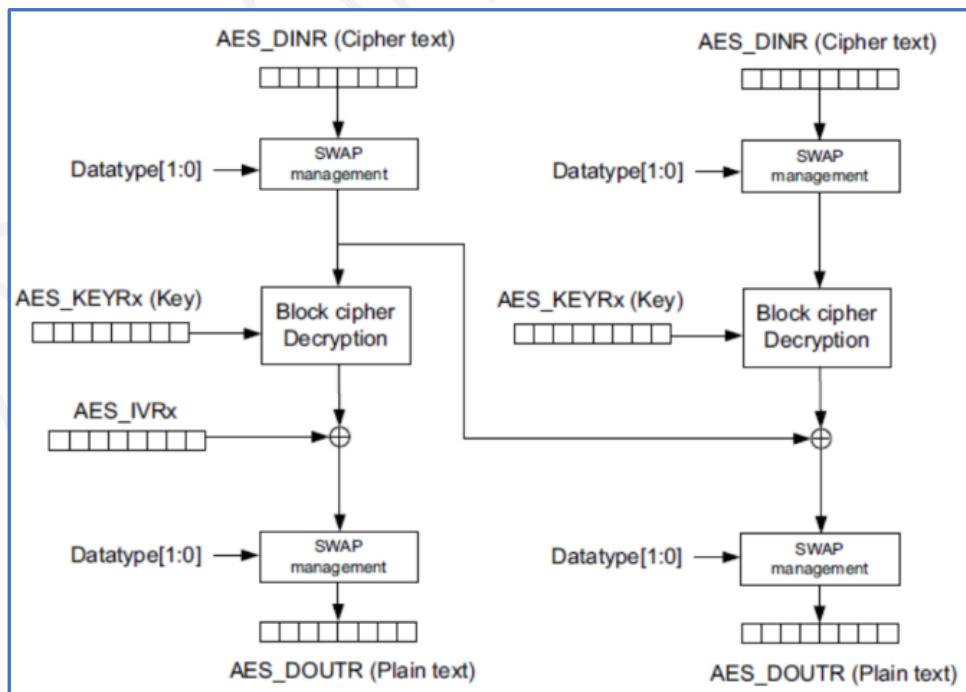


Figure 36- 4 CBC 解密模式

### 36.4-2.3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）

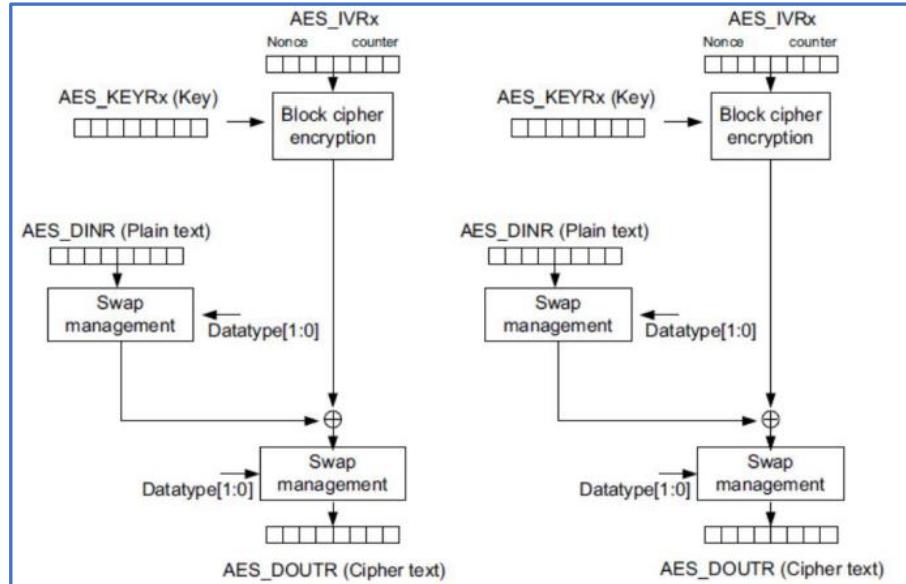


Figure 36- 5 CTR 加密模式.

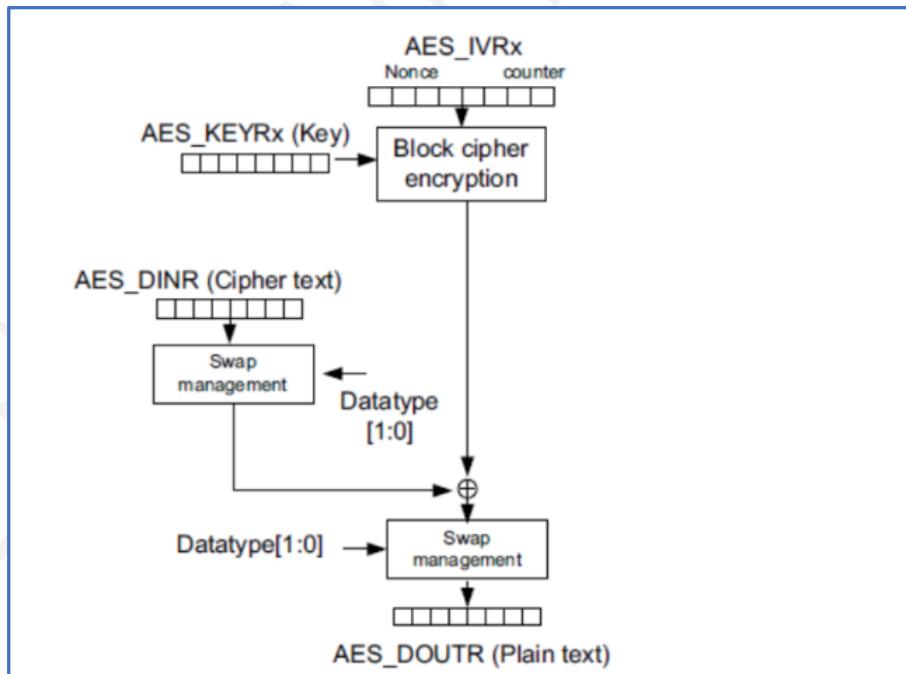


Figure 36- 6 CTR 解密模式

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

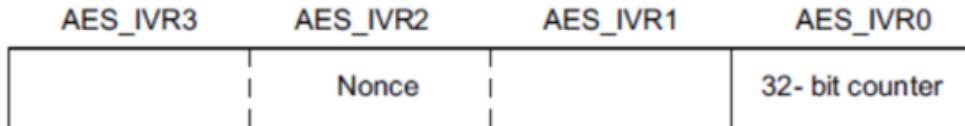


Figure 36- 7 32 位计数器+随机数组织

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。

### 36.4-3 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。

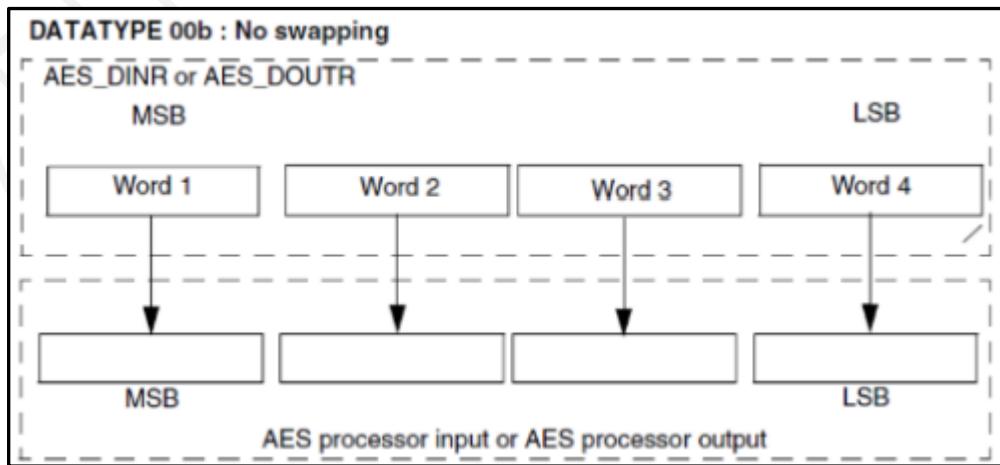


Figure 36- 8 Data Type= 2' b00

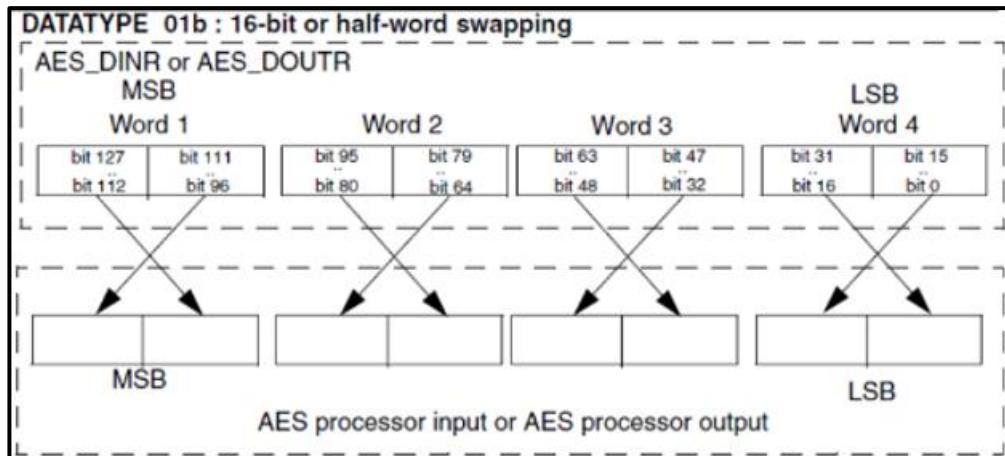


Figure 36- 9 Data Type= 2' b01

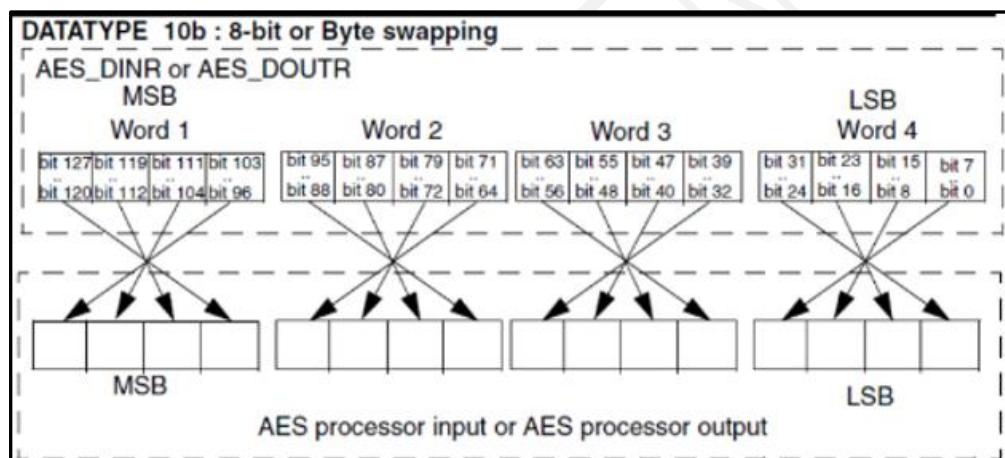


Figure 36- 10 Data Type= 2' b10

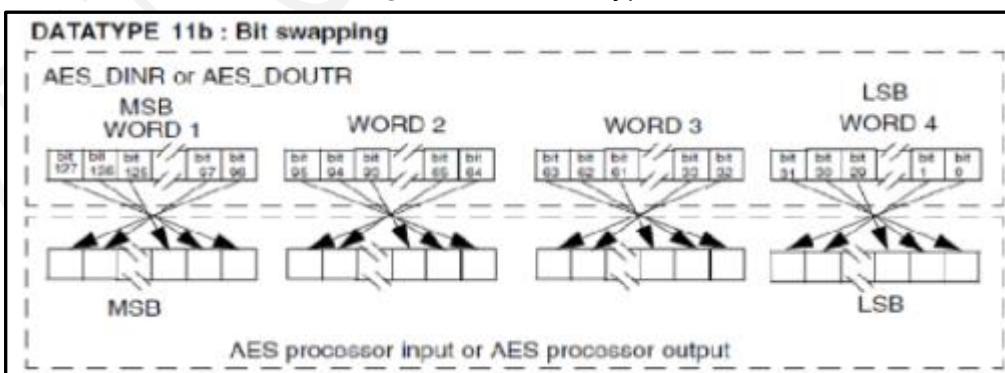


Figure 36- 11 Data Type= 2' b11

### 36.5AES 寄存器列表

地址: 0x4002 2000

偏移地址	名称	描述	默认值
0x00	AES_CR	AES 控制寄存器	0x00000000
0x04	AES_SR	AES 状态寄存器	0x00000000
0x08	AES_DINR	AES 输入数据寄存器	0x00000000
0x0C	AES_OUTR	AES 输出数据寄存器	0x00000000
0x10	AES_KEYR0	AES Key0寄存器(LSB: key[31:0])	0x00000000
0x14	AES_KEYR1	AES Key1寄存器(LSB: key[63:32])	0x00000000
0x18	AES_KEYR2	AES Key2寄存器(MSB: key[95:64])	0x00000000
0x1C	AES_KEYR3	AES Key3寄存器(MSB: key[127:96])	0x00000000
0x20	AES_IVR0	AES IVR0寄存器 (LSB: IVR [31:0])	0x00000000
0x24	AES_IVR1	AES IVR1寄存器 (LSB: IVR [63:32])	0x00000000
0x28	AES_IVR2	AES IVR2寄存器 (MSB: IVR[95:64])	0x00000000
0x2C	AES_IVR3	AES IVR3寄存器 (MSB: IVR [127:96])	0x00000000

### 36.6 寄存器说明

#### 36.6-1 AES 控制寄存器(AES\_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	DMA OUTEN	DMA INEN	ERR IE	CFF IE	ERRC	CCFC	CHMOD [1:0]	MODE [1:0]	DATATYPE [1:0]			AES EN	--	--	--
--							R/W								

位	标记	功能描述	复位值	读写
31:13	保留	--	0x0	--
12	DMAOUTEN	AES 数据输出阶段, 使用 DMA 功能 0: 不使用 DMA 1: 使用 DMA 数据输出, 当 Mode1/2/4 产生 DMA 请求时。	0x0	R/W
11	DMAINEN	AES 数据输入阶段, 使用 DMA 功能 0: 不使用 DMA 1: 使用 DMA 数据输入当, Mode1/2/4 产生 DMA 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: 使能中断	0x0	R/W
9	CCFIE	计算完成标志(CCF)中断使能 0: 禁用 CCF 中断 1: 使能 CCF 中断	0x0	R/W
8	ERRC	错误清除 写入 0 清除 RDERR , WRERR 标志。	0x0	R/W
7	CCFC	CCF 标志 Clear 清除 写入 0 清除 CCF 标志。	0x0	R/W
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: 保留 只有在禁用 AES 时才能更改 AES 链接模式。禁止在启用 AES 时写入这些位, 以避免不可预测的 AES 行为。	0x0	RW

位	标记	功能描述	复位值	读写
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。禁止在启用 AES 时写入这些位，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 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 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 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	0x0	R/W
0	AESEN	AES 启用 0: AES 禁用 1: AES 启用 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。 当 AES 计算在模式 2（密钥推导）下完成时，该位由硬件清零	0x0	R/W

### 36.6-2AES 状态寄存器(AES\_SR)

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															
--															

位	标记	功能描述	复位值	读写
31:3	保留	--	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  1: 检测读取错误</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 DMAOUTEN = 0 时有效。当 DMA_EN = 1 时，它可能保持高电平</p>	0	R/W

### 36.6-3AES 输入数据寄存器(AES\_DINR)

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
DINR[31:16]															
R/W															

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

位	标记	功能描述	复位值	读写
31:0	DINR[31:0]	<p>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。 <b>注意:</b> 该寄存器必须以 32 位数据宽度访问</p>	0x0	R/W

### 36.6-4AES 输出数据寄存器(AES\_OUTR)

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
OUTR[31:16]															
RO															

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

位	标记	功能描述	复位值	读写
31:0	OUTR[31:0]	<p>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。</p> <p>--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。</p> <p><b>注意：</b>该寄存器必须以 32 位数据宽度访问</p>	0x0	RO

### 36.6-5AES KEY0 寄存器(AES\_KEYR0) (LSB: key[31:0])

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
KEYR0[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR0[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	KEYR0[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。 注意: 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

### 36.6-6AES KEY1 寄存器(AES\_KEYR1) (LSB: key[63:32])

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
KEYR1[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	KEYR1[31:0]	数据输出寄存器 (key [63:32]) 请参考 AES_KEYR0 描述。	0x0	R/W

### 36.6-7AES KEY2 寄存器(AES\_KEYR2) (MSB: key[95:64])

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
KEYR2[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	KEYR2[31:0]	数据输出寄存器 (key [95:64]) 请参考 AES_KEYR0 描述。	0x0	R/W

### 36.6-8AES KEY3 寄存器(AES\_KEY1R) (MSB: key[127:96])

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
KEYR3[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	KEYR3[31:0]	数据输出寄存器 (key [127:96]) 请参考 AES_KEYR0 描述。	0x0	R/W

### 36.6-9AES IVR0 寄存器(AES\_IVR0)(IVR[31:0])

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
IVR0[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR0[15:0]															
R/W															

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

### 36.6-10AES IVR1 寄存器(AES\_IVR1)(IVR[63:32])

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
IVR1[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR1[15:0]															
R/W															

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 [63:32] ) 请参考 IVR0 的说明。	0x0	R/W

### 36.6-11AES IVR2 寄存器(AES\_IVR2)(IVR[95:64])

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
IVR2[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR2[15:0]															
R/W															

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] ) 请参考 IVR0 的说明。	0x0	R/W

### 36.6-12AES IVR3 寄存器(AES\_IVR3)(IVR[127:96])

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
IVR3[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVR3[15:0]															
R/W															

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 [127:96] ) 请参考 IVR0 的说明。	0x0	R/W

## 【37】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	保留（系统使用）
0x0800 0004	USERCFG1	SWD 保护位配置字
0x0800 0008	USERCFG2	保留（系统使用）

### 37.1 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															
--															

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

位	标记	功能描述	默认值	读写
31:0	保留	--	0	--

### 37.2 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															
--															

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

位	标记	功能描述	默认值	读写
31:1	保留	--	0x7FFFFFFF	--
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

### 37.3 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
保留															
--															

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

位	标记	功能描述	默认值	读写
31:0	保留	--	0	-

### 37.4 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。

## 【38】Debug 支持(DBG)

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。本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 37.2 用户配置寄存器 1\(USERCFG1\)](#)。

注意：

- 当用户没有设置 SWDP 时，PC7 和 PD1 自动被配置成 SWD 调试端口(PC7 上拉，PD1 下拉)；用户也可以通过配置 RCC\_SWDIOCR.SWDPORT 寄存器将调试接口配置成 GPIO。
- 当用户设置 SWDP 的值为 0 时，PC7 和 PD1 端口与 SWD 调试端口自动断开，即不能使用 SWD 调试功能。RCC\_SWDIOCR.SWDPORT 不能被写 1。

### 38.1 SWD 调试接口说明

#### 38.1-1 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	PC7
SWDCLK	输入	串行时钟	PD1

#### 38.1-2 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 SWDCLK 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。

### 38.2 SWD 保护位工作原理

1. 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。
2. 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。
3. 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。
4. MCU 重新复位之后, SWD 保护位即刻生效。
5. 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。

### 38.3 在低功耗模式下使用 SWD

- 在睡眠模式(**Sleep Mode**)下使用 SWD

在睡眠模式下时, 系统时钟保持工作, SWD 连接不会中断。

- 在深度睡眠模式(**Deep Sleep Mode**)下使用 SWD

1. 当 SYSCON\_CFGR0.DBGDLSP\_DIS=0(默认值)时, 在 Debug 模式下进入深度睡眠模式系统时钟会停止, SWD 连接会中断; 当 SYSCON\_CFGR0.DBGDLSP\_DIS=1 时, 在 Debug 模式下进入深度睡眠模式系统时钟不会停止, SWD 连接也就不会中断。
2. 当进入 Deep Sleep Mode 后, 通过复位, 可以通过 SWD 端口来唤醒芯片。

### 38.4 DBG 寄存器列表

基址地址: 0x4000 4C00

偏移地址	名称	描述	默认值
0x000	DBG_APBFZ	Debug 模式控制寄存器	0x00000000

**Table 38- 1 Debug 寄存器映象和复位值**

### 38.5 DBG 寄存器说明

#### 38.5-1 Debug 模式控制寄存器(DBG\_APBFZ)

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	TIM2C DBG STOP	TIM2B DBG STOP	TIM2A DBG STOP	TIM2 DBG STOP	WWDG DBG STOP	IWDG DBG STOP	BEEP DBG STOP	TIM1B DBG STOP	RTC DBG STOP	TIM1 DBG STOP	PCA DBG STOP	TIM1A DBG STOP	LPTIM DBG STOP	TIM11 DBG STOP	TIM10 DBG STOP
--	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	保留	--	0x0	--
14	TIM2CDBGSTOP	TIM2C 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
13	TIM2BDBGSTOP	TIM2B 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
12	TIM2ADBGSTOP	TIM2A 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
11	TIM2DBGSTOP	TIM2 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
10	WWDGDBGSTOP	WWDG 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
9	IWDGDBGSTOP	IWDG 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
8	BEEPDBGSTOP	BEEP 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
7	TIM1BDBGSTOP	TIM1B 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
6	RTCDBGSTOP	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
5	TIM1DBGSTOP	TIM1 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
4	PCADBGSTOP	PCA 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
3	TIM1ADBGSTOP	TIM1A 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
2	LPTIMDBGSTOP	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
1	TIM11DBGSTOP	TIM11 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
0	TIM10DBGSTOP	TIM10 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

## 【39】附录 1：SysTick 定时器(SYST)

### 39.1 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。

### 39.2 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	$1\text{ms} = 0.25\mu\text{s} * (3999+1)$

### 39.3SysTick 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 40- 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)。

### 39.4SysTick 定时器寄存器说明

#### 39.4-1SysTick 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	--	--	0
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	R/W	0
1	TICKINT	SysTick 中断使能 1: 使能中断 0: 禁止中断	R/W	0
0	ENABLE	SysTick 定时器使能1: 使能SysTick 0: 禁止SysTick	R/W	0

#### 39.4-2SysTick 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	R/W	未定义

#### 39.4-3SysTick 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。	R/W	未定义

#### 39.4-4SysTick 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

## 【40】版本修订纪录

Version	Date.	Description
V1.00	2021-1101	初稿发布
V1.01	2021-1108	Modified: (1) CRC chapter (2) QSPI 内存映射图 (3) OP2 结构框图选择
V1.02	2021-1117	Modified: (1) <a href="#">4.1 引脚定义(新增 BOOT1)</a> (2) <a href="#">4.2 引脚功能配置 include Timer ETR</a> (3) <a href="#">7.2-1 系统时钟控制模块图</a> (4) <a href="#">8.2-5,6 Timer ETR Define</a> (5) <a href="#">8.2-8 SYSCON BOOT状态寄存器</a> (6) <a href="#">11.3-8.2 外部 BOOT PIN 选择</a> (7) <a href="#">5.4 BOOTPIN 地址映射</a>
V1.03	2021-1206	Modified: (1) <a href="#">5.4BOOTPIN 地址映射</a> (2) <a href="#">29.2 ADC 框图</a> (3) <a href="#">32.3 OPA 框图</a> (4) <a href="#">32.4-3 串級 PGA</a> (5) <a href="#">8.2-2 端子 Deep Sleep 中断模式控制寄存器</a> Add (1) <a href="#">24.6-3.2 使用 DMA 发送</a> (2) <a href="#">24.6-4.2 使用 DMA 接收</a>
V1.05	2022-0121	Add (1) <a href="#">22.7.4 控制寄存器 1 (USARTx CR1)(bit 14)</a> Modified: (1) <a href="#">22.4-10 红外线数据传输(IrDA)</a> (2) <a href="#">8.2-5 TIM1,1A,1B 通道输入源选择(SYSCON_TIM_TIM1XCH1IN,TIM1XCH2IN,TIM1XCH3IN</a>
V1.06	2022-0207	Modified: (1) <a href="#">Block Diagram</a> (2) <a href="#">系统的模块框图</a>

V1.08	2023-0117	Modified: (1) <a href="#">控制寄存器 3(USART CR3)</a> (2)新增 Note 索引 (3) 新增 <a href="#">支持 QPSI 单/双模式</a>
V1.09	2023-0327	Modified: 内容和排版修正